Server SuiteScript Developer's Guide

NETSUITE™

Server SuiteScript Developer’s Guide

October 12, 2006

Trademarks

Contents

Chapter 1 Welcome

Chapter 2 Defining Scripts

Chapter 3 Server SuiteScript API

Chapter 4 Use Cases

Errors

Server SuiteScript API Library File

Chapter 1 Welcome

Welcome to NetSuite's Server SuiteScript feature. With SuiteScript you can add a powerful level of customization to your NetSuite business applications. Customizing your workflow processes, generating email notifications and automatically creating or updating records are just some of the things SuiteScript offers you the ability to do.

There are currently three different types of SuiteScript that can be created:

- **User Event:** This script is executed upon pre-defined events on records such as after creation of a record or before an update of a record. User event scripts are useful for customizing the workflow and association between your NetSuite entry forms. These can also be used for doing additional processing before records are entered or for validating entries based on other data in the system.
- **Scheduled**: This script is executed on demand or via a user-configurable schedule. Scheduled scripts are useful for bulk processing of records.
- **Dashboard**: This script is used to create custom dashboard portlets. For example, you can use SuiteScripts to create a portlet and populate with company messages based on data within the system on the fly.

Some of the API exposed in Server SuiteScript is also available in Client SuiteScript. For example, you can use the nlapiLoadRecord or nlapiSaveRecord functions to manipulate record objects client-side.

Important: The purpose of this guide is to introduce you to SuiteScript. It provides step-by-step instructions on how to create and deploy SuiteScript scripts as well as real-world examples of how companies use SuiteScript to customize NetSuite accounts for their businesses. SuiteScript requires an understanding of JavaScript code and is intended to be used by customers with some advanced programming knowledge only.

In This Guide

This guide contains the following chapters:

Chapter 1 "Welcome": provides an overview of this guide, enabling SuiteScript, governance and best practices when working with Server SuiteScript.

Chapter 2 "Defining Scripts": describes how to upload and implement Server SuiteScripts in the NetSuite UI.

Chapter 3 "Server SuiteScript API": provides details on the Server SuiteScript API, including descriptions of available functions and objects and sample code snippets.

Chapter 4 "Use Cases": details specific examples of how Server SuiteScript can be used in your NetSuite implementation with full code samples.

"Errors": provides tables of possible error messages that can be returned.

"Server SuiteScript API Library File": provides text for a library file that can be used to enable code completion of the SuiteScript API in your IDE.

Important: The **Server SuiteScript Reference Guide** is also available from the NetSuite Help Center. This guide provides much of the reference material you will need when developing your Server SuiteScripts. This includes material such as Standard Field Names, Search Field Name, Search Column Names, Search Joins, Search Operators, and Machine Field Names.

Getting Started

Enabling the Server SuiteScript Feature and Preferences

Before beginning with SuiteScript, you should enable the Server SuiteScript feature and enable the preference that causes NetSuite to display internal IDs.

- **Enabling the SuiteScript Feature**: go to Setup > Company > Enable Features and check Server SuiteScripts on the General tab.
- **Displaying Internal IDs**: go to Home > Set Preferences and check Show Internal IDs. When Show internal IDs is checked, the IDs for records are displayed in the ID column of record lists. You may need to reference record IDs in your scripts.

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

Supported Records

The following table lists each supported record and the associated record ID. Many of the available SuiteScript functions take the record ID as an argument. For example, to search for a specific record, you must provide the record ID for the type of record being searched.

Record Type	Record	Record ID
Entities	Contact	contact
	Customer	customer
	Employee	employee
	Competitor	competitor
	Lead	lead
	Partner	partner
	Prospect	prospect
	Vendor	vendor
	Job	job
Activities	Event	calendarevent
	Phone Call	phonecall
	Task	task
Transactions	Cash Sale	cashsale
	Invoice	invoice
	Item Fulfillment	itemfulfillment
	Item Receipt	itemreceipt
	Expense Report	expensereport
	Opportunity	opportunity
	Purchase Order	purchaseorder
	Sales Order	salesorder
	Estimate	estimate
	Time	timebill
	Vendor Bill	vendorbill
Support	Case	supportcase
	Solution	solution
Items	Description	descriptionitem
	Discount	discountitem
	Inventory Part	inventoryitem
	Markup	markupitem
	Non-Inventory Part	noninventoryitem
	Payment	paymentitem
	Service	serviceitem
	Subtotal	subtotalitem
Communications	Campaign	campaign
	Message	message

Record Type	Record	Record ID
	Note	note
Issue	Issue Records	issue
Customization	Custom Records	

Governance

In order to optimize Netsuite application and database servers, we have implemented the following two mechanisms to govern the usage of SuiteScript scripts:

- **Time-Based Governance**: a script is terminated during execution once a pre-defined time period has elapsed as follows:
  - User Event and Dashboard scripts: 30 seconds
  - Scheduled scripts: 15 minutes
- **Usage-Based Governance**: a script is terminated during execution if it exceeds a pre-defined amount of usage. Usage is metered depending on the type of script and type of call as follows:
  - nlapiDeleteRecord, nlapiSubmitRecord: 20 units
  - nlapiLoadRecord, nlapiCopyRecord, nlapiCreateRecord, nlapiTransformRecord, nlapiSearchRecord, nlapiRequestURL, nlapiMergeEmail, nlapiSendEmail: 10 units
  - User Event and Dashboard scripts: 300 units
  - Scheduled scripts: 3,000 units

Script Development Workflow

Setting up Your Development Environment

When developing your Server SuiteScripts, it is always useful to use an IDE tool that supports Javascript. There are many tools on the market that provide everything from simple syntax highlighting to code-completion and suggestions.

Important: A SuiteScript API file for use with your IDE is available for download from the SuiteScripts folder at Documents > Files > SuiteScripts.

Setting Up Eclipse

If you don’t already have an IDE set up or are looking for an alternate solution, this section provides details on one popular IDE that works well for developing your custom Server SuiteScripts — Eclipse with the JSEclipse plugin. Some benefits of this tool include:

- Syntax highlighting
- Code completion
- Argument auto-suggestion
- Library importing

- Error reporting

To get started with Eclipse:

1. Download and install Eclipse and the JSEclipse plugin.

   Go to the following URLs to get the latest version of each.

   - Eclipse Website
   - InterAKT Website (for JSEclipse Plugin)

2. Install Eclipse as described in the Eclipse documentation.

3. Unzip the JSEclipse Plugin and copy the com.interaktonline.jseclipse_x.x.x.jar file from the plugins folder to the Plugins folder in the Eclipse root directory.

   (where x.x.x is the version you downloaded)

4. Restart Eclipse.

5. Create a new project and add a javascript file to that project.

   Once a Javascript file is opened in the Edit pane within Eclipse, a new JSEclipse menu is displayed.

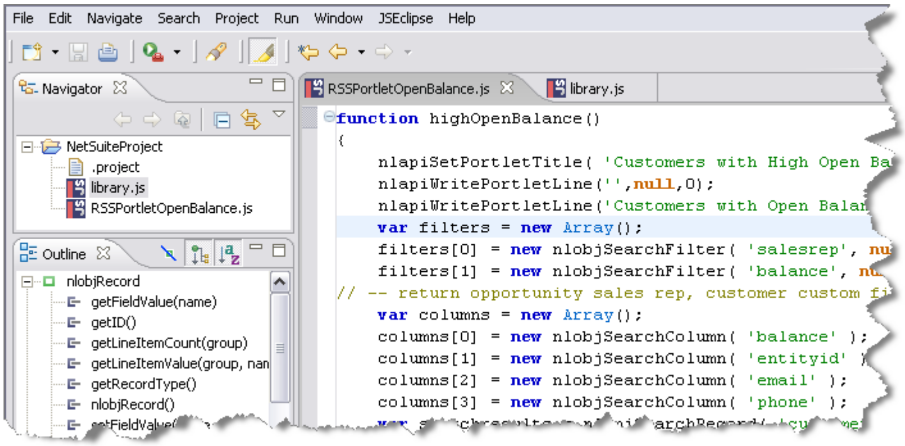
6. Before continuing you should add the SuiteScript API Library File to your project.

Adding a Library File to your IDE

Before beginning with your script development, it is a good idea to include code completion libraries whereever possible. In the section Server SuiteScript API Library File we have provided the text that can be copied directly into your Javascript file.

Important: If you add the Server SuiteScript API Library file directly to your SuiteScript file, you must remember to remove this section before publishing your file.

JSEclipse comes with code completion for most of the default libraries (document, system, etc) and you can add your own as well. Simply add the text from the section Server SuiteScript API Library File to a .js file within your JSEclipse project. JSEclipse then automatically scans for and provides code completion for the functions and methods provided in this library file.



```
function highOpenBalance()
{
    nlapiSetPortletTitle( 'Customers with High Open Ba
    nlapiWritePortletLine('',null,0);
    nlapiWritePortletLine('Customers with Open Balar
    var filters = new Array();
    filters[0] = new nlobjSearchFilter( 'salesrep', nu
    filters[1] = new nlobjSearchFilter( 'balance', nu
// -- return opportunity sales rep, customer custom fi
    var columns = new Array();
    columns[0] = new nlobjSearchColumn( 'balance' );
    columns[1] = new nlobjSearchColumn( 'entityid' )
    columns[2] = new nlobjSearchColumn( 'email' );
    columns[3] = new nlobjSearchColumn( 'phone' );
```

JSEclipse Hints

Some of the useful features of Eclipse are as follows:

- Click Ctrl-Space while typing to get a drop-down of code completion suggestions.
- Double-click on any function to highlight all instances of the function within the current file.
- Check the Problems panel as you work for potential errors in your script.
- Collapse or expand blocks of code by selecting the - or + signs next to a function.

Chapter 2 Defining Scripts

When using SuiteScript, you must first create a script record and then define the deployments for that script record. When creating the script record, you can define what script file is to be loaded, which functions are run and possible parameters. During deployment, you can define specific values for parameters, define which records the script will run against, and for Scheduled scripts, define the execution times for the script. You can have multiple deployments for each defined script.

Creating Script Records

To create a SuiteScript script record:

1. Go to Setup > Customization > Server Scripts > New.
2. Select the desired type of script.
   - **Dashboard**: a script used to create dashboard portlets. For example, you use server-side scripts to create and populate an RSS portlet on the fly.
   - **Scheduled**: a script that is executed according to a pre-defined schedule. Scheduled scripts are useful for bulk processing of records.
   - **User Event**: a script that is executed upon pre-defined events on records such as after creation of a record or before an update of a record. User event scripts are useful for customizing the workflow and association between your NetSuite entry forms.
3. Enter a Name for the script.

   Since you may have multiple deployments of a single script, ensure that the name of the script is generic enough to be relevant for all deployments. For example, a single script may be applied to several different record types so it may not make sense to include the record type in the script name.
4. If desired, enter a description of the script.
5. Select an Owner of the script.

   By default the owner is set to the currently logged in user. Once a script is created, only the owner of the script can modify it.
6. Select the script file and library script file to associate with the current script record.

   The library script file should contain any commonly used functions whereas the script file should contain functions specific to the current script record.
7. Select the function you'd like to perform from the selected script files.

   If defining a User Event script, you can select one function for each section. For example, you can have a Before Load Function and After Submit Function defined within the same script execution. These functions must exist in either the libary script file or the script file associated with the script record.

- **Before Load Functions**: useful to restrict access to records, set default values for new records or non-stored fields, or implement audit trails.
- **Before Submit Functions**: useful for performing any necessary validation, permission checking, or pre-processing prior to writing a record to the system.
- **After Submit Functions:** useful for performing any post-commit operations that require the current record to exist prior to execution — such as notificiations, synchronization, setting redirects, or process automation.

The type argument provides additional context for when a script should be executed. For more information, see “Type Arguments” on page 11.

Important: If a function has parameters associated with it that are being passed from the deployment definitions, you must use the parameter field ID, NOT the field name, in the function call.

8. In the Parameters tab, define the possible parameters to pass to the functions selected in the previous step.

   The parameters defined here are custom code fields and can have any of the characteristics of a custom field. For example, you can assign sourcing and defaulting behaviors. Once the script is deployed, you can then define individual deployments to have specific default values for each parameter.

   Note: When defining a field here, the field ID is prefaced with **custscript** to indicate that the field is associated with a script.

9. In the Unhandled Errors tab, define the notification messages for errors.

   Enable the individuals or groups you'd like to receive notification when errors occur. If desired, you can also enter a comma-deliminated list of emails of who should be notified.

   Note: Only existing groups are available in the Groups notification drop-down list. To define new groups for selection here, go to Lists > Relationships > Groups.

10. Click Save.

When this script is saved, it will still not be run until specific deployments are defined as described in the next section.

Deploying Scripts

Once a SuiteScript script record has been defined, you must deploy the script before it is actually executed. When deploying a script, you need to do the following:

- Define the records you want the script associated with.
- Define the roles where the script should execute.
- Define any deployment-specific parameter defaults.
- In the case of scheduled scripts, define the execution times.

- In the case of Dashboard scripts, enable the SuiteScript Portlet preference and select the desired Dashboard script.

Tip: Because multiple deployments can be applied to the same record, and these deployments are executed in the order specified in the UI, if an error occurs in one deployment subsequent deployed scripts may NOT be executed. When troubleshooting, you may want to make sure that you are executing only one script per record type.

Defining Deployments

To define a script deployment:

1. Go to the desired script record.

   Click Setup > Customization > Server Scripts > [Script] > Edit. Where [Script] is the desired SuiteScript script.

2. Select the Deployment tab.

3. Do one of the following:

   - For User Event scripts, select each record you would like the script to be associated with.
   - For Scheduled scripts, define the date and times for this deployment. You must also define the current deployment as Scheduled before it will be executed.

     Note: Click the **Go to Calendar** link in the upper right corner to view a calendar that lists all currently scheduled scripts.

   - For Dashboard scripts, enter a name for the script. After saving, set the new Dashboard script to display on your dashboard.

     Note: You can only define one deployment per Dashboard script.

4. Click Save.

To display Dashboard scripts on the dashboard:

1. Go to your dashboard page and click Customize this Page.

2. Enable **SuiteScript Portlet** on the Standard Content tab.

3. Click Save.

   An empty scripted portlet is displayed on your dashboard.

4. Click Set Up.

5. In the Set Up Scripted Content popup, select the desired Dashboard script from the Source dropdown list and then click Save & Close.

   The portlet will populate with data as defined in your Dashboard script.

Viewing and Organizing Deployments

There are several ways to view your script deployments:

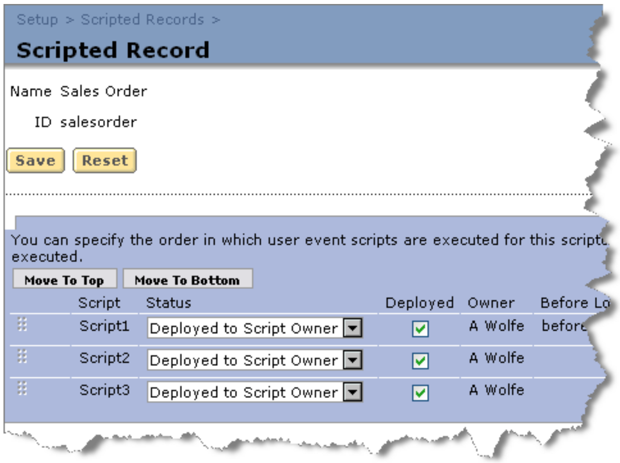
- Go directly to the script deployment by clicking Setup > Customization > Script Deployments
- View deployed scripts by clicking View Deployments in the upper right corner of the page while in a SuiteScript record.
- Click the deployments tab within a script record to see the deployments specific to that script record and then click on a specific deployment to go to the deployment record.

  Remember: In each specific deployment record you can define default parameter values for that deployment.
- View a list of records that have scripts associated with them at Setup > Customization > Scripted Records.

  By default, this list only displays records that have at least one script associated with it. Enable the Show Undeployed checkbox to view all records regardless of whether or not a script has been associated with it. Use the Script filter drop-down to list only those records associated with a specific SuiteScript.

  Note: The scripted records page lists all records with an associated SuiteScript regardless of whether or not that script is inactive.

  From the scripted records page, you can drill down in a record to specify the execution order of scripts associated with each record, edit the deployment status, and inactivate a specific deployment.



Name Sales Order

ID salesorder

You can specify the order in which user event scripts are executed for this script executed.

Script	Status	Deployed	Owner	Before Lo
Script1	Deployed to Script Owner	☑	A Wolfe	before
Script2	Deployed to Script Owner	☑	A Wolfe	
Script3	Deployed to Script Owner	☑	A Wolfe	

Note: You can NOT edit a SuiteScript deployment or definition while the script is currently executing.

Chapter 3 Server SuiteScript API

The SuiteScript API consists of a set of Functions that can be used to manipulate supported NetSuite records. These functions may return Objects which in turn can be acted on by a set of methods as described in this section.

Type Arguments

For user event scripts, you can associate an event **type** argument to each function such that the action will only be performed on create, edit, view, or delete of the record. For example, suppose that you have a script associated with a beforeLoad event for a given record. If you would like to cause an action only when the record is initially created, specify that the type should be create.

```
function beforeSaveSalesOrder(type)
{
  var newRecord = nlapiGetNewRecord();
  var cutoffRate = custscript_maximumdiscountlevel;
  var discountRate = newRecord.getFieldValue('discountrate');
  if ( type == 'Create' && discountRate != null && discountRate.length > 0
  && cutoffRate != null && cutoffRate.length > 0 )
    {
    discountRate = Math.abs( parseFloat( discountRate ) );
    ...remainder of code...
```

Note: Type is an **autogenerated** argument passed by the system. You can NOT set this as a parameter for a specific deployment like other function arguments.

Possible type arguments vary depending on the event as outlined in the following table.

Function	Argument
beforeLoad	type (create, edit, view, copy)
beforeSubmit	type (create, edit, delete, approve, reject, xedit, cancel)
afterSubmit	type (create, edit, delete, approve, reject, xedit)

Note: The portlet argument can also be set for functions. For more information, see "nlobjPortlet" on page 29.

NLAPI Functions

This section describes the set of Functions that can be invoked in your Server SuiteScripts. The following table outlines each function. For detailed information and examples for each function, click the Function name.

In addition to the NLAPI functions described below, you can use the following Client SuiteScript functions when working with User Event Scripts:

- nlapiGetFieldValue

- nlapiSetFieldValue
- nlapiGetLineItemValue
- nlapiSetLineItemValue
- nlapiGetLineItemCount

Function	Arguments	Returns	Function Description
nlapiCopyRecord	type: the recordType for the record id: the internal id for the record	nlobjRecord object	Initializes a new record using data from an existing record of a different type
nlapiCreateError	code: user defined error code details: user defined error detail	nlobjError	Use to create your own nlobjErrors.
nlapiCreateRecord	type: record type	nlapiRecord object	Instantiates a scriptable nlapiRecord object using default values for that recordtype.
nlapiDateToString	d: date object being converted into a String		Convert a Date object into a String using the current user's date format
nlapiDeleteRecord	type: record type id: internalID	N/A	Deletes a scriptable record from the database
nlapiFormatCurrency	str: string being formatted into currency	currency field value	Format a String into a currency field value
nlapiGetNewRecord	none	Returns an nlobjRecord representing the record data submitted to the system for a write operation	Available in beforeSubmit and afterSubmit user event scripts only.
nlapiGetOldRecord	none	Returning an nlobjRecord representing the state of the record prior to the current write operation.	
nlapiGetRecordId	ID for the current record	ID for the current record	Use to retrieve the internalID of the current record in a User Event script.
nlapiGetRecordType	recordtype for the current record	Recordtype for the current record	Use to retrieve the record type of the current record in a User Event SuiteScript.
nlapiGetContext	nlobjContext	nlobjContext object: returns context information (metadata) about current user and script execution	Use to branch scripts depending on the metadata. For example, you may want the script to perform in one way when a form is accessed via the UI and another when the form is accessed via Web services.

Function	Arguments	Returns	Function Description
nlapiLoadRecord	type: record type id: internalID	nlobjRecord object	Load a record from the database into an nlobjRecord object.
nlapiLogExecution	type: log type (audit, debug, error, emergency) subject: log subject details: log details	N/A	Logs a message, audit, or note against a SuiteScript script deployment.
nlapiLookupField	type: sets the recordType to perform a field lookup on id: the internal id for the record fields: set an array of column/field names to look up, or a single column/field name	field values	Convenient helper function used to perform searches for field(s) on a records.
nlapiLookupField	type: type of record to perform search on id: internal ID of the record fields: an array of column/field names that we're looking up -or- a single column/field name	values for each field in the array	Use to search for values of fields on specified record.
nlapiMergeEmail	id: The internalId of the email template baseType: The recordType for the primary record used in the merge baseId: The internalId for the primary record used in the merge altType: The recordType for the secondary record used in the merge altId: The internalId for the secondary record used in the merge	string containing the results.	Performs a mail merge operation
nlapiRequestURL	url: String (fully qualified unless Netsuite page) postdata: postdata used for a POST request. It can either be an associative array of form parameters or a String. If null, then a GET request is used headers: Custom HTTP headers to be included in the request	response: nlobjServerResponse	request an external resource (URL) over HTTP and return an nlobjServerResponse object containing the response.

Function	Arguments	Returns	Function Description
nlapiResolveURL	identifier: The primary id for this resource (recordType for RECORD, scriptId for SCRIPTLET) type: recordtype or tasklink intenalId id: The secondary id for this resource (recordId for RECORD, deploymentId for SCRIPTLET) editmode: if TRUE, returns the URL to an existing record in EDIT mode	url: String	resolves a Netsuite hosted HTTP resource into a URL
nlapiSearchRecord	type: record type id: saved search ID nlobjSearchFilter[ ] : An array of nlobjSearchFilter search field objects nlobjSearchColumn[ ] : array of nlobjSearchColumn return field objects	nlobjSearchResult[] : An array of nlapiSearchResult objects	Executes and returns the results of an existing NetSuite saved search.
nlapiSelectNode	nlapiDocument: XML node expression: x-path expression	value: Node	select Node from an XML node using Xpath
nlapiSelectNodes	nlapiDocument: XML node expression: x-path expression	value[] : Array of Nodes	select Node Array from an XML node using Xpath
nlapiSelectValue	nlapiDocument: XML node expression: x-path expression	value	select value from an XML node using Xpath
nlapiSelectValues	nlapiDocument: XML node expression: x-path expression	value[] : Array of values	select value Array from an XML node using Xpath

Function	Arguments	Returns	Function Description
nlapiSendEmail	author: author internalID (must be an employee record and defaults to script deployer ID) recipient: recipient internalID or external email address subject: email subject body: email body cc[ ] : array of e-mails to copy bcc[ ]: array of emails to blind copy records: An associative array of internal records to associate this email with. The following options are valid: • transaction - Transaction/ Opportunity • activity - Case/Campaign • entity - Entity • record - Custom record internalId • recordtype - Custom recordType internalId	N/A	Use to generate and send an email.
nlapiSetRedirectURL	identifier: The primary id for this resource (recordType for RECORD, scriptId for SCRIPTLET) type: recordtype or tasklink id id: The secondary id for this resource (recordId for RECORD, deploymentId for SCRIPTLET) editmode: if TRUE, returns the URL to an existing record in EDIT mode parameters: An associative array of additional URL parameters	N/A	Sets the redirect URL or redirects user directly (for user event load script). You can only redirect to pages within NetSuite.
nlapiStringToDate	str: string being converted to a Date		Convert a String to a Date object using the current user's date format
nlapiStringToXML	text: String	Document: org.w3c.dom.Document	load Document from String
nlapiSubmitRecord	record: nlobjRecord object doSourcing: if TRUE, enables automatic field sourcing	internalID	Submits an nlobjRecord to the database returning the ID used to save record

Function	Arguments	Returns	Function Description
nlapiTransformRecord	type: record type id: internalID newType: recordtype	nlapiRecord object	transforms an existing record from one type to another and returns a pre-populated nlapiRecord object
nlapiXMLToString	xml: org.w3c.dom.Document	String	load String from Document

nlapiSearchRecord

The nlapiSearchRecord function can be used to execute and return the results a new search or of an existing NetSuite saved search. You can then extract the desired information from the search results using the methods available on the returned nlobjSearchResults object.

Important: The nlapiSearchRecord function returns up to 500 records. Because of this, you should always include an error handling routine to verify that less than 500 records are returned. If 500 records are returned, and it is critical that the subsequent actions on the search results are performed on all records that match the criteria of the search, you should abort the operation. If it is NOT critical that the subsequent actions are performed on all results, you can simply limit the number of results to less than 500 records.

Arguments	Returns
type: record type id: saved search ID nlobjSearchFilter[ ] : An array of nlobjSearchFilter search field objects or a single nlobjSearchFilter nlobjSearchColumn[ ] : array of nlobjSearchColumn return field objects or a single nlobjSearchColumn	nlobjSearchResult[] : An array of nlobjSearchResult objects.

Executing a Search

When executing a search, you must provide the recordType. In the following code, a search for all customer records in the system is executed with the maximum limit of 10 results set.

```
function executeSearch()
{
  var searchresults = nlapiSearchRecord( 'customer', null, null, null );
  for ( var i = 0; i < Math.min( 10, searchresults.length ); i++)
  {
    var record = nlapiLoadRecord(searchresults[i].getRecordType(),
      searchresults[i].getId() );
  }
}
```

Returning Specific Fields in a Search

After performing a search, use the nlobjSearchRecord getValue method to return the values of specific record fields. In the following example, the email fields for records returned from a saved customer search are returned.

```
function findCustomerEmails()
{
  var searchresults = nlapiSearchRecord('customer', 8, null,null);
  var emailAddress = '';
  for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
  {
    var searchresult = searchresults[ i ];
    emailAddress += searchresult.getValue( 'email' );
  }
}
```

In order to increase performance, if you only need to access a specific subset of fields on a record, you should limit the returned objects to include only that subset. This can be accomplished by using the nlobjSearchColumn and nlobjSearchFilter objects in the nlapiSearchRecord function.

Filtering a Search

In order to filter a search, use the nlobjSearchFilter to specify filter criteria and pass the criteria as an argument to the nlapiSearchRecord function. In the following example only customer records that match the entityid of test1 are returned.

```
function filterCustomers()
{
  var filters = new Array();
  filters[0] = new nlobjSearchFilter( 'entityid', null, 'contains', 'test1', null
);
  var searchresults = nlapiSearchRecord('customer', 11, filters, null);
  var emailAddress = '';
  for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
  {
    var searchresult = searchresults[ i ];
  }
}
```

Note: If it is unclear which values you can filter by for a given filter variable, try performing a search that returns the value of a field as a result to see possible options.

Filtering Based on CheckBox Fields

When filtering search results for check box fields, use the **is** operator with **T** or **F** as the filter values. For example, in the following Dashboard script, all memorized Cash Sale transactions are returned.

```
function testPortlet(portlet) {
portlet.setTitle('Memorized Cash Sales');
var filters = new Array();
filters[0] = new nlobjSearchFilter('name', null, 'equalTo', '87', null);
filters[1] = new nlobjSearchFilter('memorized',null,'is','T',null);
var columns = new Array();
columns[0] = new nlobjSearchColumn('internalid');
columns[1] = new nlobjSearchColumn('memorized');
var searchresults = nlapiSearchRecord('cashsale', null, filters, columns);
for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
{
var searchResult = searchresults[i];
portlet.writeLine(i+": "+searchResult.getValue('internalid')+",
"+searchResult.getValue('memorized'),null,0);
}
}
```

Executing a Saved Search

When using the nlapiSearchRecord function to execute an existing saved search, note the following:

- The saved search must NOT include any group clauses.
- Only saved searches on record types currently supported by SuiteScript can be executed.
- Saved searches acted on by SuiteScript should be protected. If a saved search is edited after script deployment is complete, the execution of the script could fail (for example, if a saved search is edited to include a group clause). You can add security to saved search by defining access permissions in the search definition.

  Tip: You may want to include the script administrator in an email notification for any time a saved search included in the script is updated. Email notifications can be defined on the Alerts tab of the saved search definition.
- Saved search IDs are NOT displayed in the saved search page within NetSuite. To get saved search IDs, go to the saved search page and hover over the desired saved search record link. The ID is displayed as the id parameter in the URL of the browser status bar.

In the following code, a customer saved search is executed. 57 references the ID of the saved search.

```
function executeSavedSearch()
{
  var searchresults = nlapiSearchRecord('customer', 57, null, null);
  for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
  {
    var searchresult = searchresults[ i ];
  }
}
```

Searching on Custom Records

Example:

```
function searchWarranties()
{
var filters = new Array();
filters[0] = new nlobjSearchFilter( 'created', null, 'onOrAfter', 'daysAgo15' );
// -- return opportunity sales rep, customer custom field, and customer ID
var columns = new Array();
columns[0] = new nlobjSearchColumn( 'name' );
columns[1] = new nlobjSearchColumn( 'owner' );
columns[2] = new nlobjSearchColumn( 'custrecord_customer' );
columns[3] = new nlobjSearchColumn( 'custrecord_resolutiontime' );
var searchresults = nlapiSearchRecord( 'customrecord15', null, filters, columns );
for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
{
var searchresult = searchresults[ i ];
var record = searchresult.getId( );
var rectype = searchresult.getRecordType( );
var name = searchresult.getValue( 'name' );
var resolution = searchresult.getValue( 'custrecord_resolutiontime' );
var customer = searchresult.getValue( 'custrecord_customer' );
var customer_name = searchresult.getText( 'custrecord_customer' );
}
}
```

Error Codes

The following error codes can be returned by nlapiSearchRecord:

- SSS_INVALID_RECORD_TYPE
- SSS_TYPE_ARG_REQD
- SSS_INVALID_SRCH_ID
- SSS_INVALID_SRCH_FILTER
- SSS_INVALID_SRCH_FILTER_JOIN
- SSS_INVALID_SRCH_OPERATOR
- SSS_INVALID_SRCH_COL_NAME
- SSS_INVALID_SRCH_COL_JOIN

nlapiLoadRecord

The nlapiLoadRecord function can be used to load an existing record. You can then extract the desired information from the loaded record using the methods available on the returned nlobjRecord object.

Note: Only records currently supported in SuiteScript can be loaded. Refer to "Supported Records" on page 2 for a list of supported object types.

Arguments	Returns
type: record type id: record ID	nlobjRecord object

Loading a Record

In the following example, the search described in the nlapiSearchRecord section is performed but each search result object is loaded using the nlapiLoadRecord function. Then the getRecordType and getId nlobjRecord object methods are used to retrieve specific in formation about each record.

```
function executeSearch()
{
  var rec = '';
  var searchresults = nlapiSearchRecord( 'customer', null, null, null );
  for ( var i = 0; i < Math.min( 500, searchresults.length ); i++)
  {
    var record = nlapiLoadRecord(searchresults[i].getRecordType(),
      searchresults[i].getId() );
    rec = rec + record.getRecordType() ;
    rec = rec + '  -Record ID = '  +  record.getId();
  }
  nlapiSendEmail(312, 312, 'customerRecordLoaded', rec,  null);
}
```

Error Codes

The following error codes can be returned by nlapiLoadRecord:

- SSS_INVALID_RECORD_TYPE
- SSS_TYPE_ARG_REQD
- SSS_INVALID_INTERNAL_ID

- SSS_ID_ARG_REQD

nlapiCreateRecord

The nlapiCreateRecord function can be used to instantiate a new instance of a record. You can then use the methods available on the returned nlobjRecord object to populate the record with the desired information. The nlapiCreateRecord must be followed by the nlapiSubmitRecord function before the record is actually committed to the database.

Note: Values for all required fields **must** be provided or a newly instantiated record can NOT be submitted. Refer to the Server SuiteScript Reference Guide for tables that list fields and whether they are required for each supported record type. There may be additional required fields when custom forms or used. Also, note records can NOT be created as standalone records since these records are always associated with a record in the UI and do not make sense as standalone records. Similarly, Message records require an author and recipient to ensure that they are not created as standalone records.

Arguments	Returns
type: record type	nlapiRecord object

Creating a New Record

In the following example, the createTaskRecord() function causes a new task record to be created. This could be tied to an After Submit Function of a User Event and deployed to Opportunity records so that each time an Opportunity is created, a task is automatically created.

Note: You must use the nlapiSubmitRecord in conjunction with the nlapiCreateRecord in order for the new record to be committed.

```
function createTaskRecord()
{
  var taskTitle = 'Follow up regarding new Opportunity';
  var record = nlapiCreateRecord( 'task');
  record.setFieldValue( 'title', taskTitle);
  id = nlapiSubmitRecord(record, true);
}
```

Error Codes

The following error codes can be returned by nlapiCreateRecord:

- SSS_INVALID_RECORD_TYPE
- SSS_TYPE_ARG_REQD

nlapiSubmitRecord

The nlapiSumitRecord function can be used in conjunction with nlapiCreateRecord or nlapiLoadRecord in order to create or modify a record related to the current one.

Important: When using nlapiSubmitRecord in a User Event script it is possible that the related record modified or created by the script is committed to the database but the actual record initiating the script fails on save. To avoid this scenario, SuiteScripts that cause actions on records other than the current one should be set to run **after submit**.

Arguments	Returns
record: nlobjRecord object doSourcing: if TRUE, enables automatic field sourcing	record ID

Note: By default doSourcing is set to FALSE.

Submitting a Record

Expanding on the example Creating a New Record, the createTaskRecord() function now causes a new task record to be created and submitted. This could be tied to an After Submit Function of a User Event and deployed to Opportunity records so that each time an Opportunity is created, a task is automatically created.

```
function createTaskRecord()
{
  var taskTitle = 'Follow up regarding new Opportunity';
  var record = nlapiCreateRecord( 'task');
  record.setFieldValue( 'title', taskTitle);
  id = nlapiSubmitRecord(record, true);
}
```

Understanding Sourcing

When submitting a record, you can retain full control over the data that is written to the system by setting **doSourcing** to FALSE, or you can accept sourcing values from NetSuite by setting doSourcing to TRUE. When set to TRUE, fields normally dependent on values from parent fields are automatically pre-populated.

Some advantages to setting doSourcing to **TRUE** include:

- Reduces the number of fields that have to be filled out while retaining data integrity across fields
- Ensures that field values reflect what would normally be submitted when using the entering records via the UI.

Some advantages to setting doSourcing to **FALSE** include:

- You retain full control over the data that is written to the system
- Reduces overhead incurred — with soSourcing set to TRUE, all empty dependent fields on the record (including supported sublists) must be processed

Example

For example, in the UI when a customer is selected on an opportunity record, the leadsource, partner, salesrep, and any custom sourced fields are automatically populated.

If creating an opportunity using SuiteScript with doSourcing set to FALSE, the leadsource, partner, salesrep, and any custom sourced fields not specifically set by the SuiteScript code would be empty. doSourcing must be set to TRUE for these fields to automatically populate with values based on the value of the customer field.

Error Codes

The following error codes can be returned by nlapiSubmitRecord:

- SSS_INVALID_RECORD_OBJ
- SSS_RECORD_OBJ_REQD
- SSS_INVALID_SOURCE_ARG

nlapiDeleteRecord

The nlapiDeleteRecord function allows you to delete the defined record from your account.

Warning: Use caution when using the nlapiDeleteRecord function in SuiteScript scripts. Records deleted using the nlapiDeleteRecord function are **permanently** deleted from your NetSuite database!

Arguments	Returns
type: record type id: record ID	N/A

Deleting a Record

In the following code, the resultant record set from a customer saved search is deleted. Once the search is performed, methods on the nlobjSearchResult objects take the desired action. In this example, the getRecordType and getId methods are used to identify which records to delete.

```
function executeSavedSearch()
{
  var searchresults = nlapiSearchRecord('customer', 57, null, null);
  for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
  {
    var searchresult = searchresults[ i ];
    nlapiDeleteRecord(searchresults[i].getRecordType(), searchresults[i].getId();
  }
}
```

Error Codes

The following error codes can be returned by nlapiDeleteRecord:

- SSS_INVALID_RECORD_TYPE
- SSS_TYPE_ARG_REQD
- SSS_INVALID_INTERNAL_ID
- SSS_ID_ARG_REQD

nlapiTransformRecord

The nlapiTransform function can be used to transform an existing record from one type to another, returning a pre-populated nlapiRecord object. This is useful for automated order processing such as as creating item fulfillment transactions and invoices off of orders.

Arguments	Returns
type: record type id: internalID newType: recordtype	nlapiRecord object

Creating a Dependent Record

Some NetSuite record types can NOT be created as standalone records. They are always created from another record type because of relationships between the record types. The nlapiTransfromRecord API can be used to create these types of records.

For example, in the following script an Item Receipt record is created from a Purchase Order.

```
function trans()
{
  var fromrecord;
  var fromid;
  var torecord;
  var trecord;
  var qty;
  fromrecord = 'purchaseorder';
  fromid = 26 ; // Transform PO with ID = 26 ;
  torecord = 'itemreceipt';
  // Transaform a record with given id to a different record type.
  // Eg. From PO to Item Receipt
  // Get the object of the transformed record.
  trecord = nlapiTransformRecord(fromrecord, fromid, torecord,true);
  qty = trecord.getLineItemValue('item', 'quantity', 1 );
  trecord.setLineItemValue('item', 'quantity', 1, '2' );
  var idl = nlapiSubmitRecord(trecord, true);
  nlapiSendEmail(-5, -5, 'Transform Email' + 'Original Qty =
  ' + qty + ' ' + 'Record Created = ' + idl , null);
}
```

nlapiResolveURL

The nlapiResolveURL function can be used to create a URL on-the-fly by passing URL parameters from within your SuiteScript. For example, when creating a SuiteScript Dashboard script, you may want to create and display the record URLs for each record returned in a search.

When creating the URL, you can use either the RECORD reference as retrieved in a search result or a known TASKLINK. Each page in NetSuite has a unique Tasklink Id associated with it for a given record type. Refer to the Server SuiteScript for a list of available NetSuite Tasklinks.

Note: You can also discover the Tasklink for a page within NetSuite by viewing the HTML page source. Search for a string similar to the following, where LIST_SCRIPT refers to the TASKLINK.

```
onclick="nlPopupHelp('LIST_SCRIPT','help')
```

Arguments	Returns
identifier: RECORD|TASKLINK type: recordtype -or- tasklink id id: record ID editmode	url: String

Error Codes

The following error codes can be returned by nlapiResolveURL:

- SSS_INVALID_URL_CATEGORY
- SSS_CATEGORY_ARG_REQD
- SSS_INVALID_TASK_ID
- SSS_TASK_ID_REQD
- SSS_INVALID_INTERNAL_ID
- SSS_INVALID_EDITMODE_ARG

nlapiSetRedirectURL

The nlapiSetRedirectURL function can be used to customize the workflow of records within NetSuite. For a User Event SuiteScript script, you can use nlapiSetRedirectURL to send the user to a pre-defined NetSuite page on a specific event. For a Scheduled SuiteScript script, you can use nlapiSetRedirectURL to sequentially work on different records when certain conditions exist in those records. The redirect page must be a fully qualified NetSuite URL. You can NOT redirect a user to an external URL.

A record must first exist in NetSuite in order to redirect to it. If you want to redirect a user to a new record, then you must first create and submit the record before redirecting them. You must also ensure that any required fields for the new record are populated before submitting it.

Note: To return a record in edit mode, set the editmode argument to false.

Arguments	Returns
identifier: RECORD (support TASKLINK or SCRIPTLET) type: recordtype or tasklink id id: intenalID editmode: true if we should return the URL to an existing record in EDIT mode	N/A

Redirecting to Another Page

In the following example, a task record is created and populated with default values before it is redirected to. This could be deployed to an Opportunity record in a User Event script to cause the user to be redirected to a new Task record each time an Opportunity is entered.

```
function redirectTaskRecord()
{
  var taskTitle = 'New Opportunity';
  var record = nlapiCreateRecord( 'task');
  record.setFieldValue( 'title', taskTitle );
  id = nlapiSubmitRecord(record, true);
  nlapiSetRedirect( 'RECORD', 'task', id, false );
}
```

Error Codes

The following error codes can be returned by nlapiSetRedirect:

- SSS_INVALID_URL_CATEGORY
- SSS_CATEGORY_ARG_REQD
- SSS_INVALID_TASK_ID
- SSS_TASK_ID_REQD
- SSS_INVALID_INTERNAL_ID
- SSS_INVALID_EDITMODE_ARG

nlapiSendEmail

The nlapiSendEmail function can be used to send an email to an individual or group of individuals. This can be used for automatic email notifications of critical events or for message logging.

Error Codes

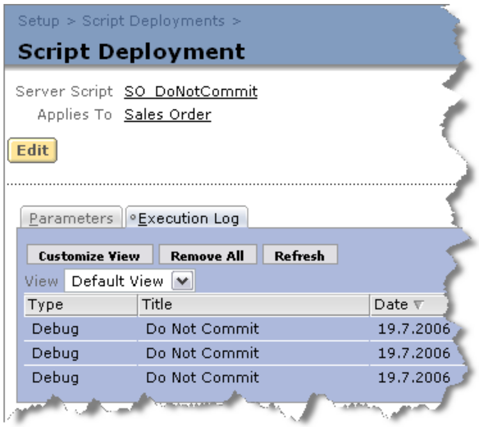
The following error codes can be returned by nlapiSendMail:

- SSS_AUTHOR_MUST_BE_EMPLOYEE
- SSS_AUTHOR_REQD
- SSS_INVALID_RECIPIENT_ID
- SSS_RECIPIENT_REQD
- SSS_INVALID_CC_EMAIL
- SSS_INVALID_BCC_EMAIL

nlapiLogExecution

The nlapiLogExecution function can be used to log a message, audit, or note against a SuiteScript script deployment. This is useful during the debugging of a script or for providing useful execution notes for auditing or tracking purposes.

Script execution logs are listed in the Execution Log tab of the Server Script record or the specific Script deployment record.



Server Script SO_DoNotCommit
Applies To Sales Order

View Default View

Type	Title	Date ▼
Debug	Do Not Commit	19.7.2006
Debug	Do Not Commit	19.7.2006
Debug	Do Not Commit	19.7.2006

Note: The Execution Log tab also lists notes returned by NetSuite such as error messages.

Arguments	Returns
type: log type (audit, debug, error, emergency) subject: log subject details: log details	N/A

nlapiGetContext

The nlapiGetContext function can be used to branch scripts depending on the context of the execution. For example, you may want the script to perform in one way when a form is accessed via the UI and another when the form is accessed via Web services.

Arguments	Returns
nlobjContext	nlobjScriptContext object: returns context information (metadata) about current user and script execution

Specifying Web Services Context

To cause a form to behave differently in Web Services versus the UI, you can do one of the following:

- Write context specific SuiteScript code and use the nlapiGetContext function to branch the code
- Disable SuiteScript in Web services

However, both Client and Server SuiteScripts are written to enforce customized business rules which may need to be enforced regardless of the mechanism by which a record is created or updated within NetSuite. This is particularly true for customers who deploy a NetFlex partner application and want to be sure their business rules are still respected. Since Client SuiteScript often has browser specific behavior that requires user action and can NOT automatically run during a Web Services call, NetSuite recommends that you disable Client SuiteScript and deploy Server SuiteScript for those business conditions that need to be enforced in all cases.

To specify that Server SuiteScript should never execute during a Web services call enable the **Disable Server-side Scripting** preference on the Web Services Preference page at Setup > Integration > Web Services.

Important: Only enable this preference when data submitted via Web services does NOT need to adhere to custom business logic and workflows that may be executed via Server SuiteScript.

nlapiGetNewRecord

Arguments	Returns
none	Returns an nlobjRecord representing the record data submitted to the system for a write operation

Objects

This section describes the list of objects that can be returned from a SuiteScript api call. Each object has various methods that can be performed against it once it is returned in the script.

For example, you could create a record using nlapiCreateRecord, use the setFieldValue functions to populate mandatory fields for a record and then use the nlapiSubmitRecord to save the record to your NetSuite account.

Note: The only constructable objects are: nlapiSearchColumn, and nlapiSearchFilter.

Object	Methods	Return Value	Description
nlobjRecord	getRecordType()	record type	
	getId()	record ID	
	setFieldValue(name, value)	field value	
	setFieldValues(name, values)	array of field values (only applicable for multiselects)	
	getFieldValue(name)	field value	
	getFieldValues(name)	array of field values (only applicable for multiselects)	
	setLineItemValue(group, name, line, value)		The group argument refers to the type of machine being referenced. Currently only Item and Expense machines are supported. For a list of field name values for each group argument, refer to the Server SuiteScript Reference Guide.

Object	Methods	Return Value	Description
	getLineItemValue(group, name, line)	line item value	The group argument refers to the type of machine being referenced. Currently only Item and Expense machines are supported. For a list of field name values for each group argument, refer to Server SuiteScript Reference Guide
	getLineItemCount(group)	number of lines in a machine	The group argument refers to the type of machine being referenced. Currently only Item and Expense machines are supported. The first line number on a machine is **1** (NOT 0).
nlobjSearchResult	getRecordType()	record type	
	getId()	record ID	
	getText(name, join)	return field text value (for dropdown fields)	
	getValue(name, join)	return field value	
	getAllFilters()	return an Array of nlapiSearchFilters	v11.x (requires labels)
	getAllColumns()	return an Array of nlapiSearchColumns	v11.x (requires labels)
nlobjSearchFilter	name, join, operator, value1, value2		When searching on Check Box fields, use the **is** operator with a **T** or **F** value to search for checked or unchecked fields respectively.
	getName()		
	getJoin()		
	getOperator()		
nlobjSearchColumn	name, join		
	getName()		
	getJoin()		
nlobjServerResponse	getCode()	HTTP response code	
	getBody()	response body as String	
	getHeader(name)	header value	
	getAllHeaders()	Associative Array of header-value pairs	
nlobjError	getCode()		
	getDetails()		
	getStackTrace()		
	getInternalID()		

Object	Methods	Return Value	Description
	getUserEvent()		
nlobjContext	getName()	user name	
	getUser()	user entity internalID	
	getRole()	user role internalID	
	getCompany()	user company internalID	
	getEmail()	user email address	
	getDepartment()	user department	
	getLocation()	user location	
	getExecutionContext	execution context: userinterface, webservices, offlineclient, smbxml	
nlobjPortlet	setTitle(title)	void	
	writeLine(text, url, indent)	void	

nlobjPortlet

To create Dashboard scripts, you set the portlet argument to specify that the action should only be performed for portlets. For example:

```
function myPortlet( portlet )
{
portlet.setTitle(‘Portlet Title’);
portlet.writeLine(‘This is my SuiteScript portlet’,null,1);
}
```

Chapter 4 Use Cases

Important: The following use case often depend on specific fields being available on the forms being used. If you are using custom forms, you should ensure that the fields required in the script are available in that form.

ERP Use Cases

The following ERP use cases are outlined in this section:

- Email Notification
- Adding Contact Information to a Sales Order
- Modifying Line Items on a Transaction

Email Notification

This **User Event** script causes an email notification to be sent when a Sales Order is entered with a discount level higher than a given value. The maximum allowed discount level is set as a **_maximumdiscountlevel** parameter of the specific deployment.

The script uses the parameters outlined in the following table. Values for these parameters must be defined for each deployment of the script. To define these parameters, go to the Parameters tab of the script definition and create a New Field for each parameter. These parameter fields are essentially custom fields. For detailed information on how to define custom fields, refer to the Customization Guide.

Note: In order to enhance readability, it is recommended that you use an underbar in the field ID definitions as shown here since each script field created is automatically prefaced with **custscript**.

Parameter Name	Field ID	Field Name
_maximumdiscountlevel	custscript_maximumdiscountlevel	Percent
_salesorderapproveremail	custscript_salesorderapproveremail	E-mail Address
_salesordername	custscript_salesordername	Free-Form Text
_salesorderapproveremail2	custscript_salesorderapproveremail2	E-mail Address

In the following script, edit the nlapSendEmail function to reflect a valid email in your account and valid recipients. The first argument (-5 in this sample) is the author of the email. The second argument (adminsToEmail) is a comma deliminated list of emails that can correspond to the internalID of for any entity in the system or any other valid email address.

The Script:

```
/* before Save trigger: test discount rate on new orders against cutoff rate and if
it exceeds it then change the status to Pending approval.
Also update memo */
```

```
function beforeSaveSalesOrder(type)
{
  var newRecord = nlapiGetNewRecord();
  var cutoffRate = custscript_maximumdiscountlevel;
  var discountRate = newRecord.getFieldValue('discountrate');
if ( type == 'Create' && discountRate != null && discountRate.length > 0  &&
cutoffRate != null && cutoffRate.length > 0 )
    {
    discountRate = Math.abs( parseFloat( discountRate ) );
    cutoffRate = Math.abs( parseFloat( cutoffRate ) );
    if ( discountRate > cutoffRate )
    {
      newRecord.setFieldValue('orderstatus','A' /* Pending Approval */);
      newRecord.setFieldValue('memo','Changed status to pending approval
      because discount exceeded '+custscript_maximumdiscountlevel );
    }
        else
    {
      newRecord.setFieldValue('orderstatus','B' /* Pending Fulfillment */);
      newRecord.setFieldValue('memo','Changed status to pending fulfillment
      since it did not exceed cutoff' );
    }
    }
}
/* after Save trigger: test discount rate on new orders against cutoff rate and if
it exceeds it then send an email to a predefined list of addresses. */
function afterSaveSalesOrder(type)
{
  var newRecord = nlapiGetNewRecord();
  var cutoffRate = custscript_maximumdiscountlevel;
  var discountRate = newRecord.getFieldValue('discountrate');
  if ( type == 'Create' && discountRate != null && discountRate.length > 0  &&
    cutoffRate != null && cutoffRate.length > 0 )
    {
    discountRate = Math.abs( parseFloat( discountRate ) );
    cutoffRate = Math.abs( parseFloat( cutoffRate ) );
    if ( discountRate > cutoffRate )
      sendDiscountWarningEmail( );
    }
}
/* Lookup customer name by internalId */
function queryCustomerName( customer )
{
  var filters = new Array();
  filters[0] = new nlobjSearchFilter( 'internalid', null, 'equalTo', customer,
    null );
  var columns = new Array();
  columns[0] = new nlobjSearchColumn( 'entityid' );

  var searchresults = nlapiSearchRecord( 'customer', null, filters, columns );
  var entityid = searchresults[ 0 ].getValue( 'entityid' );
  return entityid;
}
/* Lookup salesrep name by internalId */
function querySalesRepName( salesrep )
{
  var filters = new Array();
  filters[0] = new nlobjSearchFilter( 'internalid', null, 'equalTo', salesrep,
    null );
  var columns = new Array();
  columns[0] = new nlobjSearchColumn( 'entityid' );

  var searchresults = nlapiSearchRecord( 'contact', null, filters, columns );
  var entityid = searchresults[ 0 ].getValue( 'entityid' );
  return entityid;
}
/* Send pre-formatted email to predefined list of recipients */
function sendDiscountWarningEmail()
{
  var newRecord = nlapiGetNewRecord();
  var customerName = queryCustomerName(newRecord.getFieldValue('entity'));
  var salesrepName = querySalesRepName(newRecord.getFieldValue('salesrep'));
  var orderName = custscript_salesordername != null ? custscript_salesordername :
  'Sales Order';

  var str = salesrepName+' has entered a '+orderName+' for '+customerName+' that
  exceeds '+custscript_maximumdiscountlevel+'.\n\n';
  str += 'Your approval is required before it can be fulfilled.\n\n';
  str += 'Please log in to your NetSuite solution to approve the discount and
  order.\n';

  var adminsToEmail = custscript_salesorderapproveremail;
  if ( custscript_salesorderapproveremail2 != null )
    adminsToEmail += ';'+custscript_salesorderapproveremail2;

  /*nlapiSendEmail( -5 /* Joe Wolfe (admin)*/, adminsToEmail, 'Discount
  Warning', str ); */


/* logging a note in the execution log to indicate success */
nlapiLogExecution ('DEBUG', 'Discount Warning ' );

}
```

Adding Contact Information to a Sales Order

This **User Event** script sets a custom field on sales orders to the fax number of the customer's primary contact.

To use this script:

- Create a custom transaction body field.

  Custom Transaction Body fields can be defined at Setup > Customization > Transaction Body Fields > New. The custom field should have the following characteristics:

  - ID = **_contactfax**
  - Type = Phone Number
  - Display Subtab = Main
  - Applies To = Sale
  - Store Value = False (deselect)

  For detailed information on creating custom fields, refer to the Customization Guide.
- Set the script to execute **Before Load** on Sales Order records.

The Script:

```
function beforeLoadRecord(type)
{
  // only run this script for existing records
  if (type.toLowerCase() == 'create')
  return;
  // Get the current record
  var record = nlapiGetNewRecord();
  // Get the entity (customer) for this order
  var customer = record.getFieldValue("entity");
  // Execute a search to retrieve the primary contact's fax
  // Filter to custom id = the customer of the sales order gleaned above
  var filters = new Array();
  filters[0] = new nlobjSearchFilter("internalid",null,"anyof",customer);
  // Return the fax # from the primary contact joined search
  var results = new Array();
  results[0] = new nlobjSearchColumn("fax","contactprimary");
  // Execute the search
  var searchresults = nlapiSearchRecord( 'customer', null, filters, results );
  // Retrieve the result field
  var contactfax = searchresults[0].getValue("fax","contactprimary");
  // Set it into the record
  record.setFieldValue('custbody_contactfax',contactfax);
}
```

Modifying Line Items on a Transaction

This **User Event** script demonstrates how to modify each line item of a transaction. In this case, the Commit field on a Sales Order is set to **Do Not Commit**.

To use this script, set the script to execute Before Submit on a Sales Order record.

The Script:

```
function setLineField(type)
{
  var currentRecord;
  var lines;
  var i ;
  // Execute this only when Sales order is created or edited.
  if( ( type == 'create') || (type=='edit') )
  {
    currentRecord = nlapiGetNewRecord();
    // Get the number of line items before submit
    lines = currentRecord.getLineItemCount('item');
    for ( i = 1 ; i<= lines ; i++ )
    {
      currentRecord.setLineItemValue('item', 'commitinventory', i, '3');
    }
  }
}
```

CRM Use Cases

The following CRM use cases are outlined in this section:

- Creating Tasks on Opportunity Creation
- Redirecting to a New Task Record
- Notification of Closed Cases
- Defaulting Case Fields Based on Incoming Email Address
- High Open Balance Dashboard Portlet

Creating Tasks on Opportunity Creation

This **User Event** script causes a Task to be automatically created for the Sales rep when a new Opportunity record is created. If no Sales rep is set for the Opportunity at the time it is created, the task is assigned to a predefined default Sales Rep. A notification email is also sent to inform the Sales Rep of the new Opportunity.

To use this script:

- Change the salesrep variable to reflect an internal ID of a valid employee with the Sales Rep Role for your account.

  The Sales Role is set on the Human Resource tab of an Employee record and the Internal ID for employees can be seen on the Employee Record when the Show Internal IDs preference is enabled.
- Edit the author ID in the nlapSendEmail function to reflect a valid email in your account.
- Set the script to execute **After Submit** on Opportunity records.

The Script:

```
function taskCreator(type)
{
  var currentRecord ;
  var recordCreated;
  var salesrep;
  var customer;
  var tranNum;
  var emailText = 'A new task has been assigned to you';
  // create Task only when a new Opportunity is created
  if ( type == 'create' )
  {
    // Get the Current Record
    currentRecord = nlapiGetNewRecord();
    // Get the Sales Rep on opportunity.
    salesrep = currentRecord.getFieldValue('salesrep');
    // Get the Customer from the opportunity.
    customer = currentRecord.getFieldValue('entity');
    // Get the opportunity Id being created
    tranNum = currentRecord.getId();
    // Set default sales rep if none on opportunity
    if (salesrep == '' )
    {
      salesrep = 302;
    }
    // Create Task
     recordCreated = nlapiCreateRecord('task');
    // Set Title, Assigned to, Message and Company
    recordCreated.setFieldValue('title', 'Opportunity Follow-Up');
    recordCreated.setFieldValue('assigned', salesrep);
    recordCreated.setFieldValue('message', 'Follow up with your customer');
    recordCreated.setFieldValue('company', customer);
    recordCreated.setFieldValue('transaction', tranNum);
    nlapiSubmitRecord(recordCreated, true);
    nlapiSendEmail(-5, salesrep, 'Task Creating Email
      Notification', emailText,null);
  }
}
```

Redirecting to a New Task Record

This **User Event** script is similar to the Creating Tasks on Opportunity Creation use case, except that after creating the task record pre-populated with information from the Opportunity record, the user is actually redirected to the new task record in edit mode.

To use this script, set the script to execute after submit on Opportunity records.

The Script:

```
function taskRedirect(type)
{
  var currentRecord ;
  var recordCreated;
  var salesrep;
  var customer;
  var tranNum;
  var taskNum;
  // create Task only when a new Opportunity is created
  if ( type == 'create' )
  {
    // Get the Current Record
    currentRecord = nlapiGetNewRecord();
    // Get the Sales Rep on opportunity.
    salesrep = currentRecord.getFieldValue('salesrep');
    // Get the Customer from the opportunity.
    customer = currentRecord.getFieldValue('entity');
    // Get the opportunity Id being created
    tranNum = currentRecord.getId();
    // Create Task
    recordCreated = nlapiCreateRecord('task');
    // Set Task Title Title
    recordCreated.setFieldValue('title', 'Opportunity Follow-Up');
    // Set the sales rep
    recordCreated.setFieldValue('assigned', salesrep);
    // Set the company
    recordCreated.setFieldValue('company', customer);
    // Set the Customer
    recordCreated.setFieldValue('transaction', tranNum);
    // Save the Task record
    taskNum = nlapiSubmitRecord(recordCreated);
    // Redirect user to the task that was created in Edit mode.
    nlapiSetRedirectURL('RECORD', 'task', taskNum, true);
  }
}
```

Notification of Closed Cases

This **User Event** script causes an email to be sent whenever the status of a case is changed to Closed. The email is sent to the all emails listed in the E-mail(s) field on the Case record — which normally defaults to the email of the currently selected company.

Note: Any emails selected in the Email Employees multi-select field are not emailed this notification using this script.

To use this script:

- Set the script to execute after submit on Case records
- Edit the nlapiSendEmail author ID (-5) to a valid ID from your account

The Script:

```
function sendCaseEmail()
{
  var currentRecord;
  var status;
  var emailaddress;
  var emailMessage = 'Dear Customer - Your case has been resolved and is now
closed';
  var emailSubject = 'Case Status Notification';
  currentRecord= nlapiGetNewRecord();
  // Get the value of the Status
  status = currentRecord.getFieldValue('status');
  // check if status is closed
  if ( status == '5')
  {
    // Get email address from the case
    emailaddress = currentRecord.getFieldValue('email');
    if (emailaddress != '')
    {
      nlapiSendEmail( -5, emailaddress, emailSubject, emailMessage, null);
    }
  }
}
```

Defaulting Case Fields Based on Incoming Email Address

This **User Event** script causes a custom field, custevent_inboundmemo, on a case record to populate with a predetermined value based on the Inbound Email field when a case is submitted via email.

To use this script:

- Create a custom CRM field.

  Custom CRM fields can be defined at Setup > Customization > CRM Fields > New. The custom field should have the following characteristics:

  - ID = **_inboundmemo**
  - Type = Freeform Text
  - Display Subtab = Main
  - Applies To = Case
  - Store Value = False (deselect)

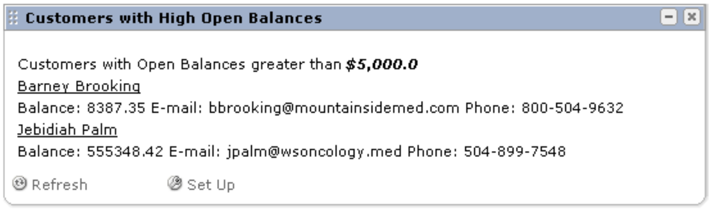
  For detailed information on creating custom fields, refer to the Customization Guide.
- Set the script to execute **Before Submit** on Case records.

The Script

```
function beforeSubmit(type)
{
  var record = nlapiGetNewRecord();
  if ( type == 'create' )
  {
    var inboundemail = record.getFieldValue('inboundemail');
    if ( inboundemail != null )
    {
      if ( inboundemail == 'info@rippleit.com' )
        record.setFieldValue('custevent_inboundmemo','INFO');
      else if ( inboundemail == 'demo@rippleit.com' )
        record.setFieldValue('custevent_inboundmemo','DEMO');
      else
        record.setFieldValue('custevent_inboundmemo','SUPPORT');
    }
  }
}
```

High Open Balance Dashboard Portlet

This **Dashboard** script searches for all Customers with an open balance greater than $5,000.00 for a specific sales rep. Once the script is defined, the SuiteScript Portlet preference must be enabled on the dashboard and this script selected to display on the dashboard.



Customers with High Open Balances

Customers with Open Balances greater than ***$5,000.0***
Barney Brooking
Balance: 8387.35 E-mail: bbrooking@mountainsidemed.com Phone: 800-504-9632
Jebidiah Palm
Balance: 555348.42 E-mail: jpalm@wsoncology.med Phone: 504-899-7548

The Script:

```
function highOpenBalance(portlet)
{
  portlet.setTitle( 'Customers with High Open Balances' );
  portlet.writeLine('Customers with Open Balances greater than
   <b><i>$5,000.0</i></b>',null,0);
  var filters = new Array();
  filters[0] = new nlobjSearchFilter( 'salesrep', null, 'anyOf', 164, null );
  filters[1] = new nlobjSearchFilter( 'balance', null, 'greaterThan',
    5000, null );
// -- return opportunity sales rep, customer custom field, and customer ID
  var columns = new Array();
  columns[0] = new nlobjSearchColumn( 'balance' );
  columns[1] = new nlobjSearchColumn( 'entityid' );
  columns[2] = new nlobjSearchColumn( 'email' );
  columns[3] = new nlobjSearchColumn( 'phone' );
  var searchresults = nlapiSearchRecord( 'customer', null, filters, columns );
  for ( var i = 0; searchresults != null && i < searchresults.length ; i++ )
  {
    var searchresult = searchresults[ i ];
    var record = searchresult.getId( );
    var rectype = searchresult.getRecordType( );
    var entityid = searchresult.getValue( 'entityid' );
    var balance = searchresult.getValue( 'balance' );
    var email = searchresult.getValue( 'email' );
    var phone = searchresult.getValue( 'phone' );
    portlet.writeLine( entityid , nlapiResolveURL('RECORD',
      'customer',record,null), 0 );
    portlet.writeLine( '<font style="font-size:8pt">Balance: '+balance +
      ' E-mail: '+email + ' Phone: '+phone+'</font>' , null, 2 );
  }
}
```

Errors

Note: In addition to the errors listed in this guide, you may also receive runtime errors generated by the javascript engine. These errors are standard javascript errors that are not NetSuite specific and should be handled the same as in any other javascript scripting environment.

Server SuiteScript Specific Errors

Error Code	Description
SSS_MISSING_SCRIPT_FUNCTION	The function defined for executing this script is missing: {function name}
SSS_AUTHOR_MUST_BE_EMPLOYEE	The author internal id or email must match an employee.
SSS_INVALID_BCC_EMAIL	One or more bcc emails are not valid.
SSS_INVALID_CC_EMAIL	One or more cc emails are not valid.
SSS_INVALID_HEADER	One or more headers are not valid.
SSS_INVALID_LOG_TYPE	Execution log type must be one of AUDIT, DEBUG, ERROR, or EMERGENCY.
SSS_INVALID_RECIPIENT_ID	Recipient internal id does not match an existing entity.
SSS_INVALID_RECORD_OBJ	The record is not a valid object.
SSS_INVALID_RECORD_TYPE	{function name}: type argument {type argument} is not a valid record or is not available in your account. Please see the documentation for a list of supported record types.
SSS_INVALID_SRCH_FILTER	An nlobjSearchFilter contains invalid search criteria: {search filter name}.
SSS_INVALID_SRCH_OPERATOR	An nlobjSearchFilter contains an invalid operator, or is not in proper syntax: {search filter name}.
SSS_INVALID_SRCH_FILTER_JOIN	An nlobjSearchFilter contains an invalid join ID, or is not in proper syntax: {search filter name}.
SSS_INVALID_SRCH_COLUMN_JOIN	An nlobjSearchColumn contains an invalid column join ID, or is not in proper syntax: {search column name}.
SSS_INVALID_SRCH_COLUMN	An nlobjSearchColumn contains an invalid column, or is not in proper syntax: {search column name}.
SSS_INVALID_SRCH_COLUMN_SUM	An nlobjSearchColumn contains an invalid column summary type, or is not in proper syntax: {search column name}.
SSS_INVALID_SCRIPTLET_ID	That scriptlet is invalid, disabled, or no longer exists.
SSS_INVALID_TASK_ID	The task ID: {task ID argument} is not valid. Please refer to the documentation for a list of supported task IDs.
SSS_INVALID_URL	The URL must be a fully qualified HTTP or HTTPS URL if it is referencing a non-Public.NetSuite resource.
SSS_INVALID_URL_CATEGORY	The URL category must be one of RECORD, TASKLINK or SCRIPTLET.

Error Code	Description
SSS_MISSING_REQD_ARGUMENT	{function name}: Missing a required argument: {argument name}
SSS_INVALID_TRANSFORM_TYPE	That type of record transformation is not allowed. Please see the documentation for a list of supported transformation types
SSS_USAGE_LIMIT_EXCEEDED	Usage limit exceeded

Error Status Codes

Error Code	Long Description or Message
ACCT_DISABLED	YOUR_ACCOUNT_HAS_BEEN_INACTIVATED_BY_AN_ADMINISTRATOR;
ACCT_DISABLED	ACCT_DISABLED;
ACCT_DISABLED	THIS_ACCOUNT_HAS_BEEN_DISABLED;
ACCT_DISABLED	THIS_COMPANY_HAS_BEEN_DISABLED_PLEASE_CONTACT_A_HREFMAILTO1_SUPPORTA_TO_REENABLE_THIS_COMPANY;
ACCT_NUMS_REQD_OR_DONT_MATCH	MISSING_ACCT__OR_ACCT_NUMBERS_DONT_MATCH;
ACCT_TEMP_DISABLED	YOU_HAVE_ENTERED_AN_INVALID_PASSWORD_ON_1_CONSECUTIVE_ATTEMPTS_ACCESS_TO_YOUR_ACCOUNT_HAS_BEEN_SUSPENDED_FOR_2_MINUTES_IF_YOU_HAVE_FORGOTTEN_YOUR_PASSWORD_PLEASE_CONTACT_CUSTOMER_SUPPORT;
ACCT_TEMP_UNAVAILABLE	TEMPORARILY_UNAVAILABLE;
ACCT_TEMP_UNAVAILABLE	WS_ACCOUNT_IS_CURRENTLY_UNAVAILABLE;
ACCT_TEMP_UNAVAILABLE	WE_ARE_CURRENTLY_PERFORMING_MAINTENANCE_ON_OUR_SYSTEM__PLEASE_TRY_AGAIN_SOON;
ACCT_TEMP_UNAVAILABLE	YOUR_ACCOUNT_IS_DISABLED_FOR_1_MORE_MINUTES_DUE_TO_2_CONSECUTIVE_FAILED_LOGIN_ATTEMPTS;
ACCT_TEMP_UNAVAILABLE	YOUR_ACCOUNT_IS_NOT_YET_READY_FOR_YOU_TO_LOG_IN__PLEASE_WAIT_AND_TRY_AGAIN;
ACCT_TEMP_UNAVAILABLE	YOUR_COMPANY_DATABASE_IS_OFFLINE;
ACCT_TEMP_UNAVAILABLE	YOUR_DATA_IS_STILL_BEING_LOADED__PLEASE_TRY_AGAIN_LATER_CONTACT_A_HREFAPPCRMSUPPORTNLBUGFORMNLTYPEBUGSPF31PROFESSIONAL_SERVICESA_IF_YOU_HAVE_QUESTIONS;
ACCT_REQD	ATTEMPTING_TO_ADJUST_PROVISIONING_FOR_A_CUSTOMER_WITHOUT_AN_EXISTING_ACCOUNT;
ACTIVE_TRANS_EXIST	THERE_ARE_ACTIVE_DIRECT_DEPOSIT_TRANSACTIONS_FOR_THIS_PAYCHECK;
ADMIN_ACCESS_REQ	AT_LEAST_ONE_ACTIVE_ADMINISTRATOR_FOR_EACH_ACCOUNT_MUST_HAVE_ACCESS;
ADMIN_ACCESS_REQ	AT_LEAST_ONE_ACTIVE_ADMINISTRATOR_FOR_THIS_ACCOUNT_MUST_HAVE_ACCESS;
ADMIN_ACCESS_REQD	ONLY_ADMINISTRATORS_MAY_ENTER_A_MEMORIZED_TRANSACTION_IN_A_CLOSED_PERIOD;
ALL_DATA_DELETE_REQD	YOU_MUST_FIRST_DELETE_ALL_THE_DATA_IN_YOUR_ACCOUNT_BEFORE_PERFORMING_THIS_ACTION;

Error Code	Long Description or Message
ALL_MTRX_SUBITMES_OPTNS_REQD	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;
ALREADY_IN_INVT	THE_FOLLOWING_1_NUMBER_IS_ALREADY_IN_INVENTORY_2;
ALREADY_IN_INVT	THE_FOLLOWING_1_NUMBERS_ARE_ALREADY_IN_INVENTORY_2;
APP_DEPRECATION_WARN	The record <record name> is deprecated. A new required field, <field name>, has been added to this record. We are setting this field to the default value of <default value>. The deprecation period will end on <end date>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.
APP_DEPRECATION_WARN	The record <record name> is deprecated. The <field name> field is now required. We are setting this field to the default value of <default value>. The deprecation period will end on <end date>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.
APP_DEPRECATION_WARN	The record <record name> is deprecated. The type of the <field name> field has changed to <new field type>. We are mapping the value of this field to <mapped value>. The deprecation period will end on <end date>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.
APP_DEPRECATION_WARN	The record <record name> is deprecated. The <field name> field has been renamed to <new field name>. We are mapping the old name to the new name. The deprecation period will end on <end date>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.
APP_DEPRECATION_WARN	The record <record name> is deprecated. The <field name> field has been removed. The deprecation period will end on <end date>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.
APP_DEPRECATION_WARN	The <record name> record is deprecated. It has been renamed to <new record name>. The deprecation period will end on <end date>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.
APP_DEPRECATION_WARN	The <record name> record is deprecated. It has been removed. The deprecation period will end on <end date>. The current version of your client will be obsolete at the end of the deprecation period, please upgrade to the latest version.
AREA_CODE_REQD	PLEASE_INCLUDE_AN_AREA_CODE_WITH_THE_PHONE_NUMBER;
ATTACH_SIZE_EXCEEDED	YOU_HAVE_EXCEEDED_THE_MAXIMUM_ATTACHMENTS_SIZE_OF_10_MB_PLEASE_REMOVE_ONE_OR_MORE_ATTACHMENTS_AND_TRY_AGAIN;
AUTO_NUM_UPDATE_DISALLWD	WE_CURRENTLY_DO_NOT_SUPPORT_AN_AUTOMATIC_NUMBERING_UPDATE_OF_MORE_THAN_1_2_RECORDS;
BASE_CRNCY_REQD	YOU_MAY_NOT_DELETE_YOU_BASE_CURRENCY;
BILL_PAY_STATUS_UNAVAILABLE	VIEW_ONLINE_BILL_PAY_STATUS_INFORMATION_IS_CURRENTLY_NOT_AVAILABLE_PLEASE_TRY_AGAIN_IN_A_FEW_MINUTES;
BILL_PAY_STATUS_UNAVAILABLE	VIEW_ONLINE_BILL_PAY_STATUS_IS_NOT_AVAILABLE_UNTIL_YOUR_BILL_PAY_REGISTRATION_IS_COMPLETE;
BILL_PMTS_MADE_FROM_ACCT_ONLY	YOUR_PAYMENT_HAS_BEEN_RECORDED_BUT_ONLINE_BILL_PAYMENTS_CAN_ONLY_BE_MADE_FROM_THE_ACCOUNT;
BILLING_ISSUES	YOUR_ACCOUNT_HAS_BEEN_LOCKED_DUE_TO_BILLING_ISSUES__YOU_MUST_CALL_YOUR_SALES_REPRESENTATIVE_AT_1800SMALLBIZ_FOR_FURTHER_ASSISTANCE;

Error Code	Long Description or Message
BILLING_ISSUES	YOUR_ACCOUNT_HAS_NOT_BEEN_FULLY_PAID_FOR_PLEASE_LOG_IN_TO_YOUR_ACCOUNT_AND_FOLLOW_THE_BILLING_PROCESS_OR_CONTACT_YOUR_ACCOUNT_MANAGER;
CALENDAR_PREFS_REQD	SET_UP_1_CALENDAR_PREFERENCES_FIRST;
CALENDAR_PREFS_REQD	SET_UP_CALENDAR_PREFERENCES_FIRST;
CAMPAGIN_ALREADY_EXECUTED	YOU_CANNOT_DELETE_EMAIL_CAMPAIGNS_THAT_HAVE_ALREADY_BEEN_EXECUTED;
CAMPAIGN_IN_USE	YOU_CANNOT_DELETE_A_CAMPAIGN_EVENT_THAT_ALREADY_HAS_ACTIVITY;
CAMPAIGN_SET_UP_REQD	THE_FOLLOWING_STEPS_NEED_TO_BE_PERFORMED_BEFORE_A_CAMPAIGN_CAN_BE_CREATED;
CANT_CREATE_NON_UNIQUE_RCRD	WS_ERROR_CREATING_NON_UNIQUE_RECORD;
CANT_DELETE_CHILD_RCRD_FOUND	WS_CHILD_RECORD_FOUND_IN_DELETE;
CANT_DELETE_CHILD_RCRDS_EXIST	THIS_RECORD_CAN_NOT_BE_DELETED_BECAUSE_IT_HAS_CHILD_RECORDS;
CANT_DELETE_CUST_PLACEHOLDER	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;
CANT_DELETE_LEGACY_CATEGORY	LEGACY_CATEGORY_CANNOT_BE_REMOVED;
CANT_DELETE_STATUS_TYPE	YOU_CANNOT_DELETE_THE_ONLY_STATUS_OF_TYPE_1;
CANT_ESTABLISH_LINK	UNABLE_TO_ESTABLISH_LINK_WITH_1;
CANT_SEND_EMAIL	UNABLE_TO_SEND_NOTIFICATION_EMAIL;
CANT_SEND_EMAIL	UNABLE_TO_SEND_NOTIFICATION_EMAIL_TO_SUPPORT_REP;
CANT_SET_CLOSE_DATE	UNABLE_TO_SET_EXPECTED_CLOSE_DATE_OF_PROSPECTLEAD_BASED_ON_CURRENT_ESTIMATESOPPORTUNITIES;
CANT_SET_STATUS	UNABLE_TO_SET_STATUS_OF_PROSPECTLEAD_BASED_ON_CURRENT_ESTIMATES;
CANT_SWITCH_ROLES_FROM_LOGIN	ROLE_SWITCHING_IS_NOT_ALLOWED_FROM_THIS_LOGIN;
CANT_UPDATE_ATTACHMENT	Updating an attachment in this context is not supported. You can explicitly update the file through an update operation on the File record.
CANT_UPDATE_STATUS_TYPE	YOU_CANNOT_UPDATE_THE_ONLY_STATUS_OF_TYPE_1;
CASE_ALREADY_ASSIGNED	THIS_CASE_CANNOT_BE_GRABBED_BECAUSE_IT_IS_ALREADY_ASSIGNED_TO_ANOTHER_REP__TO_VIEW_THE_CASE_GO_BACK_AND_CLICK_ON_THE_CASE_NUMBER;
CASE_NOT_GROUP_MEMBER	_1_THIS_CASE_RECORD_DOES_NOT_BELONG_TO_YOUR_GROUP;
CASH_SALE_EDIT_DISALLWD	THIS_CASH_SALE_CANNOT_BE_EDITED_WHILE_IT_HAS_AN_AUTOMATED_CLEARING_HOUSE_TRANSMISSION_IN_PROCESSTDTRTRTD_CLASSTEXTNBSPTDTRTRTD_CLASSTEXTNBSPTO_VIEW_THE_STATUS_OF_CASH_SALES_WITH_ACH_TRANSMISSIONS_GO_TO_TRANSACTIONS__VIEW_ELECTRONIC_FUNDS_TRANSFER_STATUS;
CC_ALREADY_SAVED	THAT_CREDIT_CARD_IS_ALREADY_SAVED__PLEASE_USE_THE_SAVED_CREDIT_CARD;

Error Code	Long Description or Message
CC_EMAIL_ADDRESS_REQD	PLEASE_GO_BACK_AND_PROVIDE_AN_EMAIL_ADDRESS_TO_CC_STORE_ORDERS_TO;
CC_NUM_REQD	PLEASE_PROVIDE_A_CREDIT_CARD_NUMBER;
COMMSSN_ALREADY_CALCLTD	A_PLANS_SCHEDULES_OR_PAST_SALES_REP_ASSIGNMENTS_CANNOT_BE_MODIFIED_ONCE_COMMISSIONS_AGAINST_THE_PLAN_HAVE_BEEN_CALCULATED;
COMMSSN_FEATURE_DISABLED	YOU_HAVE_NOT_ENABLED_THE_COMMISSIONS_FEATURE;
COMMSSN_PAYROLL_ITEM_REQD	A_COMMISSION_PAYROLL_ITEM_MUST_BE_ADDED_FOR_EACH_EMPLOYEE_TO_BE_PROCESSED_THROUGH_PAYROLL;
COMPANION_PROP_REQD	ERROR__ITEMS_DO_NOT_HAVE_COMPANION_PROPERTY_COLUMN_0;
COMPANY_RCRD_DELETED	THE_COMPANY_YOU_TRY_TO_ATTACH_THE_CONTACT_TO_HAS_BEEN_DELETED_OR_MERGED;
CONSLD_PRNT_AND_CHILD_DISALLWD	A_COMPANY_CAN_BE_A_CONSOLIDATED_CHILD_OR_A_CONSOLIDATED_PARENT_BUT_NOT_BOTH;
CONTACT_NOT_GROUP_MEMBR	_1_THIS_CONTACT_DOES_NOT_BELONG_TO_YOUR_GROUP;
COOKIES_DISABLED	YOU_HAVE_DISABLED_COOKIES_FROM_BEING_STORED_ON_YOUR_COMPUTER_OR_TURNED_OFF_PERSESSION_COOKIES_PLEASE_ENABLE_THIS_FEATURE_AND_TRY_AGAIN;
CRNCY_NOT_UPDATED	THE_FOLLOWING_CURRENCIES_WERE_NOT_UPDATED;
CRNCY_RCRD_DELETED	THIS_CURRENCY_RECORD_HAS_BEEN_DELETED__YOU_CAN_CREATE_A_NEW_CURRENCY_RECORD_AT_LISTS__CURRENCIES;
CSTM_FIELD_KEY_REQD	WS_CUSTOM_FIELD_KEY_IS_MISSING;
CSTM_FIELD_VALUE_REQD	WS_CUSTOM_FIELD_VALUE_IS_MISSING;
CUST_ACCESS_FEATURE_DISABLED	CANT_OPEN_STORE_FOR_1__THIS_COMPANY_DOES_NOT_HAVE_THE_BCUSTOMER_ACCESSB_FEATURE_ENABLED__THIS_FEATURE_IS_REQUIRED_FOR_CUSTOMERS_TO_BE_ABLE_TO_REGISTER_CHECK_OUT_AND_LOG_IN_TO_THE_SCORE;
CUST_LEAD_NOT_GROUP_MEMBR	_1_THIS_CUSTOMER_OR_LEAD_DOES_NOT_BELONG_TO_YOUR_GROUP;
DATA_MUST_BE_UNIQUE	THE_UPDATE_FAILED_BECAUSE_EVERY_ENTRY_IN_THIS_COLUMN_MUST_BE_UNIQUE;
DATA_REQD	WS_PROVIDE_A_PROPER_VALUE_FOR_THE_REQUIRED_FIELD;
DATE_EXPECTED	YOU_ENTERED_1_INTO_A_FIELD_WHERE_A_CALENDAR_DATE_WAS_EXPECTEDNPLEASE_GO_BACK_AND_CHANGE_THIS_VALUE_TO_THE_CORRECT_DATE;
DATE_PARAM_REQD	MISSING_DATE_PARAMETER;
DEFAULT_TYPE_DELETE_DISALLWD	YOU_CANNOT_DELETE_DEFAULT_TYPES;
DEPT_IN_USE	YOUR_CLASSES_CANNOT_BE_CONVERTED_TO_DEPARTMENTS_BECAUSE_YOUR_EXISTING_DEPARTMENT_RECORDS_ARE_REFERRED_TO_BY_TRANSACTIONS_OR_OTHER_RECORDS_THESE_DEPARTMENT_RECORDS_CANNOT_BE_OVERWRITTEN;
DISALLWD_IP_ADDRESS	THE_SPECIFIED_IP_ADDRESS_RULES_MUST_ALLOW_THE_LOGIN_OF_YOUR_CURRENT_IP_ADDRESS__YOUR_CURRENT_IP_ADDRESS_IS__1__FOR_INFORMATION_ON_ENTERING_IP_ADDRESS_RULES_CLICK_HELP_AT_THE_TOP_OF_THE_PAGE;
DISTRIB_REQD_ONE_DAY_BFORE	ALL_ITEMS_MUST_BE_DISTRIBUTED_AT_LEAST_ONE_DAY_BEFORE_THEY_MAY_BE_TRANSFERRED;

Error Code	Long Description or Message
DFRNT_SWAP_PRICE_LEVELS_REQD	PLEASE_SELECT_DIFFERENT_PRICE_LEVELS_TO_SWAP_PRICES;
DUE_DATE_BFORE_START_DATE	DUE_DATE_OCCURS_BEFORE_START_DATE;
DUP_PAYROLL_ITEM	THERE_IS_ALREADY_A_PAYROLL_ITEM_NAMED_1;
DUP_TRACKING_NUM	YOU_ENTERED_THE_FOLLOWING_TRACKING_NUMBER_TWICE;
DUPLICATE_INVENTORY_NUM	DUPLICATE_INVENTORY_NUMBER_FOUND_IN_ENTRY_1;
DUPLICATE_INVENTORY_NUM	DUPLICATE_INVENTORY_NUMBER_FOUND_ON_DIFFERENT_LINES_OF_TRANSACTION;
DUPLICATE_KEYS	WS_DUPLICATED_KEYS;
DUPLICATE_NAME_FOR_PRD	PLEASE_CHOOSE_A_DIFFERENT_PERIOD_NAME_1_IS_ALREADY_TAKEN;
DUPLICATE_NAME_FOR_ROLE	PLEASE_CHOOSE_A_DIFFERENT_ROLE_NAME_1_IS_ALREADY_TAKEN;
DUPLICATE_USER_NAME	A_USER_WITH_THIS_NAME_ALREADY_EXISTS;
EMAIL_ADDRS_REQD	PLEASE_ENTER_YOUR_EMAIL_ADDRESS;
EMAIL_ADDRS_REQD_TO_NOTIFY	PLEASE_ENTER_AN_EMAIL_ADDRESS_FOR_THIS_COMPANY_A_NOTIFICATION_EMAIL_WILL_BE_SENT_WHEN_THIS_CASE_RECORD_IS_SAVED;
EMAIL_ADDRS_REQD_TO_NOTIFY	THE_RECIPIENT_YOU_ARE_SENDING_THIS_EMAIL_TO_DOES_NOT_HAVE_AN_EMAIL_ADDRESS_PLEASE_ENTER_ONE_AND_TRY_AGAIN;
EMAIL_REQD	YOU_MUST_ENTER_A_VALID_EMAIL_ADDRESS_IN_ORDER_TO_EMAIL_THE_TRANSACTION;
EMPL_IN_USE	YOU_CANT_DELETE_THIS_EMPLOYEE_AS_COMMISSIONS_HAVE_BEEN_CALCULATED_FOR_THIS_EMPLOYEE;
EMPL_IN_USE	YOU_CANT_DELETE_THIS_EMPLOYEE_AS_IT_IS_OR_HAS_BEEN_REFERENCED_BY_OTHER_EMPLOYEES_AS_A_SUPERVISOR;
ERROR_PRCSSNG_TRANS	THERE_WERE_ERRORS_PROCESSING_THE_SELECTED_TRANSACTIONS__PLEASE_PROCESS_THEM_INDIVIDUALLY_FOR_MORE_INFORMATION;
EVENT_ID_NOT_FOUND	EVENT_ID_NOT_FOUND;
EXPIRED_SEARCH_CRITERIA	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;
FAX_NUM_REQD	YOU_MUST_ENTER_A_FAX_NUMBER;
FAX_NUM_REQD	YOU_MUST_ENTER_A_FAX_NUMBER_FOR_THIS_RECIPIENT_BEFORE_PERFORMING_A_FAX_MERGE_OPERATION;
FAX_NUM_REQD	YOU_MUST_ENTER_A_VALID_FAX_NUMBER_IN_ORDER_TO_FAX_THE_TRANSACTION;
FEATURE_DISABLED	THE_FEATURE_1_REQUIRED_TO_ACCESS_THIS_PAGE_IS_NOT_ENABLED_IN_THIS_ACCOUNT;
FEATURE_UNAVAILABLE	THE_1_FEATURE_IS_NOT_AVAILABLE_TO_YOUR_COMPANY;
FED_WITHHOLDING_REQD	YOUR_EMPLOYEE_RECORD_DOES_NOT_HAVE_CURRENT_FEDERAL_WITHHOLDING_INFORMATIONPPLEASE_CONTACT_YOUR_SUPERVISOR_TO_SET_UP_YOUR_RECORD_WITH_THE_APPROPRIATE_INFORMATIONP;
FIELD_CALL_DATE_REQD	MISSING_REQUIRED_FIELD_CALL_DATE;

Error Code	Long Description or Message
FIELD_DEFN_REQD	FIELD_DEFINITION_NOT_FOUND;
FIELD_REQD	YOU_MUST_FIRST_SELECT_A_FIELD;
FILE_NOT_FOUND	FILEMEDIA_ITEM_1_NOT_FOUND;
FILE_REQD	YOU_MUST_UPLOAD_A_FILE_BEFORE_CREATING_THIS_MEDIA_ITEM;
FILTER_BY_AMT_REQD	PLEASE_ENTER_AN_AMOUNT_TO_FILTER_BY;
FIRST_LAST_NAMES_REQD	PLEASE_ENTER_BOTH_YOUR_FIRST_AND_LAST_NAME;
FORM_RESUBMISSION_REQD	YOU_HAVE_LOGGED_IN_TO_A_DIFFERENT_USER_SINCE_YOU_NAVIGATED_TO_THIS_FORM__YOU_MUST_RESUBMIT_THIS_FORM_AS_THE_NEW_USER;
FULL_DISTRIB_REQD	YOU_MUST_FULLY_DISTRIBUTE_ALL_1_NUMBERS_FOR_1_NUMBERED_ITEMS;
FULL_USERS_REQD_TO_INTEGRATE	ONLY_FULL_1_USERS_CAN_INTEGRATE_WITH_PARTNERS;
GETALL_RCRD_TYPE_REQD	WS_MISSING_GETALL_RECORD_TYPE;
GROUP_DSNT_EXIST	THAT_GROUP_DOES_NOT_EXIST;
GROUP_REQD	YOU_CANNOT_PERFORM_A_BULK_MERGE_OPERATION_WITH_AN_EMPTY_GROUP;
ILLEGAL_PERIOD_STRUCTURE	ILLEGAL_PERIOD_STRUCTURE_DATE_1_IS_IN_MULTIPLE_PERIODS;
INACTIVE_RCRD_FOR_ROLE	THE_RECORD_FOR_THIS_ROLE_HAS_BEEN_MADE_INACTIVE;
INCRCT_ORD_INFO	THE_ORDER_CONTAINS_INCORRECT_INFORMATION_AND_WAS_NOT_PLACED;
INSUFCNT_NUM_PRDS_FOR_REV_REC	NOT_ENOUGH_ACCOUNTING_PERIODS_IN_RANGE_SPECIFIED_FOR_REVENUE_RECOGNITION;
INSUFCNT_OPEN_PRDS_FOR_REV_REC	NOT_ENOUGH_OPEN_ACCOUNTING_PERIODS_AVAILABLE_FOR_REVENUE_RECOGNITION;
INCRCT_ORD_INFO	THE_ORDER_CONTAINS_INCORRECT_INFORMATION_AND_WAS_NOT_PLACED;
INSUFFICIENT_CHARS_IN_SEARCH	GLOBAL_SEARCHES_MUST_CONTAIN_AT_LEAST_THREE_CHARACTERS_TO_PREVENT_EXCESSIVE_MATCHES;
INSUFFICIENT_PERMISSION	YOU_MUST_HAVE__TRANSACTIONS__FULFILL_SALES_ORDERS__PERMISSION_TO_FULFILL_SALES_ORDERS;
INSUFFICIENT_PERMISSION	YOU_MUST_HAVE_EITHER_TRANSACTIONS__INVOICE_OR_TRANSACTIONS__CASH_SALE_PERMISSION_TO_BILL_SALES_ORDERS;
INSUFFICIENT_PERMISSION	YOU_NEED_EMPLOYEE_ACCESS_IN_ORDER_TO_DELETE_THIS_RECORD;
INSUFFICIENT_PERMISSION	FOR_SECURITY_REASONS_ONLY_AN_ADMINISTRATOR_IS_ALLOWED_TO_EDIT_AN_ADMINISTRATOR_RECORD;
INSUFFICIENT_PERMISSION	THIS_ORDER_HAS_BEEN_PARTIALLY_OR_FULLY_PROCESSED_AND_MAY_NOT_BE_EDITED_BY_A_USER_WITHOUT_PERMISSION_TO_APPROVE_SALES_ORDERS;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_ACCESS_TO_THE_ACTIVITY_HISTORY_FOR_THAT_RECORD;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_ACCESS_TO_THE_MEDIA_ITEM_YOU_SELECTED;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_PERMISSION_TO_EMAIL_TRANSACTIONS;
INSUFFICIENT_PERMISSION	YOU_MUST_HAVE_EITHER_TRANSACTIONS__INVOICE_OR_TRANSACTIONS__CASH_SALE_PERMISSION_TO_FULFILL_SALES_ORDERS;
INSUFFICIENT_PERMISSION	YOU_ARE_NOT_ALLOWED_TO_APPROVE_YOUR_OWN_TRANSACTIONS;

Error Code	Long Description or Message
INSUFFICIENT_PERMISSION	_1_THE_RESTRICTIONS_ON_YOUR_ROLE_DENY_YOU_ACCESS_TO_THIS_RECORD;
INSUFFICIENT_PERMISSION	_1_THE_2_RESTRICTIONS_ON_YOUR_ROLE_DENY_YOU_ACCESS_TO_THIS_RECORD;
INSUFFICIENT_PERMISSION	_1_THE_2_RESTRICTIONS_ON_YOUR_ROLE_PREVENT_YOU_FROM_SEEING_THIS_RECORD;
INSUFFICIENT_PERMISSION	INSUFFICIENT_PRIVILEGES;
INSUFFICIENT_PERMISSION	YOUR_ROLE_DOES_NOT_HAVE_PERMISSION_TO_PROVISION_ACCOUNTS;
INSUFFICIENT_PERMISSION	USER_PERMISSION_LEVEL_COULD_NOT_BE_ESTABLISHED;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_PERMISSION_TO_PERFORM_THIS_OPERATION;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_PERMISSION_TO_VIEW_THIS_PAGE;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_PRIVILEGES_TO_PERFORM_THIS_ACTION;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_PRIVILEGES_TO_PERFORM_THIS_OPERATION;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_PRIVILEGES_TO_USE_THIS_PAGE;
INSUFFICIENT_PERMISSION	YOU_DO_NOT_HAVE_PRIVILEGES_TO_VIEW_THIS_PAGE;
INSUFFICIENT_PERMISSION	_1_THE_CUSTOMER_RESTRICTIONS_ON_YOUR_PARTNER_ROLE_PREVENT_YOU_FROM_SEEING_THIS_RECORD;
INSUFFICIENT_PERMISSION	_1_YOU_NEED_1_THE_2_PERMISSION_TO_ACCESS_THIS_PAGE_PLEASE_CONTACT_YOUR_ACCOUNT_ADMINISTRATOR;
INSUFFICIENT_PERMISSION	PERMISSION_ERROR_YOU_MAY_NOT_EDIT_THIS_ROLE;
INSUFFICIENT_PERMISSION	PERMISSION_VIOLATION_PARTNERS_DO_NOT_HAVE_ACCESS_TO_THIS_REPORT;
INSUFFICIENT_PERMISSION	PERMISSION_VIOLATION_PARTNERS_MAY_NOT_DELETE_SAVED_REPORTS;
INSUFFICIENT_PERMISSION	PERMISSION_VIOLATION_YOU_CANNOT_DELETE_SAVED_REPORTS_NOT_CREATED_BY_YOURSELF;
INSUFFICIENT_PERMISSION	WS_NO_PERMISSIONS_TO_SET_VALUE
INTEGER_REQD_FOR_QTY	QUANTITY_MUST_BE_AN_INTEGER_FOR_NUMBERED_ITEMS;
INVALID_ACCT	INVALID_LOGIN_NO_SUCH_ACCOUNT;
INVALID_ACCT	WS_INVALID_ACCOUNT_NUMBER;
INVALID_ADJUSTMENT_ACCT	THE_ACCOUNT_YOU_SELECTED_IN_ADJUSTMENT_ACCOUNT_IS_THE_SAME_AS_THE_ASSET_ACCOUNT_FOR_ONE_OF_THE_ITEMS_YOU_ARE_ADJUSTING_PLEASE_GO_BACK_AND_CHANGE_THE_ACCOUNT_NORMALLY_THE_ADJUSTMENT_ACCOUNT_WOULD_BE_AN_EXPENSE_ACCOUNT;
INVALID_AUTHORIZATION	CORRECT_AUTHORIZATION_EXCEPTION;
INVALID_BALANCE_RANGE	YOUR_BALANCE_IS_NOT_WITHIN_THE_ALLOWED_RANGE;
INVALID_BUG_NUM	BUG_NUMBER_SPECIFIED_WAS_INCORRECT__1_ISNT_A_NUMBER;
INVALID_CAMPAIGN_GROUP_SIZE	WHILE_IN_1_YOU_CAN_ONLY_SEND_2_EMAILS_PER_CAMPAIGN_EVENT__PLEASE_MODIFY_ONE_OR_MORE_OF_YOUR_TARGET_GROUPS_TO_CONTAIN_2_MEMBERS_OR_LESS_ALL_CAMPAIGN_EMAILS_WILL_BE_SENT_TO_YOUR_1_LOGIN_EMAIL_ADDRESS;
INVALID_CATGRY_TAX_AGENCY_REQ	A_VENDOR_MUST_BE_CREATED_IN_A_CATEOGRY_WITH_THE_TAX_AGENCY_CHECKBOX_CHECKED;

Error Code	Long Description or Message
INVALID_CC_EMAIL_ADDRESS	THE_EMAIL_ADDRESS_TO_CC_STORE_ORDERS_TO_IS_INVALID__PLEASE_GO_BACK_AND_CORRECT_IT;
INVALID_CC_NUM	CREDIT_CARD_NUMBER_IS_NOT_VALID__PLEASE_CHECK_THAT_ALL_DIGITS_WERE_ENTERED_CORRECTLY;
INVALID_CC_NUM	CREDIT_CARD_NUMBER_MUST_CONTAIN_ONLY_DIGITS;
INVALID_CC_NUM	CREDIT_CARD_NUMBERS_MUST_CONTAIN_BETWEEN_13_AND_16_DIGITS;
INVALID_CHARS_IN_EMAIL	EMAIL_ADDRESS_CONTAINS_INVALID_CHARACTERS;
INVALID_CHARS_IN_NAME	THE_FROM_NAME_FIELD_CANNOT_CONTAIN_APOSTROPHES_QUOTATION_MARKS_COMMAS_OR_GREATER_THAN_OR_LESS_THAN_SIGNS;
INVALID_CHARS_IN_NAME	YOU_CANNOT_USE_THE_COLON__CHARACTER_IN_THE_TOPIC_NAME__PLEASE_REMOVE_IT_;
INVALID_CHARS_IN_PARAM_FIELD	THE_ADDITIONAL_PARAMETERS_FIELD_CAN_NOT_CONTAIN_ANY_OF_THE_FOLLOWING_CHARACTERS___PLEASE_REMOVE_THEM_AND_TRY_AGAIN;
INVALID_CHARS_IN_URL	SPACES_ARE_NOT_ALLOWED_IN_THE_1URLPEXAMPLES_OF_A_VALID_1URL_AREBRBHTTPWWWMYDOMAINCOMIMAGEGIFBNBSPNBSPORNBSPNBSPBHTTPSONETWOORGUSERNAMETESTJPGB;
INVALID_CHARS_IN_URL	THE_URL_COMPONENT_YOU_HAVE_CHOSEN_CONTAINS_A_SPACE_OR_ONE_OF_THE_FOLLOWING_PROHIBITED_CHARACTER___PLEASE_REMOVE_THEM_AND_TRY_AGAIN;
INVALID_COSTING_METHOD	SERIAL_AND_LOT_ARE_THE_ONLY_COSTING_METHODS_THAT_MAY_BE_PASSED_AS_PARAMETERS_TO_THIS_PAGE;
INVALID_CREDENTIALS	NCEL_AUTHORIZATION_EXCEPTION;
INVALID_CSTM_FIELD_REF	WS_INVALID_CUSTOM_FIELD_REF;
INVALID_CSTM_RCRD_TYPE_KEY	WS_INVALID_CUSTOM_RECORD_TYPE_KEY;
INVALID_DATA	Error in record number <nskey>: Invalid field value <field>. Please refer to the XSD for enumerated list of valid field values."
INVALID_DATE_RANGE	INVALID_DATE_RANGENTHE_EVENT_1_START_TIME_2_MUST_BE_EARLIER_THAN_THE_END_TIME_3;
INVALID_DATE_RANGE	THE_DATE_RANGE_YOU_SPECIFIED_DOES_NOT_FALL_INSIDE_THAT_OF_THE_PARENT_PERIOD;
INVALID_EMAIL	EMAIL_ADDRESS_IS_NOT_VALID;
INVALID_EMAIL	YOU_HAVE_ENTERED_AN_INVALID_EMAIL_ADDRESS__PLEASE_TRY_AGAIN;
INVALID_EMAIL	YOUR_EMAIL_OR_CODE_IS_INVALID__PLEASE_TRY_AGAIN;
INVALID_END_DATE	EVENT_1_RECURRENCE_END_DATE_IS_INVALID;
INVALID_END_TIME	INVALID_END_TIME;
INVALID_FILE	VERIFY_THAT_YOU_HAVE_A_VALID_FILE_TO_UPLOAD;
INVALID_FILE_TYP	INVALID_FILE_TYPE_FILE_IS_NOT_A_COMPRESSED_ZIP_FILE;
INVALID_FILE_TYP	INVALID_FILE_TYPE_FILE_IS_NOT_A_COMPRESSEDZIP_FILE;
INVALID_FORMAT_IN_PARAM_FIELD	THE_ADDITIONAL_PARAMETERS_FIELD_IS_NOT_FORMATTED_CORRECTLY__PLEASE_REFORMAT_AND_TRY_AGAIN;

Error Code	Long Description or Message
INVALID_FROM_DATE	INVALID_FROM_DATE;
INVALID_FROM_TIME	INVALID_FROM_TIME;
INVALID_GROUP_SIZE	YOU_CANNOT_PERFORM_A_BULK_MERGE_OPERATION_WITH_A_GROUP_LARGER_THAN_500_RECORDS;
INVALID_GST_PST_AGENCIES	THE_GST_OR_PST_AGENCIES_ARE_NOT_VALID_PLEASE_REVIEW_YOUR_COMPANY_PREFERENCES;
INVALID_INVENTORY_NUM	INVALID_SET_OF_INVENTORY_NUMBERS_VALUES_MUST_BE_SEPARATED_BY_COMMAS_SPACES_TABS_OR_LINE_FEEDS;
INVALID_IP_ADDRESS_RULE	THE_FOLLOWING_IP_ADDRESS_RULE_IS_NOT_VALID_1;
INVALID_KEY_OR_REF	WS_INVALID_KEY;
INVALID_KEY_OR_REF	WS_INVALID_REFERENCE_KEY_1;
INVALID_KEY_OR_REF	WS_INVALID_REFERENCE_KEY_2;
INVALID_LOGIN	INVALID_LOGIN_ONLINE_FORM_ACCESS_IS_DISABLED;
INVALID_LOGIN	INVALID_LOGIN_SUPPLIER_ACCESS_IS_DISABLED;
INVALID_LOGIN_ATTEMPT	INVALID_LOGIN_ATTEMPT;
INVALID_LOGIN_CREDENTIALS	A_PROBLEM_OCCURED_VERIFYING_THE_PRESENTED_EMAIL_ADDRESS_PASSWORD_ROLENAME_OR_ACCOUNT_NUMBER_PLEASE_VERIFY_THESE_PIECES_OF_INFORMATION_AND_TRY_AGAIN;
INVALID_LOGIN_CREDENTIALS	YOU_HAVE_ENTERED_AN_INVALID_EMAIL_ADDRESS_OR_ACCOUNT_NUMBER_PLEASE_TRY_AGAIN;
INVALID_LOGIN_CREDENTIALS	YOU_HAVE_ENTERED_AN_INVALID_EMAIL_ADDRESS_OR_PASSWORD_PLEASE_TRY_AGAIN;
INVALID_LOGIN_CREDENTIALS	YOU_HAVE_ENTERED_AN_INVALID_LOGIN_PASSWORD__PLEASE_TRY_AGAIN;
INVALID_LOGIN_CREDENTIALS	YOU_HAVE_ENTERED_AN_INVALID_PASSWORD__PLEASE_TRY_AGAIN;
INVALID_LOGIN_CREDENTIALS	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.
INVALID_LOGIN_IP	INVALID_LOGIN__IP_ADDRESS_DOES_NOT_MATCH_ANY_OF_THE_IPADDRESS_RULES_SPECIFIED_FOR_THIS_ENTITY;
INVALID_LOT_NUM_FORMAT	LOT_NUMBERS_MUST_BE_ENTERED_USING_THIS_FORMAT;
INVALID_MARKUP_DISCOUNT	MARKUPDISCOUNT__MUST_BE_BETWEEN_999_AND_999;
INVALID_MEMRZD_TRANS	A_MEMORIZED_TRANSACTION_MAY_NOT_CONTAIN_ANY_SERIAL_OR_LOT_NUMBERS;
INVALID_NUMBER	INVALID_INTEGER_1;
INVALID_NUMBER	INVALID_NUMBER_1;
INVALID_ORD_STATUS	THIS_ORDER_HAS_BEEN_PARTIALLY_OR_FULLY_PROCESSED_AND_MAY_NOT_BE_RESET_TO_PENDING_APPROVAL;
INVALID_POST	INVALID_POST;
INVALID_PST_TAX_VALUE	PST_TAX_VALUE_IS_NOT_A_VALID_NUMBER_1;
INVALID_PSWD	YOUVE_USED_THAT_PASSWORD_BEFORE__PLEASE_CHOOSE_A_NEW_PASSWORD;
INVALID_PSWD	PASSWORD_MUST_BE_AT_LEAST_6_CHARACTERS_LONG;

Error Code	Long Description or Message
INVALID_PSWD	PASSWORD_MUST_BE_AT_LEAST_6_CHARACTERS_LONG_AND_CONTAIN_AT_LEAST_ONE_NUMBER_OR_SPECIAL_CHARACTER;
INVALID_PSWD	PASSWORD_MUST_CONTAIN_AT_LEAST_ONE_LETTER_AZ;
INVALID_PSWD	PASSWORD_MUST_CONTAIN_AT_LEAST_ONE_NUMBER_OR_SPECIAL_CHARACTER;
INVALID_PSWD	PASSWORD_PROVIDED_DOES_NOT_MATCH_SYSTEM_PASSWORD;
INVALID_PSWD	YOUR_PASSWORD_MUST_BE_AT_LEAST_6_CHARACTERS;
INVALID_PSWD	THE_CURRENT_PASSWORD_YOU_SUPPLIED_IS_INCORRECT;
INVALID_PSWD	YOUR_NEW_PASSWORD_MUST_BE_AT_LEAST_6_CHARACTERS_CONTAIN_AT_LEAST_ONE_NONLETTER_AND_BE_SUBSTANTIALLY_DIFFERENT_FROM_THE_CURRENT_PASSWORD;
INVALID_PSWD_ILLEGAL_CHAR	PASSWORD_CONTAINS_AN_ILLEGAL_CHARACTER;
INVALID_QUANTITY	SERIAL_AND_LOT_NUMBER_QUANTITIES_MUST_BE_POSITIVE;
INVALID_RCRD_TYPE	WS_INVALID_RECORD_TYPE;
INVALID_REFFERER_EMAIL	THE_REFFERER_EMAIL_ADDRESS_YOU_HAVE_ENTERED_IS_NOT_VALID__PLEASE_TRY_AGAIN;
INVALID_ROLE	WS_ROLE_IS_INVALID;
INVALID_ROLE	YOU_HAVE_ENTERED_AN_INVALID_ROLE_ID_PLEASE_TRY_AGAIN;
INVALID_ROLE_FOR_EVENT	YOU_SEEM_TO_HAVE_BEEN_INVITED_TO_THIS_EVENT_IN_A_DIFFERENT_ROLE__PLEASE_CHANGE_YOUR_ROLE_TO_VIEW_THE_EVENT;
INVALID_RQST_CONTACTS_EXIST	IT_HAS_ASSOCIATED_PRIMARY_CONTACTS;
INVALID_RQST_PARENT_REQD	IT_HAS_ASSOCIATED_CONTACT_RECORDS_THAT_WOULD_BE_LEFT_WITH_NO_PARENT_COMPANY;
INVALID_RQST_SBCUST_JOBS_EXIST	IT_HAS_ASSOCIATED_SUBCUSTOMERS_OR_JOBS;
INVALID_SCHDUL_FORMAT	TO_CREATE_A_VALID_SCHEDULE_PLEASE_ENTER_THE_BRACKET_VALUES_IN_ASCENDING_ORDERS_WITHOUT_GAPS;
INVALID_SEARCH	THAT_SEARCH_OR_MASS_UPDATE_DOES_NOT_EXIST;
INVALID_SEARCH_CRITERIA	GLOBAL_SEARCH_SUPPORTS_AT_MOST_THREE_KEYWORDS_AND_REQUIRES_AT_LEAST_ONE_KEYWORDS_ARE_COMPOSED_OF_ONLY_LETTERS_DIGITS_AND_DASHES;
INVALID_SEARCH_FIELD_OBJ	WS_ERROR_INVALID_SEARCH_CUSTOMFIELD_OBJECT;
INVALID_SEARCH_FIELD_OBJ	WS_INVALID_SEARCH_FIELD_OBJECT;
INVALID_SEARCH_MORE	WS_INVALID_SEARCHMORE_OPERATION;
INVALID_SEARCH_OPERATOR	WS_INVALID_SEARCH_OPERATOR;
INVALID_SEARCH_PAGE_INDEX	WS_INVALID_SEARCH_PAGE_INDEX;
INVALID_SEARCH_PAGE_SIZE	WS_INVALID_SEARCH_PAGE_SIZE;
INVALID_SEARCH_VALUE	WS_MISSING_SEARCH_VALUE;
INVALID_SEARCH_VALUE	WS_MISSING_SEARCH_VALUES;

Error Code	Long Description or Message
INVALID_SERIAL_OR_LOT_NUMBER	SERIAL_AND_LOT_NUMBERS_MAY_NOT_CONTAIN_THE_1_CHARACTER;
INVALID_STATE	SIGNUP_PROSPECT_STATE_1_IS_INVALID;
INVALID_SUPERVISOR	YOU_CANT_INSERT_THIS_EMPLOYEE_RECORD_AS_IT_WOULD_CREATE_A_LOOP_IN_THE_SUPERVISOR_HIERARCHY;
INVALID_SUPERVISOR	EMPLOYEES_CAN_NOT_BE_THEIR_OWN_SUPERVISOR;
INVALID_TAX_VALUE	GST_AND_PST_AMOUNT_CANNOT_BE_NEGATIVE;
INVALID_TAX_VALUE	GST_TAX_VALUE_IS_NOT_A_VALID_NUMBER_1;
INVALID_TIME	_1_IS_NOT_A_VALID_TIME
INVALID_TO_DATE	INVALID_TO_DATE;
INVALID_TRACKING_NUM	YOU_HAVE_ENTERED_A_TRACKING_NUMBER_THAT_EXCEEDS_THE_MAXIMUM_SIZE_OF_1_CHARACTERS;
INVALID_TRANS	THIS_TRANSACTION_IS_NOT_VALID;
INVALID_TRANSACTIO_DATE	THERE_ARE_NO_ACCOUNTING_PERIODS_THAT_COVER_THIS_TRANSACTION_DATE;
INVALID_TRANSACTION_DATE	TRANSACTION_DATE_1_IS_NOT_VALID;
INVALID_URL	PLEASE_BEGIN_THE_1_URL_WITH_BHTTPBNBSPNBSPORNBSPNBSPBHTTPSBPEXAMPLES_OF_A_VALID_1URL_AREBRBHTTPWWWMYDOMAINCOMIMAGEGIFBNBSPNBSPORNBSPNBSPBHTTPSONETWOORGUSERNAMETESTJPGB;
INVALID_VAT_AMOUNT	VAT_AMOUNT_CANNOT_BE_NEGATIVE;
INVALID_WS_VERSION	MESSAGE_CONTAINS_A_WEB_SERVICES_VERSION_THAT_DOES_NOT_EXIST
INVALID_YEAR_FORMAT	ILLEGAL_YEAR_FORMAT_OR_VALUE_EXAMPLES_1999_2000_2001_ETC;
INVALID_ZIP_FILE	INVALID_ARCHIVE_ZIP_FILE_MUST_CONTAIN_AT_LEAST_ONE_FILE;
INVENTORY_NUM_DISALLWD	INVENTORY_NUMBERS_ARE_ONLY_ALLOWED_ON_ITEMS_WITH_SERIAL_NUMBERED_OR_LOT_NUMBERED_ITEMS;
ITEM_ACCT_REQD	ONE_OF_THE_ITEMS_ON_THIS_TRANSACTION_HAS_AN_AMOUNT_BUT_NO_ACCOUNT__PLEASE_FIX_THE_ITEM_AND_RESUBMIT_THE_TRANSACTION;
ITEM_ACCT_REQD	ONE_OF_THE_ITEMS_ON_THIS_TRANSACTION_HAS_AN_AMOUNT_BUT_NO_ACCOUNT__PLEASE_FIX_THE_ITEM_AND_RESUBMIT_THE_TRANSACTION__IT_MIGHT_BE_THAT_YOU_HAVE_RECENTLY_ELECTED_TO_CHARGE_FOR_SHIPPING_AND_HAVE_NOT_ASSIGNED_AN_ACCOUNT_TO_THE_SHIPPING_ITEM_THAT_IS_INCLUDED_IN_THIS_TRANSACTION;
ITEM_ACCT_REQD	YOU_MUST_SPECIFY_ASSET_AND_COGS_ACCOUNTS_FOR_THIS_INVENTORY_ITEM;
ITEM_COUNT_MISMATCH	COGSCORRECTION_2_MEANS_OF_CALCULATING_THE_ITEM_COUNT_DO_NOT_MATCH_FOR_ITEM_1_VS_2;
ITEM_COUNT_MISMATCH	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;
ITEM_IS_UNAVAILABLE	ITEM_IS_UNAVAILABLE;
ITEM_NAME_MUST_BE_UNIQUE	AN_ITEM_WITH_THAT_NAME_ALREADY_EXISTS_PLEASE_CHOOSE_ANOTHER_NAME;
ITEM_PARAM_REQD_IN_URL	ERROR__ITEM_PARAMETER_IDNNN_WAS_NOT_PROVIDED_ON_THE_URL;

Error Code	Long Description or Message
LIST_KEY_REQD	WS_MISSING_LIST_KEY;
LOCATION_REQD	YOU_MUST_SPECIFY_A_LOCATION_IN_ORDER_TO_USE_1_NUMBERS_WHEN_MULTILOCATION_INVENTORY_IS_ENABLED;
LOCATIONS_IN_USE	YOUR_CLASSES_CANNOT_BE_CONVERTED_TO_LOCATIONS_BECAUSE_YOUR_EXISTING_LOCATION_RECORDS_ARE_REFERRED_TO_BY_TRANSACTIONS_OR_OTHER_RECORDS;
LOCATIONS_SETUP_REQD	YOU_MUST_FIRST_DEFINE_LOCATIONS_LISTSLOCATIONSNEW_BEFORE_YOU_CAN_DISTRIBUTE_INVENTORY;
LOCATIONS_SETUP_REQD	YOU_MUST_FIRST_DEFINE_LOCATIONS_LISTSLOCATIONSNEW_BEFORE_YOU_CAN_TRANSFER_INVENTORY;
LOGIN_DISABLED	LOGIN_ACCESS_HAS_BEEN_DISABLED_FOR_THIS_ROLE;
LOGIN_DISABLED	YOUR_ACCESS_TO_1_HAS_BEEN_DEACTIVATED_PLEASE_CONTACT_THE_COMPANYS_ADMINISTRATOR_TO_REACTIVATE_YOUR_ACCESS;
LOGIN_DISABLED	YOUR_ACCESS_TO_THIS_ACCOUNT_HAS_BEEN_REMOVED_OR_DISABLED_PLEASE_CONTACT_THE_ACCOUNT_ADMINSTRATOR;
LOGIN_DISABLED	INVALID_LOGIN_CUSTOMER_ACCESS_IS_DISABLED;
LOGIN_DISABLED_PARTNER_CTR	DISABLED_LOGIN_ADVANCED_PARTNER_CENTER_ACCESS_HAS_BEEN_DISABLED_BY_THE_ACCOUNT_ADMINISTRATOR;
LOGIN_DISABLED_PARTNER_CTR	DISABLED_LOGIN_STANDARD_PARTNER_CENTER_ACCESS_HAS_BEEN_DISABLED_BY_THE_ACCOUNT_ADMINISTRATOR;
LOGIN_EMAIL_REQD	INVALID_LOGIN_YOU_MUST_PROVIDE_AN_EMAIL_ADDRESS;
LOGIN_NAME_AND_PSWD_REQD	PLEASE_ENTER_BOTH_A_USER_NAME_AND_A_PASSWORD;
MACHN_LIST_KEY_NAMES_REQD	WS_MISSING_MACHINE_LIST_KEY_NAMES;
MANDATORY_PRD_TYPE_REQD	PLEASE_SELECT_THE_MANDATORY_PERIOD_TYPE;
MATRIX_INFO_TEMP_LOST	MATRIX_ITEM_INFORMATION_WAS_LOST_THIS_WAS_PROBABLY_DUE_TO_A_TRANSIENT_CONDITION_LIKE_A_SERVER_REBOOT_PLEASE_TRY_AGAIN;
MERGE_OPERATION_DISALLWD	YOU_CANNOT_PERFORM_MERGE_OPERATIONS_ON_RECORDS_THAT_BELONG_TO_YOUR_GROUP;
EXCEEDED_MAX_FIELD_LENGTH	
MAX_RCRDS_EXCEEDED	WS_EXCEEDED_MAX_RECORDS;
MEDIA_FILE_INVALID_JSCRIPT	MEDIA_FILE_WAS_OF_TYPE_JAVASCRIPT_AND_WOULD_NOT_COMPILE__ERROR_ON_LINE;
MEDIA_NOT_FOUND	MEDIA_ITEM_NOT_FOUND_1;
MEDIA_NOT_INITIALIZED	MEDIA_ITEM_CANNOT_BE_INITIALIZED;
MISMATCHED_SEARCH_PARENTHESIS	SEARCH_ERROR_PARENTHESES_ARE_UNBALANCED;
MISSING_ACCT_PRD	THERE_ARE_NOT_ENOUGH_PERIODS_SPECIFIED_TO_COMPLETELY_RECOGNIZE_THIS_TRANSACTION;
MISSNG_ACCT_PRD	UNABLE_TO_FIND_AN_ACCOUNTING_PERIOD_FOR_THE_ALLOCATION_DATE;
MISSNG_REV_REC_RCRD	UNABLE_TO_LOCATE_REVENUE_RECOGNITION_RECORDS;

Error Code	Long Description or Message
MISSNG_SO_REV_REC_PARAMS	UNABLE_TO_GET_REVENUE_RECOGNITION_PARAMETERS_FROM_ORIGINATING_SALES_ORDER;
MISSNG_SO_START_END_DATES	UNABLE_TO_ACQUIRE_START_AND_END_DATE_FROM_SALES_ORDER;
MLTPLE_TAX_LINES_DISALLWD	MULTIPLE_TAX_LINES_FOR_LINE_ITEM_IN_TRANSACTION;
MORIZATION_JE_ALERT	MORIZATION_JOURNAL_ENTRY_ALERT;
MORIZED_TRAN_CANNOT_DELETE	MORIZED_TRAN_CANNOT_DELETE;
MORIZED_TRAN_FAILURE_REMEDY	MORIZED_TRAN_FAILURE_REMEDY;
MORIZED_TRAN_WITH_SAME_NAME	MORIZED_TRAN_WITH_SAME_NAME;
MSNG_FIELD_OWRTE_MUST_BE_TRUE	MISSINGFIELDOVERWRITE_ATTRIBUTE_REQUIRED_AS_TRUE;
MULTISELECT_TYPE_REQD	WS_NO_MULTISELECT_TYPE_IS_DEFINED;
NAME_ALREADY_IN_USE	A_MASS_UPDATE_HAS_ALREADY_BEEN_SAVED_WITH_THAT_NAME_PLEASE_USE_A_DIFFERENT_NAME;
NAME_ALREADY_IN_USE	A_SEARCH_HAS_ALREADY_BEEN_SAVED_WITH_THAT_NAME_PLEASE_USE_A_DIFFERENT_NAME;
NAME_NOT_AUTHORIZED	RT_NAME_NOT_AUTHORIZED_EXCEPTION;
NAME_TYPE_FLDR_FIELDS_REQD	MISSING_REQUIRED_FIELDS__NAME_TYPE_AND_FOLDER;
NEGATIVE_TAX_RATE_DISALLWD	A_TAX_RATE_CANNOT_BE_NEGATIVE;
NO_DATA_FOUND	NO_DATA_WAS_FOUND;
NO_EXPENSES_FOR_PRD	THE_ALLOCATION_SOURCES_OR_DESTINATIONS_DID_NOT_HAVE_ANY_EXPENSES_ASSOCIATED_WITH_THEM_FOR_THE_SELECTED_PERIOD;
NO_MTRX_ITEMS_TO_UPDATE	THERE_ARE_NO_MATRIX_SUBITEMS_TO_UPDATE;
NO_ORD_SHPMNT	THERE_IS_NO_SHIPMENT_ON_THAT_ORDER;
NO_RCRD_FOR_USER	THERE_IS_NO_RECORD_FOR_THIS_USER_IN_THE_COMPANYS_ENTITY_TABLE_EMAILLOGINSEMAIL1_KENTITY2;
NO_SCHDUL_APPLIED	THERE_WERE_NO_SCHEDULES_THAT_NEED_TO_APPLIED_TO_THE_GIVEN_PERIOD;
NO_SCHDUL_APPLIED	THERE_WERE_NO_SCHEDULES_THAT_NEED_TO_BE_APPLIED_TO_THE_INPUT_ACCOUNTING_PERIOD;
NONZERO_AMT_REQD	YOU_DID_NOT_ENTER_NONZERO_AMOUNTS_FOR_ANY_ACCOUNTS;
NOT_IN_INVT	YOU_MAY_NOT_DISTRIBUTE_1_NUMBERS_THAT_ARE_NOT_CURRENTLY_IN_INVENTORY__YOU_ATTEMPTED_TO_DISTRIBUTE_THE_FOLLOWING_1_NUMBERS_THAT_WERE_NOT_IN_INVENTORY_2;
NONMATCHING_EMAILS	EMAIL_ADDRESSES_DONT_MATCH;
NULL_CHECK_NUMBER	NULL_CHECK_NUMBER;
NUM_ALREADY_USED	ECK_NUMBER_X_IS_ALREADY_USED;

Error Code	Long Description or Message
NUM_ITEMS_GRTR_THAN_QTY	NUMBER_OF_ITEM_NUMBERS_ENTERED_1_IS_GREATER_THAN_THE_ITEM_QUANTITY_2;
NUM_ITEMS_NOT_EQUAL_TO_QTY	NUMBER_OF_ITEM_NUMBERS_ENTERED_1_IS_NOT_EQUAL_TO_THE_ITEM_QUANTITY_2;
NUM_REQD_FOR_FIRST_LABEL	NO_NUMBER_WAS_SPECIFIED_FOR_THE_FIRST_LABEL;
ONE_ADMIN_REQD_PER_ACCT	THIS_OPERATION_WOULD_LEAVE_YOUR_ACCOUNT_WITHOUT_AN_ACTIVE_ADMINISTRATOR__IN_ORDER_TO_SUCCESSFULLY_PERFORM_THE_MASS_UPDATE_PLEASE_DESELECT_AT_LEAST_ONE_ENTITY_WITH_AN_ADMINISTRATOR_ROLE;
ONE_ADMIN_REQD_PER_ACCT	YOU_CANT_DELETE_THIS_EMPLOYEE_NO_ADMINISTRATORS_FOR_THIS_ACCOUNT_WOULD_REMAIN;
ONE_ADMIN_REQD_PER_ACCT	YOU_CANT_INACTIVATE_1_THE_ACCOUNT_WOULD_BE_LEFT_WITH_NO_ACTIVE_ADMINISTRATORS;
ONE_ADMIN_REQD_PER_ACCT	YOU_CANT_REMOVE_THE_ADMINISTRATOR_ROLE_FROM_THIS_USER_NO_ADMINISTRATORS_FOR_THIS_ACCOUNT_WOULD_REMAIN;
ONE_EMPL_REQD	AT_LEAST_ONE_EMPLOYEE_IS_REQUIRED_TO_PROCESS_PAYROLL;
ONE_RCRD_REQD_FOR_MASS_UPDATE	PLEASE_CREATE_AT_LEAST_ONE_1_BEFORE_USING_THIS_MASS_UPDATE;
ONE_ROLE_REQD	YOU_CANT_INACTIVATE_ALL_ROLES_YOU_WOULD_NOT_BE_ABLE_TO_LOG_IN;
ONLINE_BILL_PAY_SETUP_REQD	B1B_IS_NOT_SET_UP_FOR_ONLINE_BILL_PAY_TO_SET_UP_THIS_PAYEE_CLICK_GO_BACK;
ONLINE_FORM_ID_REQD	MISSING_REQUIRED_ONLINE_FORM_ID;
ONLINE_ORD_FEATURE_DISABLED	CANT_OPEN_STORE_FOR_1__THIS_COMPANY_DOES_NOT_HAVE_THE_BONLINE_ORDERINGB_FEATURE_ENABLED__THE_FEATURE_IS_REQUIRED_FOR_CUSTOMERS_TO_MAKE_ONLINE_PURCHASES;
ONLINE_ORD_FEATURE_DISABLED	CANT_OPEN_STORE_FOR_1__THIS_COMPANY_DOES_NOT_HAVE_THE_BUSE_SALES_ORDERSB_FEATURE_ENABLED__THE_FEATURE_IS_REQUIRED_FOR_CUSTOMERS_TO_MAKE_ONLINE_PURCHASES;
ONLY_ONE_CONTRIB_ITEM_REQD	ONLY_ONE_INSTANCE_OF_A_COMPANY_CONTRIBUTION_ITEM_IS_ALLOWED_ON_AN_EMPLOYEE_RECORD;
ONLY_ONE_DEDCT_ITEM_REQD	ONLY_ONE_INSTANCE_OF_A_DEDUCTION_ITEM_IS_ALLOWED_ON_AN_EMPLOYEE_RECORD;
ONLY_ONE_DISTRIB_ALLWD	YOU_MAY_NOT_DISTRIBUTE_1_NUMBERS_MORE_THAN_ONCE__YOU_ATTEMPTED_TO_DISTRIBUTE_THE_FOLLOWING_1_NUMBERS_MORE_THAN_ONCE_2;
ONLY_ONE_EARNING_ITEM_REQD	ONLY_ONE_INSTANCE_OF_AN_EARNING_ITEM_IS_ALLOWED_ON_AN_EMPLOYEE_RECORD;
ONLY_ONE_UNIT_AS_BASE_UNIT	ONLY_ONE_UNIT_MAY_BE_DESIGNATED_AS_THE_BASE_UNIT;
ONLY_ONE_UPLOAD_ALLWD	YOU_CANNOT_UPLOAD_MORE_THAN_ONE_FILE_AT_A_TIME;
ONLY_ONE_WITHLD_ITEM_REQD	ONLY_ONE_INSTANCE_OF_A_WITHHOLDING_ITEM_IS_ALLOWED_ON_AN_EMPLOYEE_RECORD;
ORD_ALREADY_APPRVD	YOU_CANNOT_CANCEL_THIS_ORDER_BECAUSE_IT_HAS_ALREADY_BEEN_APPROVED;
ORDER_DSNT_EXIST	THAT_ORDER_DOES_NOT_EXIST;

Error Code	Long Description or Message
PACKAGE_WEIGHT_REQD	ATTEMPTED_TO_CREATE_A_PACKAGE_WITHOUT_SPECIFYING_A_NONZERO_PACKAGE_WEIGHT;
PARENT_CANT_ITSELF_BE_MEMBER	PARENT_ITEM_CAN_NOT_BE_A_MEMBER_OF_ITSELF;
PARTNER_ACCESS_DENIED	PARTNERS_DO_NOT_HAVE_ACCESS_TO_THIS_ITEM;
PARTNER_CODE_ALREADY_USED	A_PARTNER_WITH_THAT_PARTNER_CODE_ALREADY_EXISTS;
PAYCHECK_IN_USE	YOU_CANNOT_CLEAR_THIS_PAYCHECK_BECAUSE_IT_IS_LINKED_TO_BY_ONE_OR_MORE_LIABILITY_PAYMENTS__YOU_MUST_DELETE_OR_VOID_THOSE_TRANSACTIONS_FIRST;
PAYEE_REQD	U_MUST_SPECIFY_A_PAYEE;
PAYEE_REQD_FOR_PMT	YOUR_PAYMENT_HAS_BEEN_RECORDED_BUT_AN_ONLINE_BILL_PAY_PAYMENT_WILL_NOT_BE_MADE_BECAUSE_NO_PAYEE_WAS_SPECIFIEDYOU_SHOULD_RETURN_TO_THE_PAYMENT_SCREEN_IF_YOU_WISH_TO_PRINT_THE_CHECK;
PAYROLL_EXPENSE_ACCT_REQD	PLEASE_SELECT_A_EXPENSE_ACCOUNT_FOR_PAYROLL_ITEM;
PAYROLL_EXPENSE_ACCT_REQD	PLEASE_SELECT_AN_EXPENSE_ACCOUNT_FOR_PAYROLL_ITEM;
PAYROLL_FEATURE_DISABLED	YOU_HAVE_NOT_ENABLED_THE_PAYROLL_FEATURE;
PAYROLL_FEATURE_UNAVAILABLE	YOU_ARE_TRYING_TO_EDIT_A_PAY_CHEQUE__PAYROLL_IS_NOT_AVAILABLE_IN_NETLEDGER_CANADA;
PAYROLL_ITEM_DELETE_DISALLWD	UNABLE_TO_REMOVE_PAYROLL_ITEM_1__THERE_ARE_EXISTING_TRANSACTIONS_FOR_THIS_PAYROLL_ITEM__YOU_MAY_MARK_IT_INACTIVE_INSTEAD;
PAYROLL_LIABILITY_ACCT_REQD	PLEASE_SELECT_A_LIABILITY_ACCOUNT_FOR_PAYROLL_ITEM;
PERMISSION_VIOLATION	PERMISSION_VIOLATION_YOU_MAY_NO_LONGER_EDIT_THIS_RECORD;
PERMISSION_VIOLATION	PERMISSION_VIOLATION_YOU_MAY_NOT_ACCESS_THIS_RECORD;
PHONE_NUM_REQD	PLEASE_PROVIDE_A_PHONE_NUMBER;
PLAN_IN_USE	THIS_PLAN_HAS_ALREADY_BEEN_USED_TO_GENERATE_COMMISSION_CALCULATIONS_AND_CANT_BE_DELETED;
PMT_ALREADY_SBMTD	THIS_PAYMENT_HAS_ALREADY_BEEN_SUBMITTED_FOR_ONLINE_BILL_PAY;
PMT_EDIT_DISALLWD	THIS_LIABILITY_PAYMENT_CANNOT_BE_EDITED_WHILE_IT_HAS_AN_AUTOMATED_CLEARING_HOUSE_TRANSMISSION_IN_PROCESS;
POSITIVE_QTY_REQD	ASSEMBLY_MEMBER_ITEMS_MUST_HAVE_POSITIVE_QUANTITIES;
POSTING_PRD_SETUP_REQD	CREATION_OF_JOURNAL_ENTRIES_REQUIRE_A_SINGLE_ACCOUNTING_PERIOD_VALUE_ACROSS_ALL_REVENUE_RECOGNITION_EVENTS__PLEASE_SETUP_A_POSTING_PERIOD_FILTER;
PRD_SETUP_REQD	YOU_MUST_CHANGE_YOUR_PERIOD_DEFINITIONS_TO_CONTAIN_FISCAL_YEARS_PLEASE_VISIT_SETUPMANAGE_ACCOUNTING_PERIODS_AND_CLICK_SET_UP_YEAR;
PRD_SETUP_REQD	YOU_MUST_DEFINE_THE_PERIODS_OF_THE_PRIOR_FISCAL_YEAR_PLEASE_VISIT_SETUPMANAGE_ACCOUNTING_PERIODS_AND_CLICK_SET_UP_YEAR;

Error Code	Long Description or Message
PREF_VENDOR_COST_REQD	DROP_SHIPSPECIAL_ORDER_ITEMS_MUST_HAVE_A_PREFERRED_VENDOR_AND_A_COST;
PREFERRED_TAX_AGENCY_REQD	A_PREFERRED_TAX_AGENCY_HAS_BEEN_DELETED__PLEASE_CHOOSE_A_NEW_ONE;
PREFERRED_TAX_AGENCY_REQD	ERROR_NO_PREFERRED_TAX_AGENCIES_HAVE_BEEN_SET_UP;
PRDS_DISALLWD_NAMES_NOT_UNIQUE	AFTER_ADDING_NEW_PERIODS_NOT_ALL_NAMES_WOULD_BE_UNIQUE;
PRIVATE_RCRD_ACCESS_DISALLWD	YOU_CANNOT_VIEW_OR_EDIT_THIS_RECORD_BECAUSE_IT_IS_MARKED_PRIVATE;
PRIVATE_STATUS_CHNG_DISALLWD	YOU_CANNOT_MAKE_THIS_CONTACT_PRIVATE;
PSWD_REQD	YOU_MUST_PROVIDE_A_PASSWORD_TO_GIVE_THIS_PERSON_ACCESS_TO_YOUR_ACCOUNT;
PSWD_REQD	PASSWORD_IS_EMPTY;
PSWD_REQD	PLEASE_TYPE_YOUR_PASSWORD_INTO_BOTH_FIELDS;
PWSDS_DONT_MATCH	PASSWORDS_DONT_MATCH;
PWSDS_DONT_MATCH	THE_PASSWORDS_YOU_ENTERED_DO_NOT_MATCH__PLEASE_REENTER_YOUR_PASSWORDS;
PWSDS_DONT_MATCH	THE_PASSWORDS_YOU_HAVE_ENTERED_DO_NOT_MATCH;
PWSDS_DONT_MATCH	NEW_PASSWORDS_DONT_MATCH;
RCRD_DSNT_EXIST	THAT_RECORD_DOES_NOT_EXIST;
RCRD_PREVSLY_DELETED	THIS_RECORD_HAS_ALREADY_BEEN_DELETED;
RCRD_PREVSLY_DELETED	THIS_RECORD_HAS_BEEN_DELETED_SINCE_THE_LIST_WAS_GENERATED;
RCRD_TYPE_DSNT_EXIST	THERE_ARE_NO_RECORDS_OF_THIS_TYPE;
RCRD_TYPE_REQD	WS_MISSING_RECORD_TYPE;
RCRD_UNEDITABLE	THAT_RECORD_IS_NOT_EDITABLE;
READY_MEMORIZED_TRANS	READY_MEMORIZED_TRAN;
RECOGNIZED_METHOD	RECOGNIZED_METHOD;
RECUR_EVENT_DISALLWD	A_YEARLY_EVENT_CANNOT_BE_ON_THE_29TH_OF_FEBRUARY;
REG_CRNCY_NOT_BASE_CRNCY	THE_CURRENCY_YOU_ARE_REGISTERED_TO_USE_IS_DIFFERENT_FROM_THE_BASE_CURRENCY_OF_THIS_COMPANY;
REQUEST_PARAM_REQD	THIS_REQUEST_IS_MISSING_A_REQUIRED_PARAMETER;
REV_REC_DATE_REQD	NO_REVENUE_RECOGNITION_START_DATE_SPECIFIED;
REVERSAL_DATE_WARNING	REVERSAL_DATE_WARNING;
ROLE_REQUIRED	In order to login, a role is required unless a default has been previously set.
ROUNDING_DIFF_TOO_BIG	ROUNDING_DIFFERENCE_TOO_BIG__TAX1_1_TAX2_2;
ROUNDING_ERROR	ROUNDING_ERROR_1;

Error Code	Long Description or Message
SCHDUL_EDIT_DISALLWD	THIS_SCHEDULE_CANNOT_BE_EDITED_AS_IT_HAS_ALREADY_BEEN_USED_FOR_COMMISSION_CALCULATIONS_PLEASE_GO_BACK_AND_SELECT_SAVE_AS_NEW_INSTEAD;
SEARCH_INTEGER_REQD	PLEASE_ENTER_AN_INTEGER_NUMBER_TO_SEARCH_ON;
SECTIONS_UNEDITABLE	ILTIN_SECTIONS_CANNOT_BE_EDITED_EXCEPTION;
SECURE_TRANS_REQD_ON_CHECKOUT	STORE_SERVER_ERROR_AS_CONFIGURED_THIS_SERVER_DOES_NOT_PERMIT_SECURE_TRANSACTIONS_REQUIRED_BY_STORE_CHECKOUT;
SESSION_TERMD_2ND_LOGIN_DECTD	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_THE_SAME_USER_USE_FILENEWWINDOW_FRO
SESSION_TERMD_2ND_LOGIN_DECTD	YOU_CAN_HAVE_A_MAXIMUM_OF_1_ACTIVE_USERS_AT_A_TIME_IN_2;
SESSION_TERMD_2ND_LOGIN_DECTD	YOU_CAN_HAVE_A_MAXIMUM_OF_1_ACTIVE_USERS_AT_A_TIME_THAT_ARE_ENABLED_FOR_OFFLINE_SALES_CLIENT;
SESSION_TIMED_OUT	WS_SESSION_TIMED_OUT;
SESSION_TIMED_OUT	YOUR_SESSION_HAS_TIMED_OUT_PLEASE_REENTER_YOUR_INFORMATION_AND_TRY_AGAIN;
SESSION_TIMED_OUT	YOUR_CONNECTION_HAS_TIMED_OUT__PLEASE_A_HREFPAGESLOGINJSP_TARGETSELFLOG_INA_AGAIN;
SO_HAS_CHILD_TRANS	SALESORDER_HAS_CHILD_TRANSACTIONS;
START_DATE_AFTER_END_DATE	THE_START_DATE_MUST_PRECEED_THE_END_DATE;
STORAGE_LIMIT_EXCEEDED	YOU_ENTERED_A_VALUE_THAT_WILL_EXCEED_THE_INTERNAL_STORAGE_LIMIT_OF_1_PLEASE_REDUCE_THE_NUMBER;
STORE_ALIAS_UNAVAILABLE	THE_STORE_ALIAS_YOU_CHOSE_1_IS_ALREADY_TAKEN__PLEASE_GO_BACK_AND_CHOOSE_ANOTHER;
STORE_DOMAIN_UNAVAILABLE	THE_STORE_DOMAIN_NAME_YOU_CHOSE_0_IS_ALREADY_TAKEN_PLEASE_GO_BACK_AND_CHOOSE_ANOTHER;
STORE_SERVER_ERROR	STORE_SERVER_ERROR_AS_CONFIGURED_THIS_SERVER_DOES_NOT_PERMIT_SHOPPING;
SUBITEM_REQD	YOU_MUST_FIRST_SELECT_THE_NEW_SUBITEMS_ON_THE_MATRIX_TAB_YOU_WANT_TO_ADD;
SUBITEM_REQD	YOU_MUST_FIRST_SELECT_THE_SUBITEMS_ON_THE_MATRIX_TAB_YOU_WANT_TO_CREATE;
SUCCESS_TRANS	THE_TRANSACTION_WAS_ENTERED_1_SUCCESSFULLY_2;
TAX_CODES_SETUP_PROBLEM	THE_TAX_CODES_HAVENT_BEEN_SET_PROPERLY;
TAX_CODES_SETUP_REQD	THE_COMPANY_IS_NOT_USABLE_ADMINISTRATOR_HASNT_SET_UP_THE_TAX_CODES;
TAX_CODES_SETUP_REQD	CANT_OPEN_STORE_FOR_1__THIS_COMPANY_DOES_NOT_HAVE_ITS_TAX_CODES_FULLY_SET_UP__THIS_IS_REQUIRED_TO_PROPERLY_CALCULATE_TAXES_ON_INTERNATIONAL_OTHERPROVINCE_AND_SAMEPROVINCE_ORDERS;
TICKET_NOT_LOCATED	THE_TICKET_1_CANNOT_BE_LOCATED_IN_THE_ERROR_DATABASE_IF_THIS_IS_FROM_A_CUSTOMER_LOGGED_CASE_THE_ERROR_MAY_NOT_YET_BE_INSERTED_INTO_THE_SYSTEM;

Error Code	Long Description or Message
TRANS_AMTS_UNBALNCD	TRANSACTION_IS_NOT_IN_BALANCE__AMOUNTSTAXESSHIPPING_1_TOTAL_AMOUNT_2;
TRANS_APPLIED_AMTS_UNBALNCD	TRANSACTION_IS_NOT_IN_BALANCE__TOTAL_TO_APPLY_OF_1_DOES_NOT_EQUAL_SUM_OF_APPLIED_2_AND_UNAPPLIED_3;
TRANS_APPLIED_AMTS_UNBALNCD	TRANSACTION_IS_NOT_IN_BALANCE__TOTAL_TO_APPLY_OF_1_DOES_NOT_EQUAL_SUM_OF_PAYMENT_2_AND_CREDITS_3;
TRANS_CLASS_UNBALNCD	TRANSACTION_OUT_OF_BALANCE_FOR_CLASS_1_TOTAL__2;
TRANS_DEPT_UNBALNCD	TRANSACTION_OUT_OF_BALANCE_FOR_DEPARTMENT_1_TOTAL__2;
TRANS_DOES_NOT_EXIST	NO_TRANSACTION_EXISTS_FOR_THAT_ENTITY;
TRANS_EDIT_DISALLWD	YOU_CANNOT_EDIT_THIS_TRANSACTION_1_DOES_NOT_SUPPORT_THE_IMPORTED_TRANSACTION;
TRANS_EDIT_DISALLWD	THIS_TRANSACTION_IS_IN_A_PERIOD_THAT_HAS_BEEN_CLOSED_YOU_MAY_NOT_EDIT_IT;
TRANS_FORGN_CUR_UNBALNCD	TRANSACTION_WAS_NOT_IN_BALANCE_FOREIGN_CURRENCY_POSTING_TOTAL__1;
TRANS_FORGN_CUR_UNBALNCD	TRANSACTION_WAS_NOT_IN_BALANCE_FOREIGN_CURRENCY_TOTAL__1;
TRANS_IN_USE	THIS_TRANSACTION_CANNOT_BE_DELETED_BECAUSE_IT_IS_LINKED_TO_BY_ONE_OR_MORE_TRANSACTIONS;
TRANS_LINE_AND_PMT_UNBALNCD	TRANSACTION_IS_NOT_IN_BALANCE__LINE_ITEM_SUM_OF_1_NOT_EQUAL_TO_PAYMENT_AMOUNT_2;
TRANS_LINES_UNBALNCD	TRANSACTION_IS_NOT_IN_BALANCE__LINE_ITEM_SUM_OF_1_DOES_NOT_EQUAL_AMOUNT_OF_2;
TRANS_LINES_UNBALNCD	TRANSACTION_IS_NOT_IN_BALANCE__LINE_ITEM_SUM_OF_1_DOES_NOT_EQUAL_APPLIED_AMOUNT_OF_2;
TRANS_LOC_UNBALNCD	TRANSACTION_OUT_OF_BALANCE_FOR_LOCATION_1_TOTAL__2;
TRANS_NOT_CLEANED	TRANSACTION_NOT_CLEANED_UP;
TRANS_NOT_COMPLETED	TRANSACTION_WAS_NOT_COMPLETE;
TRANS_UNBALNCD	TRANSACTION_IS_NOT_IN_BALANCE_1;
TRANS_UNBALNCD	TRANSACTION_IS_NOT_IN_BALANCE_12__OTHERCOUNT__3;
TRANS_UNBALNCD	TRANSACTION_WAS_NOT_IN_BALANCE_POSTING_TOTAL__1;
TRANS_UNBALNCD	TRANSACTION_WAS_NOT_IN_BALANCE_TOTAL__1;
TRANSACTION_DELETED	THE_TRANSACTION_YOU_ARE_ATTEMPTING_TO_ACCESS_HAS_BEEN_DELETED;
TYPE_NOT_RECGNZD	DIA_TYPE_NOT_RECOGNIZED;
UNDEFINED_ACCTNG_PRD	THE_ACCOUNTING_PERIOD_RANGE_1_HAS_NOT_BEEN_DEFINED_PLEASE_VISIT_A_HREFAPPSETUPPERIODFISCALPERIODSNLSETUP__ACCOUNTING__MANAGE_ACCOUNTING_PERIODSA_TO_DEFINE_THIS_PERIOD_OR_SET_UP_YOUR_YEAR;
UNDEFINED_ACCTNG_PRD	THE_COMPARISON_ACCOUNTING_PERIOD_RANGE_1_HAS_NOT_BEEN_DEFINED_PLEASE_VISIT_A_HREFAPPSETUPPERIODFISCALPERIODSNLSETUP__ACCOUNTING__MANAGE_ACCOUNTING_PERIODSA_TO_DEFINE_THIS_PERIOD_OR_SET_UP_YOUR_YEAR;

Error Code	Long Description or Message
UNDEFINED_ACCTNG_PRD	THE_DEFAULT_ACCOUNTING_PERIOD_FOR_THIS_REPORT_HAS_NOT_BEEN_DEFINED_PLEASE_VISIT_A_HREFAPPSETUPPERIODFISCALPERIODSNLSETUP__ACCOUNTING__MANAGE_ACCOUNTING_PERIODSA_TO_DEFINE_THIS_PERIOD_OR_SET_UP_YOUR_YEAR;
UNDEFINED_TAX_PRD	THE_DEFAULT_TAX_PERIOD_FOR_THIS_REPORT_HAS_NOT_BEEN_DEFINED_PLEASE_VISIT_A_HREFAPPSETUPPERIODTAXPERIODSNLSETUP__ACCOUNTING__MANAGE_TAX_PERIODSA_TO_DEFINE_THIS_PERIOD_OR_SET_UP_YOUR_YEAR;
UNEXPECTED_ERROR	AN_UNEXPECTED_ERROR_HAS_OCCURRED__TECHNICAL_SUPPORT_HAS_BEEN_ALERTED_TO_THIS_PROBLEM;
UNEXPECTED_ERROR	WS_AN_UNEXPECTED_ERROR_OCCURRED;
UNEXPECTED_ERROR	WS_DTO_JAVA_CLASS_IS_NOT_DEFINED;
UNEXPECTED_ERROR	WS_FAILED_TO_INSTANTIATE_CLASS;
UNEXPECTED_ERROR	WS_NO_DTO_CLASS_IS_DEFINED;
UNEXPECTED_ERROR	WS_SCHEMA_CREATION_FAILED_1_DOES_NOT_HAVE_A_MATCHING_XML_SCHEMA_TYPE;
UNEXPECTED_ERROR	WS_SERVER_ERROR_MISSING_DATABASE_ENTRIES_IN_WSRECORDELEMENT_AND_WSNAMESPACE_TABLES;
UNEXPECTED_ERROR	WS_SERVER_ERROR_NO_FORM_REQUEST_CLASS_IS_DEFINED;
UNEXPECTED_ERROR	AN_ERROR_OCCURRED_WHILE_PROCESSING_ITEM_OPTIONS;
UNEXPECTED_ERROR	ACTION_EXCEPTION;
UNEXPECTED_ERROR	ERROR;
UNEXPECTED_ERROR	ERROR_1;
UNEXPECTED_ERROR	PROBLEM_DURING_COMMISSION_CALCULATION;
UNITS_TYP_IN_USE	THIS_UNITS_TYPE_IS_USED_BY_1_2__YOU_MUST_DELETE_THE_2_AND_ALL_ASSOCIATED_TRANSACTIONS_IN_ORDER_TO_DELETE_THIS_UNITS_TYPE;
UNKNWN_ALLOCTN_FREQ_TYP	UNABLE_TO_DETERMINE_ALLOCATION_FREQUENCY_TYPE;
UNKNWN_ALLOCTN_SCHDUL_FREQ_TYP	UNABLE_TO_DETERMINE_ALLOCATION_SCHEDULE_FREQUENCY_TYPE;
UNRECOGNIZED_METHOD	UNRECOGNIZED_METHOD_1;
UNSUPRTD_DOC_TYP	UPLOAD_UNSUPPORTED_DOC_TYPE_ERROR;
UNSUPPORTED_WS_VERSION	MESSAGE_CONTAINS_AN_UNSUPPORTED_WEB_SERVICES_VERSION
UPDATE_DISALLWD	WS_UPDATE_IS_NOT_ALLOWED;
UPDATE_PRICE_AMT_REQD	PLEASE_SPECIFY_AN_AMOUNT_TO_UPDATE_PRICES;
URL_ID_PARAM_REQD	URL_IS_MISSING_THE_ID_PARAMETER_THE_FILE_COULD_NOT_BE_RETRIEVED;
URL_REQD	YOU_MUST_ENTER_A_URL_FOR_THIS_MEDIA_ITEM;
USER_DISABLED	USER_DISABLED;
VALID_PRD_REQD	INSERT_TRANSACTION_FAILURE_NO_VALID_OPEN_1_PERIOD_FOR_DATE__2;
VALID_PRD_REQD	UPDATE_TRANSACTION_FAILURE_NO_VALID_OPEN_1_PERIOD_FOR_DATE__2;
VENDOR_TYPE_REQD	NO_VENDOR_TYPE_WAS_SPECIFIED__IF_CREATING_A_TAX_AGENCY_PLEASE_ENSURE_THAT_THE_VENDOR_TYPE_IS_ACTIVE_AND_MARKED_AS_A_TAX_AGENCY;

Error Code	Long Description or Message
VOIDING_REVERSAL_DISALLWD	YOU_MAY_NOT_CREATE_A_VOIDING_REVERSAL_FOR_TRANSACTIONS_WITH_INVENTORY_IMPACT;
WS_CONCUR_SESSION_DISALLWD	WS_INVALID_CONCURRENT_SESSION;
WS_FEATURE_REQD	WS_NO_WEB_SERVICES_FEATURE;
WS_LOG_IN_REQD	WS_LOG_IN_FIRST;

Server SuiteScript API Library File

Following is a list of all of Server SuiteScript Functions. Import this into your IDE in order to enable code completion of the SuiteScript API in your IDE. For more information, see Setting up Your Development Environment.

```
/* Core SuiteScript Functions */
function nlapiLoadRecord(type, id) { ; } /* load a record from the system. */
function nlapiCreateRecord(type) { ; } /* instantiate a new record from the
system. */
function nlapiCopyRecord(type, id) { ; } /* instantiate a new record using values
from an existing record. */
function nlapiTransformRecord(type, id, transformType) { ; } /* create a new
record using values from an existing record of a different type. */
function nlapiSubmitRecord(record, doSourcing) { ; } /* submit a record to the
system for creation or update */
function nlapiDeleteRecord(type, id) { ; } /* delete a record from the system. */
function nlapiSearchRecord(type, id, filters, columns) { ; } /* perform a record
search using an existing search or filters and columns */
function nlapiLookupField(type,id,fields) { ; } /* fetch the value of a field or
set of fields for a record. */
function nlapiResolveURL(type, identifier, id, editmode) { ; } /* resolve a URL to
an object in the system. */
function nlapiSetRedirectURL(type, identifier, id, editmode, parameters) { ; } /*
redirect the user to a page. */
function nlapiRequestURL(url, postdata, headers, callback, target) { ; } /*
request a URL to an external or internal resource. */
function nlapiGetContext() { ; } /* return context information about the current
user/script. */
function nlapiGetUser() { ; } /* return the internalId for the current user. */
function nlapiGetRole() { ; } /* return the internalId for the current user's
role. */
function nlapiGetDepartment() { ; } /* return the internalId for the current
user's department. */
function nlapiGetLocation() { ; } /* return the internalId for the current user's
location. */
function nlapiGetRecordType() { ; } /* return the recordtype corresponding to the
current page or userevent script. */
function nlapiGetRecordId() { ; } /* return the internalId corresponding to the
current page or userevent script. */
function nlapiCreateError(code,details) { ; } /* create a user-defined error
object that can be thrown */
function nlapiGetFieldValue(fldnam) { ; } /* return the value of a field on the
current record on a page or userevent script. */
function nlapiSetFieldValue(fldnam,value,firefieldchanged) { ; } /* set the value
of a field on the current record on a page or userevent script. */
function nlapiGetLineItemValue(type,fldnam,linenum) { ; }/* return the value of a
line item field on the current record on a page or userevent script. */
function nlapiSetLineItemValue(type,fldnam,linenum,value) { ; }/* set the value of
a line item field on the current record on a page or userevent script. */
function nlapiGetLineItemCount(type) { ; } /* return the number of line items in a
line item group on the current record on a page or userevent script. */
/* Server SuiteScript Functions */
function nlapiLogExecution(type, title, details) { ; } /* create an entry in the
script execution log. */
function nlapiGetNewRecord() { ; } /* return an record object containing the data
being submitted to the system for the currenr record. */
```

```
function nlapiGetOldRecord() { ; } /* return an record object containing the
current record's data prior to the write operation. */
function nlapiCreateForm(title) { ; } /* return a new form that can be used to
build a custom page. */
function nlapiCreateList(title) { ; } /* return a new list that can be used to
build a custom page. */
function nlapiSendEmail(from, to, subject, body, cc, bcc, records) { ; } /* send
out an email and associate it with records in the system. */
function nlapiMergeEmail(id, baseType, baseId, altType, altId) { ; } /* perform a
mail merge operation using an email template and up to 2 records. */
/* Helper SuiteScript Functions */
function nlapiStringToDate(str) { ; } /* convert a String into a Date object. */
function nlapiDateToString(d) { ; } /* convert a Date object into a String using
the current user's date format preference. */
function nlapiAddDays(d, days) { ; } /* add days to a Date object. */
function nlapiAddMonths(d, months) { ; } /* add months to a Date object. */
function nlapiFormatCurrency(str) { ; } /* format a number for data entry into a
currency field. */
function nlapiEncrypt(s) { ; } /* encrypt a String using an asymetric encryption
algorithm. */
function nlapiEscapeXML(text) { ; } /* escape a String for use in an XML document.
*/
function nlapiStringToXML(text) { ; } /* convert a String into an XML document. */
function nlapiXMLToString(xml) { ; } /* convert an XML document into a String. */
function nlapiSelectValue(node, xpath) { ; } /* select a value from an XML node
using Xpath. */
function nlapiSelectValues(node, xpath) { ; } /* select an array of values from an
XML node using Xpath. */
function nlapiSelectNode(node, xpath) { ; } /* select a node from an XML node using
Xpath. */
function nlapiSelectNodes(node, xpath) { ; } /* select an array of nodes from an
XML node using Xpath. */

/* nlobjRecord: scriptable record used for accessing and manipulating records. */
function nlobjRecord( ) { ; }
nlobjRecord.prototype.getId = function( ) { ; } /* return the internalId of the
current record or NULL for new records. */
nlobjRecord.prototype.getRecordType = function( ) { ; } /* return the recordType
corresponding to this record. */
nlobjRecord.prototype.setFieldValue = function( name, value ) { ; } /* set the
value of a field. */
nlobjRecord.prototype.setFieldValues = function( name, value ) { ; } /* set the
values of a multi-select field. */
nlobjRecord.prototype.getFieldValue = function( name ) { ; } /* return the value
of a field. */
nlobjRecord.prototype.getFieldValues = function( name ) { ; } /* return the
selected values of a multi-select field as an Array. */
nlobjRecord.prototype.getAllFields = function( ) { ; } /* return an Array of all
fields. */
nlobjRecord.prototype.setLineItemValue = function( group, name, line, value ) { ;
} /* set the value of a line item field. */
nlobjRecord.prototype.getLineItemValue = function( group, name, line ) { ; } /*
return the value of a line item field. */
nlobjRecord.prototype.getLineItemCount = function( group ) { ; } /* return the
number of lines in a line item group. */
/* nlobjSearchFilter: filter objects used to defined search criteria. */
function nlobjSearchFilter( name, join, operator, value, value2 ) { ; }
nlobjSearchFilter.prototype.getName = function( ) { ; } /* return the name of this
search filter. */
nlobjSearchFilter.prototype.getJoin = function( ) { ; } /* return the join id for
this search filter. */
nlobjSearchFilter.prototype.getOperator = function( ) { ; } /* return the filter
operator used. */
/* nlobjSearchColumn: column objects used to return data via search. */
```

```
function nlobjSearchColumn( name, join, summary ) { ; }
nlobjSearchColumn.prototype.getName = function( ) { ; } /* return the name of this
search column. */
nlobjSearchColumn.prototype.getJoin = function( ) { ; } /* return the join id for
this search column. */
nlobjSearchColumn.prototype.getSummary = function( ) { ; } /* return the summary
type (avg,group,sum,count) of this search column. */
/* nlobjSearchResult: search result row object. */
function nlobjSearchResult( ) { ; }
nlobjSearchResult.prototype.getId = function( ) { ; } /* return the internalId for
the record returned in this row. */
nlobjSearchResult.prototype.getRecordType = function( ) { ; } /* return the
recordtype for the record returned in this row. */
nlobjSearchResult.prototype.getValue = function( name, join, summary ) { ; } /*
return the value of this return column. */
nlobjSearchResult.prototype.getText = function( name, join, summary ) { ; } /*
return the text value of this return column if it's a select field. */
/* nlobjContext: current user and script context information. */
function nlobjContext( ) { ; }
nlobjContext.prototype.getName = function( ) { ; }  /* return the name of the
current user. */
nlobjContext.prototype.getUser = function( ) { ; } /* return the internalId of the
current user. */
nlobjContext.prototype.getRole = function( ) { ; } /* return the internalId of the
current user's role. */
nlobjContext.prototype.getEmail = function( ) { ; } /* return the email address of
the current user. */
nlobjContext.prototype.getCompany = function( ) { ; } /* return the account ID of
the current user. */
nlobjContext.prototype.getDepartment = function( ) { ; } /* return the internalId
of the current user's department. */
nlobjContext.prototype.getLocation = function( ) { ; }  /* return the internalId
of the current user's location. */
 nlobjContext.prototype.getExecutionContext = function( ) { ; } /* return the
internalId of the current user's role. */
/* nlobjError: system or user-defined error object. */
function nlobjError( ) { ; }
nlobjError.prototype.getCode = function( ) { ; } /* return the error code for this
system or user-defined error. */
nlobjError.prototype.getDetails = function( ) { ; } /* return the error
description for this error. */
nlobjError.prototype.getStackTrace = function( ) { ; } /* return a stacktrace
containing the location of the error. */
nlobjError.prototype.getUserEvent = function( ) { ; } /* return the userevent
script name where this error was thrown. */
nlobjError.prototype.getInternalId = function( ) { ; } /* return the internalid of
the record if this error was thrown in an aftersubmit script. */
/* nlobjResponse: ServletResponse used in SERVLET scripts -or- returned by a call
to nlapiRequestURL. */
function nlobjResponse( ) { ; }
nlobjResponse.prototype.setHeader = function( name, value ) { ; } /* sets a custom
response header. */
nlobjResponse.prototype.setRedirectURL = function( category, type, id, editmode,
parameters ) { ; } /* sets the redirect URL for the response. */
nlobjResponse.prototype.write = function( output ) { ; } /* write information
(text/xml/html) to the response. */
nlobjResponse.prototype.writePage = function( pageobject ) { ; } /* write a page
(nlobjForm or nlobjList). */
nlobjResponse.prototype.getHeader = function( ) { ; } /* return the value of a
header returned via nlapiRequestURL. */
nlobjResponse.prototype.getAllHeaders = function( ) { ; } /* return an Array of
all headers returned via nlapiRequestURL. */
nlobjResponse.prototype.getCode = function( ) { ; } /* return the response code
returned via nlapiRequestURL. */
```

```
nlobjResponse.prototype.getBody = function( ) { ; } /* return the response body
returned via nlapiRequestURL. */
nlobjResponse.prototype.getError = function( ) { ; } /* return the nlobjError
thrown via a client call to nlapiRequestURL. */
/* nlobjRequest: ServletRequest in SERVLET scripts and available in beforeLoad
scripts (for browser requests). */
function nlobjRequest( ) { ; }
nlobjRequest.prototype.getParameter = function( name ) { ; } /* return the value
of a request parameter. */
nlobjRequest.prototype.getParameterValues = function( name ) { ; } /* return the
values of a request parameter as an Array. */
nlobjRequest.prototype.getAllParameters = function( ) { ; } /* return an Array
containing all the request parameters. */
nlobjRequest.prototype.getLineItemValue = function( group, name, line ) { ; } /*
return the value of a line item value. */
nlobjRequest.prototype.getLineItemCount = function( group ) { ; } /* return the
number of lines in a line item group. */
nlobjRequest.prototype.getHeader = function( name ) { ; } /* return the value of a
header. */
nlobjRequest.prototype.getAllHeaders = function( ) { ; } /* return an Array
containing all the request headers. */
nlobjRequest.prototype.getBody = function( ) { ; } /* return the entire body of
the request. */
nlobjRequest.prototype.getURL = function( ) { ; } /* return the URL being
requested. */
nlobjRequest.prototype.getMethod = function( ) { ; } /* return the METHOD of the
request: GET or POST. */
/* nlobjPortlet: scriptable dashboard portlet. */
function nlobjPortlet( ) { ; }
nlobjPortlet.prototype.setTitle = function( title ) { ; } /* set the portlet
title. */
nlobjPortlet.prototype.writeLine = function( text, url, indent ) { ; } /* write a
line with optional indenting, URL. */
nlobjPortlet.prototype.setHtml = function( html ) { ; } /* set the entire contents
of the PORTLET (will be placed inside a <TD>...</TD>). */
nlobjPortlet.prototype.addColumn = function( name, type, label, align ) { ; } /*
add a column (nlobjColumn) to this portlet and return it. */
nlobjPortlet.prototype.addRow = function( row ) { ; } /* add a row (Associative
Array of name-value pairs) to this portlet. */
nlobjPortlet.prototype.addRows = function( rows ) { ; } /* add multiple rows
(Array of nlobjSearchResults or rows) to this portlet. */
/* nlobjList:  scriptable list page. */
function nlobjList( ) { ; }
nlobjList.prototype.setTitle = function( title ) { ; } /* set the page title. */
nlobjList.prototype.setStyle = function( style ) { ; } /* set the global style for
this list: grid|report|plain|normal. */
nlobjList.prototype.setScript = function( script ) { ; } /* set the Client
SuiteScript used for this page. */
nlobjList.prototype.addColumn = function( name, type, label, align ) { ; } /* add
a column (nlobjColumn) to this list and return it. */
nlobjList.prototype.addRow = function( row ) { ; } /* add a row (Associative Array
of name-value pairs) to this portlet. */
nlobjList.prototype.addRows = function( rows ) { ; } /* add multiple rows (Array
of nlobjSearchResults or rows) to this portlet. */
nlobjList.prototype.addButton = function( name, label, script ) { ; } /* add a
button (nlobjButton) to the footer of this page. */
nlobjList.prototype.addLink = function( title, url ) { ; } /* add a navigation
cross-link to the top-right corner of the page. */
/* nlobjColumn: scriptable list column. */
function nlobjColumn( ) { ; }
nlobjColumn.prototype.setLabel = function( label ) { ; } /* set the header for
this column. */
nlobjColumn.prototype.setURL = function( value, dynamic ) { ; } /* set the base
URL (optionally defined per row) for this column. */
```

```
nlobjColumn.prototype.addParamToURL = function( param, value, dynamic ) { ; } /*
add a URL parameter (optionally defined per row) to this column's URL. */
/* nlobjForm: scriptable form page. */
function nlobjForm( ) { ; }
nlobjForm.prototype.setTitle = function( title ) { ; } /* set the page title. */
nlobjForm.prototype.setScript = function( script, library, pi, sr, fc, vf, ps, vl,
rc, li ) { ; } /* set the Client SuiteScript used for this page. */
nlobjForm.prototype.setValues = function( values ) { ; } /* set the values for all
the fields on this form. */
nlobjForm.prototype.addLink = function( title, url ) { ; } /* add a navigation
cross-link to the top-right corner of the page. */
nlobjForm.prototype.addButton = function( type, name, label, script ) { ; } /* add
a button to this form. */
nlobjForm.prototype.addResetButton = function( label ) { ; } /* add a reset button
to this form. */
nlobjForm.prototype.addSubmitButton = function( label ) { ; } /* add a submit
button to this form. */
nlobjForm.prototype.addTab = function( name, label ) { ; } /* add a tab (nlobjTab)
to this form and return it. */
nlobjForm.prototype.addField = function( name,type,label,source,tab ) { ; } /* add
a field (nlobjField) to this form and return it. */
nlobjForm.prototype.addSubTab = function( name,label,tab ) { ; } /* add a subtab
(nlobjTab) to this form and return it. */
nlobjForm.prototype.addSubList = function( name,type,label,tab ) { ; } /* add a
sublist (nlobjSubList) to this form and return it. */
nlobjForm.prototype.insertTab = function( tab, nexttab ) { ; } /* insert a tab
(nlobjTab) before another tab (name). */
nlobjForm.prototype.insertField = function( field, nextfld ) { ; } /* insert a
field (nlobjField) before another field (name). */
nlobjForm.prototype.insertSubTab = function( subtab, nextsublist ) { ; } /* insert
a subtab (nlobjTab) before another subtab or sublist (name). */
nlobjForm.prototype.insertSubList = function( sublist, nextsublist ) { ; } /*
insert a sublist (nlobjSubList) before another subtab or sublist (name). */
nlobjForm.prototype.getTab = function( name ) { ; } /* return a tab (nlobjTab) on
this form. */
nlobjForm.prototype.getField = function( name ) { ; } /* return a field
(nlobjField) on this form. */
nlobjForm.prototype.getSubTab = function( name ) { ; } /* return a subtab
(nlobjTab) on this form. */
nlobjForm.prototype.getSubList = function( name ) { ; } /* return a sublist
(nlobjSubList) on this form. */
/* nlobjField: scriptable form/sublist field. */
function nlobjField( ) { ; }
nlobjField.prototype.setLabel = function( label ) { ; } /* set the label for this
field. */
nlobjField.prototype.setAlias = function( alias ) { ; } /* set the alias used to
set the value for this field. Defaults to field name. */
nlobjField.prototype.setDefaultValue = function( value ) { ; } /* set the default
value for this field. */
nlobjField.prototype.setMandatory = function( mandatory ) { ; } /* make this field
mandatory. */
nlobjField.prototype.setMaxLength = function( maxlength ) { ; } /* set the
maxlength for this field. */
nlobjField.prototype.setDisplayType = function( type ) { ; } /* set the display
type for this field: inline|normal|hidden|disabled. */
nlobjField.prototype.setLayoutType = function( type, breaktype ) { ; } /* set the
layout type (outside|startgroup|midgroup|endgroup|normal) and break type
(startcol|startrow|none). */
nlobjField.prototype.setDisplaySize = function( width, height ) { ; } /* set the
width and height for this field. */
nlobjField.prototype.setPadding = function( padding ) { ; } /* set the amount of
emppty vertical space (rows) between this field and the previous field. */
nlobjField.prototype.addSelectOption = function( value, text, selected ) { ; } /*
add a select option to this field (valid for select/multiselect fields). */
```

```
/* nlobjSubList: scriptable sublist (line item group). */
function nlobjSubList( ) { ; }
nlobjSubList.prototype.setLabel = function( label ) { ; }  /* set the label for
this sublist. */
nlobjSubList.prototype.setHelp = function( help ) { ; } /* set helper text for
this sublist. */
nlobjSubList.prototype.setDisplayType = function( type ) { ; } /* set the
displaytype for this sublist: hidden|normal. */
nlobjSubList.prototype.setLineItemValue = function( field, line, value ) { ; } /*
set the value of a cell in this sublist. */
nlobjSubList.prototype.setLineItemValues = function( values ) { ; } /* set values
for multiple lines (Array of nlobjSearchResults or rows) in this sublist. */
nlobjSubList.prototype.getField = function( name ) { ; } /* return a field
(nlobjField) on this sublist. */
nlobjSubList.prototype.addField = function( name,type,label,source ) { ; } /* add
a field (column) to this sublist. */
/* nlobjTab: scriptable tab or subtab. */
function nlobjTab( ) { ; }
nlobjTab.prototype.setLabel = function( label ) { ; } /* set the label for this tab
or subtab. */
nlobjTab.prototype.setHelp = function( help ) { ; } /* set helper text for this tab
or subtab. */
```