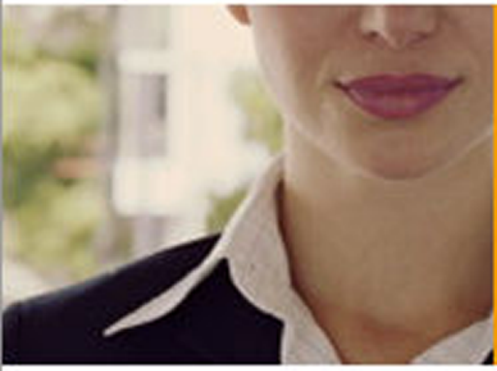


SuiteBuilder (Customization) Guide

September 3, 2013

Version 2013 Release 2

NETSUITE

Table of Contents

1. SuiteBuilder Overview 1
    Customizing Your NetSuite Account 6
2. Custom Fields 18
    Kinds of Custom Fields 18
        Custom CRM Fields 19
        Custom Entity Fields 20
        Custom Item Fields 21
        Custom Transaction Body Fields 22
        Custom Transaction Column Fields 23
        Adding Custom Fields to Transaction Forms 24
        Custom Transaction Item Options 24
        Custom Item Number Fields 26
        Other Custom Fields 27
    Creating a Custom Field 28
        Creating Custom Fields with Values Derived from Summary Search Results 30
        Assigning Custom Fields to Specific Record Types 33
        Setting Display Options for Custom Fields 33
        Setting Validation and Defaulting Properties 37
        Creating Dynamic Defaults and Dynamic Hyperlinks 40
        Creating Formula Fields 41
        SQL Expressions 46
        Setting Sourcing Criteria 52
        Understanding the ‘Source Filter by' Field 54
        Setting Filtering Criteria 55
        Multiple Dependent Dropdowns 56
        Restricting Access to Custom Fields 60
        Tracking Changes to Custom Fields 62
        Creating Display-Only Custom Fields 62
        Table of Custom Field Type Descriptions 63
        Available Standard Fields and Field Types 67
        Renaming Custom Fields 74
        Mass Updating Custom Fields 75
    Custom Lists 75
        Creating a Custom List 76
        Adding Translations for Custom Lists 77
3. Custom Forms 79
    Creating Custom Entry and Transaction Forms 81
        Storing Custom Forms with Transactions 84
        Configuring Subtabs for Custom Entry and Transaction Forms 86
        Configuring Field Groups 86
        Configuring Fields or Screens 89

Configuring Buttons and Actions 92
Configuring Printing Fields 95
Configuring Lists 97
Configuring QuickViews 98
Moving Fields and Lists Between Subtabs 104
Associating Custom Code (Client SuiteScript) Files With Custom Forms 105
Defining Preferred Entry and Transaction Forms 107
Linking Transaction Forms 108
Custom Sublists 109
Creating Custom Subtabs 113
Advanced PDF/HTML Templates (Beta) 114
Enabling the Advanced PDF/HTML Templates (Beta) Feature 115
Reviewing Available Advanced Templates 116
Setting Custom Forms to Use Advanced Templates 117
Customizing Advanced Templates in the Template Editor 119
Current Limitations for the Advanced PDF/HTML Templates Feature 133
Transaction Form PDF Layouts 134
Configuring Borders and Placement 136
Formatting Label Text 137
Formatting Data Text 137
Defining Custom Elements 138
Using a Standard #10 Window Envelope With Transactions 139
Transaction Form HTML Layouts 141
Understanding Totalling on Transaction Form Layouts 142
Defining Preferred Forms 143
Customizing Address Forms 145
Adding Disclaimers to Transaction Forms 145
Customizing Multiple Page Transaction Forms 146
Setting Transaction Forms to Print in Landscape 147
4. Custom Records 149
Creating Custom Record Types 149
Enabling Optimistic Locking for Custom Records 154
Adding Subtabs to a Custom Record 156
Numbering Custom Record Types 157
Adding Fields to Custom Record Types 158
Adding Custom Forms for a Record 158
Adding Custom Online Forms for a Record 159
Setting Permissions for a Custom Record Type 159
Applying Role-Based Restrictions to Custom Records 164
Creating Links to Custom Records 165
Choosing an Icon for a Custom Record 166
Adding Translations for Custom Records 171
Online Custom Record Forms 174
Creating Online Custom Record Forms 175

Linking Online Custom Record Forms to My Web Site 178
Creating HTML Templates for Online Custom Record Forms 179
Understanding Parent - Child Record Relationships 182
Creating Built-in Child Records 183
Using Child Records 184
Sourcing With Custom Records 185
Using Custom Records 189
5. Custom Centers 194
Creating Custom Centers 194
Creating Center Tabs 194
Creating Center Categories 196
Creating Center Links 198
Translating Custom Centers, Tabs, Categories, and Links 200
6. Deploying Upgraded Forms 204
Custom Form Deployment Process (Summary) 204
Custom Form Deployment Process (In Detail) 205
Deploying Upgraded Custom Forms 208
Previewing Undeployed Custom Forms 208
Editing the Layout of Custom Forms Prior to Deployment 209
Avoid Editing Custom Forms in Tandem 212
Testing Undeployed Custom Forms 212
Deploying Custom Forms 214
Deploying Skipped Custom Forms 214
Understanding Form Deployment Statuses 215
Understanding Form Layout Enhancement Upgrade Logic 216
Upgrade Logic for Custom Forms 216
Upgrade Logic for Subtabs 218
Upgrade Logic for Fields (Diagram) 218
Upgrade Logic for the Custom Field Record Page 219
Field Ordering 222

Chapter 1 SuiteBuilder Overview

With the NetSuite SuiteBuilder customization tools you can tailor NetSuite to your individual business needs.

These features allow you to easily adapt NetSuite to meet your company's business process. You can control the information that is accessed and entered by each user of your NetSuite account. SuiteBuilder provides a simple point and click interface for creating custom fields and forms, custom record types, transaction form layouts, and custom centers.

Additionally, the SuiteBundler feature allows you to share customization objects with other NetSuite users.

Who Uses SuiteBuilder?

There are three primary types of user of SuiteBuilder customization features:

- **Administrators** - When a company first begins using NetSuite, administrators spend time customizing transaction forms, adding custom record types, and setting up custom centers for roles within the company to get the rest of the company using the application. Some of this customization work is unique to a specific business. Some of this work can be accomplished by installing previously created 'bundles', or packaged customizations.

  Bundles are created using SuiteBundler. Finding and installing bundles that have been created by other similar businesses can greatly reduce the work required to get a company started using NetSuite.

  Administrators also assign roles to each NetSuite user. These roles determine which information each user has access to in your NetSuite account.

- **IT Staff** - For a company that uses NetSuite, the issues addressed by the IT department often include requests for changes to the NetSuite account. These issues can range from small tasks such as adding a field to a form to larger work items like creating custom record types.

  IT staff members also may have access to the administrative tools that allow them to audit changes to your NetSuite account, manage data in the system, and schedule batch processing jobs.

- **Developers** – For developers of partner solutions and independent software vendors (ISVs), most time with NetSuite is spent coding SuiteScript and SuiteTalk, but in order to work with these features, developers need a solid understanding of how customization objects interact with their code.

NetSuite Anatomy

Making changes to NetSuite is a simple point-and-click process. Whether you have experience with other software applications or not, you can get up and running with NetSuite in no time. First, however, it helps to understand the custom elements you have to work with and how they work together.

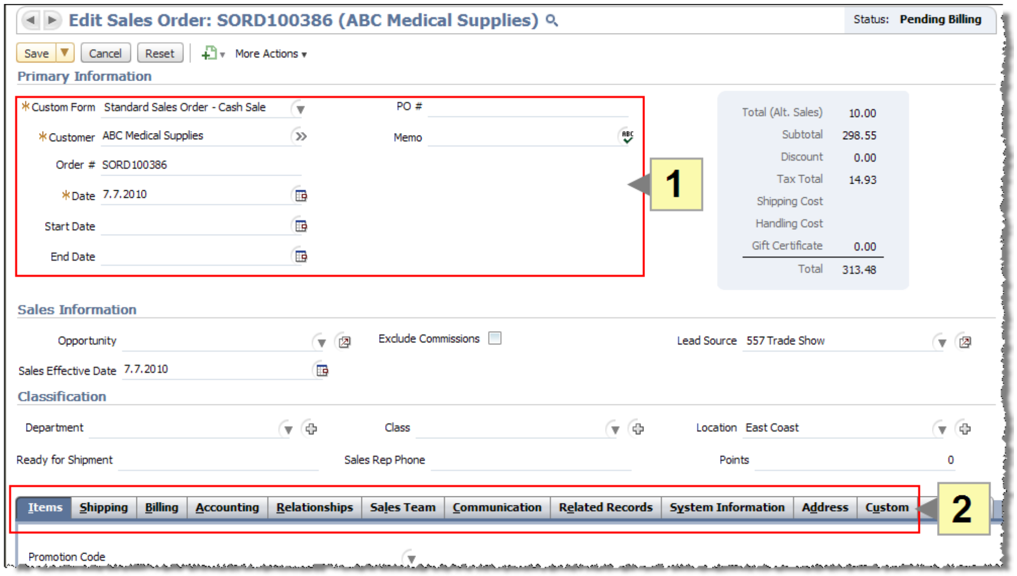
NetSuite uses the following terminology:

- **Record** – a single entry of information related to a single business concept.
- **Form** – page through which you enter records and transactions. Forms contain fields and usually have subtabs.
- **Field** – place on a record or transaction where information is entered.
- **Subtab** – section of a record or transaction that groups similar fields.

  An example of a standard subtab is the Address subtab where the shipping and billing addresses are entered on transactions and records.
- **Custom list** – list of values that can be selected in a custom field.
- **Sublist** – the results of a saved search displayed on a custom or standard record. Sublists can also be generated through parent-child relationships.
- **Script** – SuiteScript JavaScript file that runs against a specific form or record type or that creates a custom portlet. Scripts can also be scheduled to execute periodically.
- **Role** – set of permissions that can be assigned to a NetSuite user.
- **Center** – configuration of NetSuite created for a specific group of roles with similar tasks.
- **Center tab** – section of NetSuite that groups similar links and other information. Standard tabs include Home, Reports, Documents, Activities, and Setup. You can also create custom center tabs.

Understanding forms is key to understanding how to work in NetSuite. The image below shows a sales order form with the following elements highlighted:

1. fields
2. subtabs



Edit Sales Order: SORD100386 (ABC Medical Supplies)

Status: Pending Billing

Primary Information

*Custom Form Standard Sales Order - Cash Sale

PO #

*Customer ABC Medical Supplies

Memo

Order # SORD100386

*Date 7.7.2010

Start Date

End Date

Total (Alt. Sales)	10.00
Subtotal	298.55
Discount	0.00
Tax Total	14.93
Shipping Cost	
Handling Cost	
Gift Certificate	0.00
Total	313.48

Sales Information

Opportunity

Exclude Commissions □

Lead Source 557 Trade Show

Sales Effective Date 7.7.2010

Classification

Department

Class

Location East Coast

Ready for Shipment

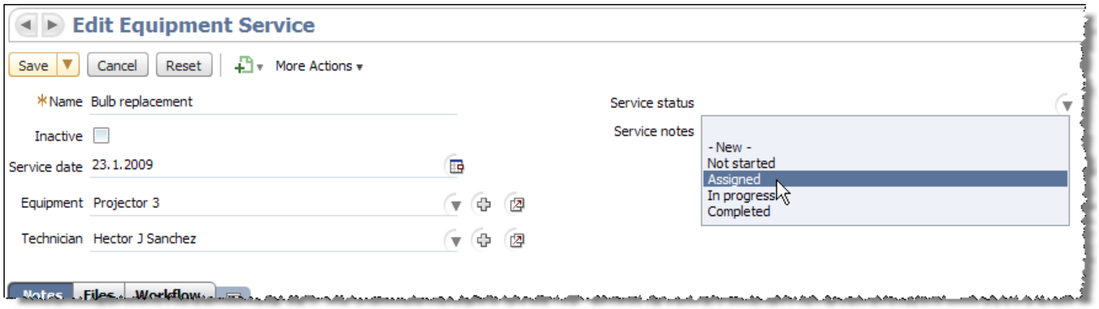
Sales Rep Phone

Points 0

Promotion Code

In the example above, notice that some of the fields allow the person filling out the form to choose from a list of values, and some allow the entry of text or numbers. The Sales Rep Phone field actually pulls the phone number from the employee record of the employee chosen in the Sales Rep field.

While many of the fields shown above allow you to choose from lists defined in your account – Customer, Department, and Class, for example – you can also create your own lists of values to choose from. The example below shows a custom field that uses a custom list.



Edit Equipment Service

*Name Bulb replacement

Inactive □

Service date 23.1.2009

Equipment Projector 3

Technician Hector J Sanchez

Service status

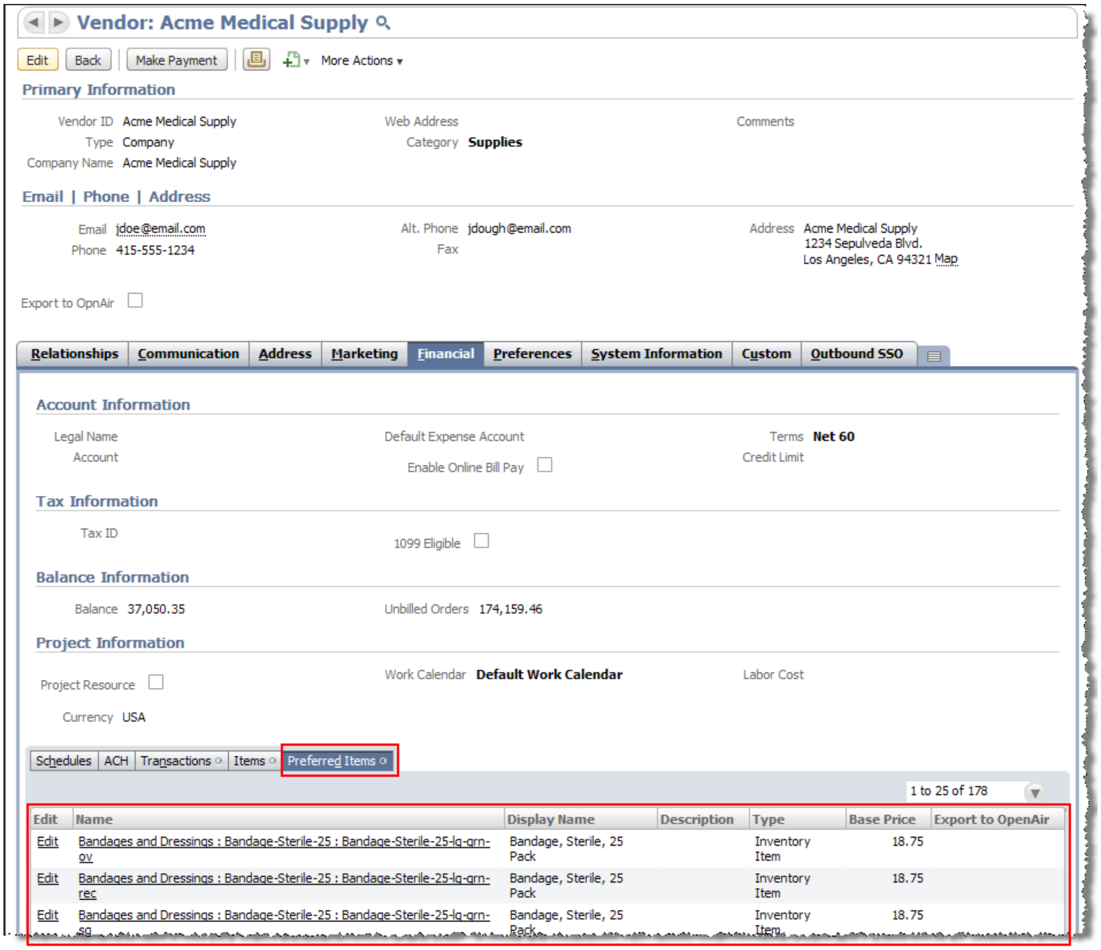
Service notes

- New -
Not started
Assigned
In progress
Completed

For more information on fields, see Custom Fields.

You can create custom versions of most forms in NetSuite to include the fields – both standard and custom – you want to show. Adding custom fields to forms allows you to capture information that is not included in NetSuite by default.

With custom sublists, you can add lists of information to a form. Sublists present saved search results on forms subtabs. In the example below, Wolfe Electronics has added the **Preferred Item** sublist to vendor records. This sublist shows the items most often sold by the vendor, along with unit price and quantity on hand.



Vendor: Acme Medical Supply

Primary Information

Vendor ID Acme Medical Supply
Type Company
Company Name Acme Medical Supply
Web Address
Category **Supplies**
Comments

Email | Phone | Address

Email jdoe@email.com
Phone 415-555-1234
Alt. Phone jdough@email.com
Fax
Address Acme Medical Supply
1234 Sepulveda Blvd.
Los Angeles, CA 94321 Map

Export to OpnAir □

Account Information

Legal Name
Account
Default Expense Account
Enable Online Bill Pay □
Terms **Net 60**
Credit Limit

Tax Information

Tax ID
1099 Eligible □

Balance Information

Balance 37,050.35
Unbilled Orders 174,159.46

Project Information

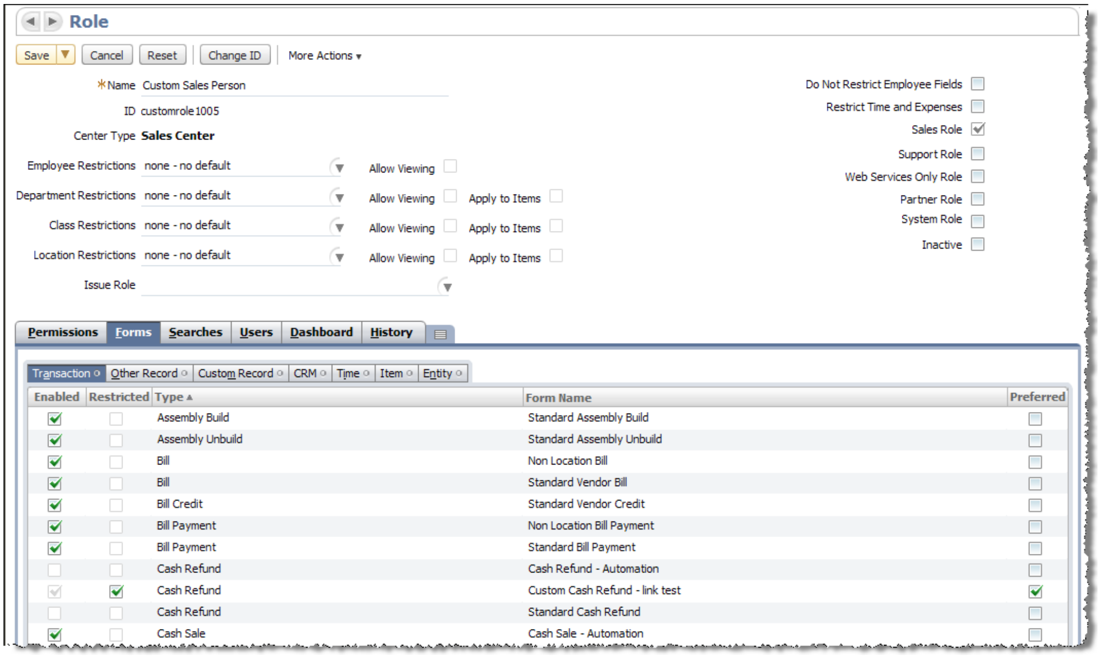
Project Resource □
Currency USA
Work Calendar **Default Work Calendar**
Labor Cost

1 to 25 of 178

Edit	Name	Display Name	Description	Type	Base Price	Export to OpenAir
Edit	Bandages and Dressings : Bandage-Sterile-25 : Bandage-Sterile-25-lg-grn-ov	Bandage, Sterile, 25 Pack		Inventory Item	18.75	
Edit	Bandages and Dressings : Bandage-Sterile-25 : Bandage-Sterile-25-lg-grn-rec	Bandage, Sterile, 25 Pack		Inventory Item	18.75	
Edit	Bandages and Dressings : Bandage-Sterile-25 : Bandage-Sterile-25-lg-grn-sq	Bandage, Sterile, 25 Pack		Inventory Item	18.75	

Customizing forms also allows you to adapt the information captured to specific roles in your company. When you create a form for a role, you choose which fields and subtabs appear on the form and which are hidden. You also have the flexibility of setting a custom form as the preferred (default) form or to restrict the role to only use a specific form.

The Custom Sales Rep role shown below is restricted to use only the Custom Cash Refund form when issuing refunds to customers. Note that the boxes in the Restricted and Preferred columns are checked.



*Name Custom Sales Person

ID customrole1005

Center Type **Sales Center**

Employee Restrictions none - no default Allow Viewing □

Department Restrictions none - no default Allow Viewing □ Apply to Items □

Class Restrictions none - no default Allow Viewing □ Apply to Items □

Location Restrictions none - no default Allow Viewing □ Apply to Items □

Issue Role

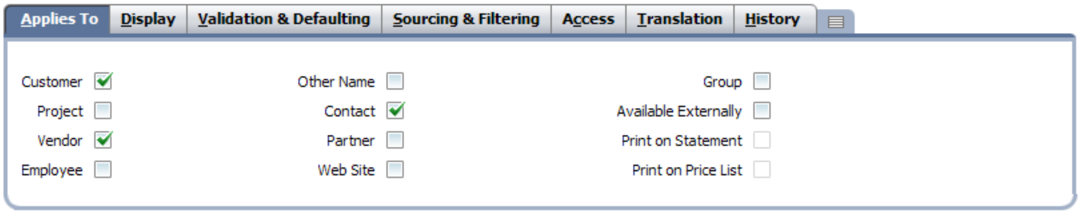
Do Not Restrict Employee Fields □
Restrict Time and Expenses □
Sales Role ☑
Support Role □
Web Services Only Role □
Partner Role □
System Role □
Inactive □

Enabled	Restricted	Type ▲	Form Name	Preferred
☑	□	Assembly Build	Standard Assembly Build	□
☑	□	Assembly Unbuild	Standard Assembly Unbuild	□
☑	□	Bill	Non Location Bill	□
☑	□	Bill	Standard Vendor Bill	□
☑	□	Bill Credit	Standard Vendor Credit	□
☑	□	Bill Payment	Non Location Bill Payment	□
☑	□	Bill Payment	Standard Bill Payment	□
□	□	Cash Refund	Cash Refund - Automation	□
☑	☑	Cash Refund	Custom Cash Refund - link test	☑
□	□	Cash Refund	Standard Cash Refund	□
☑	□	Cash Sale	Cash Sale - Automation	□

For more information on forms, see Custom Forms.

SuiteBuilder allows standard NetSuite records to be customized by adding custom fields, subtabs, or sublists or through the removal of existing fields. NetSuite's standard records can support many of your business needs, but with SuiteBuilder you can create your own custom record types to fill in the gaps.

The image below shows the **Applies To** subtab of a custom entity field. This is where you choose which standard records your custom field appears on.



Customer ☑ Other Name □ Group □
Project □ Contact ☑ Available Externally □
Vendor ☑ Partner □ Print on Statement □
Employee □ Web Site □ Print on Price List □

Just as with standard records, custom records can be designed to be associated with other custom records or with standard records. This includes instances where you might need a simple lookup relationship where one type of record refers to another as well as a situation where a tighter parent-child relationship is needed.

You can create custom forms for your record types to give you finer control on how different roles within your company are able to interact with your custom records. For more information on record types, see Custom Records.

If you need even more flexibility and automation, you can use SuiteScript, our JavaScript-based API. You can use SuiteScript to perform validations and calculations on forms, to access and update records when fields are changed or pages are submitted, or to automatically create or update records. For more information on what you can do with SuiteScript, see the help topic *SuiteScript - The Basics*.

Now, that you have a general idea of how NetSuite's customization objects work together, read Customizing Your NetSuite Account for an overview of what is possible with SuiteBuilder.

Customizing Your NetSuite Account

Now that you understand what you can do with NetSuite's customization features, you are ready to begin setting up your account.

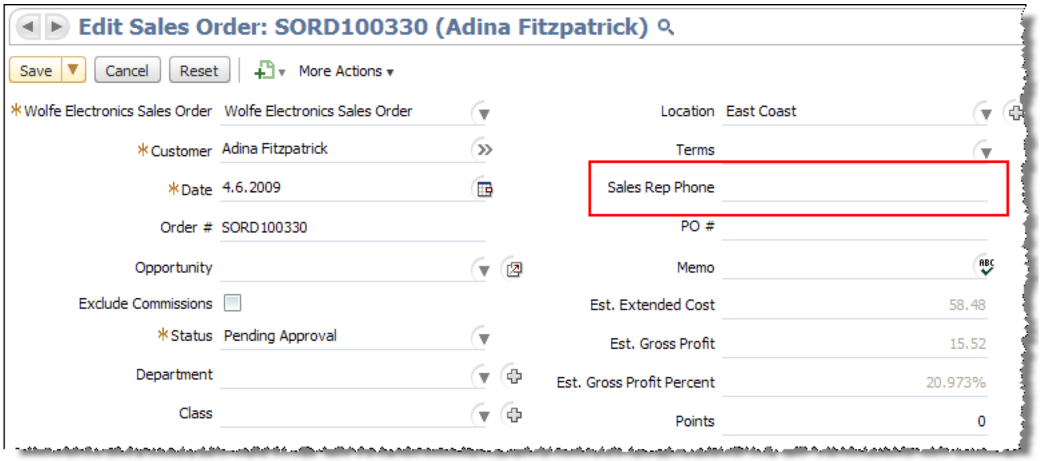
- Building Custom Forms
- Creating Custom Record Types
- Custom Centers
- SuiteBundler

Building Custom Forms

All information in NetSuite is entered through forms. There are forms for every record you can store in your NetSuite account, including transactions, relationships like customers, contacts, and vendors, as well as CRM records like events, tasks, and campaigns. NetSuite provides a full array of standard forms out of the box, but with SuiteBuilder, you can make the changes you need to ensure that you are capturing the information from forms that your business needs.

Forms are made up of fields in which you enter data. If a piece of information you need is not included on a form, you can create your own custom fields and place them there. When the form is filled out and submitted, the information in the custom field is stored just like any standard field.

On the custom sales order form below, you can see that a custom field has been added to show the phone number of the sales rep who made the sale.

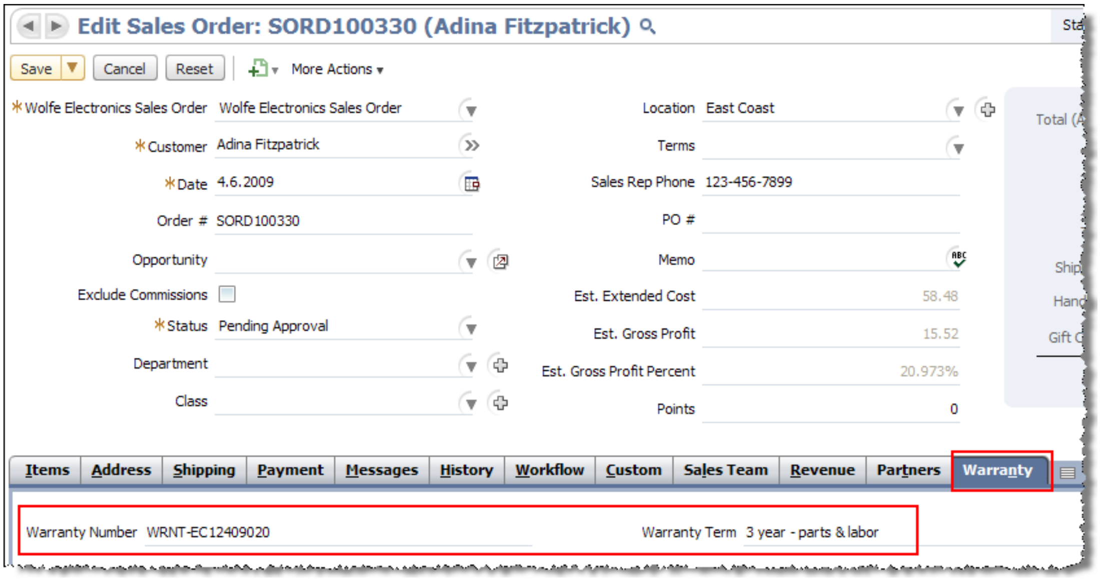


Edit Sales Order: SORD100330 (Adina Fitzpatrick)

*Wolfe Electronics Sales Order	Wolfe Electronics Sales Order	Location	East Coast
*Customer	Adina Fitzpatrick	Terms	
*Date	4.6.2009	Sales Rep Phone	
Order #	SORD100330	PO #	
Opportunity		Memo	
Exclude Commissions	□	Est. Extended Cost	58.48
*Status	Pending Approval	Est. Gross Profit	15.52
Department		Est. Gross Profit Percent	20.973%
Class		Points	0

Most forms are divided into subtabs. Subtabs group fields with similar information in one place. For example, the Address subtab is where shipping and billing addresses are entered. You can create subtabs for custom fields.

On the sales order form below, a custom **Warranty** subtab has been added to track warranty information through the **Warranty Number** and **Warranty Term** custom fields.



Edit Sales Order: SORD100330 (Adina Fitzpatrick)

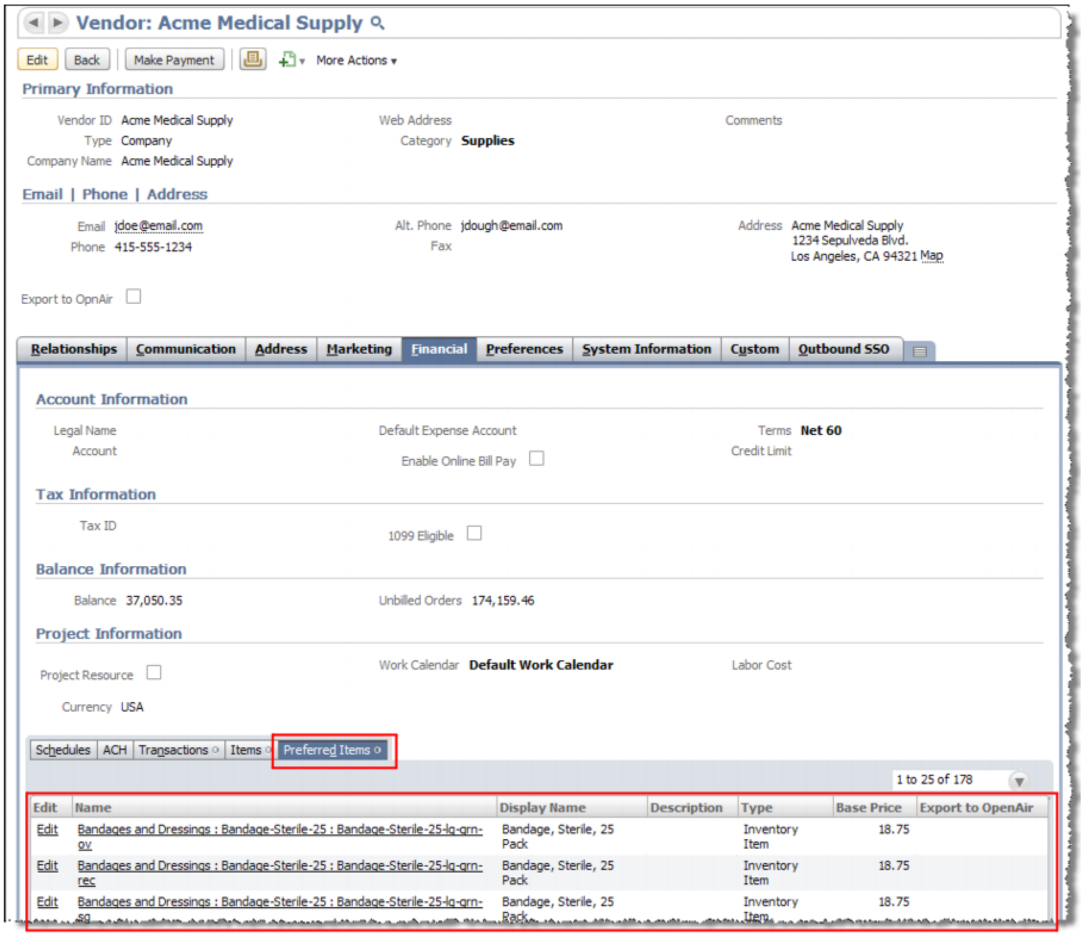
*Wolfe Electronics Sales Order	Wolfe Electronics Sales Order	Location	East Coast
*Customer	Adina Fitzpatrick	Terms	
*Date	4.6.2009	Sales Rep Phone	123-456-7899
Order #	SORD100330	PO #	
Opportunity		Memo	
Exclude Commissions	□	Est. Extended Cost	58.48
*Status	Pending Approval	Est. Gross Profit	15.52
Department		Est. Gross Profit Percent	20.973%
Class		Points	0

Warranty Number WRNT-EC12409020
Warranty Term 3 year - parts & labor

You can also add custom sublists to your forms. Sublists display saved search results relating to the record on which they are shown. For example, you might add a custom sublist to a custom vendor form that shows all of the items that most often purchased from the vendor.

First, you create an item saved search. On the Results subtab of the search, you add the columns you want to show on the sublist. On the Available Filters subtab, you select **Vendor: Name/**

ID. This filter is used by the sublist to only show the items purchased from a vendor. For more information on saved search filters, see the help topic *Selecting Available Filters for Saved Searches*.



Vendor: Acme Medical Supply

Primary Information

Vendor ID Acme Medical Supply
Type Company
Company Name Acme Medical Supply
Web Address
Category **Supplies**
Comments

Email | Phone | Address

Email jdoe@email.com
Phone 415-555-1234
Alt. Phone jdough@email.com
Fax
Address Acme Medical Supply
1234 Sepulveda Blvd.
Los Angeles, CA 94321 Map

Export to OpnAir □

Account Information

Legal Name
Account
Default Expense Account
Enable Online Bill Pay □
Terms **Net 60**
Credit Limit

Tax Information

Tax ID
1099 Eligible □

Balance Information

Balance 37,050.35
Unbilled Orders 174,159.46

Project Information

Project Resource □
Currency USA
Work Calendar **Default Work Calendar**
Labor Cost

1 to 25 of 178

Edit	Name	Display Name	Description	Type	Base Price	Export to OpenAir
Edit	Bandages and Dressings : Bandage-Sterile-25 : Bandage-Sterile-25-lg-grn-ov	Bandage, Sterile, 25 Pack		Inventory Item	18.75	
Edit	Bandages and Dressings : Bandage-Sterile-25 : Bandage-Sterile-25-lg-grn-rec	Bandage, Sterile, 25 Pack		Inventory Item	18.75	
Edit	Bandages and Dressings : Bandage-Sterile-25 : Bandage-Sterile-25-lg-grn-sg	Bandage, Sterile, 25 Pack		Inventory Item	18.75	

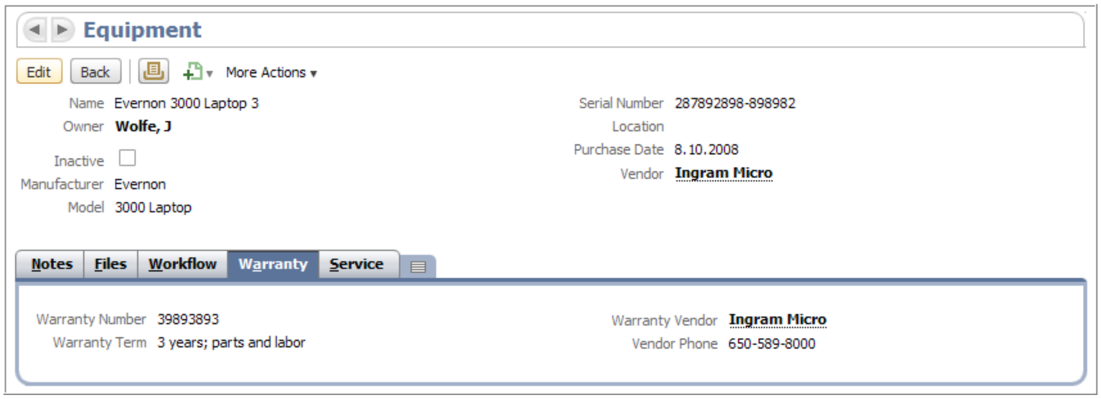
For more information, see Custom Fields, Custom Sublists, and Custom Forms.

Creating Custom Record Types

When you need to create records that are specific to the needs of your business but are not included in NetSuite out of the box, you can create custom record types.

When you create a custom record type, you must also define the custom fields that make up the record.

For example, your company might need to track information on computer and electronic equipment. Since there is not a standard record of this type in NetSuite, you could create a custom record type called Equipment and add custom fields in which you enter serial numbers, location, purchase date, and service and warranty information.



Name Evernon 3000 Laptop 3
Owner **Wolfe, J**
Inactive □
Manufacturer Evernon
Model 3000 Laptop

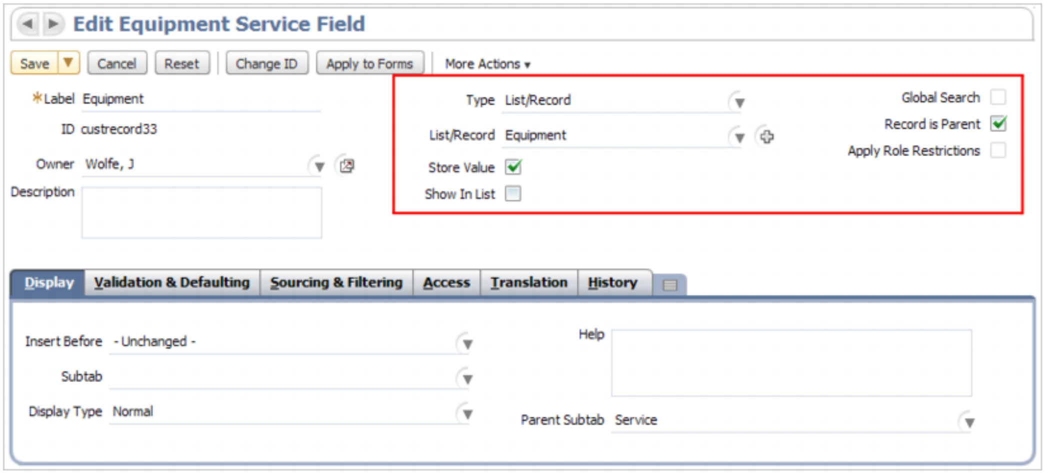
Serial Number 287892898-898982
Location
Purchase Date 8.10.2008
Vendor **Ingram Micro**

Warranty Number 39893893
Warranty Term 3 years; parts and labor

Warranty Vendor **Ingram Micro**
Vendor Phone 650-589-8000

Your custom records can be attached to standard records and other custom records. For example, on the equipment record type shown above, you might want to track details each time you service your equipment. You create a new custom record, Equipment Service, used to track these details.

To create this parent-child relationship, you add a field to the Equipment Service record in which you can select the Equipment record it is associated with. The field type is List/Record, and Equipment is chosen in the List/Record field. Checking the Record is Parent box creates the parent-child relationship.



*Label Equipment
ID custrecord33
Owner Wolfe, J
Description

Type List/Record
List/Record Equipment
Store Value ☑
Show In List □

Global Search □
Record is Parent ☑
Apply Role Restrictions □

Insert Before - Unchanged -
Subtab
Display Type Normal

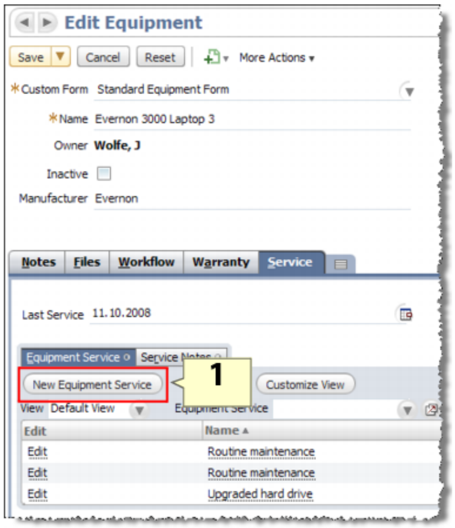
Help
Parent Subtab Service

When service is performed, the technician opens the Equipment record, clicks the Service subtab, and clicks a button to create the Equipment Service record.

1. Click the New Equipment Service button to create a new service record.

2. Enter the service information in the new record. The Equipment field is filled in automatically.

3. The child service record shows on the parent Equipment record.



*Custom Form Standard Equipment Form

*Name Evernon 3000 Laptop 3

Owner **Wolfe, J**

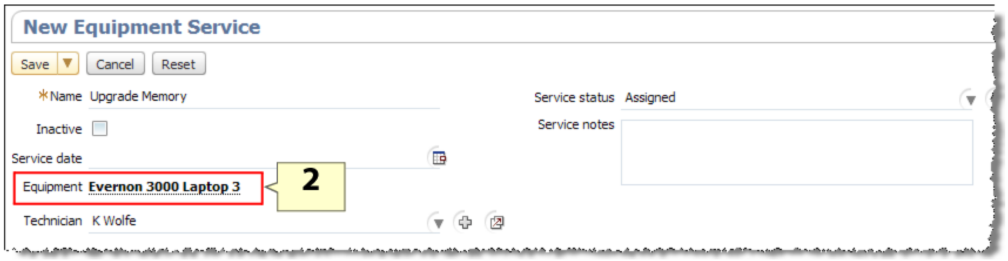
Inactive □

Manufacturer Evernon

Last Service 11.10.2008

View Default View Equipment Service

Edit	Name ▴
Edit	Routine maintenance
Edit	Routine maintenance
Edit	Upgraded hard drive



*Name Upgrade Memory

Service status Assigned

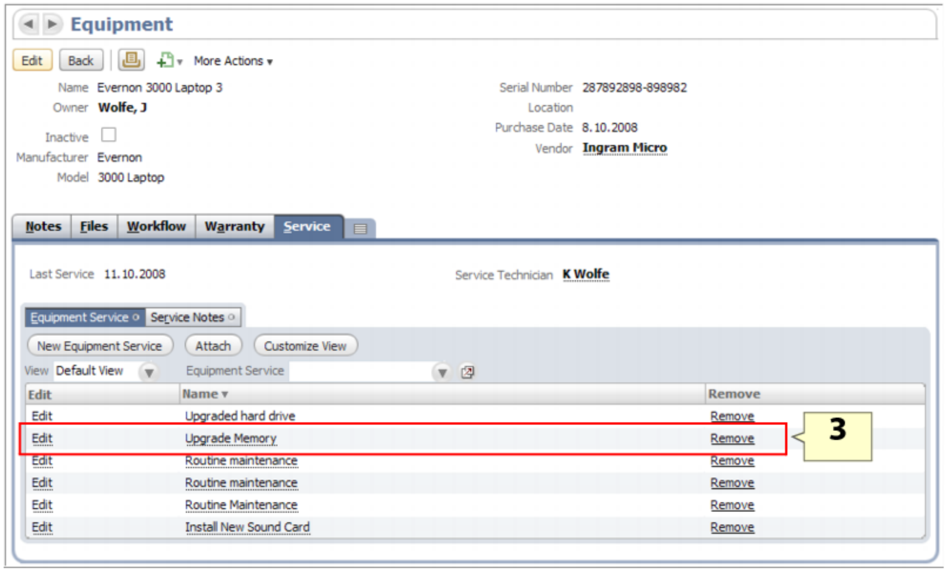
Inactive □

Service notes

Service date

Equipment **Evernon 3000 Laptop 3**

Technician K Wolfe



Name Evernon 3000 Laptop 3

Owner **Wolfe, J**

Inactive □

Manufacturer Evernon

Model 3000 Laptop

Serial Number 287892898-898982

Location

Purchase Date 8.10.2008

Vendor **Ingram Micro**

Last Service 11.10.2008

Service Technician **K Wolfe**

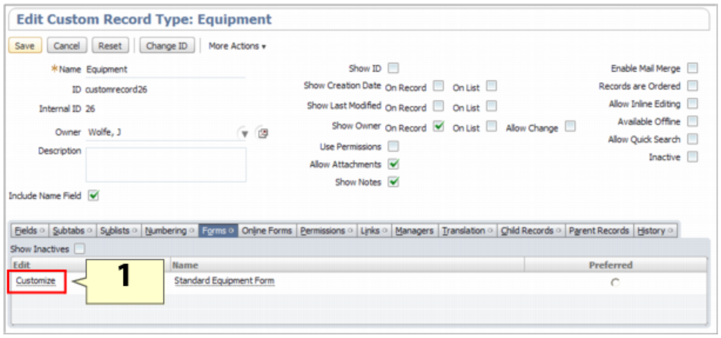
View Default View Equipment Service

Edit	Name ▾	Remove
Edit	Upgraded hard drive	Remove
Edit	Upgrade Memory	Remove
Edit	Routine maintenance	Remove
Edit	Routine maintenance	Remove
Edit	Routine Maintenance	Remove
Edit	Install New Sound Card	Remove

Just as with standard transactions and records, you can customize the forms used to enter custom records and even set up forms for different roles. Customizing a form involves determining which fields show, how these fields are arranged, and which roles use the form.

The equipment record in the example above is most useful to the IT staff who have to purchase, maintain, and track the equipment, but there is some information that is generally useful to others in the company. A custom form is created for the record.

1. On the Custom Record Type page on the Forms subtab, click Customize next to the form.
2. On the Custom Entry Form page, clear the box in the Show column next to the Warranty and Service subtabs.
3. Make sure the Use Permissions box is checked, and on the Permissions subtab, restrict your non-IT roles to use only the new custom equipment form.
4. When someone with a non-IT role views the equipment record, the hidden fields and subtabs are not shown.



Edit Custom Record Type: Equipment

*Name Equipment
ID customrecord26
Internal ID 26
Owner Wolfe, J
Description
Include Name Field ☑

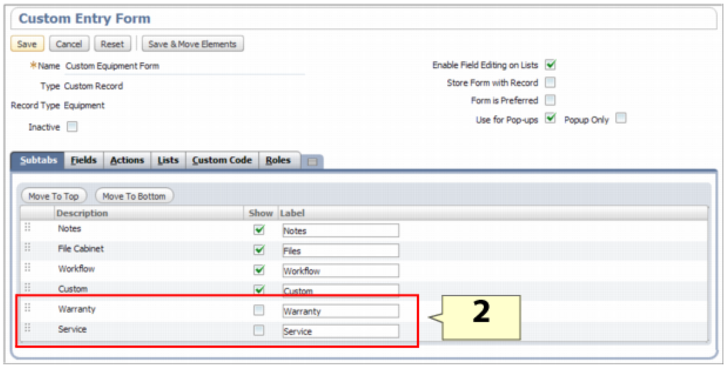
Show ID □
Show Creation Date On Record □ On List □
Show Last Modified On Record □ On List □
Show Owner On Record ☑ On List □ Allow Change □
Use Permissions □
Allow Attachments ☑
Show Notes ☑

Enable Mail Merge □
Records are Ordered □
Allow Inline Editing □
Available Offline □
Allow Quick Search □
Inactive □

Show Inactives □

Edit	Name	Preferred
Customize	Standard Equipment Form	○

1



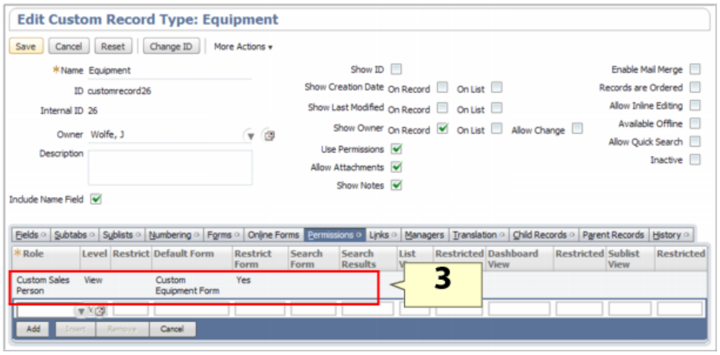
Custom Entry Form

*Name Custom Equipment Form
Type Custom Record
Record Type Equipment
Inactive □

Enable Field Editing on Lists ☑
Store Form with Record □
Form is Preferred □
Use for Pop-ups ☑ Popup Only □

Description	Show	Label
Notes	☑	Notes
File Cabinet	☑	Files
Workflow	☑	Workflow
Custom	☑	Custom
Warranty	□	Warranty
Service	□	Service

2



Edit Custom Record Type: Equipment

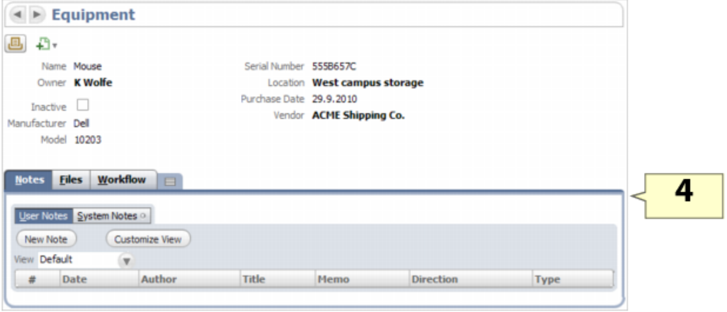
*Name Equipment
ID customrecord26
Internal ID 26
Owner Wolfe, J
Description
Include Name Field ☑

Show ID □
Show Creation Date On Record □ On List □
Show Last Modified On Record □ On List □
Show Owner On Record ☑ On List □ Allow Change □
Use Permissions ☑
Allow Attachments ☑
Show Notes ☑

Enable Mail Merge □
Records are Ordered □
Allow Inline Editing □
Available Offline □
Allow Quick Search □
Inactive □

*Role	Level	Restrict	Default Form	Restrict Form	Search Form	Search Results	List V	Restricted	Dashboard View	Restricted	Sublist View	Restricted
Custom Sales Person	View		Custom Equipment Form	Yes								

3



Equipment

Name Mouse
Owner **K Wolfe**
Inactive □
Manufacturer Dell
Model 10203

Serial Number 555B657C
Location **West campus storage**
Purchase Date 29.9.2010
Vendor **ACME Shipping Co.**

View Default

#	Date	Author	Title	Memo	Direction	Type

4

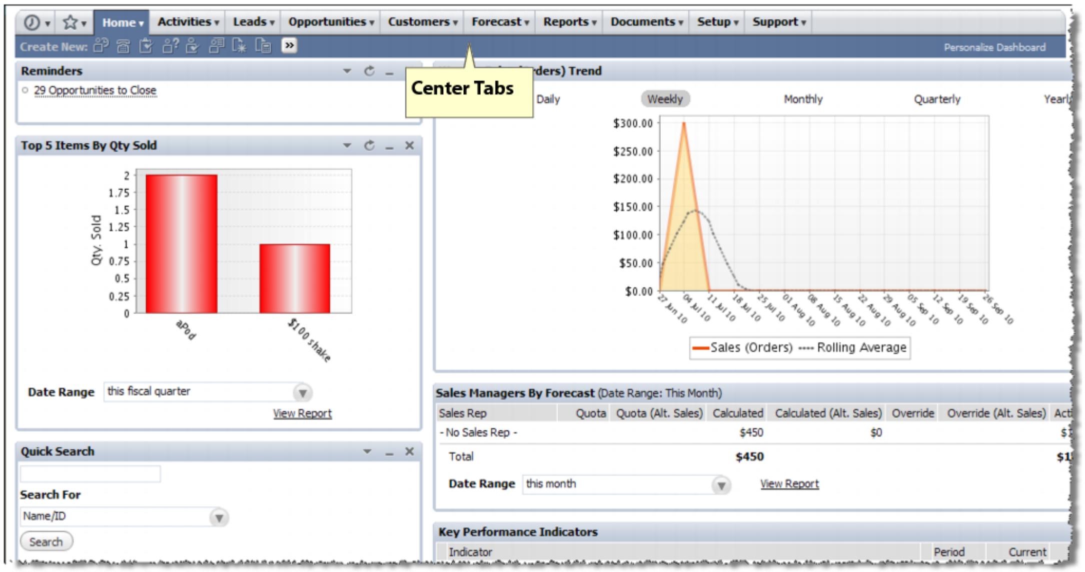
With this custom record form, employees outside of your IT department can view basic information about this piece of equipment like manufacturer and model and can read the owners manual, but they cannot see information related to service and warranty that is only relevant to your IT staff.

For more information, see Custom Records.

Custom Centers

When employees are assigned roles in NetSuite, they are granted access to the NetSuite pages necessary to complete their work.

Each standard role has access to what is called a **center**. A center is a configuration of NetSuite created for a group of roles with similar tasks. For example, all sales roles – sales reps, sales managers, and sales administrators – use the Sales Center by default. While the information available to each role differs, the basic layout of the Sales Center is the same for each of the standard sales roles.



Date Range: this fiscal quarter

View Report

Search For: Name/ID

Sales Managers By Forecast (Date Range: This Month)

Sales Rep	Quota	Quota (Alt. Sales)	Calculated	Calculated (Alt. Sales)	Override	Override (Alt. Sales)	Acti
- No Sales Rep -			$450	$0			$1
Total			$450				$1

Date Range: this month

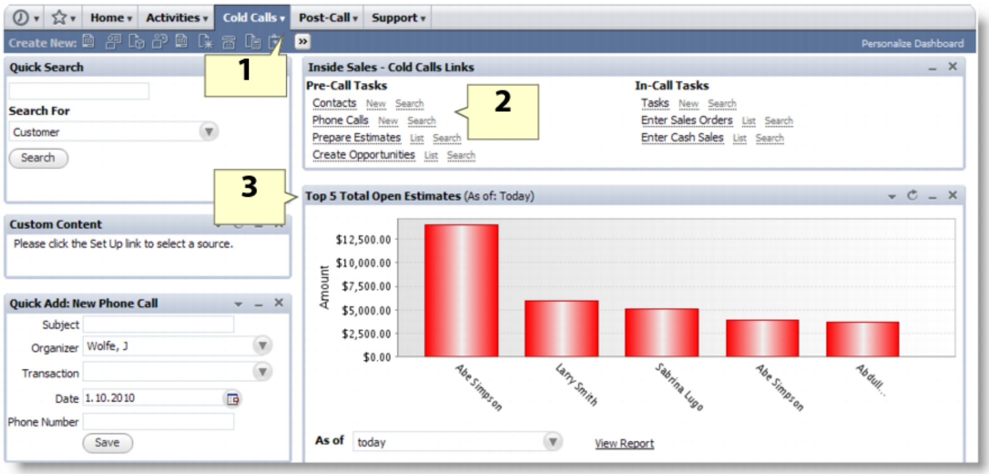
View Report

With SuiteBuilder, you can create custom centers and assign them to custom roles. This lets you completely control a role's experience with NetSuite. You can create custom tabs, add portlets, and add links to the tasks needed by the role.

The example below is a custom center created for inside sales reps. Notice the following:

1. There are two custom center tabs in this center: Cold Calls and Post-Call.
2. This links portlet contains categories and links to the pages users with this role need. “Pre-Call Tasks” is a category.

3. You choose the portlets that can be shown on each tab in the center.



Customer

Please click the Set Up link to select a source.

Subject
Organizer Wolfe, J
Transaction
Date 1.10.2010
Phone Number

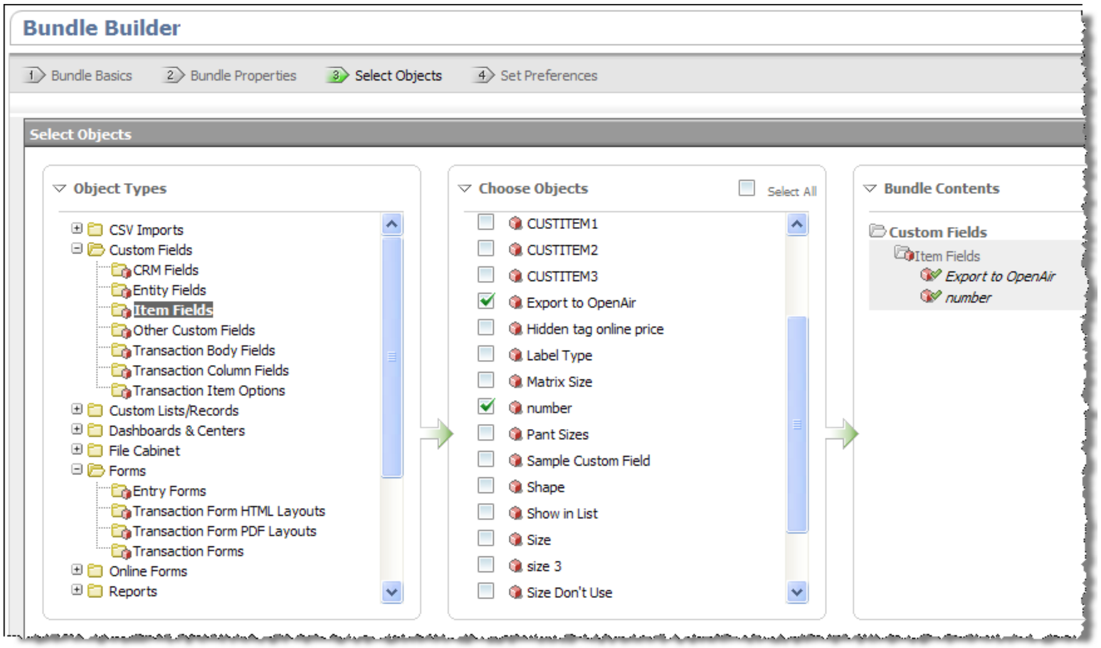
As of today

You can only grant custom center access to roles you made from scratch. You cannot give custom center access to standard roles or to customized version of standard roles.

For steps on creating custom centers, see Custom Centers.

SuiteBundler

The SuiteBundler feature allows you to share customization objects with other NetSuite accounts. When SuiteBundler authors create a bundle, they select all of the objects they want to include.



☐ Select All

☐ CUSTITEM1
☐ CUSTITEM2
☐ CUSTITEM3
☑ Export to OpenAir
☐ Hidden tag online price
☐ Label Type
☐ Matrix Size
☑ number
☐ Pant Sizes
☐ Sample Custom Field
☐ Shape
☐ Show in List
☐ Size
☐ size 3
☐ Size Don't Use

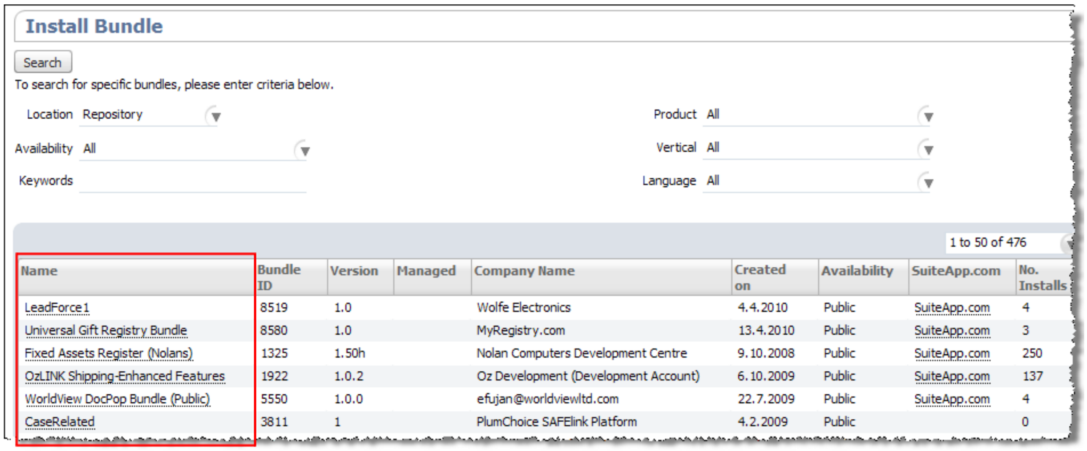
You can install any bundle that is shared publicly in the SuiteSource Repository, or you can install a bundle that has been shared with your account specifically by an administrator in another account.

You can also install bundles created in the accounts of other NetSuite customers. For example, while attending a NetSuite administrators workshop, you meet the administrator of a company much like yours, and in the course of your conversation, your new acquaintance mentions her recent work on a customer survey in NetSuite. The survey uses custom record types, custom fields and forms, as well as some saved searches that compile the results into reports. She offers to bundle these objects and to share them with you, saving you what could have been many days of work.

If your company uses the NetSuite Sandbox feature for development and testing, you can create a bundle of all of the parts of your customization in a sandbox account and install that bundle in your company's NetSuite production account. The bundle can be deployed from one sandbox account to another as well. You can also choose to share your bundle with other NetSuite companies, maybe saving someone else the time and energy required to build customizations from scratch.

You can peruse the list of bundles that are available to you on the Install Bundles page. Be sure to carefully review the documentation for a bundle before installing it.

In the name column of the Install Bundle page, click a bundle to go to the details page for that bundle. On the details page, be sure to click the Documentation link to read about bundle before installing it.



To search for specific bundles, please enter criteria below.

Location: Repository
Availability: All
Keywords:
Product: All
Vertical: All
Language: All

1 to 50 of 476

Name	Bundle ID	Version	Managed	Company Name	Created on	Availability	SuiteApp.com	No. Installs
LeadForce1	8519	1.0		Wolfe Electronics	4.4.2010	Public	SuiteApp.com	4
Universal Gift Registry Bundle	8580	1.0		MyRegistry.com	13.4.2010	Public	SuiteApp.com	3
Fixed Assets Register (Nolans)	1325	1.50h		Nolan Computers Development Centre	9.10.2008	Public	SuiteApp.com	250
OzLINK Shipping-Enhanced Features	1922	1.0.2		Oz Development (Development Account)	6.10.2009	Public	SuiteApp.com	137
WorldView DocPop Bundle (Public)	5550	1.0.0		efujan@worldviewltd.com	22.7.2009	Public	SuiteApp.com	4
CaseRelated	3811	1		PlumChoice SAFElink Platform	4.2.2009	Public		0

For more information, see the help topic *SuiteBundler Overview*.

Chapter 2 Custom Fields

Custom fields are fields that you can add to your records and transactions to record information specific to your business needs. Record custom fields can be added to existing and custom subtabs on the entry forms you use to enter records in your NetSuite account. Transaction custom fields can be added to the top (body) or the line items (columns) of transactions.

To create a custom field you need to do the following:

- Set the basic properties for the field
- Assign the field to the desired forms
- Set the display properties for the field
- Set the desired validation and defaulting properties
- Create dynamic defaults and hyperlinks
- Set any sourcing criteria for the field
- Set any access restrictions to the field based on department, role, or subsidiary
- Set any filtering criteria for the field

For step-by-step information on how to create and define a custom field, see Creating a Custom Field.

Kinds of Custom Fields

Following are the different types of custom record and transaction fields you can create:

- Custom CRM Fields – Used to add fields to CRM records. These records include Activity, Marketing and Support records — such as tasks, events, campaigns or cases.
- Custom Entity Fields – Used to add fields to entity records. These records include Relationship and Employee records — such as customers, vendors, employees, contacts, partners or groups.
- Custom Item Fields – Used to add fields to item records. These records include Accounting and Web site item records — such as inventory, non-inventory, service, other charge, group, kit/package and assembly/bill of materials item records.
- Custom Transaction Body Fields – Used to add fields to the body of transaction records. These records include purchase, sale, journal entry and expense report records — such

as sales orders, invoices, purchase orders, opportunities, Web store transactions or item receipts.

- Custom Transaction Column Fields – Used to add fields to the columns of transaction records. These fields display in the line-item columns of transaction records and include fields such as expense items, purchase items, sales items, store items or opportunity items.
- Custom Transaction Item Options – Used to add fields to the line items of your transaction records such as purchase items, sales items and Web store items. When adding a custom field to the line items of a transaction, you apply the field to the type of line item.
- Custom Item Number Fields – Used to add fields to serial and lot numbered inventory records to track information specific to each item or workflow unique to your business.This could include quality control procedures or recall information.
- Other Custom Fields – Use to add fields to custom records not defined by the above categories, including campaign events, classes, competitors, departments, and locations.

You can differentiate between custom fields and standard fields on pages in NetSuite if you enable the Show Internal IDs preference at Home > Set Preferences under the Defaults section of the General subtab. With this preference enabled, when you click on a field, the field level help popup shows a field ID in the bottom right corner. If the field you clicked on is a custom field, the ID from the custom field record is shown.

Custom CRM Fields

Custom CRM fields are fields that you can add to your CRM records to gather information specific to your business needs.

These records include:

- task records
- phone call records
- campaign records
- case records
- solution records
- event records

To create or modify custom CRM fields:

1. Go to Setup > Customization > CRM Fields.

In the Custom CRM Fields page, each custom field is listed, with columns providing detailed information about the field and which records the field has been applied to.

2. To edit an existing custom CRM field, click the field name in the description column and then modify the field definition as desired.

3. To add a new custom CRM field, click **New**.

Note: As of Version 2009 Release 1, custom CRM fields can be indexed for global search. To include a custom field in global searches, check the Global Search box on its record. You cannot index a custom field for global search if None is selected for any Level for Search/Reporting option on the Access subtab of the custom field record.

For a detailed description of all types of custom fields that you can create, see Kinds of Custom Fields . For detailed information on how to create or modify custom CRM fields, see Creating a Custom Field.

Custom Entity Fields

Custom entity fields are fields that you can add to your entity records to gather information specific to your business needs. Entity custom fields can be added to existing and custom subtabs on the entry forms you use to enter entity records in your NetSuite account.

These records include the following relationship and employee records:

- customer records
- project records
- vendor records
- other name records
- contact records
- partner records
- entity group records
- employee records
- Web site registration

To create or modify custom Entity fields:

1. Go to Setup > Customization > Entity Fields.

In the Custom Entity Fields page, each custom field is listed, with columns providing detailed information about the field and which records the field has been applied to.

2. To edit an existing custom Entity field, click the field name in the description column and then modify the field definition as desired.
3. To add a new custom Entity field, click **New**.

Note: As of Version 2009 Release 1, custom Entity fields can be indexed for global search. To include a custom field in global searches, check the Global Search box on its record. You cannot index a custom field for global search if None is selected for any Level for Search/Reporting option on the Access subtab of the custom field record.

For a detailed description of all types of custom fields that you can create, see Kinds of Custom Fields . For detailed information on how to create or modify custom Entity fields, see Creating a Custom Field.

Custom Item Fields

Custom Item fields are fields that you can add to your Item records to gather information specific to your business needs.

These records include the following Accounting and Web site item records:

- inventory item
- non-inventory items
- service items
- other charges
- item groups
- kit/packages
- assembly/bill of materials

To create or modify custom Item fields:

1. Go to Setup > Customization > Item Fields.

   In the Custom Item Fields page, each custom field is listed, with columns providing detailed information about the field and which records the field has been applied to.

2. To edit an existing custom Item field, click the field name in the description column and then modify the field definition as desired.

3. To add a new custom Item field, click **New**.

Note: As of Version 2009 Release 1, custom Item fields can be indexed for global search. To include a custom field in global searches, check the Global Search box on its record. You cannot index a custom field for global search if None is selected for any Level for Search/Reporting option on the Access subtab of the custom field record.

For a detailed description of all types of custom fields that you can create, see Kinds of Custom Fields . For detailed information on how to create or modify custom Item fields, see Creating a Custom Field.

Custom Transaction Body Fields

Custom Transaction Body fields are fields that you can add to the body of transaction records to gather information specific to your business needs.

These records include:

- purchase transactions
- sales transactions
- opportunities
- journal entries
- expense reports
- Web store checkout
- item receipts
- item fulfillments

To create or modify custom Transaction Body fields:

1. Go to Setup > Customization > Transaction Body Fields.

   In the Custom Transaction Body Fields page, each custom field is listed, with columns providing detailed information about the field and which records the field has been applied to.

2. To edit an existing custom Transaction Body field, click the field name in the description column and then modify the field definition as desired.

3. To add a new custom Transaction Body field, click **New**.

Note: As of Version 2009 Release 1, custom Transaction Body fields can be indexed for global search. To include a custom field in global searches, check the Global Search box on its record. You cannot index a custom field for global search if None is selected for any Level for Search/Reporting option on the Access subtab of the custom field record.

For a detailed description of all types of custom fields that you can create, see Kinds of Custom Fields . For detailed information on how to create or modify custom Transaction Body fields, see Creating a Custom Field.

Custom Transaction Column Fields

Custom Transaction Column fields are fields that you can add to the line items of your transaction records to gather information specific to your business needs.

These records include:

- expense items
- purchase items
- sale items
- store items
- journal entries
- expense reports
- item receipts or fulfillments
- opportunity items
- time cards

To create or modify custom Transaction Column fields:

1. Go to Setup > Customization > Transaction Column Fields.

   In the Custom Transaction Column Fields page, each custom field is listed, with columns providing detailed information about the field and which records the field has been applied to.

2. To edit an existing custom Transaction Column field, click the field name in the description column and then modify the field definition as desired.
3. To add a new custom Transaction Column field, click **New**.

For a detailed description of all types of custom fields that you can create, see Kinds of Custom Fields . For detailed information on how to create or modify custom Transaction Column fields, see Creating a Custom Field.

Adding Custom Fields to Transaction Forms

After you create a custom transaction field, it automatically appears on standard forms. However, it does not automatically appear on custom forms.

To add a custom transaction field to a custom form:

1. Go to Setup > Customization > Transaction Forms.
2. Click Edit next to your custom form.
3. On the Custom Transaction Form page, click the Screen Fields subtab.
4. Click the subtab you specified on the custom field record and locate your custom field name in the Description column.
5. Check the box in the Show column next to your custom field if you want this field to show when the transaction is being entered.
6. Click Printing Fields.
7. Click the subtab for the area of the printed form your field appears in.

   For body fields, click Body.

   For column fields and item options, click Columns.
8. Check the box in the Print/E-mail column next to your custom field if you want this field to show up when you print or e-mail this form.
9. Enter a label for each field as you want them to print.
10. When you have finished, click Save.

Your custom transaction field now shows up on your custom transaction form.

To use a custom transaction form, go to the Transactions page and click the link for the type of transaction you want to enter. When the data-entry page appears, select the appropriate form in the Custom Form field.

Custom Transaction Item Options

Transaction item options are fields that you can add to the line items of your transaction records to gather information specific to your business needs.

Item options can refer to characteristics of an item. For example, you might record different colors of an item as item options.

You can create item options for:

- purchase items
- sales items
- opportunities
- Web store items
- kits and assemblies

To create or modify custom Transaction Item Option fields:

1. Go to Setup > Customization > Lists, Records, & Fields > Transaction Item Option.

   In the Custom Transaction Item Option Fields page, each custom field is listed, with columns providing detailed information about the field and which records the field has been applied to.
2. To edit an existing custom Transaction Item Option field, click the field name in the description column and then modify the field definition as desired.
3. To add a new custom Transaction Item Option field, click **New**.
4. Complete fields on the Item Option page and click Save.

Note: You can define a formula for an item option's default value (on the Validation & Defaulting subtab), but formula references to non-item option fields are not supported. You may be able to save an item option successfully with this kind of reference, but users will encounter errors when they try to save a line item with this item option selected.

Offer your customers free promotional gift wrapping

Once you have enabled the Item Options feature, you can offer your customers complementary gift wrapping on select items as a promotion. With item options, create a Gift Wrapping checkbox on items so customers can indicate if they would like gift wrapping.

To create an item option for gift wrapping:

1. Go to Setup > Customization > Transaction Item Options > New.
2. In the Label field, enter the name of your option, Gift Wrapping.
3. In the Type field, choose Check Box to create a check box field.

4. On the Applies To subtab, check the Sale and Web Store boxes for this option to appear on sales transactions and in your Web site.
5. In the Items field, select the individual items you want to offer gift wrapping for.

   You can choose multiple items by holding down the Ctrl key as you select the items with the mouse.
6. On the Display subtab, in the Label for Input field, enter the name for this option as you want it to appear to customers on your Web site.
7. When you have finished, click Save.

Now, you can offer customers complementary gift wrapping for items. The choice your customers make appears in the Options column of each sales transaction.

For a detailed description of all types of custom fields that you can create, see Kinds of Custom Fields . For detailed information on how to create or modify custom Transaction Item Option fields, see Creating a Custom Field.

Custom Item Number Fields

Custom item number fields are fields that you can add inventory records to track information specific to each item or workflow unique to your business. For example, you can track the status and results of quality control procedures specific to each serialized item, or you can track recall information on lot records.

These records include:

- serial numbered items
- lot numbered items

To create or modify custom item number fields:

1. Go to Setup > Customization > Lists, Records and Fields > Item Number Fields.

   In the Custom Item Number Field page, each custom field is listed, with columns providing detailed information about the field and which records the field has been applied to.
2. To edit an existing custom item number field, click the field name in the description column and then modify the field definition as desired.

   To add a new custom item number field, click **New**.

When you create a lot numbered or serialized inventory item or edit an existing record, you can apply this field to the item as long as you have not set the field to apply to all items. On the item record, click the Custom subtab. In the Inventory Number Options field, press CTRL and select all of the item number fields you want to apply to the item, and click Save.

For a detailed description of all types of custom fields that you can create, see Kinds of Custom Fields . For detailed information on how to create or modify custom item number fields, see Creating a Custom Field.

Other Custom Fields

Other Custom Fields are used for records that do not have custom forms associated with them. You can add Other Custom Fields to these record types to gather information specific to your business needs.

Other Custom Fields can be added to the following record types:

- Account
- Bin
- Campaign Event
- Class
- Competitor
- Department
- Expense Category
- Location
- Product
- Product Version
- Promotion Code
- Role
- Tax Code
- Workflow

To create or modify Other custom fields:

1. Click Setup > Customization > Other Custom Fields.

   In the Other Custom Fields page, each custom field is listed, with columns providing detailed information about the field and which records the field has been applied to.

2. To edit an existing custom field, click the field name in the description column and then modify the field definition as desired.
3. To add a new custom field, click **New**.

For a detailed description of all types of custom fields that you can create, see Kinds of Custom Fields . For detailed information on how to create or modify custom Other fields, see Creating a Custom Field.

Creating a Custom Field

To record information specific to your business needs, you can create different types of custom record and transaction fields.

See Kinds of Custom Fields for an overview of the different types of custom fields.

To create a custom field:

1. Go to Setup > Customization > *[Custom Field]* > New, where *[Custom Field]* is the desired field type. The Custom Field page is displayed for the selected type.

   To easily edit all settings for a custom field, click the Expand Tabs icon.
2. In the Label field, enter a name or description for the field.

   This label is displayed by the field on the transactions you select.
3. If desired, enter a unique ID for this field.

   This field can be set to an alphanumeric string up to 30 characters long. This string must not include any spaces but can include underscores ("_").

   NetSuite automatically modifies the ID entered into this field with a prefix that corresponds to the type of field being created as listed below.

   - CRM: custevent
   - Entities: custentity
   - Items: custitem
   - Transaction Column: custcol
   - Transaction Body: custbody

   Note: If you do NOT enter an ID now, NetSuite automatically assigns one.

   In order to easily identify the location and purpose of the custom field, it is recommended that you establish a naming convention such as using an underscore followed by the field name. For example, all custom CRM fields would result in a name custevent_fieldname.

Note: Entering your own IDs makes it easier to maintain them in your SuiteScript. Custom field IDs are displayed for each field in the Custom Field lists and on the Record Type page when creating a new record.

4. Select the owner of this custom field.

   Only the owner can modify this record. Your name is selected by default.

5. Enter a description of this field.

6. In the Type field, select the kind of custom field you want to create.

   For more information on custom field types, see Table of Custom Field Type Descriptions.

   The type of field you select determines the other options you can set on this page.

7. If you have chosen List/Record or Multiple Select in the Type field, select the list or record in the List/Record field that contains the items for the list field.

   Important: The displayed value set for this type of field is not strictly filtered by standard restrictions. Be aware that users may select values from records for which they do not have permission. However, they cannot edit the records represented by these values.

8. Check the Show In List box to have this field automatically show in the list of records the field is applied to.

9. You can index this field for global searches if Store Value is checked, and if you have chosen any of the following in the Type field: Currency, Decimal Number, Email Address, Free-Form Text, Help, Hyperlink, Inline HTML, Integer Number, Percent, Phone Number, Text Area.

   You cannot index a custom field for global search if None is selected for any Level for Search/Reporting option on the Access subtab of the custom field record.

   After you check the Global Search box for a custom field, each global search compares keywords to that custom field's values, in addition to comparing with record name and ID field values. For more information, see the help topic *Global Search*.

10. If you are creating a List/Record custom field, check the Record is Parent box to indicate that the record type selected is a parent record.

    This field is used to create a parent-child relationship between two record types.

    For more information on parent-child relationships, see Understanding Parent - Child Record Relationships.

11. Once you have created a custom field, you should define which record types the field can be used in. See Assigning Custom Fields to Specific Record Types.

Note: If you are creating an Other custom field, you do not assign the field to a specific record type. Continue with Setting Display Properties.

Creating Custom Fields with Values Derived from Summary Search Results

You can create a custom field that can display values derived from summary saved search results. Each summary search field displays a rollup value for a selected search results field; this value is dynamically calculated each time a form containing the summary search field is displayed. For example, you can use a summary search field to display the total quantity of all line items on a transaction. The summary search field provides an alternative to using SuiteScript to calculate a custom field's values. Note that this functionality is very similar to that of custom sublists, except the search results are displayed in a field rather than a sublist.

For details about custom fields with summary search derived values, see the following:

- Steps for Creating a Summary Search Custom Field
- Kinds of Custom Fields
- Custom Field Data Types that Support Summary Search Derived Values
- Example Summary Search Custom Field
- Current Limitations for Summary Search Custom Fields

Steps for Creating a Summary Search Custom Field

To create a custom field with a value derived from a summary search:

1. Create or edit a summary saved search that rolls up to the result you want to display in the custom field. (For instructions, see the help topic *Defining a Saved Search*.)

    - Add search criteria to filter out any records/lines you do not want included in the rollup value. (For instructions, see the help topic *Defining Advanced Search Filters*.)
    - Define a search results field for which values will be rolled up, and select a summary type. (Count, Sum, Minimum, Maximum, and Average are supported; Group is not supported.) For example, you could set Quantity as the results field and set a summary type of Sum. (For information about summary types, see the help topic *Summary Type Descriptions*.)
    - Define an available filter field. This field is used to filter search results to include only those records with available filter field values that match the available filter field value of the current record. The value displayed for the summary search custom field is a rollup of search results field values for the records that have matching available filter field values. (For more information, see the help topic *Selecting Available Filters for Saved Searches*.)

2. Create or edit the custom field. (For general instructions, see Creating a Custom Field.)

    - Select this search on the custom field's Validation and Defaulting subtab.
    - Optionally, you can select a "Compare To" field. You can use this field in cases where you want to put the custom field on a form for a record type that is different from the summary search record type. For example, if you want to put a custom entity field showing the result of a customer record summary search on an employee custom form, you could select an employee record field whose values could be matched to the values for the search's Available Filter field. Search results field values for all records with matching values for the Compare To field and Available Filter field would be used to calculate the value of the summary search custom field.
    - Do not enable the Store Value option, as values for this kind of custom field are not stored.

Kinds of Custom Fields that Support Summary Search Derived Values

You can select a summary search to provide rollup values for the following kinds of custom fields:

- Entity Fields
- CRM Fields
- Transaction Body Fields
- Other Custom Fields
- Custom Record Custom Fields

Custom Field Data Types that Support Summary Search Derived Values

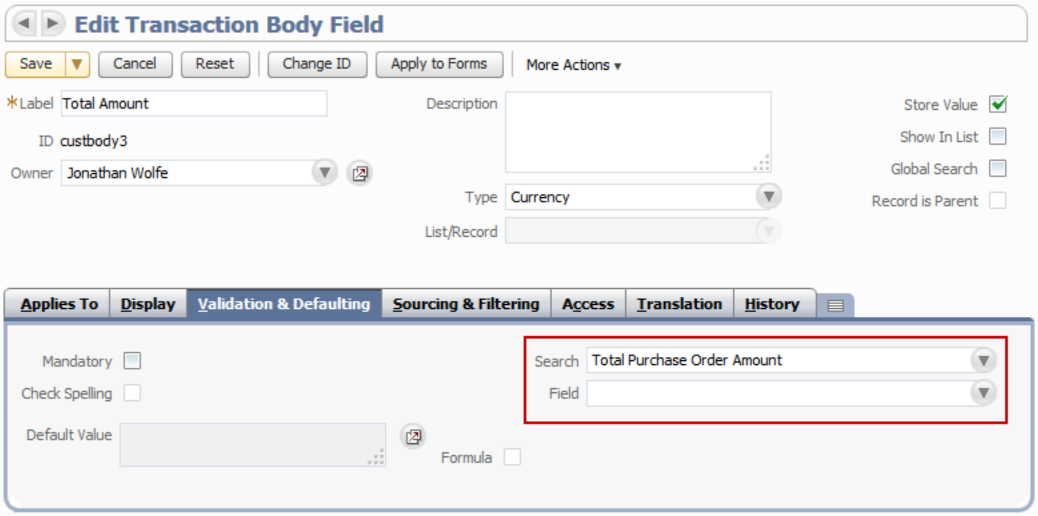
You can select a summary search to provide rollup values for custom fields of the following types:

- Currency
- Date
- Date/Time
- Decimal Number
- Email Address
- Free-Form Text
- Hyperlink

- Integer Number
- Long Text
- Percent
- Phone Number
- Rich Text
- Text Area
- Time of Day

Example Summary Search Custom Field

The following example shows the creation of a custom field to be displayed on custom purchase order forms, with the value of the field calculated by a purchase order summary search that sums the values for the purchase order line items' Amount field. No Compare To field is needed, because the Internal ID field set as the Available Filter for the search can simply be matched to the Internal ID of the purchase order record displayed on the form.



Edit Transaction Body Field

*Label Total Amount

ID custbody3

Owner Jonathan Wolfe

Description

Type Currency

List/Record

Store Value ☑

Show In List □

Global Search □

Record is Parent □

Mandatory □

Check Spelling □

Default Value

Formula □

Search Total Purchase Order Amount

Field

Current Limitations for Summary Search Custom Fields

- Custom field values from summary search results are never stored. Field values are always calculated dynamically at run time. A user may be able to edit values, depending on the display options set for the custom field, but the edited values will not be stored. Also, because its values are not stored, this kind of field is not available in search results, including lists based on saved searches.
- Calculated values for this kind of field may be different for users with different permissions. Summary search results are rolled up for the records to which the current

user has access. Because users with different permissions may have access to different sets of records, the calculated value of this field may vary per user.

Assigning Custom Fields to Specific Record Types

Custom fields can be assigned to display on specific record types. When the field is assigned to a record type, it is automatically available as a possible field when creating a custom form for that record type.

Important: You must select the record types the field is available on. If no record types are selected, the field can NOT be used. Once a record type is selected, the field is automatically displayed on all forms of that record type including any custom forms of that type. Of course, custom forms can then be edited to not show the new custom field.

To apply a custom field to a record type:

1. Click the Applies To subtab.
2. Check the boxes to indicate the records you want this field to display on.

   Note the following:

   - Record custom fields can be added to existing and custom subtabs on the entry forms you use to enter records in your NetSuite account.
   - Transaction custom fields can be added to the top (body) or the line items (columns) of transactions. When adding a custom field to the body of a transaction, you apply the field to the type of transaction.
   - When adding a custom field to the line items of a transaction, you apply the field to the type of line item.
3. Once you have defined a custom field to display on specific record types, you should define the display properties for that field. See Setting Display Options for Custom Fields.

Setting Display Options for Custom Fields

For each custom field, you can specify the exact location within the form that the field is to be displayed relative to other fields and subtabs on the page.

To set the display properties of a custom field:

1. Click the Display subtab.

   The fields on this subtab depend on the kind of field you are modifying.
2. In the Insert Before field, select where to place your new field on records.

   This field lists custom fields of the same type that have already been created.

Note: This only affects the placement of fields on standard forms and on the placement of newly created fields. To change the arrangement of fields on a custom form, you must edit the custom form.

You can also choose to have the custom field span the column at the top or bottom respectively. This is especially useful to add a text area field at the top or bottom of a tab or section which provides explanatory information.

Note: Fields can span the width of the page, above or below the other fields of that tab or section but they do not have to span the page. This is determined by the size of the field.

3. In the Subtab field, select the subtab you want this field to display on.

   For example, select Main if you want this field to display in the top portion of the record.

   Important: If a subtab is NOT selected, the field is automatically displayed on a custom subtab for the record. Select a subtab that makes sense for the type of information the custom field stores.

4. Select the Display Type.

   Display types allow you to specify how your custom fields behave in NetSuite. You can use display types to make fields for informational purposes only that are not stored in your account. You can also create fields that are not editable or that have default information or custom code calculations.

   Following are the available display types.

   - **Normal** : A normal field can be edited. You can use this field with custom code calculations, defaulting and sourcing information.

     For example, you create a custom field on employee records for a spending limit. The Spending Limit field has a display type of normal because you want to edit and store the information entered in this field.

   - **Disabled** : A disabled field cannot be edited. You can use this field with custom code calculations, defaulting and sourcing information only. Any field with a display type of disabled that does not have default, sourced or custom code information will not display on forms.

     For example, you create a custom field on expense reports for a spending limit. The Spending Limit field on expense reports is sourced from the Spending Limit custom field on employee records. When an employee enters an expense report, they automatically see their spending limit on the expense report. The field has a display type of disabled because you want the information in this field to be updated but not edited.

Note: You cannot disable a mandatory field unless the field has a default value.

- **Inline Text** : An inline text field is for informational purposes only. The information in this field is the result of custom code calculation, defaulting or sourcing information only. Note that a value for a field with an Inline Text display type will be interpreted as HTML code and will be evaluated by the browser when a record containing this field is displayed. Inline text fields of the List/Record or Check Box type are not available for use with custom code. These fields are also not available for transaction column fields.

  For example, you create a Tax ID custom field for your purchase orders. When creating the custom field, you can enter your tax ID in the Default Value field. Your Tax ID then appears on purchase orders. The field has a display type of inline text because your tax ID is for informational purposes only, and you do not want the information stored with each transaction.

- **Hidden** : A hidden field cannot be seen on the record or transaction you apply it to. You can perform a search to display the value of this field. The information in this field is the result of custom code calculations and defaulting information. You must use custom code or set a default for this field. Note that you can also define a field as shown in its custom field definition, and then selectively hide it on a form by form basis.

  For example, you can use a hidden field to store your support reps case quota. The support reps do not see the field but the information can still be searched and reported on. For more information on custom code, read Using Custom Code With Forms.

  Be aware that in SuiteScript, only user event, scheduled, and Suitelet scripts can set the value of a custom field that has a display type of hidden.

Warning: Hiding a custom field is a display convenience only. This is NOT field level security. Hidden custom fields are embedded in the page output and viewable in the page source.

5. Enter field height and width properties.

   The values entered for height and width affect the **display** size of the field only. The **allowed** size of the custom field (number of characters or lines allowed to be entered) is controlled by default options. For more information on defaulting options, see Setting Validation and Defaulting Properties.

   - **Height** : Enter a height for the custom field in number of lines. The default is three lines.

   - **Width** : Enter a width for your custom field in characters.

Important: Be aware that field configurations on each custom form interact with the values you set here. Actions such as setting up field groups, adding column breaks, and setting fields to be Same Row as Previous on custom forms can cause custom field text to be displayed differently than you might expect based on the custom field width. For details about configuring fields on custom forms, see Configuring Field Groups, Configuring Fields or Screens, and Associating Related Fields on Custom Forms.

6. If desired, in the Link Text field, enter text to display instead of a URL for hyperlink fields.

   Users see this text instead of the URL on records and transactions. Clicking the text links to the URL.

7. (Decimal Number, Integer Number, and Percent fields only) Review the Apply Formatting setting and change if desired.

   - When this preference is enabled, the custom field's values use the formatting set in the Number Format and Negative Number Format preferences at Company, Subsidiary, or User level. For details about these preferences, see the help topic *Setting Number Formats*.
   - When this preference is disabled, the custom field's values are not formatted, meaning they have no commas or decimal points. (This setting is appropriate for Integer Number type fields used as unique identifiers.)

   Note: Default settings are as follows: disabled for custom fields created prior to Version 2012 Release 1 upgrade, in order to preserve pre-upgrade behavior, and enabled for custom fields created after Version 2012 Release 1 upgrade.

8. In the Help field, enter a brief description of the kind of information you want entered in this field.

   This help is available when the name of the field is clicked.

   You can enter HTML in this field. Be sure to begin your code with <**html**> and end the code with </**html**>.

   Note: It is recommended that you always enter information in this field to increase the usability of customized forms or records.

9. In the Label for Input field, enter the label for this item option as you want it to appear to customers on your Web site.

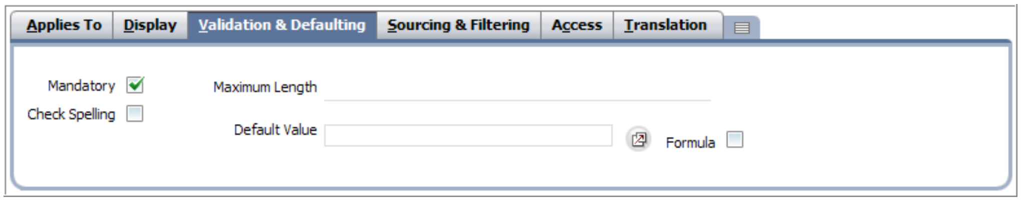
10. Once you have set the display properties, you should define any validation and defaulting properties. See Setting Validation and Defaulting Properties.

Setting Validation and Defaulting Properties

Validation options are constraints that can be placed on your custom fields to help control the information that is entered in the field. Defaults are values you specify for your custom fields that display populate automatically when a record or transaction is first created.

To set validation and defaulting properties:

1. Click the Validation and Defaulting subtab.



Mandatory ☑
Check Spelling □
Maximum Length
Default Value
Formula □

2. Set the validation options. (These options may vary according to the kind of custom field selected and its data type.)

   Possible options include the following:

   - **Mandatory** : Enable to require information to be entered in the custom fields before a record or transaction can be saved.

     For example, you can create a mandatory contact number field for your sales transactions. Then, when sales reps enter transactions, a contact number for the customer MUST be entered before the transaction can be saved.

     Note: You can also set fields to mandatory when customizing a form. When using the same custom field on different forms, consider applying required field settings in your custom form rather than on the field itself. This is useful in cases where in one form the field information is required, but in another the field information is not required.

   - **Minimum Value** : Set the minimum number that can be entered in the custom field. A record or transaction cannot be saved with a value below the minimum set here.

     For example, you can create a sales quota field for your employee records. Set a limit of $500.00 when creating the custom field and any amount less than 500 cannot be entered in the field.

     You can set a minimum value for a percent field that is less than 0. If you do not enter a minimum amount for a percent field, the minimum is 0.

   - **Maximum Value** : Set the maximum number that can be entered in the custom field. A record or transaction cannot be saved with a value above the maximum set here.

For example, you can create a spending limit custom field for your employee records. Set a limit of $700.00 when creating the custom field to ensure that no employee can be given a spending limit of more than $700.00.

You can set a maximum value for a percent field that exceeds 100. If you do not enter a maximum value for a percent field, the maximum is 100.

- **Maximum Length** : Set the maximum number of characters that can be entered in the custom field. A record or transaction cannot be saved if the information entered in this field exceeds the number of allowed characters.

  For example, you can create a gift message field for your sales transactions to record a special message from your customers. You can limit the number of characters to prevent messages from becoming too long.

- **Formula** : Enable to validate SQL formula expressions when this field is defined as a formula field. For more information on defining a custom formula field, see Creating Formula Fields.
- **Search** : Select a summary search from this dropdown if you want values for this custom field to be based on search results. Only available for kinds of custom fields and data types that support summary search derived values. For details, see Creating Custom Fields with Values Derived from Summary Search Results.
- **Field** : If you want values for this custom field to be based on summary search results, you can optionally select a comparison field to join related records in cases where you want to put the custom field on a form for a record type that is different from the summary search record type. Only available for kinds of custom fields and data types that support summary search derived values. For details, see Creating Custom Fields with Values Derived from Summary Search Results.

3. If desired, enter default parameters for this field.

   In order to set values into an Inline Text or Disabled field, you **must** specify a default value or source the information for the field from another field.
   Depending on the field type, various default values can be specified. Possible options include:

   - **Default Checked** : Check this box to indicate that the check box custom field should display checked by default. The custom fields can still be cleared on individual transactions and records.

     For example, you can add a Subscribe to Newsletter check box to your customer records. When you set the field to default checked, new customers are automatically subscribed to your newsletter.

   - **Default Value** : If desired, enter a value to display in this field by default. The value can still be changed on individual records and transactions if the field is not locked.

When working with Free-form text, Text area, Rich Text or Hypertext fields, you can include NetSuite tags in the default definition. These tags are populated with field values when the page is loaded or saved. For more information, see Creating Dynamic Defaults and Dynamic Hyperlinks.

For example, you can add a spending limit field to your employee records with a default value of $150.00. When you enter new employee records, the field automatically fills with this amount.

Default values defined here are only applied at creation time for any given record. Once a record has been created, subsequent edits to that record store the previously stored value unless it also has been edited.

- **Dynamic Default** : Dynamic default allows you to select from preset defaulting options specific to the kind of field you are creating. There are three types of dynamic defaults:

    - **Current Date/Time** : For Date fields, select this to have your custom field automatically filled with the current date or time.
      For example, you can add an information only date field to your sales orders. Setting the field to current date and time allows you to track when your sales reps are most productive.

      Important: The date and time for this type of field are based on the logged in user's NetSuite Time Zone preference, set at Home > Set Preferences, not on the browser client time zone.

    - **Current User** : For Employee List/Record fields, select this to have the name of the employee entering the record or transaction automatically filled in this field. For example, if you have data entry employees as well as sales reps, you can add a field to your transactions to record who enters the transaction as well as the sales rep who made the sale.

    - **Current User's Supervisor** : For Employee List/Record fields, select this to have the name of the supervisor selected on the employee record automatically filled in this field. For example, you can create a field for your task records that lists the Assignee's supervisor. This can help your management team stay informed of the projects their teams are working on.

- **Default Selection** : Set a selection list to display in the custom field by default. The choices are limited to the list selected in the List/Record field when creating this custom field.

  For example, you can create a custom field to record advertising preferences for your customers. You can set a default of e-mail if you know that the majority of your customers prefer to receive ads by e-mail. When a customer loads the page, E-mail is displayed in the selection list by default but they can choose Fax or Mail if desired.

4. Once you have validation and defaulting properties, you should set any sourcing criteria. See Setting Sourcing Criteria.

Creating Dynamic Defaults and Dynamic Hyperlinks

When working with Free-form text, Text area, Rich Text or Hypertext fields, you can include NetSuite tags in the default definition. These tags are populated with field values when the page is loaded or saved.

To include NetSuite tags in the default definition, simply enclose each tag within curly braces such that field tags in a dynamic default are defined as **{TAG}**, where TAG is the ID of the field. Each field in NetSuite has a unique ID and therefore a unique tag definition.

Note: Because field IDs are incorporated into tag definitions for fields, when creating custom fields that will be used in dynamic defaults, it is useful if you specify IDs for each field and use consistent naming conventions that make sense in your business environment. If the default NetSuite IDs are accepted when creating your custom fields, the tags may not make sense in your dynamic defaults making it more difficult to know exactly what the field references.

Dynamic Defaults are evaluated and each NetSuite tag is substituted on page load and page save. However, if you check the **Store Value** check box, the tag substitution values are saved when the page is created as a true default. This means that the default value is saved and is NOT dynamically changed with changes to the fields on the page, allowing you to create a dynamic default that retains its initial value. The field must be edited manually, or updated with custom code to change its initial value.

Important: If you need to ensure that NetSuite tags defined in a dynamic default are substituted on each page load and save, uncheck **Store Value**.

NetSuite Tags

Currently any field on the page which has a custom code ID can be used in a NetSuite tag as well as the following special values:

- **{useremail}** : currently logged in user's e-mail
- **{today}** : current date
- **{nlversion}** : the fully qualified internal number release number
- **{nlsessionid}** : the browser's session ID; useful for passing a session to a web service in a WebLink
- **{nluser}** : the currently logged in user's ID
- **{nlrole}** : the currently logged in user's role ID

Custom code IDs for standard NetSuite fields are itemized in the *SuiteScript Developer Guide*. To determine the custom code ID of custom fields on your forms, go to the Custom Field page for the type of field — for example Setup > Customization > CRM Fields. The custom code ID is displayed in the ID column.

Dynamic Hyperlinks

You can also create a Web link by defining dynamic defaults for hyperlink fields. This is especially useful when the exact URL is unknown until information is collected for the record or if information specific to the current logged in session is required as part of an URL parameter. When creating a WebLink, enter the http address as usual followed by **?=** and the desired NetSuite tags embedded in curly braces.

For example, suppose that you want to include an address look-up feature on a customer form. Create a custom Entity field with the following parameters specified:

- Label: Map
- ID: _map
- Type: Hyperlink
- Store Value: Unchecked
- Applies To: Customer
- Display / Subtab: Main
- Display / Link Text: Click Here for Google Map
- Validation & Defaulting / Default Value:

  http://maps.google.com/maps?q={billaddr1}%20{billcity}%20{billstate}%20{billzip}

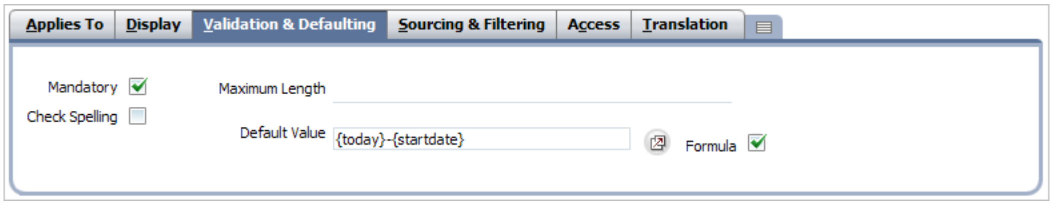
The default value includes NetSuite tags that identify the specific address of the current customer. These tags are resolved when the page is loaded so that the URL will direct the user to the customer's address as defined in the current customer record.

Note: When creating dynamic WebLinks, ensure that NetSuite tags embedded in the default value definition represent required fields. If the fields are NOT required, and the associated form does NOT include a value for the tag, then the resulting URL will be invalid.

Creating Formula Fields

In addition to defining a custom field to pre-populate with dynamic data as described in Creating Dynamic Defaults and Dynamic Hyperlinks, you can define fields to be dynamically **calculated** based on the values returned in the dynamic fields.

To define formula fields, click the Validation & Defaulting subtab of the custom field. Check the Formula box. In the Formula field, use NetSuite Tags to define the dynamically defaulted fields to be used in the calculation and use SQL Expressions to define the formula.



Mandatory ☑

Check Spelling □

Maximum Length

Default Value {today}-{startdate}

Formula ☑

Important: The **Formula** box must be **checked** in order for this field to be processed as a formula and, as with any defaulted field, the **Store Value** checkbox must be **deselected** in order to dynamically recalculate the value each time the field is viewed. Also, when a record is loaded custom formula fields are calculated, but if changes to fields used in the formula definition are made while the record is still loaded, the formula field is NOT recalculated to reflect these changes until the next time the record is loaded.

During validation, the following inline errors can be returned:

- ERROR: Field Not Found - returned when either a custom field or search formula is not recognized by the system.
- ERROR: Invalid Formula - returned when there is a syntax or datatype error in the custom formula field.

When custom Formula fields are returned as a result of search criteria, the displayed value is the result of the dynamically calculated value at the time the search is performed. You can also define search criteria as formula fields without using a custom formula field. For more information, see the help topic *Using Formulas in Search*.

Warning: If a field on a record is referred to by a formula custom field, you cannot edit the referenced field with Direct List Editing.

Note: Knowledge of SQL will help you to fully leverage the flexibility and power of SQL functions to define complex formulas, but NetSuite's Formula popup windows can help you to correctly define formula expressions. These popups include a Function dropdown that allows you to select SQL functions to be included in expressions, and Filter or Field dropdowns that allow you to select field names and have their IDs included in expressions. For more details, refer to SQL Expressions. Also, you can refer to the SuiteScript Reference Guide for tables of NetSuite field IDs.

Using Foreign References in Formula Fields

When creating a formula field, you can reference data contained in fields on related records.

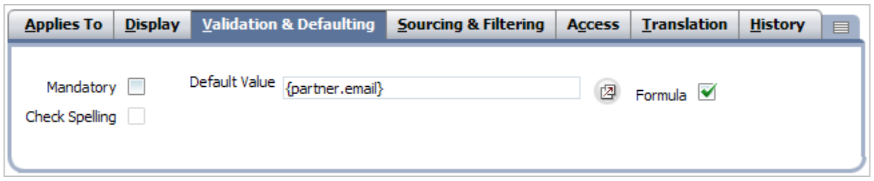
For example, if you created a custom entity field to apply to customer records, you could add a formula field that referenced a field on the employee record of the sales rep assigned to the customer.

Note: When referencing fields on other records, you are restricted to the records with search joins.

The format for formula field references is:

{fieldOnAppliedRecord.fieldOnJoinedRecord}

For example, if you wanted to display the partner email address on customer records, the format for the formula would be:



Mandatory ☐

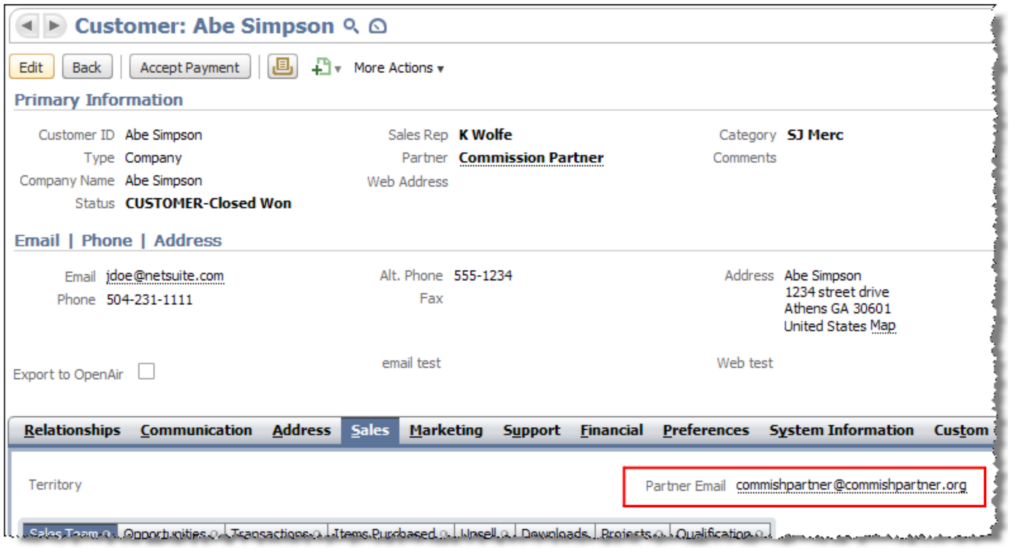
Check Spelling ☐

Default Value {partner.email}

Formula ☑

partner is the field ID for the Partner field on the customer record. **email** is the field ID for the email field on the partner record.

This example displays the email address on the record for the partner assigned to each customer.



Customer ID Abe Simpson

Type Company

Company Name Abe Simpson

Status CUSTOMER-Closed Won

Sales Rep K Wolfe

Partner Commission Partner

Web Address

Category SJ Merc

Comments

Email jdoe@netsuite.com

Phone 504-231-1111

Alt. Phone 555-1234

Fax

Address Abe Simpson
1234 street drive
Athens GA 30601
United States Map

Export to OpenAir ☐

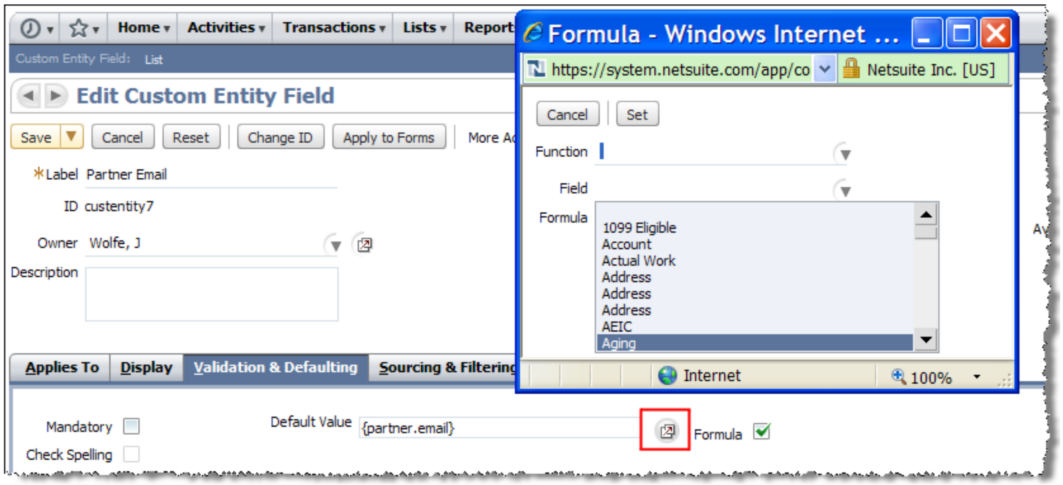
email test

Web test

Territory

Partner Email commishpartner@commishpartner.org

Note: Knowledge of SQL will help you to fully leverage the flexibility and power of SQL functions to define complex formulas, but you can click Set Formula next to the Formula check box to add SQL functions or field IDs to your formula.



*Label Partner Email
ID custentity7
Owner Wolfe, J
Description

Mandatory □
Check Spelling □
Default Value {partner.email}
Formula ☑

https://system.netsuite.com/app/co
Netsuite Inc. [US]

Function
Field
Formula
1099 Eligible
Account
Actual Work
Address
Address
Address
AEIC
Aging

Note: For more details, refer to SQL Expressions. Also, you can refer to the SuiteScript Reference Guide for tables of NetSuite field IDs.

Using List/Record Field IDs in Formula Fields

You can reference the ID value for any List/Record type field in a formula field. Use the format {field_name.ID}.

Using Transaction Memo Fields in Formulas

Transactions that have line items, such as sales orders, may have values for both a memo body field and memo line item fields. Note that two different memo field IDs are available for use in custom field formulas. These fields are basically the same, except for a difference in behavior when the memo body field value is blank. In this case:

- The **memo** field returns the first non-empty line item memo field's value when the memo body field value is blank.
- The **memobody** field always returns the memo body field value, even if it is blank.

Formula Field Example

For example, suppose you want to display the remaining credit available to a customer on the customer record. Create a custom entity field of the type Currency called **Remaining Credit**.

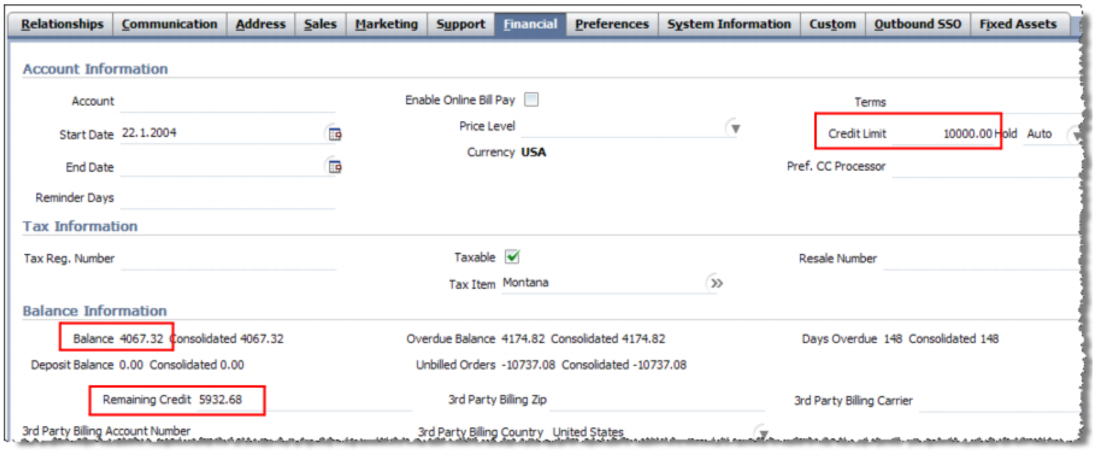
Apply the field to the Customer record and set to display on the Financial subtab. Then, define the field with the following formula in the Validation & Defaulting subtab:

{creditlimit}-nvl({balance},0)

(where creditlimit and balance are standard customer fields and the nvl NULL handling function forces the value to be set to the second parameter when the field is NULL)

Make sure that you enable the Formula field and clear the Store Value box to ensure that the value is always dynamically recalculated as a formula.

When a customer record is viewed, the Remaining Credit field returns a calculated value based on the credit limit and customer balance fields. So in the following image, you can see that Remaining Credit returns the result of $10,000.00 - $4067.32.



Account Information

Account

Enable Online Bill Pay □

Terms

Start Date 22.1.2004

Price Level

Credit Limit 10000.00 Hold Auto

Currency **USA**

End Date

Pref. CC Processor

Reminder Days

Tax Information

Tax Reg. Number

Taxable ☑

Resale Number

Tax Item Montana

Balance Information

Balance 4067.32 Consolidated 4067.32

Overdue Balance 4174.82 Consolidated 4174.82

Days Overdue 148 Consolidated 148

Deposit Balance 0.00 Consolidated 0.00

Unbilled Orders -10737.08 Consolidated -10737.08

Remaining Credit 5932.68

3rd Party Billing Zip

3rd Party Billing Carrier

3rd Party Billing Account Number

3rd Party Billing Country United States

Creating a Formula Field to Display Transaction Line Numbers

You can display line numbers on the Items subtab of transactions when they are viewed online, and in printed transactions, by creating a custom field that uses the **{linenumber}** formula, and applying this field to transaction forms.

1. To create a custom line number field, go to Setup > Customization > Transaction Column Fields > New.
2. Enter a label for the field, select a Type of Integer Number, and clear the Store Value box so it is not checked.
3. On the Applies to subtab, check boxes for the transactions that should display line numbers, and check the Print on Standard Forms box.
4. On the Display subtab, select a Display Type of Disabled.

5. On the Validation & Defaulting subtab, check the Formula box, and enter {linenumber} as the Default Value.
6. Save the new field.
7. To enable the line number field to be printed on a custom transaction form, edit the form, and on the Printing Fields, Columns subtab, check the Print/Email box for the new field.

Note the following:

- Line numbers display when a transaction record is in View mode. In Edit mode, line number value is shown as 1.
- Line numbers correspond to the printed or viewed results, meaning they are contiguous even when transaction lines are omitted.
- If you are using a line number formula field for viewed and printed transactions, and you also want to include the line number in search results, set up the search as follows to ensure that search results match viewed and printed transaction items:
    - Set a criteria of Main Line = No (false).
    - Filter out transaction line items related to taxes.
    - Add the Item field as a results field.
    - Add the Amount (Gross) field as a results field; do not use the Amount field.
    - Add a Formula(Numeric) field as a results field, with the following formula expression: RANK() OVER (PARTITION by {internalid} ORDER BY {linesequencenumber})

SQL Expressions

The following tables outline the SQL functions that can be used in NetSuite search formulas and custom formula fields:

- Numeric Functions
- Character Functions Returning Character Values
- Character Functions Returning Number Values
- Datetime Functions
- NULL-Related Functions
- Decode
- Sysdate

- CASE
- Aggregate Functions

Important: This section provides some material for convenience only. It is assumed that you are familiar with the implementation of SQL expressions. For a complete reference to SQL expressions, click here (requires oracle account activation). Note that not all of the expressions described at this URL are supported in NetSuite.

Numeric Functions

Function	Syntax	Description
ABS	ABS(n)	returns the absolute value of n
ACOS	ACOS(n)	returns the arc cosine of n
ASIN	ASIN(n)	returns the arc sine of n
ATAN	ATAN(n)	returns the arc tangent of n
ATAN2	ATAN2(n1 { , | / } n2)	returns the arc tangent of n1 and n2
BITAND	BITAND(expr1, expr2)	computes an AND operation on the bits of expr1 and expr2
CEIL	CEIL(n)	returns smallest integer greater than or equal to n
COS	COS(n)	returns the cosine of n
COSH	COSH(n)	returns the hyperbolic cosine of n.
EXP	EXP(n)	returns e raised to the nth power
FLOOR	FLOOR(n)	returns largest integer equal to or less than n.
LN	LN(n)	returns the natural logarithm of n,
LOG	LOG(n2, n1)	returns the logarithm, base n2, of n1
MOD	MOD(n2, n1)	returns the remainder of n2 divided by n1
NANVL	NANVL(n2, n1)	returns an alternative value n1 if the input value n2 is not a number
POWER	POWER(n2, n1)	returns n2 raised to the n1 power
REMAINDER	REMAINDER(n2, n1)	returns the remainder of n2 divided by n1
ROUND (number)	ROUND(n [, integer])	returns n rounded to integer places to the right of the decimal point
SIGN	SIGN(n)	returns the sign of n
SIN	SIN(n)	returns the sine of n
SINH	SINH(n)	returns the hyperbolic sine of n
SQRT	SQRT(n)	returns the square root of n
TAN	TAN(n)	returns the tangent of n

Function	Syntax	Description
TANH	TANH(n)	returns the hyperbolic tangent of n
TRUNC (number)	TRUNC(n1 [, n2])	returns n1 truncated to n2 decimal places

Character Functions Returning Character Values

Function	Syntax	Description
CHR	CHR(n [USING NCHAR_CS])	returns the character having the binary equivalent to n as a VARCHAR2 value
CONCAT	CONCAT(char1, char2)	returns char1 concatenated with char2
INITCAP	INITCAP(char)	returns char, with the first letter of each word in uppercase, all other letters in lowercase
LOWER	LOWER(char)	returns char, with all letters lowercase
LPAD	LPAD(expr1, n [, expr2])	returns expr1, left-padded to length n characters with the sequence of characters in expr2
LTRIM	LTRIM(char [, set])	removes from the left end of char all of the characters contained in set
REGEXP_REPLACE	REGEXP_REPLACE(source_char, pattern [, replace_string [, position [, occurrence [, match_parameter]]]])	allows you to search a string for a regular expression pattern
REGEXP_SUBSTR	REGEXP_SUBSTR(source_char, pattern [, position [, occurrence [, match_parameter]]])	allows you to search a string for a regular expression pattern
REPLACE	REPLACE(char, search_string [, replacement_string])	returns char with every occurrence of search_string replaced with replacement_string
RPAD	RPAD(expr1 , n [, expr2])	returns expr1, right-padded to length n characters with expr2, replicated as many times as necessary

Function	Syntax	Description
RTRIM	RTRIM(char [, set])	removes from the right end of char all of the characters that appear in set
SOUNDEX	SOUNDEX(char)	returns a character string containing the phonetic representation of char
SUBSTR	{ SUBSTR | SUBSTRB | SUBSTRC | SUBSTR2 | SUBSTR4 } (char, position [, substring_length])	return a portion of char, beginning at character position, substring_length characters long
TRANSLATE	TRANSLATE(expr, from_string, to_string)	returns expr with all occurrences of each character in from_string replaced by its corresponding character in to_string
TREAT	TREAT(expr AS [REF] [schema.]type)	changes the declared type of an expression
TRIM	TRIM([{ { LEADING | TRAILING | BOTH } [trim_character] | trim_character } FROM] trim_source)	allows you to trim leading or trailing characters (or both) from a character string
UPPER	UPPER(char)	returns char, with all letters uppercase

Character Functions Returning Number Values

Function	Syntax	Short Description
ASCII	ASCII(char)	returns the decimal representation in the database character set of the first character of char
INSTR	{ INSTR | INSTRB | INSTRC | INSTR2 | INSTR4 } (string , substring [, position [, occurrence]])	searches string for substring
LENGTH	{ LENGTH	returns the length of char

Function	Syntax	Short Description
	| LENGTHB | LENGTHC | LENGTH2 | LENGTH4 } (char)	
REGEXP_INSTR	REGEXP_INSTR (source_char, pattern [, position [, occurrence [, return_option [, match_parameter]]]])	allows you search a string for a regular expression pattern
TO_NUMBER()	TO_NUMBER(expr [, fmt [, 'nlsparam']])	converts a formatted TEXT or NTEXT expression to a number

Datetime Functions

Function	Syntax	Short Description
ADD_MONTHS	ADD_MONTHS(date, integer)	returns the date plus integer months
LAST_DAY	LAST_DAY(date)	returns the date of the last day of the month that contains date
NEXT_DAY	NEXT_DAY(date, char)	returns the date of the first weekday named by char that is later than the date
ROUND (DATE)	ROUND(date [, fmt])	returns date rounded to the unit specified by the format model fmt
TO_CHAR()	TO_CHAR({ datetime | interval } [, fmt [, 'nlsparam']])	converts a datetime or interval value to a value of VARCHAR2 datatype in the format specified by the date format fmt
TO_DATE()	TO_DATE(char [, fmt [, 'nlsparam']])	converts a formatted TEXT or NTEXT expression to a DATETIME value
TRUNK (DATE)	TRUNC(date [, fmt])	returns date with the time portion of the day truncated to the unit specified by the format model fmt

NULL-Related Functions

Function	Syntax	Short Description
COALESCE	COALESCE(expr [, expr]...)	returns the first non-null expr in the expression list
NULLIF	NULLIF(expr1, expr2)	compares expr1 and expr2. If they are equal, then the function returns null. If they are not equal, then the function returns expr1.
NVL	NVL(expr1, expr2)	allows you to replace null with the second parameter

Function	Syntax	Short Description
NVL2	NVL2(expr1, expr2, expr3)	If expr1 is not null, then NVL2 returns expr2. If expr1 is null, then NVL2 returns expr3.

Decode

Function	Syntax	Short Descriptions
DECODE	DECODE(expr, search, result [, search, result]... [, default])	Compares expr to each search value one by one. If expr is equal to a search, the corresponding result is returned. If no match is found, default is returned.

Sysdate

Function	Syntax	Short Description
SYSDATE	SYSDATE	returns the current date and time set

CASE

Function	Syntax	Short Description
CASE	CASE { expr WHEN comparison_expr THEN return_expr [WHEN comparison_expr THEN return_expr]... | WHEN condition THEN return_expr [WHEN condition THEN return_expr]... } [ELSE else_expr] END	returns value based on different conditions

Aggregate Functions

Function	Syntax	Short Description
DENSE_RANK	DENSE_RANK(expr [, expr]...) WITHIN GROUP (ORDER BY expr [DESC | ASC] [NULLS { FIRST | LAST }] [,expr [DESC | ASC] [NULLS { FIRST | LAST }]]...)	Computes the rank of a value with respect to other values and returns the rank as a NUMBER. Always results in consecutive rankings
KEEP()	KEEP(DENSE_RANK { FIRST | LAST } ORDER BY expr [NULLS { FIRST | LAST }])	Qualifies an aggregate function indicating that only the FIRST or LAST values of the function are returned
RANK	RANK(expr [, expr]...) WITHIN GROUP (ORDER BY expr [DESC | ASC] [NULLS { FIRST | LAST }] [, expr [DESC | ASC] [NULLS { FIRST | LAST }]]...	Computes the rank of a value in a group of values. Can result in non-consecutive rankings if values are the same.

Function	Syntax	Short Description
	)	

Setting Sourcing Criteria

A custom field can source information from another record in your account. The information populated into the custom field is then dependent on fields associated with a record selected on another field within that form. Sourcing enhances your NetSuite forms by reducing data-entry errors and ensuring that your customers and employees always have the most current information.

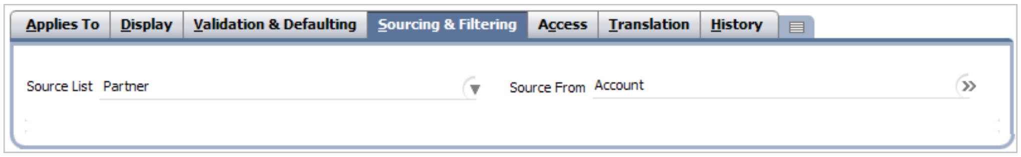
You can source from both standard and custom fields.

For example, two custom fields—*Sales Rep* and *Sales Rep Email*—are placed on a custom case form. When a company record is selected in the Company field, the Sales Rep already defined in the selected company record is sourced to the Sales Rep field on the case form and the Sales Rep E-mail address field defaults to the E-mail address defined for the sourced Sales Rep.

Note: The information is sourced into the custom field only when the record is created or if the specific fields involved are altered when editing the record. In the example above, if you change the sales rep selected on the customer, the sourced field would change to show the email address of the new sales rep.

To set sourcing and filtering criteria:

1. Edit the custom field you want to add sourcing and filtering criteria to.
2. Be sure you have not checked the Store Value box.
3. Click the Sourcing and Filtering subtab.



Source List Partner

Source From Account

4. In the Source List field, select the field that references the record you would like to source information from.

   For example, you are creating a custom field to appear on the customer record to show the email address of the sales rep assigned to the customer. In the Source List field, you would select Sales Rep.

   When working with entity fields, you can also define the field to source from a field on the parent record by selecting Parent in the Source List dropdown.

Note: You CANNOT source information for a Multiple Select field type.

5. In the Source From field, select the field you want to source from that is found on the record you selected in the Source List field.

   Any fields available on the record you select in the Source List field can be selected in the Source From field.

   In the example above, you would select Email in the Source From field.

   The value stored in the field selected here populates the field when the record is selected. The field selected here must be consistent with the Type selected for the custom field. For example, if you select E-mail as the Type of field and then select an Address field in the Source From list, an error is thrown.

   Note: If your field is a List/Record field, the field selected for the Source From field MUST be in the record type selected as the List/Record.

6. If your field is a List/Record field, you can filter the choices that can be selected.

   When a List/Record Type field is being defined, you can choose to populate the custom field with values that meet specific parameters in the sourced list or record. First, select the desired item to filter by in the Source Filter by field. Then choose an item from the Source List and, optionally from the Source From fields. When you choose an element from the source list, it will fill the option-sourced custom field with all elements where the source filter by field matches the source list (or the source from field of the source list).

   The record you are sourcing from must be associated with the type of record you want to appear in your custom field.

   Note: The field selected for the Source Filter by field MUST be in the record type selected as the List/Record.

   For more information, see Understanding the 'Source Filter by' Field.

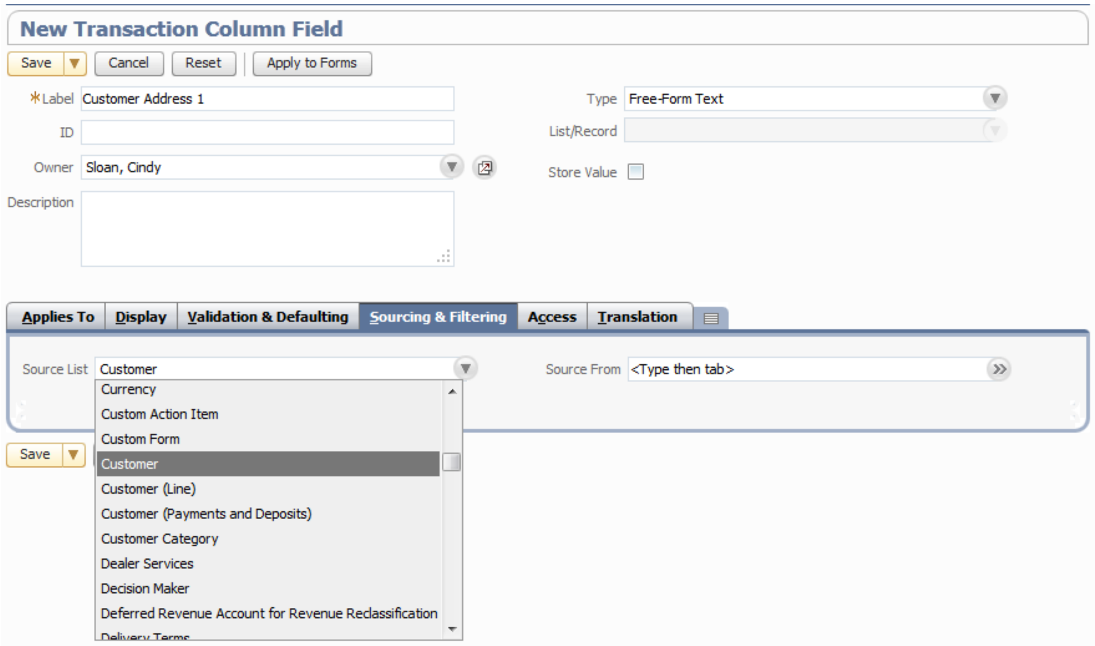
7. Once you have set the sourcing criteria, you should set any filtering criteria. See Setting Filtering Criteria.

Note: A custom field with a sourcing relationship is not available for mass updates or inline editing. See the help topic *Making Mass Changes or Updates*.

Note about Custom Transaction Column Field Sourcing

Before Version 2012 Release 2, you could source a custom transaction column field's values from a body field by selecting <Record_Name> (Line) in the Source List dropdown on the Sourcing and Filtering subtab of the Transaction Column Field page. Now, in order to source a field's values from a body field, you need to select <Record_Name> in this dropdown. If you select <Record_Name> (Line), sourcing is from a field in the sublist and if there is no such field, values for the sourced custom transaction column field are blank.

The following example shows a custom field applied to the Items sublist on Sales Order records. Because this sublist does not include a Customer field, if Customer (Line) is selected, the values for the new custom field will be blank. Selecting Customer sources the new custom field's values from the Customer body field on Sales Orders.



New Transaction Column Field

*Label: Customer Address 1
ID:
Owner: Sloan, Cindy
Description:
Type: Free-Form Text
List/Record:
Store Value: ☐

Source List: Customer
Source From: <Type then tab>

Understanding the ‘Source Filter by' Field

Following are some examples of how you can use the Source Filter by field to create dynamic custom fields.

Example 1: Linking Two Transactions

Suppose that you want to link two transactions, such as an invoice and a subsequent credit.

First, add a transaction custom field to customer credits (using a custom form to limit it only to the credit form). The field should be a List/Record Transaction Type and source Entity in the Source List and Source Filter by fields. When you select a customer on the credit memo, it will populate the new list field with only invoices from that customer.

Example 2: Combining Static and Dynamic Filtering

Suppose that you want to create a Linked Order field on a Sales Order that allows you to choose another Sales Order from that customer.

First, add a transaction custom field to sales order forms. The field should be a List/Record Transaction Type and source Entity in the Source List and Source Filter by fields. Add a static Filter to filter the drop down to only transactions of a particular type (Sales Orders). When you create a new sales order, select a customer, and the Linked Order drop down populates with sales orders from that customer.

Example 3: Filtering Against the Source From Value

Suppose that you want to have a field on a task record that lists all of the subordinates of a sales rep associated to a selected opportunity.

First, add a custom field to a Task record. The field should be a List/Record Entity Type (for example Employee). Since you want only the Sales Reps associated with the selected opportunity, define the field Source List as Opportunity. Then filter the Source From by Sales Rep and the Source Filter by as Supervisor. The resultant List filters to cases where the Source Filter By value (Supervisor) equals the Sourced From value (Sales Rep).

Note: The field selected for the Source Filter by field MUST be in the record type selected as the List/Record.

Setting Filtering Criteria

When creating a list/record or multiple select custom field, you can filter the choices available in that field on records and transactions based on selections made in other fields. This allows you to tailor the exact choices offered to users entering records and transactions.

You can filter based on the selections in multiple dropdown fields. For more information, see Multiple Dependent Dropdowns.

Note: Filtering only applies to lists of records. It does NOT apply to custom lists.

To filter a list/record or multiple select custom field:

1. Click the Sourcing and Filtering subtab.
2. In the Filter Using field, choose a field to filter on.

   The field you choose here is a field on the record you selected in the List/Record field.

   The field selected limits the results according to the filter criteria you define. For example, if you were limiting an employee list field to only show sales reps, you would select the [Is] Sales Rep field in this column since that is the name of the field on the employee record.
3. If the field chosen in the Filter Using column is a check box, check the Is Checked box to show only records with that box checked.

In the sales rep example, you would check the box in the Is Checked column.

4. In the **Compare Type** column, select how you want the information compared to the criteria you set.

   For example, select **equals** to ensure that the information you set as criteria matches the selections available in the list exactly.

5. In the **Compare Value to** column, enter the value you want to filter the list by.

   For example, if you selected State in the Filter Using field, you can enter GA to filter the list to only records with the state listed as Georgia.

6. If the field chosen in the Filter Using field is a list, choose the value you want to show in the **Value Is** column.

7. Check the **Is Not Empty** box to include all records with a value entered in your filter field.

   This option is not available for check box fields.

8. Check the **Is Empty** box to include all records with no value entered in your filter field.

9. In the **Compare To Field** column, choose which field on the record selected in the List/Record field you want to compare to the field in the **Filter Using** column.

10. Click Add/Edit.

11. Repeat these steps to add additional filters to this custom field.

    The more filters you add, the less choices are offered in the field. Each selection must match each filter to be included.

12. Click Save.

Next, you can define who has access to your custom field. For more information, see Restricting Access to Custom Fields.

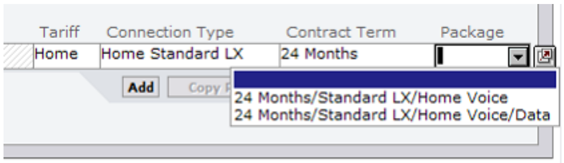
Multiple Dependent Dropdowns

HELP_TOPIC_OUT_OF_DATE_ALERT

You can filter the choices in a dropdown field based on the selections in one or more other fields.

This makes the following examples possible:

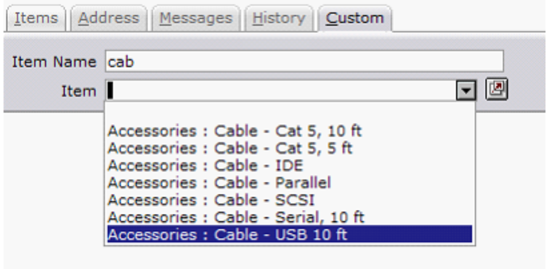
- You add three custom transaction column fields to a transaction form and add an additional field that is filtered based on the selections in the other three.



Tariff	Connection Type	Contract Term	Package
Home	Home Standard LX	24 Months	

24 Months/Standard LX/Home Voice
24 Months/Standard LX/Home Voice/Data

- You create a custom field to help you filter your company's long list of items. By entering a part of the name of the item in this custom field, the item list is filtered to show only items with names containing the text you entered.

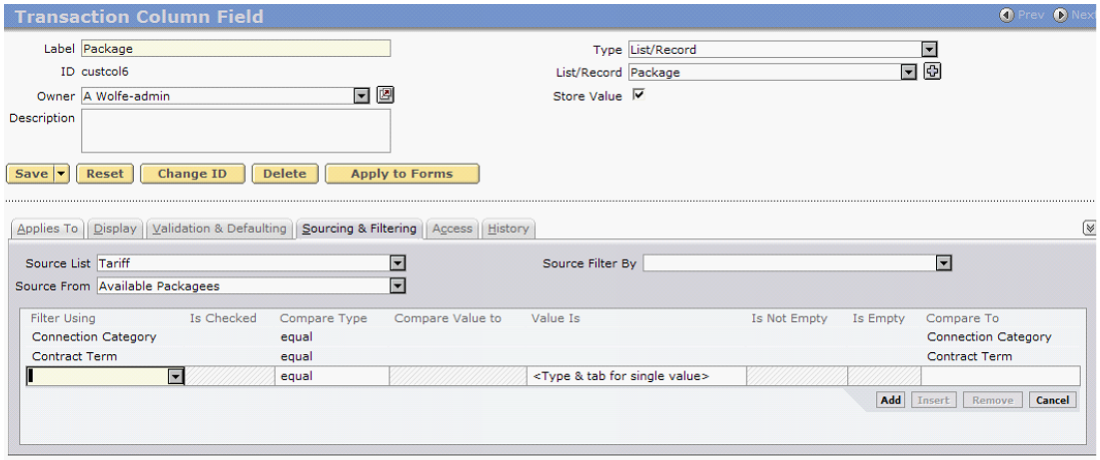


Item Name: cab

Item:

Accessories : Cable - Cat 5, 10 ft
Accessories : Cable - Cat 5, 5 ft
Accessories : Cable - IDE
Accessories : Cable - Parallel
Accessories : Cable - SCSI
Accessories : Cable - Serial, 10 ft
Accessories : Cable - USB 10 ft

Setting up these field dependencies is accomplished through the use of the new **Compare To Field** column on the Sourcing & Filtering subtab on the custom field page. With this column, you can compare values entered in multiple fields on a transaction or record with the values defined on a custom record.



Transaction Column Field

Label	Package	Type	List/Record
ID	custcol6	List/Record	Package
Owner	A Wolfe-admin	Store Value	☑
Description			

Source List: Tariff
Source From: Available Packagees
Source Filter By:

Filter Using	Is Checked	Compare Type	Compare Value to	Value Is	Is Not Empty	Is Empty	Compare To
Connection Category		equal					Connection Category
Contract Term		equal					Contract Term
		equal		<Type & tab for single value>			

The following simple example illustrates the use of multiple dependent dropdowns.

A company sells t-shirts of multiple sizes (small and large), colors (black and white), and styles (v-neck and crew). The company sells any combination of size and color, but the selection of styles is limited to a set of color-size combinations. For example, the company does not sell large, black, v-neck shirts, but it does sell large, black, crew cut shirts.

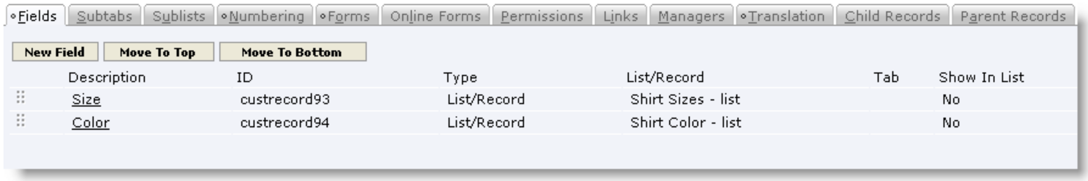
Color and Size are list/record transaction column fields that refer to custom lists. Shirt Style is a custom record type that has the Color and Size fields. There is also Shirt Style transaction column field that refers to the list of Shirt Style custom records.

To understand how this new feature works, it helps to think in terms of **controlling** vs. **dependent** fields. **Controlling** fields are the fields that are used to determine the selections that are available in **dependent** fields.

The Style transaction column field is filtered according to what is selected in the Color and Size transaction column fields. Color and Size are the **controlling** fields; Style is the **dependent** field.

First create custom lists for color and size.

Next, you create a custom record type with two fields – **Size** and **Color** – that refer to the corresponding custom lists.



Description	ID	Type	List/Record	Tab	Show In List
Size	custrecord93	List/Record	Shirt Sizes - list		No
Color	custrecord94	List/Record	Shirt Color - list		No

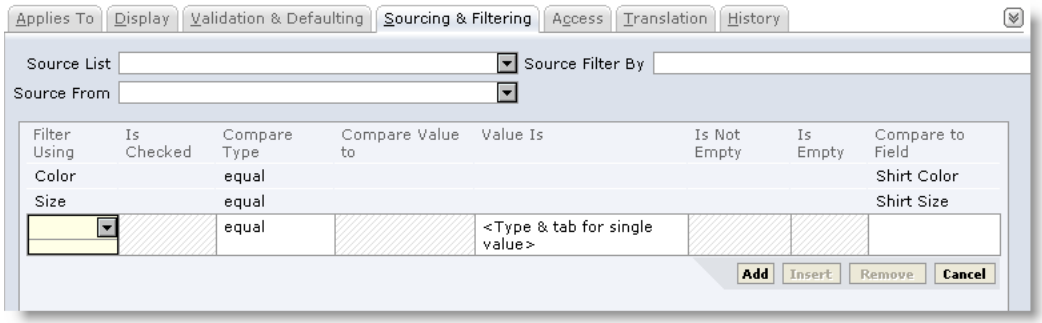
Next, you create a custom record for each **Style** combination you sell. These custom records are named:

- large, black, crew
- small, black, v-neck
- large, white, crew
- small, white, crew
- large, white, v-neck
- small, white, v-neck

Finally, you create three custom transaction column fields:

- **Color** – of type List/Record with your color list selected in the List/Record field
- **Size** – of type List/Record with your size list selected in the List/Record field
- **Style** – of type List/Record with your Style custom record selected in the List/Record field

For the **Style** field, on the Sourcing & Filtering subtab, in the Filter Using column, select **Color** – the field on your custom record that references the **Color** custom field. Set the Compare Type to **equal**, and select Color in the Compare To Field column. Do this for the **Size** field also.



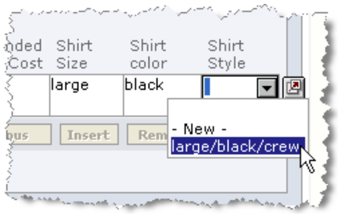
Source List

Source Filter By

Source From

Filter Using	Is Checked	Compare Type	Compare Value to	Value Is	Is Not Empty	Is Empty	Compare to Field
Color		equal					Shirt Color
Size		equal					Shirt Size
		equal		<Type & tab for single value>			

Now, the Style field is filtered by the selections in the **Size** and **Color** fields.



ded Cost	Shirt Size	Shirt color	Shirt Style
	large	black	

To create a field that is filtered by multiple other List/Record fields, complete the following steps:

First, create the custom lists and record type:

1. Create custom lists for each of the controlling fields.
2. Create a custom record type for the options you want to be available in the dependent field. This custom record must contain List/Record fields that reference the custom lists you created in step 1.
3. Create a custom record for each option you want to be available in the dependent field.

Next, add custom fields to the transaction or record:

1. Create custom fields for each of the controlling fields.
2. Create the same kind of custom field of type List/Record with the custom record type from step 2 selected in the List/Record field.

   On the dependent field record:

   1. On the Sourcing & Filtering subtab, in the Filter Using column, select one of the controlling fields.
   2. In the Compare Type field, select the qualifier you want to use to determine the filtering. For example, if you want the selections in the fields to match, select **equal**.

3. In the Compare To column, select the field on your custom record that refers to the field you selected in the Filter Using column.
4. Repeat these steps for each controlling field.
5. Click Save.

You can create multiple layers of dependencies. In the previous example, you might have an another field that is filtered based on your selection in the Style field. You will need a custom record type for each dependent field just as you created for the Style.

Restricting Access to Custom Fields

You can control who can access the information in custom fields, allowing you to maintain the security of your business information. The access you define determines how it can be accessed both on the record as well as through search results and reports.

Access to a field can be based on role, department, or subsidiary. The following custom access levels can be assigned to each department and subsidiary.

- **Edit** – The field and its contents can be viewed and changed.
- **View** – The field can be seen, but its contents cannot be changed. (This permission level affects how the form is accessed on records.)
- **Run** – The field can be seen through reports and search results, but it's contents cannot be changed. (This permission is only applicable to reports and search.)
- **None** – The field cannot be seen, and its contents cannot be changed.

For cases when different access levels are defined for a user's role, department, or subsidiary, the highest level of access is granted. For example, an employee is assigned to a department that has Edit access to a custom field, and the employee's role has been granted View access. The employee has the higher level of access – in this case, Edit access.

In addition to search and reporting, the access level granted to a custom field includes instances where it is referenced by online forms, mail merge operations, and when it is sourced by other custom fields, or referred to by formula fields.

Note: If you take away the Administrator role's access to a custom field, the field will not be accessible to scripts that are run by an administrator. To access the field via scripting, you must edit and restore administrator access to the field.

You can set the level of access you want to grant by default to custom fields. The default access level applies to the roles, departments, and subsidiaries, that you do not define on the Role, Department, and Subsidiary subtabs.

To set default access, edit the custom field record, and click the Access subtab. In the Default Access Level, set the level of access you want to give by default. In the Default Level for Search/Reporting field, select the level of access you want to give through search and reporting.

The access you define on the Role, Department, and Subsidiary subtabs overrides the default access levels.

To set role, department, or subsidiary access restrictions:

1. Edit the custom field record.
2. Click the Access subtab.
3. In the Default Access Level field, set the access level you want to grant to roles, departments, and subsidiaries that you do not specifically define below.
4. In the Default Level for Search/Reporting field, set the level of access you want to grant through search and reports to roles, departments, and subsidiaries that you do not specifically define below.
5. Click the Role, Department, or Subsidiary subtab.
6. In the first column, select the role, department, or subsidiary you want to define access for.
7. In the Access Level column, select the level of access you want to grant.
8. In the Level for Search/Reporting column, select the level of access you want to give this role, department, or subsidiary via search and reporting.
9. Click Add.
10. Repeat these steps for each role, department or subsidiary.
11. Click Save.

Important: The above procedure describes how to limit access to a custom field. You also can check the Apply Role Restrictions box to limit access to an entire custom record, according to a custom field's values. This setting extends access restrictions based on class, department, location, or subsidiary, that are set on role definition pages. For example, if you have set restrictions for roles based on locations, and you want to apply these same restrictions to Subscription custom records, you can check Use Role Restrictions for the Subscription record type's custom field that stores location values. See Applying Role-Based Restrictions to Custom Records.

Access Level History

For auditing purposes, you can view any changes that have been made to custom field access levels.

To view custom field access changes, open the custom field record. Click the Access subtab, and click the History subtab.

You can view the date and time a change was made, the user who made the change, and the changes that were made.

Bundling Fields With Access Restrictions

If you include a custom field in a bundle that has access restrictions, custom roles that you have given access to are not automatically included in the bundle. If, however, you include those custom roles in the same bundle, the access restrictions are preserved.

The access level assigned to standard roles is preserved when you bundle a custom field that has access restrictions.

Custom field restrictions based on subsidiary or departments are not carried over into the target account since departments and subsidiaries cannot be included in a bundle.

Tracking Changes to Custom Fields

In the History tab for each custom field, information on every saved change is displayed with the following summary information:

- Date/Time
- User
- Label
- Field Type
- Record Type (if Type is List/Record)
- Store Value

Creating Display-Only Custom Fields

You can create custom fields that display information but do not store that information with any record or transaction. You can create a display only field of any type with any display, default, validation or sourcing options.

A display-only field is especially useful when used in addition to sourcing information. Fields that use sourcing with a display only field always display the most accurate information. This is because the field is not stored so the information must be retrieved from the source field every time the page is loaded.

To create a display-only custom field:

1. Either edit a custom field, or create a new custom field.

   If you want to create a new field, see Kinds of Custom Fields to determine the type of field you want to create and find step-by-step instructions for each type of custom field.
2. Clear the Store Value box.
3. Click Save.

The information entered in this custom field is no longer stored in your account.

For more information on using display types, see Setting Display Options for Custom Fields.

Table of Custom Field Type Descriptions

When creating a custom field, you select the type of field you want to create, depending on the kind of information you want to store in the field.

The following table outlines all available custom field types. For information on creating a custom field, see Creating a Custom Field.

Field Type	Description	Example
Check box	This lets you record a true or false answer by placing a check mark in the custom field.	For example, you want to track orders by delivery. You create a check box custom field called Delivery that appears on sales transactions. Then, you check the Delivery box if orders are for delivery.
Currency	This lets you enter currency amounts in a custom field.	For example, you want to track the spending limit of each employee in your company. You create a currency custom field that appears on employee records and enter the spending limit for each employee.
Date	This lets you enter or pick dates. Note that changing Date fields to Date/Time fields will automatically convert Dates in existing records to Date/Times with the time value defaulting to midnight of you company's time zone. All existing SuiteScript that referenced the Date field will need to be manually updated.	For example, you want to record projected start dates on estimates. You create a date custom field called Projected Start Date. Dates entered in this type of custom field must follow the date format selected at Setup > Company > General Preferences. **Note:** The date and time for this type of field are based on the logged in user's NetSuite Time Zone preference, set at Home > Set Preferences, not on the browser client time zone.
Date/Time	This field type lets you combine date and time values in one field.	For example, you may want a single field to contain date and time “timestamp” data.

Field Type	Description	Example
	After a user enters a date/time value, the data is rendered in the user's preferred date and time format, as well as the user's time zone. Also note that time values are stored in NetSuite down to the second. If you choose, you can also add Date/ Time custom fields in SuiteScript using the nlobjForm. **addField** (name, type, label, sourceOrRadio, tab) method. **Important:** If you use this method, for the *type* parameter, you must specify 'datetimetz'.	**Note:** The date and time for this type of field are based on the logged in user's NetSuite Time Zone preference, set at Home > Set Preferences, not on the browser client time zone.
Decimal Number	This lets you enter a decimal number in a custom field.	For example, you want to record the distance your customers are from your nearest retail outlet. You create a decimal number custom field to store this information on your employee records.
Document	This lets you select a file cabinet document to preview or download. The field is searchable and can be added to search results. *Note:* The user of the document custom field must have file cabinet access in order to view, select or upload documents.	
Entity	You can create custom fields of type **Entity**. To create a custom entity field, set the field type to List/Record, and set the List/Record type to Entity. By doing so the new “generic” custom entity field will automatically include all records of type contact, customer, employee, group, other name, partner and vendor. If you choose, you can then use sourcing and filtering capabilities to limit the choices in the custom field's dropdown to a subset of the supported entities. For example, on the custom field's definition page, on the Filtering and Sourcing subtab, specify Type in the Filter Using column. In the Value Is column, specify customer, partner, and vendor. Then, when the custom field of type Entity appears on your record, only entities of type customer, partner, and	

Field Type	Description	Example
	vendor will appear as available options in the field's dropdown.	
E-mail address	This lets you enter an e-mail address in a custom field. The e-mail address field creates a link to open your default e-mail client.	For example, you create a custom record type for intern records. You can include a field for their e-mail addresses. Then, you can send e-mail to interns by clicking e-mail addresses on their records.
Free-Form Text	This lets you enter up to 300 characters of text in a custom field.	For example, you want to offer your customers monogramming on certain items. You can create a Free-Form Text field as an item option to record the monogram they want. **Note:** Calculations within free-form text fields display values in scientific notation. To avoid this, use the following syntax to round digits in free-form text fields: ROUND({cost}/0.65,3)
Help	This lets you place helpful text on record pages where your employees enter information. The help that appears is for informational purposes only. It is not stored in your account.	When you create a Help custom field, choose Help in the Type field and enter your text in the Help field.
Hyperlink	This lets you enter a URL in a custom field that links to another Web page. Hyperlink fields should begin with http://, https:// or ftp:// . **Note:** You are also permitted to enter URLs beginning with file:// or \\, but it is not recommended, because clicking on these types of links is not supported, due to security concerns.	For example, you want to link to a vendor's Web site. You create a hyperlink custom field called Web Site that appears on vendor records. First, you enter the company's URL. Then, when you return to the vendor's record, you can click the URL to open the vendor's site in a new window. You can enter up to 999 characters in this field.
Image	This lets you attach an image to a record.	For example, you can attach pictures of your employees to their records.
Inline HTML	This allows you to use HTML to define a custom field to be included on custom pages generated by Suitelets.	
Integer Number	This allows you to enter integers in a custom field.	For example, you can record a special ID number on your company records.
List/Record	This lets you attach lists and records to custom fields. In the List/Record field, select the list or kind of record you want to attach. This also allows you	For example, you want to track orders by the employee who works on the them. You create a custom body field for sales transactions. In the List/Record field, you attach your employee list. On sales

Field Type	Description	Example
	to choose from a list of **public** saved searches.	transactions, you can then select the appropriate employee for the order.
Long Text	This is a text area that can hold up to 100,000 characters. Use Long Text Custom fields where you need greater than 4,000 characters of text for a text area, and where you do not want the field to contain rich text formatting. *Note:* A free form text or text area field can be converted to a long text field but once done, the field can NOT be converted back.	For example, use a Long Text custom field in a custom record to hold the text of a item Warranty, or add to an Online Form to hold the text of a End User License Agreement.
Multiple Select	This lets you create a field where you can make multiple selections from a list or list of records.	For example, you want to track how customers heard about your business. You create a multiple select custom field that appears on customer records and contains a list of possible sources for how customers heard about your business. Then, sales reps use this field to select the different places customers heard about your business.
Password	This lets you create a field that is encrypted in the database. When you view the record it will always show a fixed number of characters regardless of how long the password is. When validating, you pull the encrypted password value into a hidden field and use custom code to encrypt the value the user typed and compare it with the actual encrypted value.	For more information about SuiteScript, contact your account representative.
Percent	This lets you create a field to store a percentage. The information entered in this field can only be an integer from 0	100. The percent sign (%) is automatically added to the information.
Phone Number	This lets you create a field for a telephone number on records and transactions.	For example, you can include a contact number on an event record using a CRM field.
Rich Text	This allows you to enter and format up to 100,000 characters of text in a custom field.	For example, you can include a small bio of your employees with information like marital status, family members and other special information. You can also format the text to create lists, make text bold, italic and underlined and change the font size and color.
Text Area	This lets you enter up to 4,000 characters of text into a custom field.	For example, you want to record special information about how you closed a customer. You could enter it into a text area custom field that appears on customer records.

Field Type	Description	Example
Time of Day	This allows you to enter the time of day into a custom field.	For example, you create a Best Time To Call custom field, and then your sales representatives could enter the best time of day to call a prospect.

Changing Field Types

When you edit a custom field and change the field type, the data that you have previously entered in that field is preserved whenever possible.

The retention of data follows the following rules:

- **Changing to a Different Field Type**

  If you change the field type to a fundamentally different field type, existing data you had entered in that field is deleted. A warning message is displayed when you make this kind of change.

  For example, changing a phone number field to a rich text field would preserve your data, while changing a rich text field to a percent field would not.

- **Changing the Source List**

  If you change the source list for a list/record field or a multiple select field, your existing data in that field is not preserved. A warning message is displayed when you make this kind of change.

- **Switching Numeric Field Types**

  If you change a decimal, currency, or percent type field to an integer field, your data is rounded off to the nearest whole number.

Available Standard Fields and Field Types

Each kind of custom field has standard fields that can be sourced from. Following are tables that list available standard fields and the required field type for each kind of custom field.

Transaction Body Fields

Source From Entity

The type of entity used to source from depends on the type of transactions you apply your field to. A **purchase** transaction sources from **vendor** records. A **sales** transaction sources from **customer** records, and an **expense report** sources from **employee** records.

In the List/Record field, you must choose the Employee list for sourcing to work properly.

Field	Type	Sales Transactions	Purchase Transactions	Expense Reports
Name	Free-Form Text	×	×	×
Bill To	Text Area	×	×	×
Ship To	Text Area	×	×	
Phone	Phone Number	×	×	×
Fax	Phone Number	×	×	
E-mail	E-mail Address	×	×	×
City	Free-Form Text	×	×	×
State	Free-Form Text	×	×	×
Zip	Free-Form Text	×	×	×
Country	Free-Form Text	×	×	×
Sales Rep	List/Record*	×		
Expected Close Date	Date	×		
Renewal Date	Date	×		
Contact	Free-Form Text	×	×	
Alt. Contact	Free-Form Text	×	×	
Alt. Phone	Phone Number	×	×	
Balance	Currency	×	×	
Credit Limit	Currency	×	×	
Account	Free-Form Text	×	×	
1099 Eligible	Check Box		×	
Tax ID	Free-Form Text		×	
Legal Name	Free-Form Text		×	
Supervisor	List/Record*			×
Soc. Sec. #	Free-Form Text			×

Source From Ship To

When you choose Ship To in the Source From field, information is sourced from the customer record chosen in the **Ship To** field of **purchase** transactions.

Field	Type
Name	Free-Form Text
Bill To	Text Area
Ship To	Text Area
Phone	Phone Number
Fax	Phone Number

Field	Type
E-mail	E-mail Address
City	Free-Form Text
State	Free-Form Text
Zip	Free-Form Text
Country	Free-Form Text
Sales Rep	List/Record*
Expected Close Date	Date
Renewal Date	Date
Contact	Free-Form Text
Alt. Contact	Free-Form Text
Alt. Phone	Phone Number
Balance	Currency
Credit Limit	Currency

Source From Sales Rep

When you choose Sales Rep in the Source From field, information is sourced from the employee record chosen in the **Sales Rep** field of **customer** records.

Field	Type
Name	Free-Form Text
Bill To	Text Area
Phone	Phone Number
E-mail	E-mail Address
City	Free-Form Text
Country	Free-Form Text
Supervisor	List/Record*
Soc. Sec. #	Free-Form Text

Transaction Column Fields

Source From Item

In the List/Record Field, you must choose the Vendor list for sourcing to work properly.

Field	Type
Name	Free-Form Text
Display Name	Free-Form Text
Vendor Name	Free-Form Text

Field	Type
Online Name	Free-Form Text
Available Online	Check Box
Base Price	Currency
Cost	Currency
Preferred Vendor	List/Record*
On Hand	Decimal Number

Source From Customer

When you choose Customer in the Source From field, information is sourced from the customer record chosen in the **Customer:Project** field of **purchase** transactions.

In the List/Record field, you must choose the Employee list for sourcing to work properly.

Field	Type
Name	Free-Form Text
Bill To	Text Area
Ship To	Text Area
Phone	Phone Number
Fax	Phone Number
E-mail	E-mail Address
City	Free-Form Text
State	Free-Form Text
Country	Free-Form Text
Sales Rep	List/Record*
Expected Close Date	Date
Renewal Date	Date
Contact	Free-Form Text
Alt. Contact	Free-Form Text
Alt. Phone	Phone Number
Balance	Currency
Credit Limit	Currency
Account	Free-Form Text

Transaction Item Options

Source From Item

In the List/Record Field, you must choose the Vendor list for sourcing to work properly.

Field	Type
Name	Free-Form Text
Display Name	Free-Form Text
Vendor Name	Free-Form Text
Online Name	Free-Form Text
Available Online	Check Box
Base Price	Currency
Cost	Currency
Preferred Vendor	List/Record*
On Hand	Decimal Number

CRM Fields

Source From Company or Contact

When you choose Company in the Source From field, information is sourced from the record chosen in the **Company** field of **case** records.

When you choose Contact in the Source From field, information is sourced from the record chosen in the **Contact** field of **case** records.

In the List/Record field, you must choose the Employee list for sourcing to work properly.

Field	Type
Name	Free-Form Text
Bill To	Text Area
Ship To	Text Area
Phone	Phone Number
Fax	Phone Number
E-mail	E-mail Address
City	Free-Form Text
State	Free-Form Text
Country	Free-Form Text
Sales Rep	List/Record*
Expected Close Date	Date
Renewal Date	Date
Contact	Free-Form Text
Alt. Contact	Free-Form Text
Alt. Phone	Phone Number

Field	Type
Balance	Currency
Credit Limit	Currency
Account	Free-Form Text

Source From Item

When you choose Item in the Source From field, information is sourced from the record chosen in the **Item** field of **case** records.

In the List/Record Field, you must choose the Vendor list for sourcing to work properly.

Field	Type
Name	Free-Form Text
Display Name	Free-Form Text
Vendor Name	Free-Form Text
Online Name	Free-Form Text
Available Online	Check Box
Base Price	Currency
Cost	Currency
Preferred Vendor	List/Record*
On Hand	Decimal Number

Source From Assigned

When you choose Assigned in the Source From field, information is sourced from the record chosen in the **Assigned To** field of **case** records.

In the List/Record field, you must choose the Employee list for sourcing to work properly.

Field	Type
Name	Free-Form Text
Bill To	Text Area
Phone	Phone Number
E-mail	E-mail Address
City	Free-Form Text
Country	Free-Form Text
Supervisor	List/Record*
Soc. Sec. #	Free-Form Text

Entity Fields

Source From Sales Rep or Supervisor

When you choose Sales Rep in the Source From field, information is sourced from the record chosen in the **Sales Rep** field of **customer** records.

When you choose Supervisor in the Source From field, information is sourced from the records chosen in the **Supervisor** field of **employee** records

In the List/Record field, you must choose the Employee list for sourcing to work properly.

Field	Type
Name	Free-Form Text
Bill To	Text Area
Phone	Phone Number
E-mail	E-mail Address
City	Free-Form Text
State	Free-Form Text
Zip	Free-Form Text
Country	Free-Form Text
Supervisor	List/Record*
Soc. Sec. #	Free-Form Text

Item Fields

Source From Preferred Vendor

When you choose Preferred Vendor in the Source From field, information is sourced from the record chosen in the **Preferred Vendor** field of **item** records.

Field	Type
Name	Free-Form Text
Bill To	Text Area
Ship To	Text Area
Phone	Phone Number
Fax	Phone Number
E-mail	E-mail Address
City	Free-Form Text
State	Free-Form Text

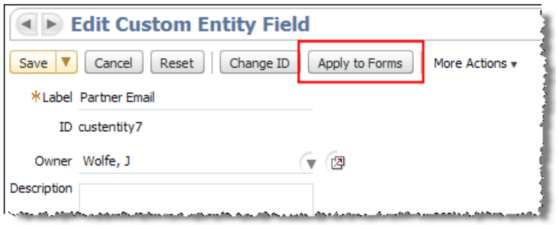
Field	Type
Zip	Free-Form Text
Country	Free-Form Text
Contact	Free-Form Text
Alt. Contact	Free-Form Text
Alt. Phone	Phone Number
Balance	Currency
Credit Limit	Currency
Account	Free-Form Text
1099 Eligible	Check Box
Tax ID	Free-Form Text
Legal Name	Free-Form Text

Renaming Custom Fields

When you change the name of a custom field, the name is **not** automatically updated on custom forms that contain the field.

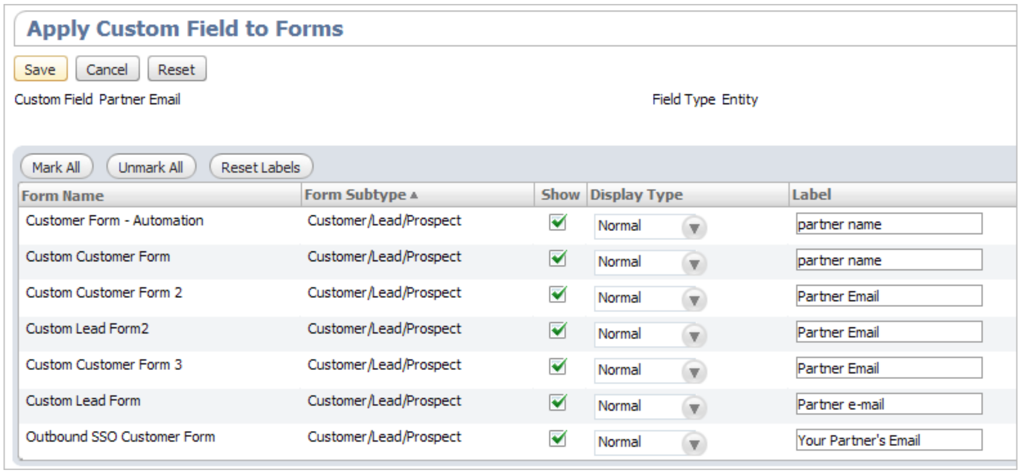
To apply the new name to custom forms:

1. After making changes to your custom field, click Apply to Forms.



*Label Partner Email
ID custentity7
Owner Wolfe, J
Description

2. On the Apply Custom Field to Forms page, change the field label for each form that you want to reflect the change.



Custom Field Partner Email

Field Type Entity

Form Name	Form Subtype ▲	Show	Display Type	Label
Customer Form - Automation	Customer/Lead/Prospect	☑	Normal	partner name
Custom Customer Form	Customer/Lead/Prospect	☑	Normal	partner name
Custom Customer Form 2	Customer/Lead/Prospect	☑	Normal	Partner Email
Custom Lead Form2	Customer/Lead/Prospect	☑	Normal	Partner Email
Custom Customer Form 3	Customer/Lead/Prospect	☑	Normal	Partner Email
Custom Lead Form	Customer/Lead/Prospect	☑	Normal	Partner e-mail
Outbound SSO Customer Form	Customer/Lead/Prospect	☑	Normal	Your Partner's Email

3. Click Save.

Mass Updating Custom Fields

To be available for mass update, a custom field must meet the following criteria:

- It must support inline editing.
- It must be displayed on your preferred form for the record type being updated.
- It must be stored.
- It must not have a sourcing relationship.

Custom Lists

A custom list is a list of values that you can use in custom fields on your forms and records. Custom lists allow you to set up predefined choices for your employees and customers to select when entering transactions and records. You can create an unlimited number of custom lists and an unlimited number of values for each list.

To go to the Lists page click Setup > Customization > Lists. On this page:

- To edit an existing custom list, click the list and edit as desired
- To create a new custom list, click **New**
- To show all lists, click the Inactive box

To save time, you should create any custom lists or subtabs that may be needed when implementing the more advanced customizations for your account.

For help on defining a custom list, see Creating a Custom List.

Important: Custom lists are intended for use with small, fixed, related sets of data. It is recommended that each custom list include no more than 1000 records.

Creating a Custom List

A custom list is a list of values that you can use in custom fields on your forms and records. Custom lists allow you to set up predefined choices for your employees and customers to select when entering transactions and records. You can create an unlimited number of custom lists and an unlimited number of values for each list.

To create a custom list:

1. Go to the New Lists page.

   Click Setup > Customization > Lists > New. The Custom List page is displayed.
2. In the Name field, enter a name for the list.
3. If you use custom code, enter a unique ID for this custom list.
4. Select the owner of this custom list.

   You are selected by default as the owner.

   Only the owner and other users with edit or full permission levels may modify this custom list.
5. Enter a description of this custom list.
6. By default, values are listed in the order in which they are entered. To list values in alphabetical order, click that radio button.
7. Check the Matrix Options List box if this list is for matrix items.

   If you check this box, an Abbreviation column is added to the Values list.

   Note: The Accounting Matrix Items feature must be enabled to use this option. If NOT enabled, the Matrix Options List box is NOT displayed.
8. In the Value field, enter a value for the list.
9. Click Add.
10. Repeat steps 5 and 6 to add additional values to the list.
11. Click Save.

The custom list can now be used in your custom fields. For details, see Creating a Custom Field and Adding Translations for Custom Lists.

Important: Custom lists are intended for use with small, fixed, related sets of data. It is recommended that each custom list include no more than 1000 records.

Adding Translations for Custom Lists

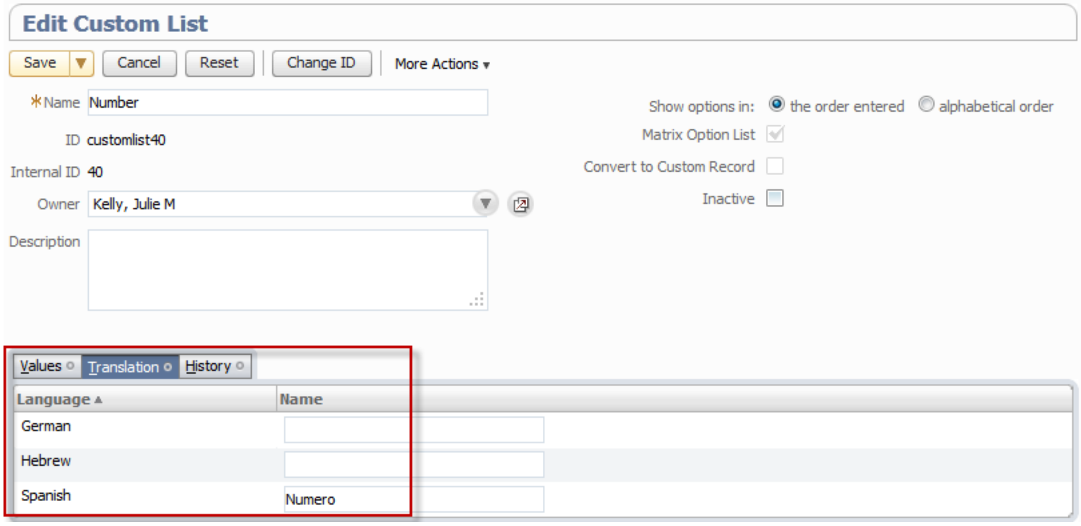
If the Multi-Language feature is enabled in your account, you can translate the name of a custom list, and its available values, so that they match the language of the NetSuite user interface. For details, see the following:

- Translating a Custom List Name
- Translating Custom List Values

Important: Before you can add these translations, you need to select translation languages at Setup > Company > General Preferences, on the Languages subtab. This subtab lists both system-supported languages that can be used for the NetSuite user interface (and are available at Home > Set Preferences), and additional languages that can used for Web site translations only (and are not available at Home > Set Preferences). You should only enter translations for system-supported languages, because these are the only languages that can be displayed in the user interface. For details, see the help topic *Setting Up Multiple Languages*.

Translating a Custom List Name

You can define translations for a custom list name on the Translation subtab of the custom list page.



Edit Custom List

*Name	Number
ID	customlist40
Internal ID	40
Owner	Kelly, Julie M
Description	
Show options in:	● the order entered ○ alphabetical order
Matrix Option List	☑
Convert to Custom Record	□
Inactive	□

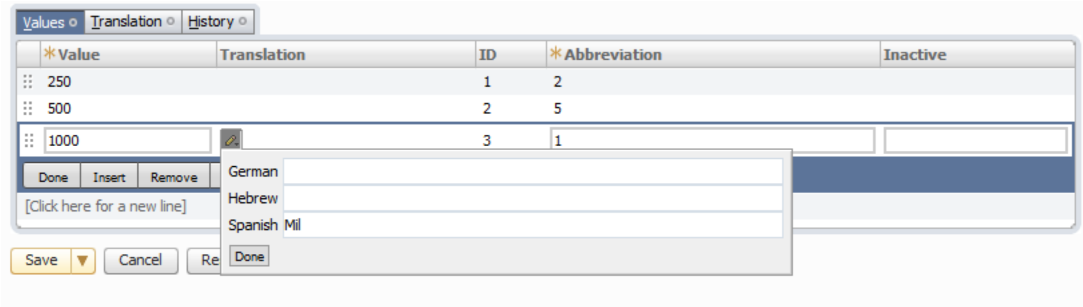
Values ○ Translation ○ History ○

Language ▴	Name
German	
Hebrew	
Spanish	Numero

Note: The maximum length for a custom list name's translation is 30 characters.

Translating Custom List Values

You can define translations for a custom list's values, on the Values subtab of the custom list page:



*Value	Translation	ID	*Abbreviation	Inactive
250		1	2	
500		2	5	
1000		3	1	

German	
Hebrew	
Spanish	Mil

Note: The maximum length for a custom list value's translation is 60 characters.

Chapter 3 Custom Forms

Forms are the pages used to enter information into the NetSuite database. The standard set of forms provided with your NetSuite account can be customized to better suit your business needs. For example, you may want to reorganize subtabs or rename fields to better match your business workflow and terminology. Once you have created a custom form, it can be set as the preferred or default form for a page, or selected as needed from a custom form dropdown list.

Note: Form preferences are controlled by settings on the custom forms page as well as settings defined for each role. Any settings defined for a given role override the preferred form settings on the forms page.

Custom Entry Forms

An entry form is a form you use to enter information and create entity records in NetSuite. You can create your own custom entry forms by starting with an existing form and then rearranging and renaming fields and subtabs, hiding or disabling certain fields, hiding or renaming buttons, making specific fields mandatory, adding custom fields or applying custom code.

For detailed information on how to customize an entry form, see Creating Custom Entry and Transaction Forms.

Once you have created a custom entry form, you can set it as the default form for your users' roles.

Custom Transaction Forms

A transaction form is the form you use to enter and print transactions in NetSuite. You can create your own custom transaction forms by starting with an existing form and then rearranging and renaming fields and subtabs, hiding or disabling certain fields, hiding or renaming buttons, making specific fields mandatory, adding custom fields or applying custom code.

For detailed information on how to customize a transaction form, see Creating Custom Entry and Transaction Forms.

Once you have created a custom transaction form, you can set it as the default form for your users' roles.

You can also link transaction forms together to create transaction workflows. For more information, see Linking Transaction Forms.

Transaction Form Layouts

You can create custom layouts for your custom transaction forms. With custom layouts, you can customize the look and feel of the transaction forms you print or email to those with whom you do business.

You can define different custom transaction layouts for printing forms as PDF and for printing forms as HTML. To define whether to print using PDF or HTML, go to Home > Set Preferences > Transactions, check or clear the Print Using HTML box, and click Save.

Using custom layouts, you can hide and show fields, move and resize fields and change the font and colors on your forms.

You can choose from standard or classic layouts to use for your custom layouts. Standard layouts offer a more streamlined, updated look with labels appearing next to the information and without a background color. Classic layouts include labels above information and a black background color.

You can create custom layouts for the following types of printed forms:

- Bill of Materials
- Check
- Item Label
- Mailing Label
- Packing Slip
- Payment Receipt
- Payment Voucher
- Picking Ticket
- Price List
- Remittance Slip
- Return Form
- Shipping Label
- Statement
- Transaction

To create custom layouts, go to Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and click the Customize link next to a layout. Make your changes and click Save. You can choose default layouts to apply to one or more types of forms by checking box(es) in the Preferred column at Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and clicking Submit.

For more details, see Transaction Form PDF Layouts and Transaction Form HTML Layouts.

Advanced PDF/HTML Templates

Advanced PDF/HTML templates provide an alternate model for customizing printed and emailed records. These templates support more customization capabilities than transaction form layouts, also known as legacy layouts. You can use these templates to produce either PDF or HTML output, depending upon the settings of your print and email preferences.

Important: In order to use Advanced PDF/HTML Templates, you must enable the related feature. The beta version of this feature supports a selected set of records. This feature currently supports Cash Refund, Cash Sale, Credit Memo, Customer Deposit, Invoice, Packing Slip, Payment Voucher, Picking Ticket, Purchase Order, Quote (Estimate), Return Authorization, and Sales Order templates.

You can set custom forms for supported transaction types to use standard advanced PDF/HTML templates provided by NetSuite or custom templates that you have created in a template editor available in the NetSuite user interface. This editor supports both rich text editing and HTML source editing, and uses industry-standard tools and syntax.

For more details, see Advanced PDF/HTML Templates (Beta).

Creating Custom Entry and Transaction Forms

To create a custom entry or transaction form:

1. Do one of the following to select the desired form to customize.

   - Go to the type of form you want to customize, and click Customize.
   - Click Customize or Edit next to the form name at Setup > Customization > Entry Forms.
   - Click Customize or Edit next to the form name at Setup > Customization > Transaction Forms.

   Note: Forms labeled as (External) are used in the Customer Center and My Account section of your Web site.

2. In the Name field, enter a name for your custom form.
3. Set the custom form properties.

   Options vary depending on the type of form being customized but may include the following.

For both Entry and Transaction Forms:

- **Form is Preferred** – check to make this form your preferred form. Only one form can be defined as the preferred form per transaction type. Checking this box clears any previously defined preferred forms. The preferred form is automatically used when entering transactions of this type. For details on how preferred forms are defined, see Defining Preferred Forms.

  Note the following about marking an entry or transaction form Preferred for the Customer Center role:

  - External forms, meaning forms with names appended with (External), can be marked Preferred for Customer Center roles, but not for other roles.
  - Forms that are not external cannot be marked Preferred for Customer Center roles.
  - When a non-online order form is marked Preferred for Customer Center, it is saved as the form for the order. However, an online form is not saved as the form for an order, even if it is preferred; instead the preferred non-online order form is used.
- **Store Form with Record** – check to store a reference to this form with each record created by it. When viewed or edited, any record that was originally created with this form is displayed using this form rather than the user's preferred form.
  - For custom entry forms, the Store Form with Record option is disabled by default.
  - The Store Form with Record option is only available for a subset of transaction forms. For this subset, this option is enabled by default. For other transaction forms, this option is not available and custom forms are always stored with records. For more information, see Storing Custom Forms with Transactions.

For Entry Forms Only:

- **Enable Field Editing on Lists** –check to allow direct list editing of this form. Direct list editing allows users to edit fields on this form from within the record view. When enabled, fields that can be edited from within the record view display the direct list editing icon.
- **Use for Pop-ups** – check to use this form in popups when you enter information on-the-fly. This capability is available for entity forms, item forms, and custom record forms only. For each type of form, only one form can be set as the pop-up form for each type of form. When set for a new form, the form previously defined as the pop-up form is unset.

For Transaction Forms Only:

- **Allow Add Multiple** – uncheck to hide the Allow Add Multiple button on transaction item lists. You should turn off the Add Multiple button on any forms that rely on custom code line item validation scripts. This is because, the Add Multiple button is displayed on item machines and allows you to add multiple items at a time to the item list. However, when items are added in this manner, any Validate Line custom code events defined for the form will NOT fire.

- **Printing Type** - (Available for cash refund, cash sale, credit memo, customer deposit, invoice, packing slip, payment voucher, picking ticket, purchase order, quote (estimate), return authorization, and sales order forms only when the Advanced PDF/HTML Templates feature is enabled.) The Legacy option that is selected by default allows you to set the custom form to use legacy PDF layouts and HTML layouts. Choose the Advanced option to set the custom form to use an advanced PDF/HTML template instead. For more details, see Advanced PDF/HTML Templates (Beta).
- **Advanced PDF/HTML Template** - (Available for cash refund, cash sale, credit memo, customer deposit, invoice, packing slip, payment voucher, picking ticket, purchase order, quote (estimate), return authorization, and sales order forms only when the Advanced PDF/HTML Templates feature is enabled, and the Printing Type is set to Advanced) — select a template for your form.
- **Transaction Form PDF Layouts** – select a layout for your form. To customize layouts, click the Custom Layouts link in the upper-right corner of the page. For more details, see Transaction Form PDF Layouts.
- **Transaction Form HTML Layouts** – select a layout for your form. Standard and Classic layouts exist for all the standard form types other than shipping label. These are assumed to be printed using PDF. To customize layouts, click Customize. For more details, see Transaction Form HTML Layouts.
- **Remittance Slip** – specify which remittance slip is used on invoices, statements, return authorizations and packing slips. To prevent the current transaction from printing with a remittance slip, select None.

  Note: To use this feature, the Print Remittance Form with Invoices & Statements preference must be turned on.
- **Disclaimer** – enter a policy statement or message. You can enter up to 4,000 characters, including spaces, for this message.
- **Address** – enter an address to be used only on this form. If you do NOT enter an address, the default address entered at Setup > Company > Company Information is used.
- **Logo** – select a logo to be used only on this form. You must first upload the image to you File Cabinet at Documents > Files > Images. If you do NOT select a logo, the default logo selected at Setup > Company > Company Information is used.
- **Columns Space** – This number is the maximum number of inches of printable space allowed on your form. The measurement is determined by the Page Width of the layout you choose. You can change the page width by creating a custom layout. To do this, click the Custom Layouts link in the upper right-hand corner of the page.

- **Columns Width** – This is the total of all the columns on your form. This measurement is determined by the values that you enter on the column tab for your custom form. If your columns' width totals more than your columns' space, NetSuite will adjust the widths proportionally to fit on the page.

1. Once you have created a custom form, you should configure the subtabs. See Configuring Subtabs for Custom Entry and Transaction Forms.

Important: As you are configuring your custom form, be sure to consider whether tax data is required for transactions for which you want to use the form. Only a form that includes required tax-related fields can be used for a transaction with tax consequences. Note that you cannot control naming for tax fields through form customization; you must go to Setup > Accounting > Set Up Taxes. See the help topic *Customizing Tax Fields on Transaction Forms*.

Note: If you create or edit custom project forms when the Advanced Projects feature is enabled, be aware that these forms may be altered if you later disable this feature. Immediately after you disable Advanced Projects, you need to review custom project forms to see if they have been changed, and if necessary, edit them to fit your requirements. For information about working with this feature, see the help topics *Enabling Project Features* and *Advanced Projects*.

Storing Custom Forms with Transactions

The Store Form with Record option, available for custom entry and some custom transaction forms, indicates whether a reference to the form should be stored with each record created by it. When this reference is stored, whenever a record is viewed or edited, that form is used, rather than the user's preferred form. This option is enabled by default for custom transaction forms.

For some transaction types, the custom form is always stored with each transaction where it was used. For these transaction types, the Store Form with Record option is not available on custom forms, so users cannot change it.

The following table lists custom transaction form types and whether users have the choice of changing the Store Form with Record option from the default of True.

Custom Transaction Form Type	Can Choose Not to Store Form with Record?
Assembly Build	Yes
Assembly Unbuild	Yes
Bill of Materials	No, form always stored with record
Bill Payment	No, form always stored with record

Custom Transaction Form Type	Can Choose Not to Store Form with Record?
Cash Refund	No, form always stored with record
Cash Sale	No, form always stored with record
Check	Yes
Credit Card Charge	Yes
Credit Memo	No, form always stored with record
Customer Deposit	No, form always stored with record
Customer Refund	Yes
Deposit	Yes
Estimate (Quote)	No, form always stored with record
Expense Report	Yes
Inventory Adjustment	Yes
Inventory Cost Revaluation	Yes
Inventory Worksheet	Yes
Invoice	No, form always stored with record
Item Fulfillment	No, form always stored with record
Item Receipt	No, form always stored with record
Journal Entry	Yes
Opportunity	Yes
Packing Slip	No, form always stored with record
Payment	No, form always stored with record
Picking Ticket	No, form always stored with record
Price List	No, form always stored with record
Purchase Order	No, form always stored with record
Remittance Slip	No, form always stored with record

Custom Transaction Form Type	Can Choose Not to Store Form with Record?
Return Authorization	No, form always stored with record
Return Form	No, form always stored with record
Sales Order	No, form always stored with record
Shipping Label	No, form always stored with record
Statement	No, form always stored with record
Transfer Order	Yes
Vendor Bill	Yes
Vendor Credit	Yes
Vendor Return Authorization	Yes
Work Order	Yes

Configuring Subtabs for Custom Entry and Transaction Forms

On the Subtab subtab, you can select which subtabs to display on your form and provide a custom heading for each subtab.

To modify the available subtabs:

1. In the Show column, check the boxes for the subtabs you want to display on the form. Clear the boxes for the subtabs you don't want to display.
2. In the Label column, edit the headings for the subtabs as desired.
3. To rearrange the subtabs, click the desired line to drag and drop it to the desired position or click Move to Top / Move to Bottom.
4. Once you have configured the subtabs, you should configure the fields or screens. See Configuring Fields or Screens.

Configuring Field Groups

Click the Field Groups subtab to customize the field groups that appear on your forms. Note that the Field Groups subtab appears on custom form pages only **after** you have upgraded and deployed the standard forms in your account.

Note: If you are not familiar with the term field groups, see the help topic *Field Groups* in the NetSuite Help Center.

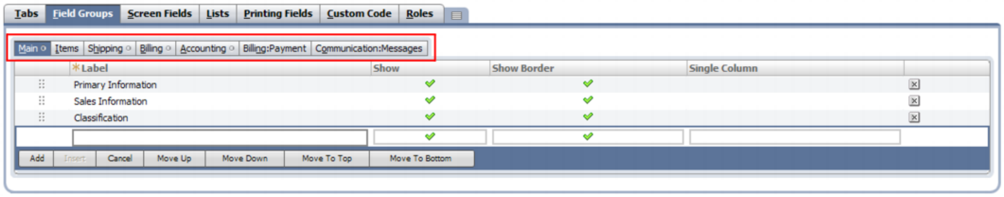
You can use the Field Groups subtab to change the UI label of a field group, the fields within the field group, and the placement of a field group on the page. You can also use the Field Groups subtab to create new field groups and organize all fields into specific field groups.

Be aware that the order of field groups on the Field Groups subtab determines the order of field groups on your form. For example, in the figure below, the Primary Information field group will appear on top of the Sales Information and Classification field groups.

Be aware that adding custom fields to field groups can cause field text to be displayed differently than you might expect based on the display size attributes set for custom fields, as described in Configuring Field Groups.

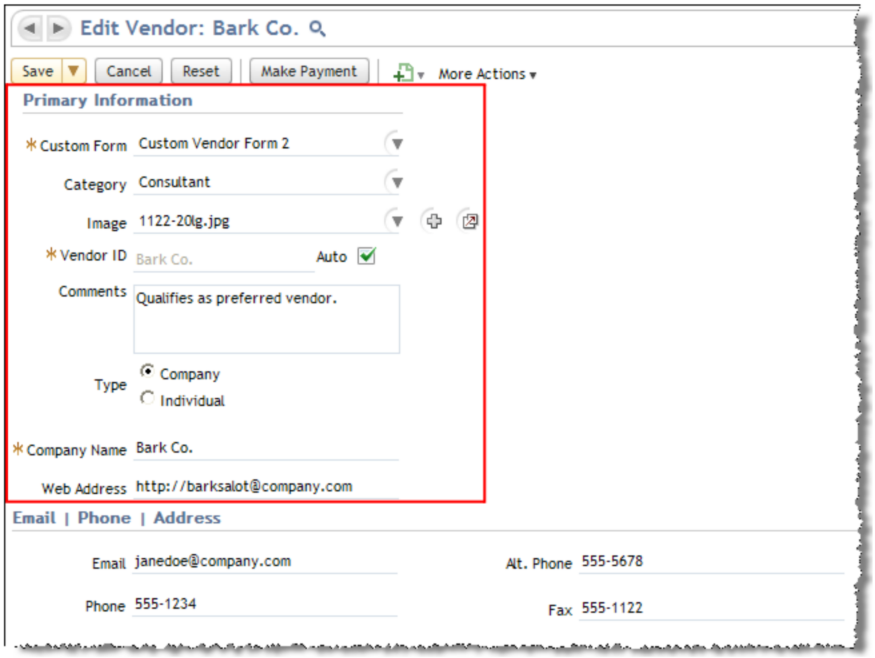
To work with field groups:

1. On the Field Groups tab, click the Main subtab to customize field groups that appear in the main body area of a page. Click any of the other Field Group subtabs to customize the field groups that appear on a form's subtabs.



*Label	Show	Show Border	Single Column	
Primary Information	✓	✓		☒
Sales Information	✓	✓		☒
Classification	✓	✓		☒
	✓	✓		

2. In the Label column, edit field group headings.
3. In the Show column, uncheck a field group to hide it on the form.
4. In the Show Border column, uncheck a field group to hide the field group title and title line.
5. In the Single Column column, check a field group if you want it to display vertically rather than horizontally, which is the default display. The following figure provides an example of the Primary Information field group vertically displayed.



Edit Vendor: Bark Co.

Primary Information

* Custom Form	Custom Vendor Form 2
Category	Consultant
Image	1122-20lg.jpg
* Vendor ID	Bark Co. Auto ☑
Comments	Qualifies as preferred vendor.
Type	● Company ○ Individual
* Company Name	Bark Co.
Web Address	http://barksalot@company.com

Email | Phone | Address

Email	janedoe@company.com	Alt. Phone	555-5678
Phone	555-1234	Fax	555-1122

Note that if you select Single Column for three fields groups together, the field groups will appear side by side in the UI, as three single (field group) columns make a row.

6. Click the X icon in the last column to delete the field group. Clicking the X will not remove the fields from your form; they will continue to display at the bottom of the page. However, the fields will not be assigned to any field group, as that field group no longer exists.

7. Once you have configured your field groups, you can click the Fields subtab (for Entry forms) or the Screen Fields subtab (for Transaction forms) to configure the fields that appear in each field group. See Configuring Fields or Screens.

Field Groups on Custom Forms of Custom Records

You can add field groups to both new and existing custom record custom forms. Note that the following steps are for adding field groups to an **existing** custom record.

To add field groups to the custom form of a custom record:

1. Go to Setup > Customization > Record Types.
2. On the Record Types list page, select your record type.
3. Click the Forms subtab.

4. On the Forms subtab, click Edit next to the custom record's custom form. (Note that if you do not yet have a custom form, click Customize next to the record type to create a new custom form.)
5. Next, click the Field Groups subtab.
6. Provide a UI label for the first field group, set all other field group attributes, and click Add to add the new field group.
7. After adding all field groups, click Save.
8. Once you have configured your field groups, you can click the Fields subtab (for Entry forms) or the Screen Fields subtab (for Transaction forms) to configure the fields that appear in each field group. See Configuring Fields or Screens.

Configuring Fields or Screens

In the Fields subtab of entry forms or Screen Fields subtab of transaction forms, fields available on each subtab for the current form can be configured. These fields can also be moved to display on a different subtab or in different field groups.

Note: Even if you have disabled a subtab as described in Configuring Subtabs, you can still configure fields for that subtab. This is useful should you later decide to include the subtab on the form.

To configure fields for each available tab:

1. In the Show column, check the boxes next to the fields you want to display. Clear the boxes for the fields you don't want to display.

   Warning: Hiding fields on a form can hide other related fields. For example, hiding the Credit Card Approved box on the sales order form also hides fields related to Address Verification System (AVS).

2. In the Quick Add column (entry forms only), check the boxes next to the fields you want to define as direct list editable and available in Quick Add portlets.

   If a field is checked here, and the Enable Field Editing on Lists is also checked, then that field can be edited directly and the record is saved when the user clicks off of the field. Direct list editable fields are also available when attaching contacts or scheduling activities to records. For more information, *Using Inline Editing see the help topic Using Direct List Editing*.

   Quick Add portlets can be made available on users' dashboards to allow the quick addition of a selected record type without shifting to the record type page. For more information, *Using the Quick Add Portlet*.

3. In the Mandatory column, check the boxes next to the fields you want to **require** on your entry form.

Note: You cannot clear mandatory boxes for fields that NetSuite requires or fields defined as mandatory in the custom field definition.

4. In the Display Type column, select the display type for each field.

   For more information about display types, see Setting Display Options for Custom Fields.

5. Check the Default Checked box for each checkbox field you want checked by default.

6. In the Label field, edit the name of any of the field as desired.

   - If you change the label of a field on the Screen Fields subtab, the new label is also applied to the field on the Printing Fields subtab.
   - In accounts where the Multi-Language feature is enabled, the label you edit here applies only to your current language, as set at Home > Set Preferences. To modify the label in other languages, you need to reset your language preference and reedit the label for each one. Note that two different settings are available for languages such as English, French, and Spanish, for example English (U.S.) and English (International).

7. In the Field Groups dropdown, select the field you want the field to appear in.

8. Check any box in the Column Break column to insert a column break after that field.

   Be aware that adding a custom field to a field group or inserting a column break after a field can cause field text to be displayed differently than you might expect based on the display size attributes set for custom fields, as described in Configuring Field Groups.

9. If you want to include a blank line before a field, enter the number of blank lines in the Space Before column.

10. If you want to associate a field with the field immediately above it, check the Same Row as Previous box. An associated field shares the same Show/Hide setting as the previous field, belongs to the same Field Group as the previous field, and is displayed together on the form with the previous field. See Associating Related Fields on Custom Forms.

11. Rearrange the fields as desired.

    - Move each field to the desired subtab. Click Move Elements Between Subtabs. For details, refer to Moving Fields and Lists Between Subtabs.
    - Rearrange the order of the fields on each subtab. Click the desired line and then drag and drop it to the desired position or click Move to Top / Move to Bottom.

12. For transaction forms only, set the following options:

    - In the Default Checked column, set the default value for check box fields. This option is only available for toggle fields where the item is either checked or unchecked. If you enable the Default Checked box, the field is automatically checked unless the user specifically unchecks it.

- Click the Columns subtab, and arrange the line-item columns for your transaction entry form by dragging and dropping lines as desired. The order of the columns on the screen does NOT have to match the order of the printed columns of your form.

13. On the Columns subtab in the Items Filter field, select a saved search you want to use to filter the items that appear in the Items dropdown on this form.

    For information on creating a saved search, see the help topic *Defining a Saved Search*.

14. Click Save.

15. If desired, click New Field to create any new custom fields.

    Ensure that you save any changes before creating New Fields from within the form. Selecting New Field causes you to leave the custom form to go to the custom field page. For detailed information on creating custom fields for your forms, see Custom Fields.

16. Once you have configured the fields or screens, you should configure the printing fields (for transaction forms only). See Configuring Printing Fields for details.

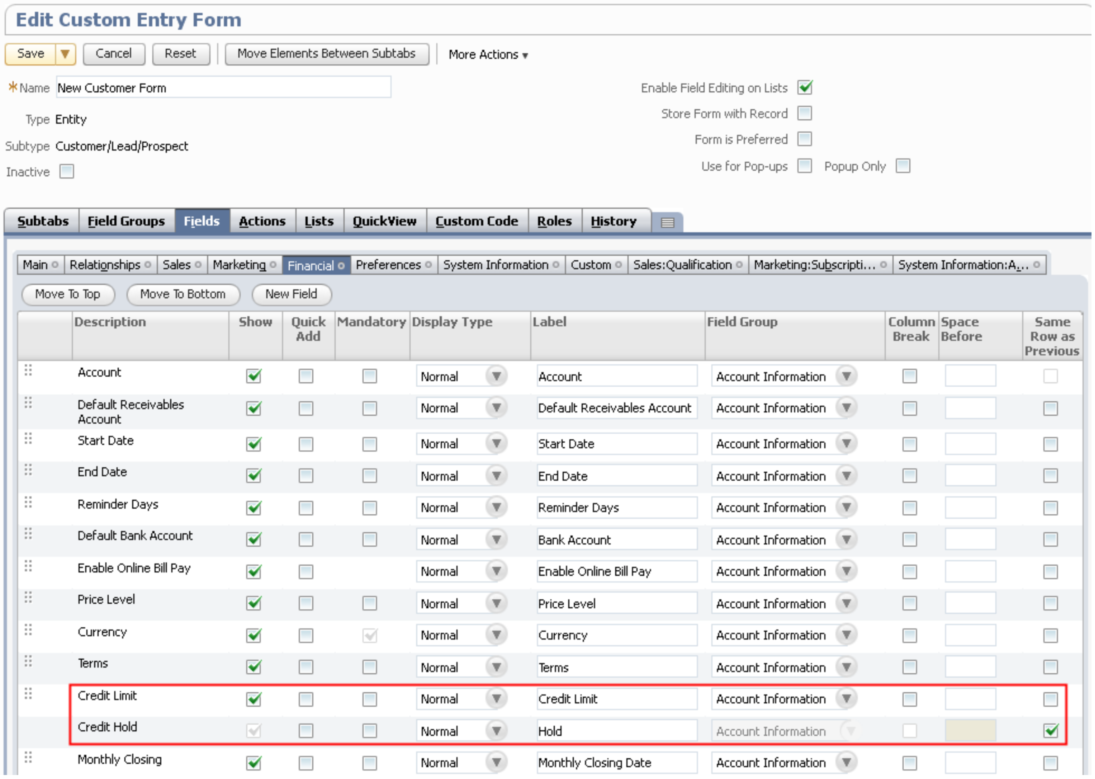
Important: As you are configuring your custom form, be sure to consider whether tax data is required for transactions for which you want to use the form. Only a form that includes required tax-related fields can be used for a transaction with tax consequences. Note that you cannot control naming for tax fields through form customization; you must go to Setup > Accounting > Set Up Taxes. See the help topic *Customizing Tax Fields on Transaction Forms*.

Associating Related Fields on Custom Forms

You can define an association among closely related fields on custom entry and transaction forms. Your definition of one or more fields as associated with a previous field on the form causes the associated fields to:

- Share the same Show/Hide setting as the previous field
- Belong to the same Field Group as the previous field
- Be displayed together on the form with the previous field

You define associated fields on the Fields subtab of a custom form, by checking the Same Row as Previous box. You also can remove a field's association with another field by clearing this box.



*Name New Customer Form

Type Entity

Subtype Customer/Lead/Prospect

Inactive □

Enable Field Editing on Lists ☑

Store Form with Record □

Form is Preferred □

Use for Pop-ups □ Popup Only □

Description	Show	Quick Add	Mandatory	Display Type	Label	Field Group	Column Break	Space Before	Same Row as Previous
Account	☑	□	□	Normal	Account	Account Information	□		□
Default Receivables Account	☑	□	□	Normal	Default Receivables Account	Account Information	□		□
Start Date	☑	□	□	Normal	Start Date	Account Information	□		□
End Date	☑	□	□	Normal	End Date	Account Information	□		□
Reminder Days	☑	□	□	Normal	Reminder Days	Account Information	□		□
Default Bank Account	☑	□	□	Normal	Bank Account	Account Information	□		□
Enable Online Bill Pay	☑	□		Normal	Enable Online Bill Pay	Account Information	□		□
Price Level	☑	□	□	Normal	Price Level	Account Information	□		□
Currency	☑	□	☑	Normal	Currency	Account Information	□		□
Terms	☑	□	□	Normal	Terms	Account Information	□		□
Credit Limit	☑	□	□	Normal	Credit Limit	Account Information	□		□
Credit Hold	☑	□	□	Normal	Hold	Account Information	□		☑
Monthly Closing	☑	□	□	Normal	Monthly Closing Date	Account Information	□		□

Be aware that setting custom fields to be Same Row as Previous on custom forms can cause field text to be displayed differently than you might expect based on the display size attributes set for custom fields, as described in Configuring Field Groups.

Important: As of Version 2012 Release 1, all fields previously defined as child fields are defined as associated fields of former parent fields. This change provides greater flexibility, as you can remove the association if desired.

Configuring Buttons and Actions

On the Actions subtab you can configure standard NetSuite buttons and custom buttons. Use the Standard Actions subtab to customize built-in, standard NetSuite buttons. (See Working with Standard Buttons for details.) Use the Custom Actions subtab to create or modify custom buttons you have associated with client SuiteScripts. (See Working with Custom Buttons for details.)

Also note that you can use point-and-click customization to translate the button labels for both standard and custom buttons. For details, see Translating Buttons to Other Languages.

Note: You can also use SuiteScript to hide and rename buttons. See the help topic *Button IDs* for a list of standard buttons that are supported in SuiteScript. Additionally, you can use the *Remove Button* action in SuiteFlow to conditionally hide buttons from specific users or when a record is in a certain state.

Important: Customizing the Save, Edit, Cancel, Back, and Reset buttons is not supported in SuiteScript or in point-and-click customization.

Once you have configured your buttons, you should configure the printing fields. See Configuring Printing Fields.

Working with Standard Buttons

Use the following steps to configure standard buttons through point-and-click customization.

To customize standard buttons:

1. On the Custom Form page, go to Actions > Standard Actions.
2. In the Label field, enter a UI label for the button.
3. In the Show column, select whether you want the button to appear or to be hidden on the form. Note that buttons are shown by default. To hide a button, uncheck the checkbox next to the button.
4. In the Display As column, choose Button to display the button as an inline button. Choose Menu if you want the button to appear as an action in the More Actions menu or as a menu item in a button group. See Understanding Button and Action Layout for more details.
5. Click Save.

Working with Custom Buttons

You can add custom buttons to forms to execute client SuiteScript. Note that for the Custom Actions subtab to appear, you must have the Client SuiteScript feature enabled in your account.

Also note that you must have already added the client SuiteScript file to the Custom Code subtab on the form. See Associating Custom Code (Client SuiteScript) Files With Custom Forms for details.

Important: Custom buttons do not appear when a record is in View mode. Custom buttons appear only when a record is in Edit mode.

To add a custom button to associate with client SuiteScript:

1. On the Custom Form page, go to Actions > Custom Actions.

2. In the Label field, enter the UI label for the button.

   Although you can enter up to 99 characters, it is recommended that you keep button names brief in order to save space.

3. In the Function field, enter the name of the function you want executed when the button is clicked. The function can exist in your client SuiteScript file or any library file you have attached to the Custom Code subtab.

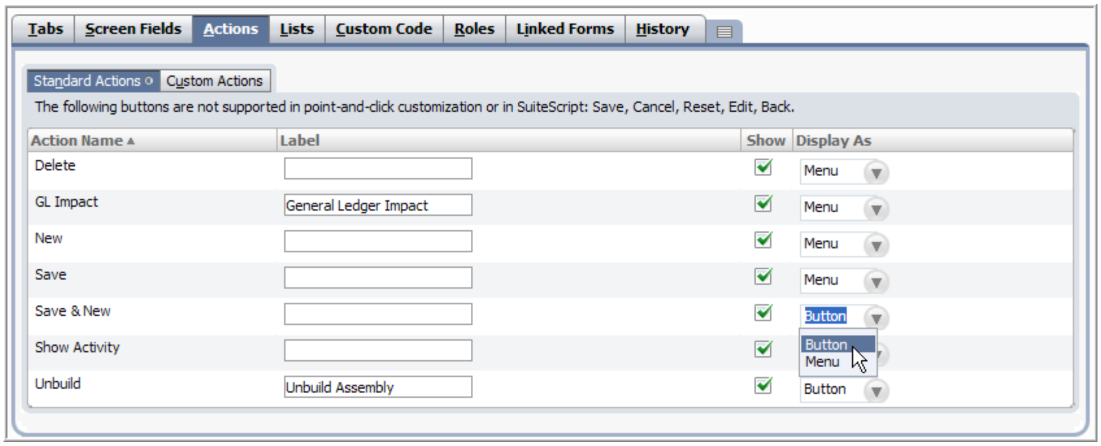
4. In the Display As column, choose Button to display the button as an inline button. Choose Menu if you want the button to appear as an action in the More Actions menu. Custom buttons, when changed to Menu, will always be grouped in the More Actions menu.

5. Click Save.

Understanding Button and Action Layout

On the Action subtab, use the **Display As** dropdown menu to customize the layout of a button or action. (See the first figure below.)

When you first click the Actions subtab, the display type for each action defaults to either Button or Menu. **Button** means that the action will display as a clickable button on the top and bottom of a form. **Menu** means that the action will display as an action in the More Actions menu, OR as a menu item in a button group (See the second figure below.)

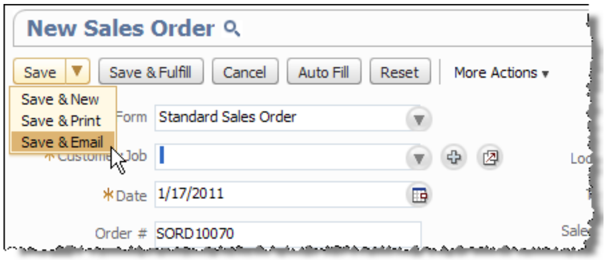


The following buttons are not supported in point-and-click customization or in SuiteScript: Save, Cancel, Reset, Edit, Back.

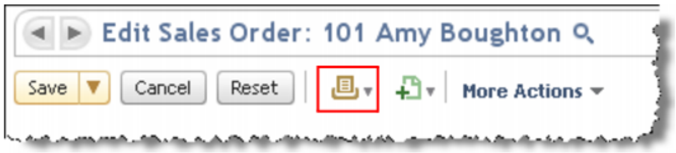
Action Name ▴	Label	Show	Display As
Delete		☑	Menu
GL Impact	General Ledger Impact	☑	Menu
New		☑	Menu
Save		☑	Menu
Save & New		☑	Button
Show Activity		☑	Button Menu
Unbuild	Unbuild Assembly	☑	Button

In the case of certain buttons such as the “Save &” buttons (Save & New and Save & Print, etc), the display type of **Menu** means that they will display as an item in a button group. (See the following figure.) **Menu** does not mean the button will display as an item in the More Actions menu.

Actions are grouped in sub-groups represented from left to right. Core actions such as Save appear as buttons and appear in the left-most group. Functional buttons (actions specific to a record) are in a middle group, and More Action actions are in the right-most group.



Also note, if you choose to display a custom button as **Button**, it will appear inline in front of the Print button (see figure). If you choose to display both standard and custom buttons as an action in the **More Actions** menu, the action will appear at the top of the More Actions list.



Translating Buttons to Other Languages

If you have the Multi-Language feature enabled, you can provide custom translations of standard and custom buttons on your forms. Note, however, you must have the Multi-Language feature enabled in your account to translate your buttons into multiple languages.

Note: To enable the Multi-Language feature, go to Setup > Company > Enabled Feature. On the Company subtab, select the Multi-Language checkbox, and then click Save.

To provide a custom translation for a button, be sure you have logged into NetSuite in the language you want to translate to, navigate to the custom form page, click the Actions subtab, and then perform the translation in place using the Label field.

Configuring Printing Fields

(Transaction Forms Only)

On the Printing Fields subtab, you can customize the way your form appears when printed. The Printing Fields subtabs displayed represent the different areas of a printed form.

Important: If the Advanced PDF/HTML Templates feature is enabled and you have set Printing Type to Advanced for a form, the Printing Fields subtab is not available for the custom form. This subtab is not needed when an advanced PDF/HTML template is used to define print formatting. There is no need for a separate listing of printing fields on the custom form record, because advanced templates explicitly include the fields to be printed. You can change the fields to be included in a custom advanced template by modifying the actual template in the template editor. See Customizing Advanced Templates in the Template Editor.

To configure how each section of your printed form displays:

1. For all subtabs:

   - In the Print/E-mail column, check the boxes next to fields you want to appear on printed and e-mailed forms. Clear the boxes for fields you don't want to appear on printed and e-mailed forms.
   - In the Label column, edit the labels of the fields as desired.

   Note: If you change the label for a field on the Printing Fields subtab, the label is also automatically applied to the field on the Screen Fields subtab.

2. For the Body and Columns subtabs:

   - In the Width column, enter the width for transaction column fields appearing on your printed and emailed forms.

     Note: To change the width of custom body fields, you must make the change on the Custom Form page. The width you set on the custom field does not affect printed transaction body fields.

   - Rearrange the fields as desired. Drag and drop each line item to the desired position.

3. Click Save.

Important: If the Advanced Taxes feature is enabled in your account, or if you are using NetSuite OneWorld, which requires Advanced Taxes, you cannot directly rename tax fields on a custom transaction form. To change names of tax fields that display on the custom form, you must rename them in the appropriate language(s) on the Field Naming subtab of the Set Up Taxes page, at *Setup > Accounting > Taxes > Set Up Taxes (Administrator)*. See the help topic *Customizing Tax Fields on Transaction Forms*.

Of the fields on the header subtab, some represent actual **values** that are inserted and some represent **labels** of field data that is inserted. For the fields that are values, the defaults from the company setup page are used unless overridden on the printing fields tab.

Following are the available fields listed by type:

Values:

- Company Name
- Company Phone
- Company URL
- Form Title
- Page Number

Labels:

- Business Number
- Acct. No.
- Date
- Doc. No.
- Bill To
- Ship To

Once you have configured the fields or screens, you should configure the printing fields. See Configuring Lists.

Configuring Lists

This applies to **entry forms** only.

On the Lists subtab, lists available on each subtab for the current form can be configured.

To configure lists for each available tab:

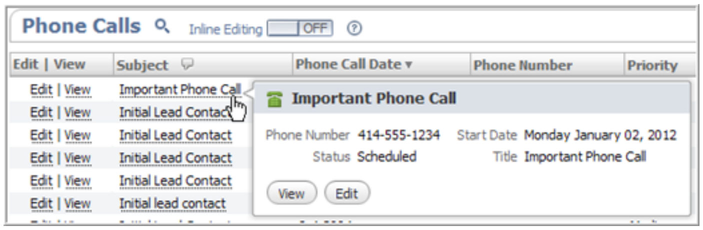
1. In the Show column, check the boxes for the lists you want to display on the form. Clear the boxes for the lists you don't want to display.
2. In the Label column, edit the headings for the lists as desired.
3. Rearrange the lists as desired.
    - Move each list to the desired subtab. Click Move Elements Between Subtabs > Lists. For details, refer to Moving Fields and Lists Between Subtabs.
    - Rearrange the order of the lists on each subtab. Click the desired line and drag and drop it to the desired position or click Move to Top / Move to Bottom.

4. Click Save.

Once you have configured the lists, you should configure the borders and placement. See Configuring Borders and Placement.

Configuring QuickViews

For both entry and transaction forms you will use the QuickViews tab to customize which fields appear in your QuickViews. The figure below shows an example of a QuickView for a Phone Call record.



Edit | View	Subject	Phone Call Date ▾	Phone Number	Priority
Edit | View	Important Phone Call			
Edit | View	Initial Lead Contact			
Edit | View	Initial Lead Contact			
Edit | View	Initial Lead Contact			
Edit | View	Initial Lead Contact			
Edit | View	Initial lead contact			

☎ Important Phone Call

Phone Number 414-555-1234 Start Date Monday January 02, 2012
Status Scheduled Title Important Phone Call

Body fields and custom fields are supported in QuickViews. However, you cannot add sublist fields. Also, the maximum number of fields you can add to a QuickView is 20.

Note that you can also configure QuickViews on forms associated with custom records. See Configuring QuickViews for Custom Records for information.

Important: Confisguring QuickViews on entry and transaction forms is easy if you have deployed any upgraded forms in your account. See Configuring QuickViews for Upgraded Forms for information.

Important: If you have not upgraded and deployed the forms in your account, you can still indirectly affect which fields appear in a QuickView, however, the process is not as straightforward. The benefits of QuickView customization are meant to come with upgraded forms. If you have not upgraded the forms in your account, yet you would like to update your QuickViews, see Configuring QuickViews for Non-Upgraded Forms.

Configuring QuickViews for Upgraded Forms

QuickView customization is easy if you have upgraded the custom forms in your account to include field groups and all other look-and-feel enhancements introduced in NetSuite Version 2010 Release 2. Once a form has been upgraded, a **QuickView** configuration tab will appear on the form customization page. You will use this tab to add, remove, and rearrange the fields in a QuickView.

To configure QuickViews for upgraded entry and transaction forms:

1. Go to Setup > Customization > Entry Form [or Transaction Form].

2. Click the Customize link next to the Standard version of the form.

   If you already have an existing custom form, click Edit next to this form.

   Important: If you are not sure if the form you want to work with has been upgraded, see How do I tell if the custom forms in my account are upgraded?

3. On the Edit Custom Entry [or Transaction] Form page that appears, click the QuickViews tab.

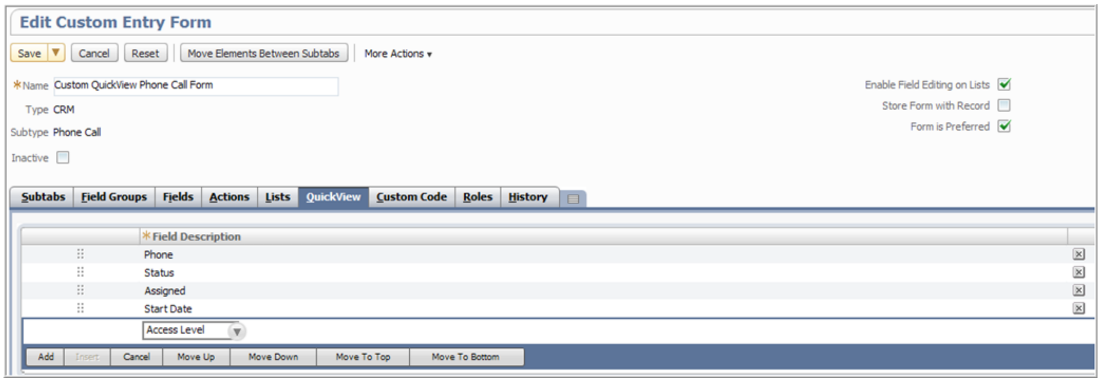
   If the page does not have a QuickView tab, this means that the form you want to work with has not yet been upgraded/deployed.

4. In the Field Description column, add the fields you want to appear in the QuickView for records that use this form. After adding each new field, be sure to click the Add button (see figure below).

   If you want to remove a field, click the X icon that appears to the far right side of the field.

5. Select the Form is Preferred checkbox.

6. Click Save when you are finished configuring your QuickView fields.



Edit Custom Entry Form

*Name Custom QuickView Phone Call Form

Type CRM

Subtype Phone Call

Inactive □

Enable Field Editing on Lists ☑

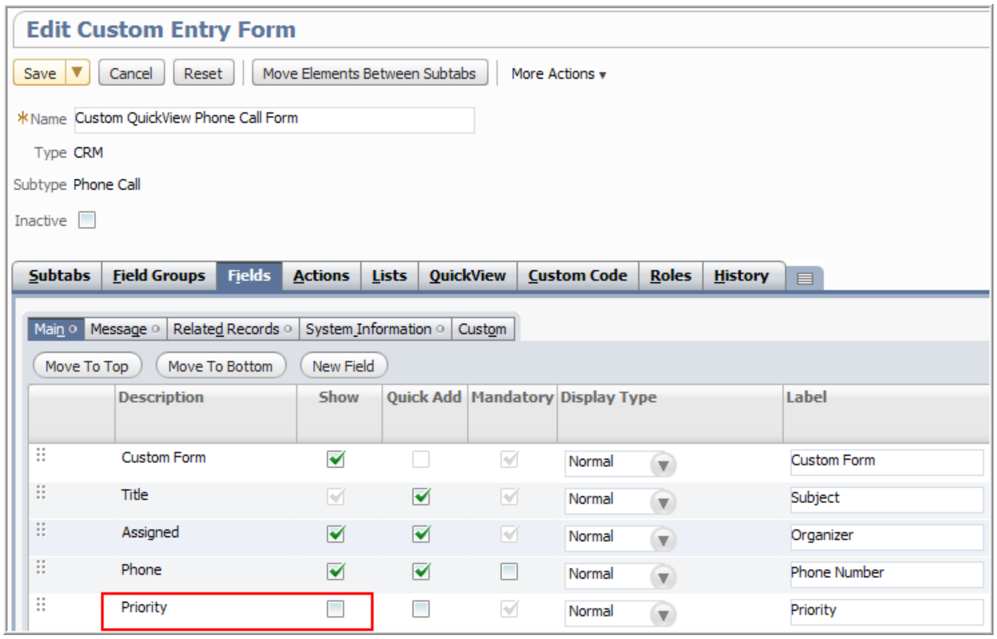
Store Form with Record □

Form is Preferred ☑

*Field Description
Phone
Status
Assigned
Start Date
Access Level

Please be aware of these additional guidelines when customizing your QuickViews:

- If you customize a form's QuickView and select Store Form with Record, these customizations will supersede any customization made to a form set to Form is Preferred.
- During form customization, if you make field updates on the Fields tab, you must save the form customization page before those updates will appear on the QuickView tab.
- On the Fields tab, if a field is not set to Show (see figure below), the field will not appear in the QuickView, even if you add the field on the QuickView tab. For example, in this figure the Priority field is not set to Show. This means that even if Priority is added on the QuickView tab, the Priority field will not appear in the QuickView of a Phone Call record.



Edit Custom Entry Form

Save ▼ | Cancel | Reset | Move Elements Between Subtabs | More Actions ▾

*Name	Custom QuickView Phone Call Form
Type	CRM
Subtype	Phone Call
Inactive	□

Subtabs | Field Groups | **Fields** | Actions | Lists | QuickView | Custom Code | Roles | History

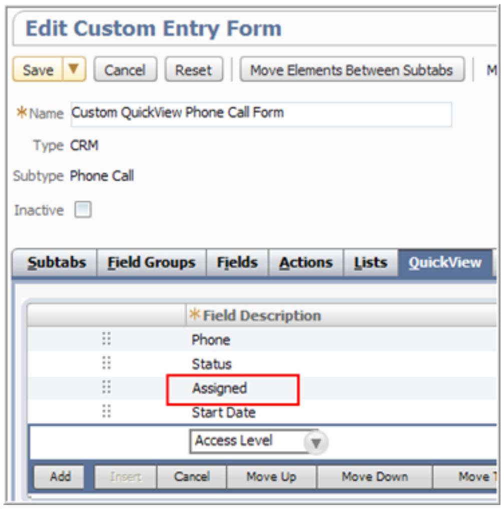
Main ○ | Message ○ | Related Records ○ | System Information ○ | Custom

Move To Top | Move To Bottom | New Field

Description	Show	Quick Add	Mandatory	Display Type	Label
Custom Form	☑	□	☑	Normal	Custom Form
Title	☑	☑	☑	Normal	Subject
Assigned	☑	☑	☑	Normal	Organizer
Phone	☑	☑	□	Normal	Phone Number
Priority	□	□	☑	Normal	Priority

- The fields appearing on the QuickView tab match the fields listed in the **Description** column on the Fields tab. However, in the QuickView it is a field's custom label that appears.

  The figures below show the Assigned field on the QuickView tab. This field name matches what is listed in the **Description** column on the Fields tab. However, when the QuickView appears for the Phone Call record, it is Organizer that shows. This is the custom label given to the Assigned field.



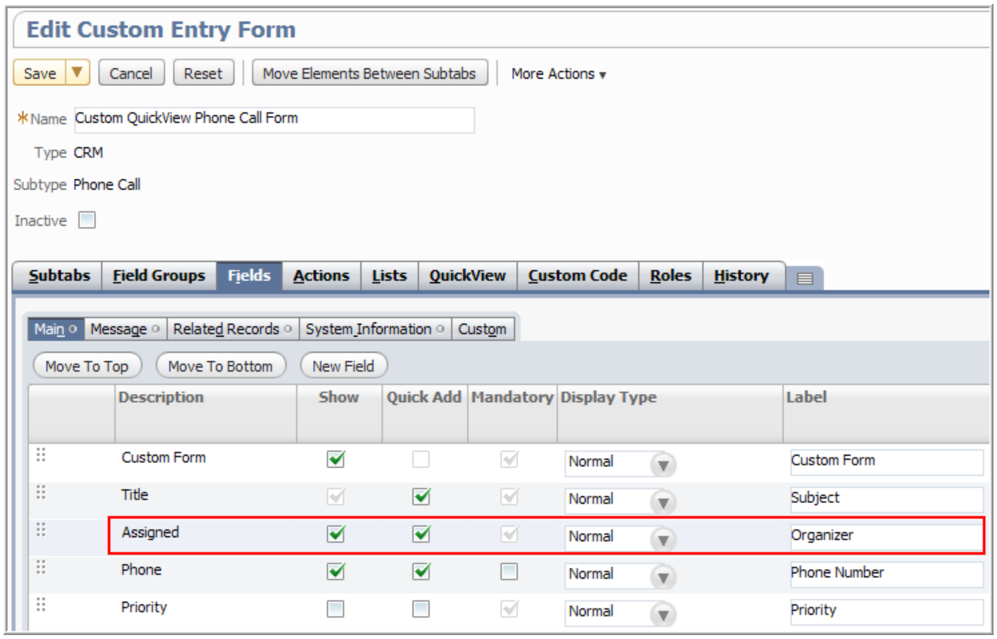
Edit Custom Entry Form

Save ▼ | Cancel | Reset | Move Elements Between Subtabs | M

*Name	Custom QuickView Phone Call Form
Type	CRM
Subtype	Phone Call
Inactive	□

Subtabs | Field Groups | Fields | Actions | Lists | **QuickView**

*Field Description
Phone
Status
Assigned
Start Date
Access Level



Edit Custom Entry Form

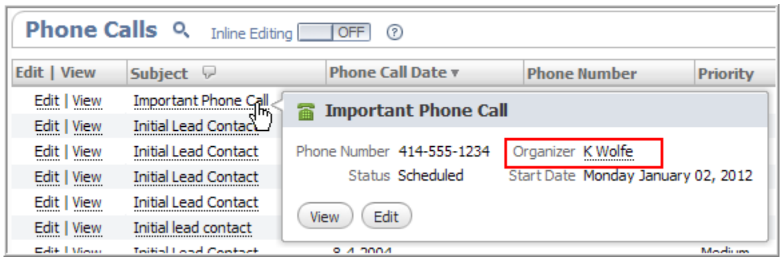
*Name Custom QuickView Phone Call Form

Type CRM

Subtype Phone Call

Inactive □

Description	Show	Quick Add	Mandatory	Display Type	Label
Custom Form	☑	□	☑	Normal	Custom Form
Title	☑	☑	☑	Normal	Subject
Assigned	☑	☑	☑	Normal	Organizer
Phone	☑	☑	□	Normal	Phone Number
Priority	□	□	☑	Normal	Priority



Phone Calls

Edit | View	Subject	Phone Call Date ▾	Phone Number	Priority
Edit | View	Important Phone Call			
Edit | View	Initial Lead Contac			
Edit | View	Initial Lead Contact			
Edit | View	Initial Lead Contact			
Edit | View	Initial Lead Contact			
Edit | View	Initial lead contact			
Edit | View	Initial Lead Contact	8.4.2004		Medium

☎ Important Phone Call

Phone Number 414-555-1234

Organizer K Wolfe

Status Scheduled

Start Date Monday January 02, 2012

- Fields that appear on a record ONLY when the record is in Edit mode do not appear in QuickViews. Fields that appear on a record while it is in View mode do appear in QuickViews.

  One thing to note is that the QuickView dropdown (on the QuickView configuration tab) includes **all** the fields that are on a record. The dropdown includes fields that appear on a record while it is in View mode and Edit mode. If you add a field to your customized QuickView that appears only when the record is in Edit mode, the field will **NOT** appear in the QuickView when you hover over links to this record. Although there are not many fields in NetSuite that appear on records only when they are in Edit mode, this is one thing to be aware of. If you add a field to a QuickView, yet the field does not appear when the QuickView displays, it may be that you have added a field that is visible only when the record is in Edit mode.

Configuring QuickViews for Non-Upgraded Forms

If you have not upgraded the custom forms in your account to include the look-and-feel changes introduced in Version 2010 Release 2, you cannot directly edit or customize the default QuickView fields assigned by NetSuite. Non-upgraded forms do not include the QuickViews configuration tab on entry and transaction form customization pages.

You can, however, indirectly affect which fields appear in a QuickView by creating an equivalent upgraded form. You can then use the QuickViews tab on the newly upgraded form to make QuickView customizations. You must then set this form to Form is Preferred. By doing do, even the records in your account that use the non-upgraded version of the form will show the QuickView customizations of the upgraded form.

Also note the following when attempting to customize QuickViews for non-upgraded forms:

- A non-upgraded stored form will show the customizations of an upgraded preferred form or standard form if there is no preferred form.
- Even if you click Store Form with Record on a non-upgraded form, the QuickViews that appear for records using this form will show the QuickView customizations (if any) on the upgraded form.
- If a standard form is marked as preferred, then all Quickviews for both non-upgraded and upgraded forms will show the NetSuite default QuickView fields for that record type.

Configuring QuickViews for Custom Records

Unlike built-in standard records, there is no default QuickView for custom records. Because the fields on every custom record are unique, there is no way for the system to set default QuickView fields for all custom records. Therefore, you must create custom record QuickViews yourself by following the steps below.

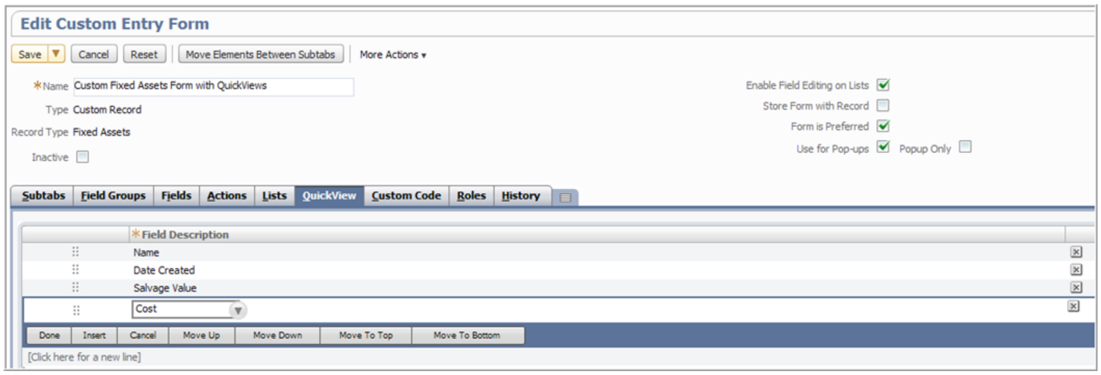
To configure QuickViews for custom records:

1. Go to Setup > Customization > Record Types.
2. Select the custom record type you want to display QuickViews.
3. On the Edit Custom Record Type page, click the Forms tab.
4. Click the Customize link next to the Standard version of the custom record form.

   If you already have an existing custom form, click Edit next to this form.
5. On the page that appears, click the QuickViews tab.
6. In the Field Description column, add the fields you want to appear in the QuickView for this custom record type. After adding each new field, be sure to click the Add button.

If you want to remove a field, click the X icon that appears to the far right side of the field.

7. Click Save when you are finished configuring your QuickView fields.

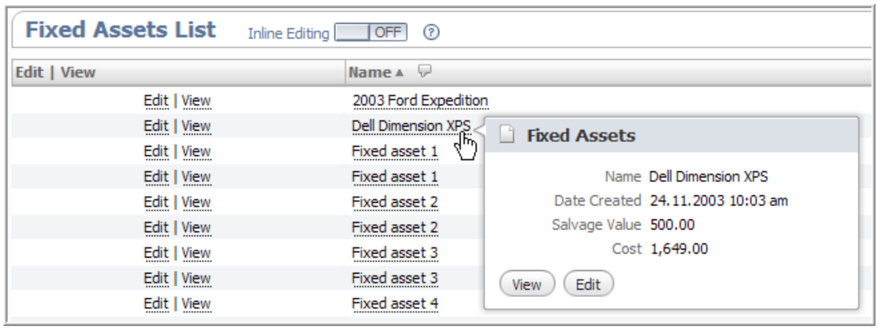


*Name Custom Fixed Assets Form with QuickViews
Type Custom Record
Record Type Fixed Assets
Inactive □

Enable Field Editing on Lists ☑
Store Form with Record □
Form is Preferred ☑
Use for Pop-ups ☑ Popup Only □

*Field Description
Name
Date Created
Salvage Value
Cost

If you go to the List view of this custom record type, you can hover over a record to see the newly configured QuickView. The figure below shows the QuickView for the Fixed Asset custom record.



Edit | View	Name ▲
Edit | View	2003 Ford Expedition
Edit | View	Dell Dimension XPS
Edit | View	Fixed asset 1
Edit | View	Fixed asset 1
Edit | View	Fixed asset 2
Edit | View	Fixed asset 2
Edit | View	Fixed asset 3
Edit | View	Fixed asset 3
Edit | View	Fixed asset 4

Fixed Assets

Name Dell Dimension XPS
Date Created 24.11.2003 10:03 am
Salvage Value 500.00
Cost 1,649.00

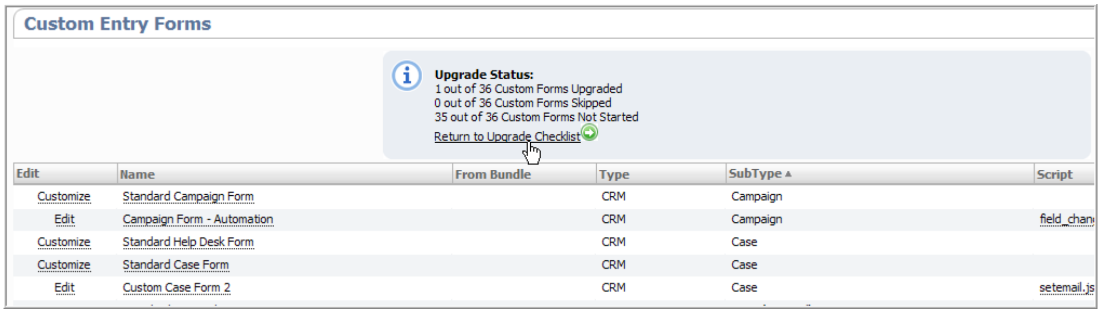
How do I tell if the custom forms in my account are upgraded?

Records that use upgraded forms will have all fields organized into field groups. (If you are not sure what field groups look like, see Configuring Field Groups.)

If you have noticed any records in your account with a field group layout, then at least one form in your account (the form associated with that record type) has been upgraded.

If you are still not sure if any of the forms in your account have been upgraded, Go to Setup > Customization > Entry Forms [*or Transaction Forms*]. If any of the forms in your account have been upgraded, you will see the **Upgrade Status** message at the top of your entry or transaction form page (see figure below). If no forms have been upgraded, the message title will say **Information** and provide details on the form layout enhancements introduced in NetSuite Version 2010 Release 2.

If forms in your account *have* been upgraded and you want to see specifically which forms, click the Return to Upgrade Checklist link.



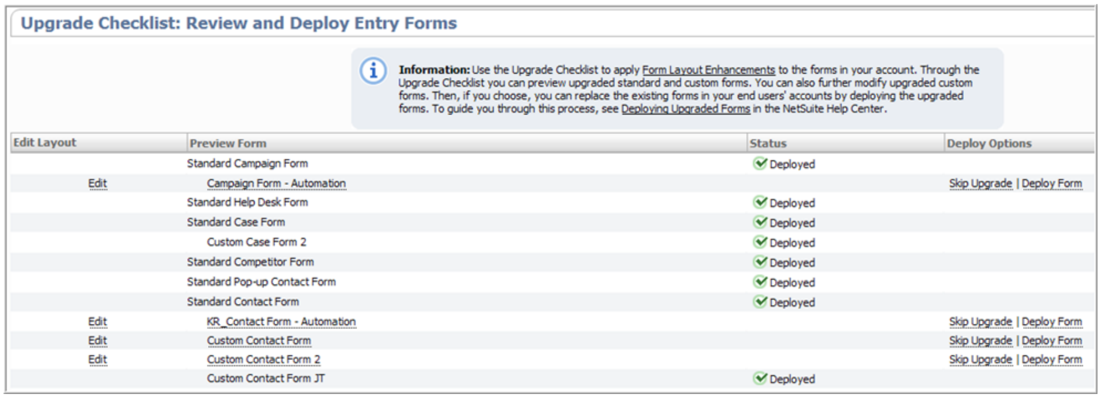
Custom Entry Forms

Upgrade Status:
1 out of 36 Custom Forms Upgraded
0 out of 36 Custom Forms Skipped
35 out of 36 Custom Forms Not Started
Return to Upgrade Checklist

Edit	Name	From Bundle	Type	SubType ▲	Script
Customize	Standard Campaign Form		CRM	Campaign	
Edit	Campaign Form - Automation		CRM	Campaign	field_chan
Customize	Standard Help Desk Form		CRM	Case	
Customize	Standard Case Form		CRM	Case	
Edit	Custom Case Form 2		CRM	Case	setemail.js

The figure below shows specifically which forms have been upgraded and that their upgrades have been **Deployed** into NetSuite. Notice that a number of other forms have not been upgraded/deployed. These are the forms with the **Skip Upgrade | Deploy Form** links.

To customize a QuickView for any form, the form must be **Deployed**.



Upgrade Checklist: Review and Deploy Entry Forms

Information: Use the Upgrade Checklist to apply Form Layout Enhancements to the forms in your account. Through the Upgrade Checklist you can preview upgraded standard and custom forms. You can also further modify upgraded custom forms. Then, if you choose, you can replace the existing forms in your end users' accounts by deploying the upgraded forms. To guide you through this process, see Deploying Upgraded Forms in the NetSuite Help Center.

Edit Layout	Preview Form	Status	Deploy Options
	Standard Campaign Form	✓ Deployed	
Edit	Campaign Form - Automation		Skip Upgrade | Deploy Form
	Standard Help Desk Form	✓ Deployed	
	Standard Case Form	✓ Deployed	
	Custom Case Form 2	✓ Deployed	
	Standard Competitor Form	✓ Deployed	
	Standard Pop-up Contact Form	✓ Deployed	
	Standard Contact Form	✓ Deployed	
Edit	KR_Contact Form - Automation		Skip Upgrade | Deploy Form
Edit	Custom Contact Form		Skip Upgrade | Deploy Form
Edit	Custom Contact Form 2		Skip Upgrade | Deploy Form
	Custom Contact Form JT	✓ Deployed	

Moving Fields and Lists Between Subtabs

You can move fields and lists (the Contacts list on entry forms, for example) between subtabs on entry forms.

After you have made changes on the Custom Form page, click Save & Move Elements to move fields and lists on the form to other subtabs.

To move fields and lists between subtabs:

1. Click Save & Move Elements.

   Your form is saved, and the Move Form Elements page opens.

On the Fields subtab are lists of fields that appear on each subtab on the form you are customizing that include fields.

Subtabs with lists (for example, the Contacts subtab on entry forms or the Items subtab on transactions) are shown on the Lists subtab of the Move Form Elements page.

2. For each field on each subtab, select the subtab you want that field to show on. Note that you can also select multiple fields and move them to a subtab to set the dropdown on those fields. At any time you can click Save and Move More, which saves your field changes, and reloads the page with the fields now moved to the specified subtab.
3. Click the Lists subtab.
4. Select the subtab you want each list to appear on.

   It is possible to move a field or list to a subtab that is NOT displayed since it does not contain items of that type (fields or lists respectively). When a field or list is moved to a previously empty, undisplayed subtab, you can NOT move the field back until the Move Form Items page is saved. If you need to move a field again, you can return to the Move Form Items page and move it after you have saved.
5. Click Save to return to the Custom Form page.

Fields and lists are now shown on the subtabs you selected. If a field or list has been moved to a subtab that is not shown on the form, those items are no longer available on the form.

Associating Custom Code (Client SuiteScript) Files With Custom Forms

In the Custom Code subtab, define any existing client SuiteScript files to use with this form. When the script's executing function is called, the actions defined within the script (as well as any built-in NetSuite actions for that form type) are performed. For example, if your form calls a Page Init function, the actions defined by that script, as well as any predefined NetSuite actions, are performed prior to the page loading.

Important: Before associating client SuiteScript files with a form, you must first create the script. For information on creating client SuiteScripts, see the help topic *Client Scripts* in the NetSuite Help Center.

Note that, if you choose, you can add a custom button to the form to trigger the client script. See Working with Custom Buttons for steps on adding custom buttons to forms.

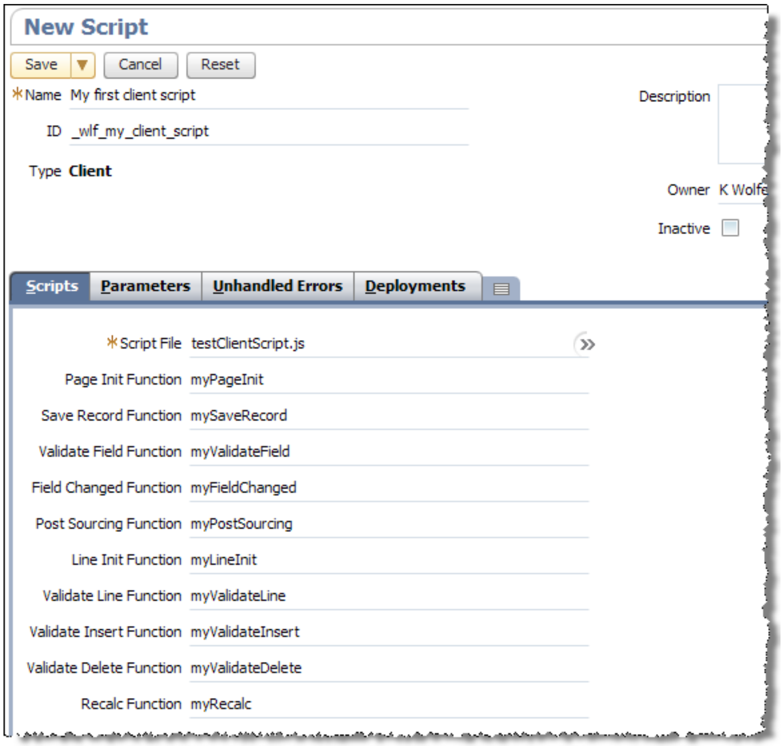
To associate client SuiteScript files with a form:

1. In the Script File field, select the client SuiteScript file that contains the desired scripts for this form.

Note that you must upload your file to the file cabinet before you can select it.

2. In one or more of the client event type fields, enter the name(s) of the function(s) you want to execute. When entering function names, do not include parentheses or arguments. For example, type **sampleFunction** for a function that appears as sampleFunction(param1, param2) in your SuiteScript file.

    - In the Page Init Function field, enter the name of the executing function to be called when this form is first loaded.
    - In the Save Record Function field, enter the name of the executing function to be called when this record is saved.
    - In the Validate Field Function field, enter the name of the executing function to be called when a field on this entry form is changed.
    - In the Field Changed Function field, enter the name of the executing function to be called when a change made to a field is accepted.
    - In the Post Sourcing Function field, enter the name of the function that will run on Post Sourcing events. These events occur following a field change once all the field's child field values are sourced from the server. This enables fieldChange style functionality to occur once all dependent field values have been set.
    - In the Line Init Function field, enter the name of the function that will run on Line Init events. These events occur when an existing line is selected.
    - In the Validate Line Function field, enter the name of the function that will run on Validate Line events. These events occurs prior to a line being added to a sublist (inlineeditor or editor sublists only). It can be thought of as the saveRecord equivalent for sublist line items.
    - In the Validate Insert Function field, enter the name of the function that will run on Validate Insert events. These events occur when you insert a line into an edit sublist. The UI equivalent of this event is when a user selects an existing line in a sublist and then clicks the Insert button. Note that returning false on a validateInsert blocks the insert.
    - In the Validate Delete Function field, enter the name of the function that will run on Validate Delete events. These events occur when you try to remove an existing line from an edit sublist. Returning false blocks the removal.
    - In the Recalc Function field, enter the script name to be called from the attached script file. This script is called when a line item is added. For example, after entering the information you add an item to a transaction.



*Name	My first client script
ID	_wlf_my_client_script
Type	Client
Description	
Owner	K Wolfe
Inactive	□

*Script File	testClientScript.js
Page Init Function	myPageInit
Save Record Function	mySaveRecord
Validate Field Function	myValidateField
Field Changed Function	myFieldChanged
Post Sourcing Function	myPostSourcing
Line Init Function	myLineInit
Validate Line Function	myValidateLine
Validate Insert Function	myValidateInsert
Validate Delete Function	myValidateDelete
Recalc Function	myRecalc

3. If your client SuiteScript file references any functions in a library file, add the library file(s) in the Library Script File dropdown. (You must click the Add button to add your file.)

4. Click Save.

Defining Preferred Entry and Transaction Forms

On the Roles subtab, enable the Preferred box for roles that should have this form set as preferred. Preferred forms are selected by default in the Custom Form field when a transaction or record is created.

Note the following about marking an entry or transaction form Preferred for the Customer Center role:

- External forms, meaning forms with names appended with (External), can be marked Preferred for Customer Center roles, but not for other roles.
- Forms that are not external cannot be marked Preferred for Customer Center roles.
- When a non-online order form is marked Preferred for Customer Center, it is saved as the form for the order. However, an online form is not saved as the form for an order, even if it is preferred; instead the preferred non-online order form is used.

Note: You can also define preferred forms from the Manage Roles page. **Restrictions** to a form must be defined from the Manage Roles page. For more information, see the help topic *Setting Default Forms for Roles*.

Linking Transaction Forms

On the Linked Forms subtab, you can control which transaction form is used when you convert one transaction into another. You can create a chain of transaction forms that mirror your business workflow.

For example, a company has three custom sales order forms that are each used for a certain set of items they sell. When one of these sales orders is used to create a picking ticket, the specific picking ticket form created for each type of sales order is used. The employee creating the picking ticket does not have to search the custom form dropdown to find the proper picking ticket form.

To set up this form workflow, the company administrator edits the custom sales order form, and selects the picking ticket form on the Linked Forms subtab.

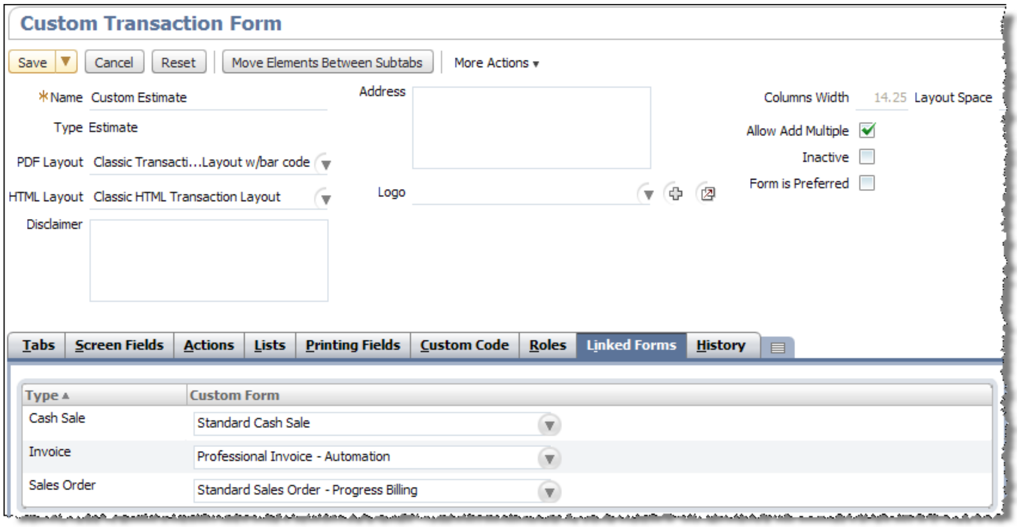
When you convert a transaction you created with a custom transcation form, the custom form set on the Linked Forms subtab is selected by default. In the example above, When one of the sales orders is used to print a picking ticket, the custom picking ticket form selected on the sales order form is used by default.

To set up linked forms:

1. Open the list of forms that contains the form you want to create a linkage from.
2. Click Customize or Edit next to the form you want to link from.
3. Enter a name for your custom form.
4. Click the Linked Forms subtab.

   A list of the transactions that you can convert to from the type of transaction form you are customizing is shown. For example, if you are customizing an estimate form, you can set linkages for cash sales, invoices, and sales orders.
5. In the Custom Form column, select the standard or custom form for each transaction.

   If you want to link to the preferred form, do not select a linked form for that transaction type.



*Name: Custom Estimate
Type: Estimate
PDF Layout: Classic Transacti...Layout w/bar code
HTML Layout: Classic HTML Transaction Layout
Disclaimer:
Address:
Logo:
Columns Width: 14.25 Layout Space
Allow Add Multiple ☑
Inactive □
Form is Preferred □

Type	Custom Form
Cash Sale	Standard Cash Sale
Invoice	Professional Invoice - Automation
Sales Order	Standard Sales Order - Progress Billing

6. Click Save.

When the transaction is converted, the correct transaction form is used automatically.

Note: In the event that the person is assigned a role that is restricted to use only specific transaction forms, the forms set for that role take precedence over the forms you have set on the Linked Forms subtab.

Custom Sublists

You can add a custom sublist to any transaction or entry form, including forms for custom record types. Custom sublists present information related to the record you are viewing, based on results from a selected saved search of the form record type or a related record type. The following are examples:

- You could attach a customers sublist on customer records that lists other customers with the same sales rep, based on a customers search with a filter of Sales Rep.
- You could attach an employees sublist on employee records listing other employees with the same supervisor, based on an employees search with a filter of Supervisor.
- You could attach a transactions sublist on customer records listing transactions for that customer, based on a transactions search with a filter of customer name.

The creation of custom sublists is controlled by the Custom Sublists permission. You can assign this permission to a custom role on the Setup subtab of the role record.

For details about creating saved searches to be used as custom sublists and applying custom sublists to records, see:

- Creating Saved Searches for Custom Sublists
- Applying Custom Sublists to Standard Records
- Applying Custom Sublists to Custom Record Types

Creating Saved Searches for Custom Sublists

To be used for a custom sublist, a saved search must have at least one field defined as an available filter. Custom sublist results are joined to the form record based on this field's values. The custom sublist results that are displayed on each form are those that have the same value for the available filter field as the record currently displayed on the form does.

For example, in a saved search for a transactions sublist on a customer record, you could set an available filter of Name (the customer name) so that all transactions for the currently displayed customer show in the sublist.

For details about available filters, see the help topic *Selecting Available Filters for Saved Searches*

Be aware of the following when you choose a field to be the available filter for a custom sublist search:

- If the saved search has more than one available filter, only the first one listed on the Available Filters subtab is used to filter the custom sublist.
- The available filter field must be of the List/Record type, meaning it is displayed in the user interface as a dropdown (select). Constant value fields and numeric fields will not work.

  Note that the searches you can select for custom sublists do not even include searches where the first available filter is check box or a date, because this type of field will not work.
- If the sublist represents a different record than the form record, the available filter field should be a logical choice to produce joins with the form record. Many fields that are listed on a search's Available Filters subtab are not sensible choices for this type of join.

  For example, in a saved search for a transactions sublist on customer record, it would not work to set an available filter of Type. This choice would most likely result in no joins with the customer record, and a blank sublist would display.

  To be a good choice as an available filter, a field should identify a record type.

  - This field can the same as the form record, for example, the Name field for a transactions sublist on a customer record that displays all of that customer's transactions.
  - Or, this field can be a field on the form record that is a list of records, for example, the Location field, for a transactions sublist on a customer record that displays the customer's transactions for a particular location only.

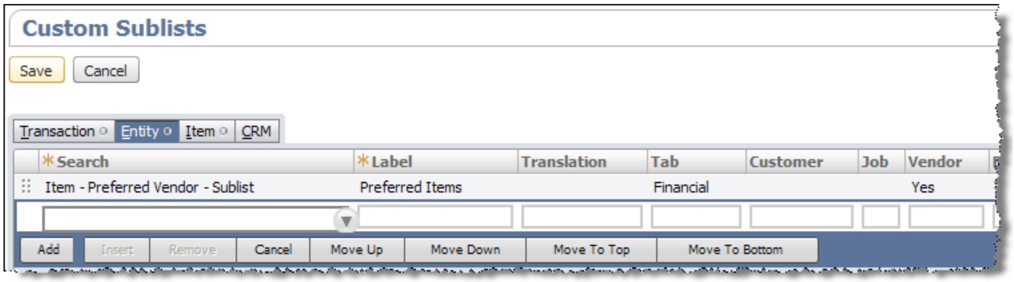
Applying Custom Sublists to Standard Records

You can add a custom sublist to any entity or transaction record, including custom transaction and entry forms.

For example, you might want to show a list of customers on the sales order form that are assigned to the sales rep on the order.

To add a custom sublist to a standard record:

1. First, create a saved search for the information you want to show. The results of this search should include the information you want to show on your sublist.
2. Go to Setup > Customization > Lists, Records, & Fields > Sublists.
3. Click the subtab that corresponds with the kind of record to which you want to add the sublist.



*Search	*Label	Translation	Tab	Customer	Job	Vendor
Item - Preferred Vendor - Sublist	Preferred Items		Financial			Yes

4. In the Search column, select the saved search that returns the results you want to appear on the record.
5. Enter a label for this sublist.
6. In the Tab column, select the subtab you want this sublist to appear on.
7. Check the box for each record you want this sublist to appear on.

   This sublist shows on the standard and custom forms of the types you select.
8. In the Field column, select the field you are filtering the sublist by.
9. Click Add.
10. Repeat these steps for each sublist you want to attach to a record.
11. Click Save.

Now, these search results show on the record you selected.

Applying Custom Sublists to Custom Record Types

You can also add custom sublists to custom record types.

First, you create a saved search the results of which you want to show on your custom records. Then, you add the search as a custom sublist to your custom record type.

To add a sublist to a custom record type:

1. First, go to Setup > Customization > Lists, Records, & Fields > Record Types, and click Search next to the record type you want display in the custom sublist.
2. The results of this search should include the information you want to show on your sublist.
3. Click Create Saved Search.

   For information about creating saved searches, see the help topic *Defining a Saved Search*.
4. Check the Advanced Search box on the search page if you are not already in advanced search mode.
5. Enter a name for the search.
6. Set criteria for the search.
7. Add at least one available filter to the search.

   You can only attach a saved search as a sublist if the search has at least one available filter. The sublist that is shown is filtered by the first filter listed on the saved search. Be sure that the search and the filter both apply to the record you are attaching the sublist to.

   For information about setting available filters, see the help topic *Selecting Available Filters for Saved Searches*.
8. Click Save.
9. Go to Setup > Customization > Lists, Records, & Fields > Record Types, and click the name of the record type you want to add the sublist to.
10. Click the Sublists subtab.
11. In the Search column, select the saved search.
12. In the Label column, enter a label for the subtab the search results will display under.
13. (When the Multi-Language feature is enabled) In the Translation column, enter one or more translations of the label.
14. In the Tab column, choose the custom subtab this sublist will display under.
15. In the Field column, select the field you are filtering this sublist by.
16. Click Add.
17. Click Save.

Creating Custom Subtabs

Custom subtabs are used to organize custom fields on your transaction, entity, CRM, and item records. You should first create custom subtabs and then assign any custom fields to the custom subtabs. For each custom subtab, you can define an existing subtab as the parent subtab. This parent-child definition allows you to include an additional layer of information for your subtab categories.

Important: Be aware that for custom subtabs to appear on a record, you must assign a field to the subtab.

To create a custom subtab:

1. Go to Setup > Customization > Subtabs.
2. Click the tab for the type of record for which you want to create a new subtab.

   The following options are available:

   - **Transaction** — subtabs for cash refund, cash sale, credit memo, estimate, invoice, opportunity, purchase order, return authorization, sales order, and other transaction records.
   - **Entity** — subtabs for customer, project, vendor, employee, other name, contact, partner, and group records.
   - **Item** — subtabs for inventory, non-inventory, group, other charge, assembly/bill of materials, kit/package, service item and other item records.
   - **CRM** — subtabs for task, phone call, event, case, campaign, solution, and other CRM records .
3. Enter the name for the subtab in the Title field.
4. If desired, enter translations for this name in the Translation field.

   - Before you can add translations, you need to select translation languages at Setup > Company > General Preferences, on the Languages subtab. This subtab lists both system-supported languages that can be used for the NetSuite user interface (and are available at Home > Set Preferences), and additional languages that can used for Web site translations only (and are not available at Home > Set Preferences). You should only enter translations for system-supported languages, because these are the only languages that can be displayed in the user interface. For details, see the help topic *Setting Up Multiple Languages*.
   - The maximum length for a custom subtab's translation is 50 characters.
5. If desired, designate this subtab as a child of an existing subtab in the Parent field, by selecting an existing subtab from the dropdown list.

- This list consists of any standard subtabs associated to the selected record type as well as any custom subtabs that you have defined for that record type.
- Once defined as a child subtab, a custom subtab is not available to be selected as a parent for another custom subtab. In other words, you CANNOT create a child-grandchild relationship.

6. Click Add.
7. Repeat these steps for each subtab you want to create.
8. Click Save.

Rearrange your custom subtabs by using the Move Up, Move Down, Move To Top, and Move To Bottom buttons. Insert new subtabs using the Insert button.

Now, when you create a custom field, you can select your new subtab in the Subtab field. After assigning the custom field to the subtab, the subtab will then displays on the record.

Advanced PDF/HTML Templates (Beta)

The Advanced PDF/HTML Templates (Beta) feature supports an alternate model for customizing printed and emailed records. This model supports more customization capabilities than transaction form layouts, also known as legacy layouts. You can use advanced templates to produce either PDF or HTML output, depending upon the settings of your print and email preferences.

The beta version of this feature is intended to introduce a new model for print customization, allowing you to try it out on a selected set of records. This feature currently supports cash refund, cash sale, credit memo, customer deposit, invoice, packing slip, payment voucher, picking ticket, purchase order, quote (estimate), return authorization, and sales order templates. You can set custom forms for these supported types to use standard advanced templates or your customized advanced templates. The NetSuite user interface provides a template editor where you can customize templates.

Custom forms that use advanced templates are printed in the locale of the current user, set in the PDF Language field at Home > Set Preferences. You also can print these forms in the locale of the customer associated with a transaction, for transaction types that support this option at Transactions > Management > Print Checks and Forms. (Note that there is a current known limitation with East Asian languages that use the CJK font. See Current Limitations for the Advanced PDF/HTML Templates Feature.)

Each printed form that uses an advanced template automatically includes a company logo, based on the image file defined as the Company Logo (Forms) field at Setup > Company > Company Information. For instructions for defining this image file, see the help topic *Setting Up Company Information*.

Note: The Advanced PDF/HTML Templates feature was formerly known as the Advanced Printing feature.

Review the following to get started with the Advanced PDF/HTML Templates (Beta) feature:

- Enabling the Advanced PDF/HTML Templates (Beta) Feature
- Reviewing Available Advanced Templates
- Setting Custom Forms to Use Advanced Templates
- Customizing Advanced Templates in the Template Editor

Because the Advanced PDF/HTML Templates feature is a beta version, it includes a number of limitations. For a list, see Current Limitations for the Advanced PDF/HTML Templates Feature.

Important: You can choose whether or not to enable the Advanced PDF/HTML Templates feature. Even when this feature is enabled, no changes are made to legacy layouts that are currently in use. Transaction Form PDF Layouts and Transaction Form HTML Layouts continue to be supported, and you can continue to associate them with custom forms as desired. The Advanced PDF/HTML Templates feature provides an additional option to associate advanced templates with custom forms.

Using Advanced Template Formatting Programmatically

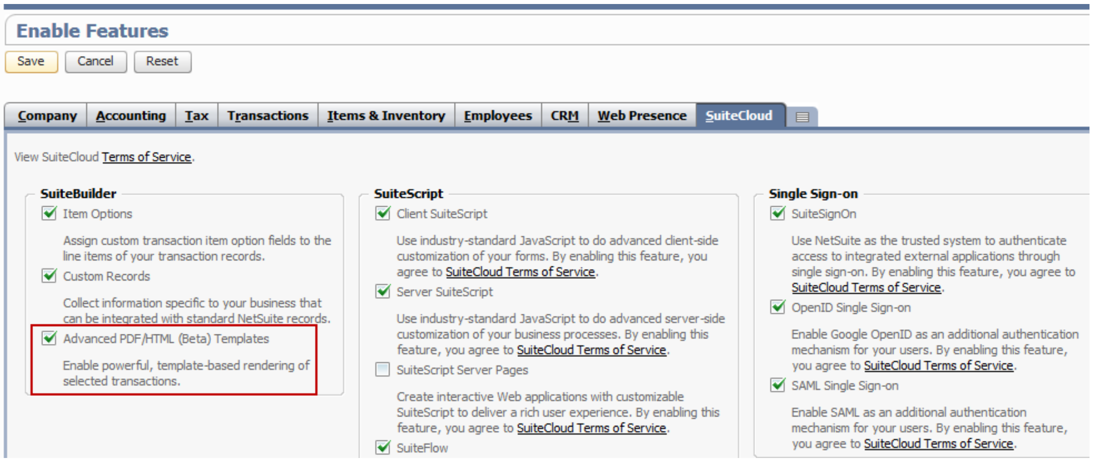
SuiteScript supports a template engine object and related methods so you can apply advanced template format capabilities programmatically. For details, see the help topics *nlapiCreateTemplateRenderer()* and *nlobjTemplateRenderer*.

In addition to this function and object, the SuiteScript function *nlapiPrintRecord(type, id, mode, properties)* supports the use of advanced templates. If you associate an advanced template with the custom form saved for a transaction and use this API to print the transaction, the advanced template is used to format the printed transaction.

Enabling the Advanced PDF/HTML Templates (Beta) Feature

The Advanced PDF/HTML Templates feature must be enabled for access to advanced templates and the template editor.

To enable the Advanced PDF/HTML Templates feature, go to Setup > Company > Enable Features, and on the SuiteCloud tab, under SuiteBuilder, check the box for Advanced PDF/HTML Templates (Beta).



Enable Features

Save | Cancel | Reset

Company | Accounting | Tax | Transactions | Items & Inventory | Employees | CRM | Web Presence | SuiteCloud

View SuiteCloud Terms of Service.

SuiteBuilder

☑ Item Options

Assign custom transaction item option fields to the line items of your transaction records.

☑ Custom Records

Collect information specific to your business that can be integrated with standard NetSuite records.

☑ Advanced PDF/HTML (Beta) Templates

Enable powerful, template-based rendering of selected transactions.

SuiteScript

☑ Client SuiteScript

Use industry-standard JavaScript to do advanced client-side customization of your forms. By enabling this feature, you agree to SuiteCloud Terms of Service.

☑ Server SuiteScript

Use industry-standard JavaScript to do advanced server-side customization of your business processes. By enabling this feature, you agree to SuiteCloud Terms of Service.

☐ SuiteScript Server Pages

Create interactive Web applications with customizable SuiteScript to deliver a rich user experience. By enabling this feature, you agree to SuiteCloud Terms of Service.

☑ SuiteFlow

Single Sign-on

☑ SuiteSignOn

Use NetSuite as the trusted system to authenticate access to integrated external applications through single sign-on. By enabling this feature, you agree to SuiteCloud Terms of Service.

☑ OpenID Single Sign-on

Enable Google OpenID as an additional authentication mechanism for your users. By enabling this feature, you agree to SuiteCloud Terms of Service.

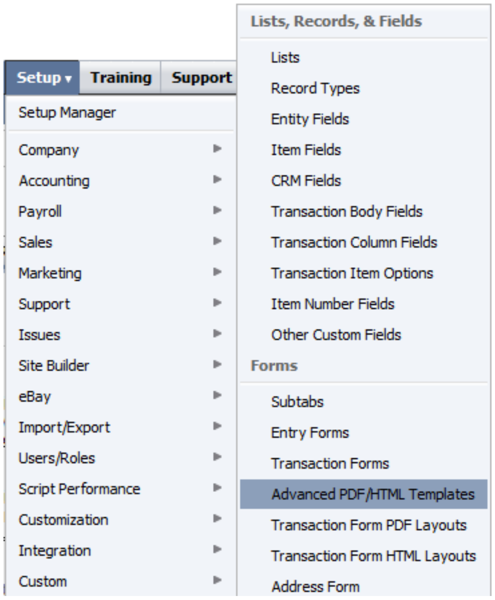
☑ SAML Single Sign-on

Enable SAML as an additional authentication mechanism for your users. By enabling this feature, you agree to SuiteCloud Terms of Service.

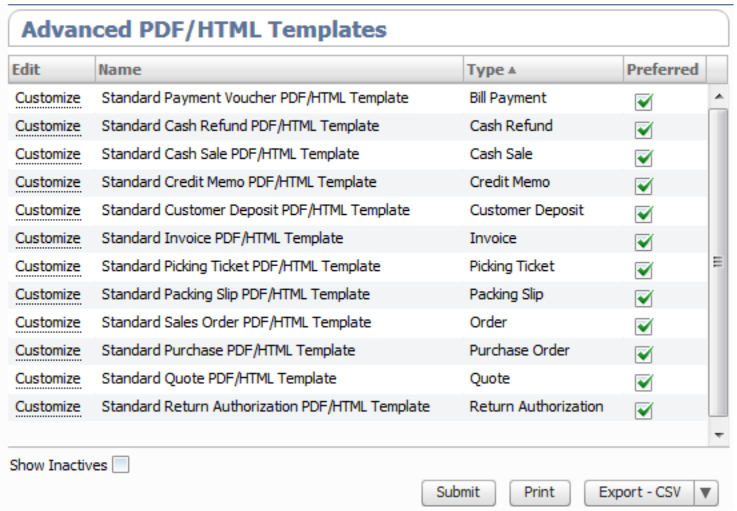
Reviewing Available Advanced Templates

When the Advanced PDF/HTML Templates feature is enabled, an Advanced PDF/HTML Templates option is available in the Setup > Customization menu. You can click this option to view a list of the advanced templates in your account.

Note: The Advanced PDF/HTML Templates menu option is only available to users who have the Custom PDF Layouts permission. This is a Setup type permission. For details about permissions, see the help topic *Understanding NetSuite Permissions*.



The Advanced PDF/HTML Templates list page includes standard advanced templates and any customized advanced templates that have been created in the account.



Advanced PDF/HTML Templates

Edit	Name	Type ▲	Preferred
Customize	Standard Payment Voucher PDF/HTML Template	Bill Payment	☑
Customize	Standard Cash Refund PDF/HTML Template	Cash Refund	☑
Customize	Standard Cash Sale PDF/HTML Template	Cash Sale	☑
Customize	Standard Credit Memo PDF/HTML Template	Credit Memo	☑
Customize	Standard Customer Deposit PDF/HTML Template	Customer Deposit	☑
Customize	Standard Invoice PDF/HTML Template	Invoice	☑
Customize	Standard Picking Ticket PDF/HTML Template	Picking Ticket	☑
Customize	Standard Packing Slip PDF/HTML Template	Packing Slip	☑
Customize	Standard Sales Order PDF/HTML Template	Order	☑
Customize	Standard Purchase PDF/HTML Template	Purchase Order	☑
Customize	Standard Quote PDF/HTML Template	Quote	☑
Customize	Standard Return Authorization PDF/HTML Template	Return Authorization	☑

Show Inactives □

By default, the standard advanced template is the preferred advanced template for a print type. When you choose the Advanced printing type for a custom form, the preferred advanced template is selected by default as the template for that form. See Setting Custom Forms to Use Advanced Templates.

Setting Custom Forms to Use Advanced Templates

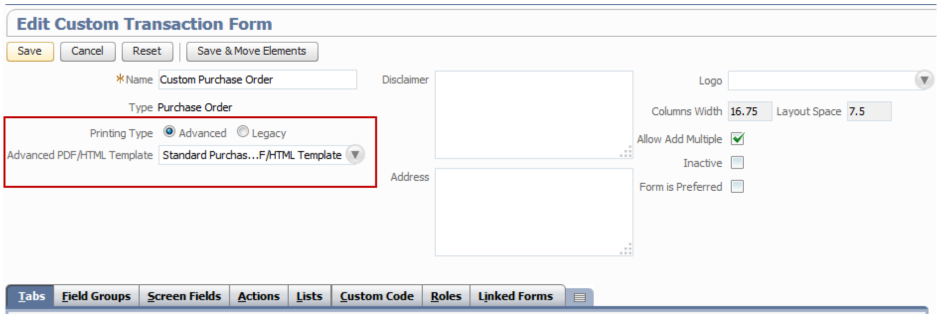
When the Advanced PDF/HTML Templates feature is enabled, you can set custom forms for cash refunds, cash sales, credit memos, customer deposits, invoices, packing slips, payment vouchers, picking tickets, purchase orders, quotes (estimates), return authorizations, and sales orders to use advanced templates. When you set a custom form to use an advanced template, that template defines the print formatting and contents for transactions that use that custom form.

To define an advanced template for a custom form:

1. Go to Setup > Customization > Transaction Forms.
2. On the Custom Transaction Forms page, click the Edit link for an existing cash refund, cash sale, credit memo, customer deposit, invoice, packing slip, payment voucher, picking ticket, purchase order, quote (estimate), return authorization, or sales order form, or click the Customize link for a standard form.
3. On the Edit page for the custom form, review the Printing Type options.
    - The Legacy option allows legacy PDF layouts and HTML layouts to be associated with the custom form.
    - The Advanced option allows an advanced template to be associated with the custom form. (Note that you select one advanced template to be used for both PDF and HTML layouts.)

4. Select an advanced template from the dropdown.

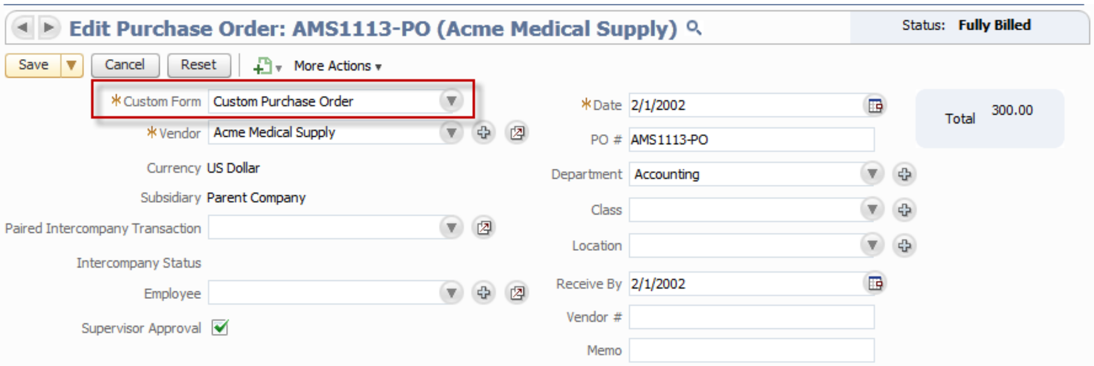
    - The Advanced PDF/HTML Template dropdown lists the standard advanced template and any custom advanced templates for the transaction type.
    - The preferred advanced template is selected by default in the dropdown.



Edit Custom Transaction Form

*Name: Custom Purchase Order
Type: Purchase Order
Printing Type: ● Advanced ○ Legacy
Advanced PDF/HTML Template: Standard Purchas…F/HTML Template
Disclaimer
Address
Logo
Columns Width: 16.75
Layout Space: 7.5
Allow Add Multiple: ☑
Inactive: □
Form is Preferred: □

For example, the screenshot above indicates that any purchase order transaction that has the Custom Purchase Order form selected for Custom Form will use the advanced template “Standard Purchase PDF/HTML Template”. Thus, formatting and contents for printed and emailed versions of the purchase order shown in the following screenshot would be defined by this template.



Edit Purchase Order: AMS1113-PO (Acme Medical Supply)

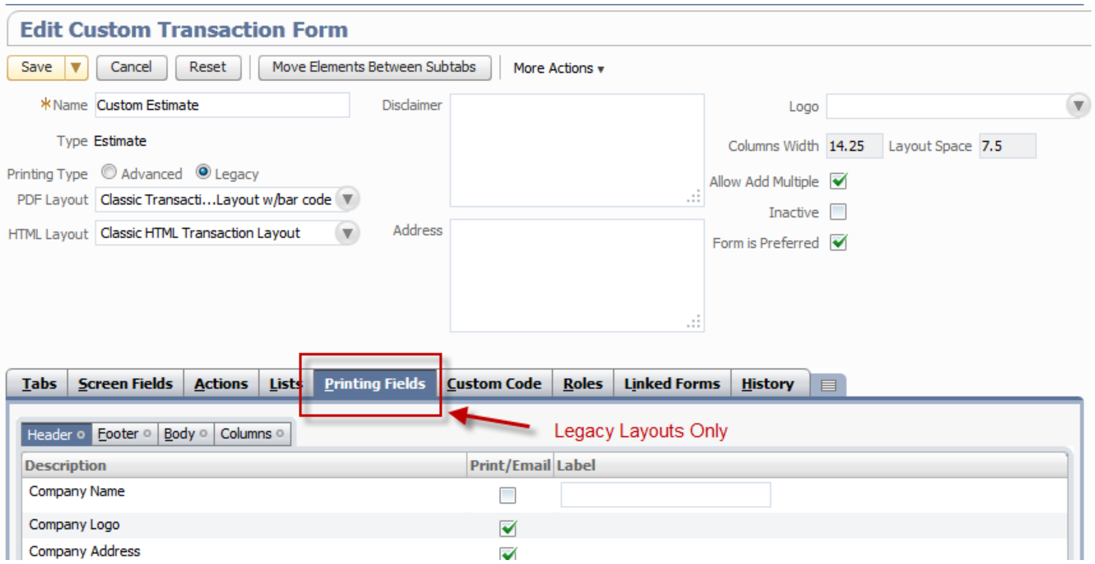
Status: Fully Billed

*Custom Form: Custom Purchase Order
*Vendor: Acme Medical Supply
Currency: US Dollar
Subsidiary: Parent Company
Paired Intercompany Transaction
Intercompany Status
Employee
Supervisor Approval: ☑
*Date: 2/1/2002
PO #: AMS1113-PO
Department: Accounting
Class
Location
Receive By: 2/1/2002
Vendor #
Memo
Total: 300.00

Important: At any time after you have set a custom form to use an advanced template, you can switch the form back to using legacy layouts. Just edit the custom form and choose Legacy for Printing Type, and select from the PDF Layout and HTML Layout dropdowns. All previously available legacy layouts are still available.

Printing Fields Subtab Not Used with Advanced Templates

Once you select Advanced for a custom form's Printing Type, notice that the Printing Fields subtab is no longer available for the custom form. Because advanced templates explicitly include the fields to be printed, there is no need for a separate listing of these fields on the custom form record. You can change the fields to be included in a custom advanced template by modifying the actual template in the template editor. See Customizing Advanced Templates in the Template Editor.



Edit Custom Transaction Form

*Name: Custom Estimate
Type: Estimate
Printing Type: ○ Advanced ● Legacy
PDF Layout: Classic Transacti…Layout w/bar code
HTML Layout: Classic HTML Transaction Layout
Disclaimer:
Address:
Logo:
Columns Width: 14.25
Layout Space: 7.5
Allow Add Multiple: ☑
Inactive: □
Form is Preferred: ☑

Description	Print/Email	Label
Company Name	□	
Company Logo	☑	
Company Address	☑	

Customizing Advanced Templates in the Template Editor

NetSuite provides a WYSIWYG template editor where you can review the formatting and contents of standard advanced templates, and edit them to create custom advanced templates. The template editor supports current industry standards for HTML-based editing, including rich text editing and HTML source editing. For more details, see:

- Viewing an Advanced Template in the Template Editor
- WYSIWYG View and Rich Text Editing
- HTML Source Editing

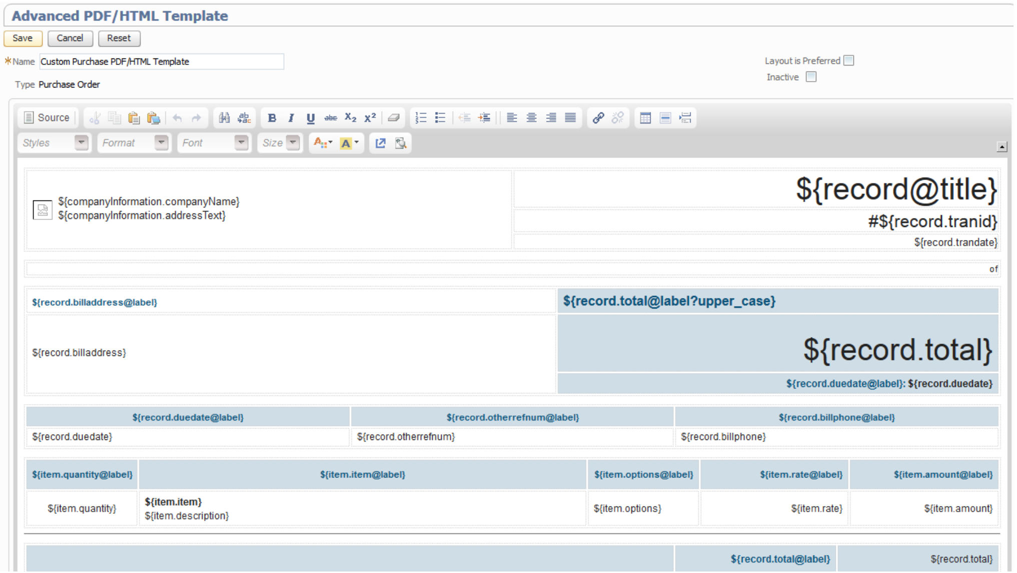
Note: The template editor uses FreeMarker-based syntax. FreeMarker documentation is available from this link, or go to:

http://freemarker.sourceforge.net/docs/index.html.

Viewing an Advanced Template in the Template Editor

To see an advanced template in the template editor, go to Setup > Customization > Advanced PDF/HTML Templates and:

- Click the Customize link for a standard advanced template to review it in the template editor and create a customized version of it, or
- Click the Edit link for a custom advanced template to review it in the template editor and make further changes as desired.



Advanced PDF/HTML Template

*Name Custom Purchase PDF/HTML Template

Type Purchase Order

Layout is Preferred □

Inactive □

${companyInformation.companyName}
${companyInformation.addressText}

${record@title}

#${record.tranid}

${record.trandate}

of

${record.billaddress@label}

${record.billaddress}

${record.total@label?upper_case}

${record.total}

${record.duedate@label}: ${record.duedate}

${record.duedate@label}	${record.otherrefnum@label}	${record.billphone@label}
${record.duedate}	${record.otherrefnum}	${record.billphone}

${item.quantity@label}	${item.item@label}	${item.options@label}	${item.rate@label}	${item.amount@label}
${item.quantity}	**${item.item}** ${item.description}	${item.options}	${item.rate}	${item.amount}

	${record.total@label}	${record.total}

Note: The default naming convention is “Custom <Print Type> PDF/HTML Template” for the first custom template created for a print type. For each subsequent template for that type, the default is to add a sequence number to the name. For example, the standard advanced template for purchase orders is named “Standard Purchase PDF/HTML Template”, the first custom advanced template for purchase orders has a default name of “Custom Purchase PDF/HTML Template”, and the next custom advanced templates for purchase orders have default names of “Custom Purchase PDF/HTML Template 2” and “Custom Purchase PDF/HTML Template 3”.

WYSIWYG View and Rich Text Editing

In the default WYSIWYG view of the template editor, you can scroll vertically and horizontally to review exactly which fields will be printed. You can add and remove fields as desired, and use built-in rich text editing functions to format contents. Rich text editing buttons have tooltips, so it is easy to find the one you need.

The template editor uses a simple syntax to get fields to be printed. Note that although this syntax may appear similar to SuiteScript, the template editor does not support the execution of SuiteScript APIs. See Understanding Syntax for Advanced Template Fields.

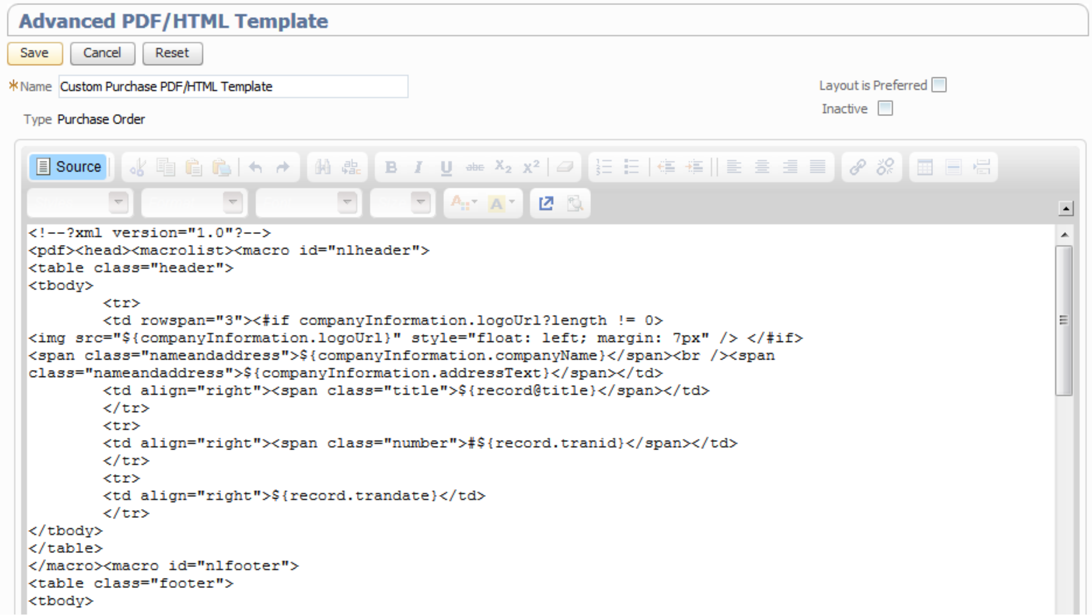
The following are example customizations you can do in the WYSIWYG view:

- Adding and Removing Fields in Advanced Templates

- Formatting Text on Advanced Templates
- Aligning and Justifying Text on Advanced Templates
- Adding Tables to Advanced Templates

HTML Source Editing

You can click the Source button in the template editor to view the HTML source for an advanced template.



*Name Custom Purchase PDF/HTML Template
Layout is Preferred □
Inactive □
Type Purchase Order

```
<!--?xml version="1.0"?-->
<pdf><head><macrolist><macro id="nlheader">
<table class="header">
<tbody>
        <tr>
        <td rowspan="3"><#if companyInformation.logoUrl?length != 0>
<img src="${companyInformation.logoUrl}" style="float: left; margin: 7px" /> </#if>
<span class="nameandaddress">${companyInformation.companyName}</span><br /><span
class="nameandaddress">${companyInformation.addressText}</span></td>
        <td align="right"><span class="title">${record@title}</span></td>
        </tr>
        <tr>
        <td align="right"><span class="number">#${record.tranid}</span></td>
        </tr>
        <tr>
        <td align="right">${record.trandate}</td>
        </tr>
</tbody>
</table>
</macro><macro id="nlfooter">
<table class="footer">
<tbody>
```

You can make template edits directly in this source text if you have sufficient knowledge of HTML. For example, if you want to include an image on your printed transactions, you can simply add an <img> tag to the template source text.

The following are example customizations you can do by editing HTML source:

- Adding Striping to Advanced Templates
- Adding Headers and Footers to Advanced Templates
- Adding Bar Codes to Advanced Templates
- Changing Page Size for Advanced Templates
- Changing Margins on Advanced Templates

Note: Some HTML tags may be displayed as literals in printed text. For example, if a field contains the character for a line break, then the line break literal value of "<br />" is displayed.

Understanding Syntax for Advanced Template Fields

The template editor uses a simple syntax to get fields to be printed. Note that although this syntax may appear similar to SuiteScript, the template editor does not support the execution of SuiteScript APIs.

Syntax for Body Fields

For most template body fields, simplified syntax is **${<object>.<field ID>}**.

- The following objects are currently supported:
  - **companyInformation** - company information for the current NetSuite account

    For example, **${companyInformation.companyname}** is the name associated with the NetSuite account.
  - **record** - current transaction record

    For example **${record.trandate}** is the date of the transaction.
- Syntax for body fields can also include an additional component, such as:
  - **${<object>.<field ID>@label}** - indicates the label of a field should be printed

    For example, **${record.trandate@label}** indicates the label for the transaction date should be printed. The label is the text that displays next to the field on the transaction form; in this case, "Date".

Syntax for Sublist Fields

Syntax for sublist (line item) fields is similar to body field syntax. Sublist fields are denoted with syntax like:

- **${item.item@label}**
- **${item.item}**
- **${item.quantity}**

Adding and Removing Fields in Advanced Templates

You can see the fields included on an advanced template by reviewing the template in the template editor. You can add and remove fields as desired, and also add currency symbols in front of fields, by following the steps in the following procedures:

- Adding a Field to an Advanced Template
- Adding a Currency Symbol to an Advanced Template Field
- Removing a Field from an Advanced Template

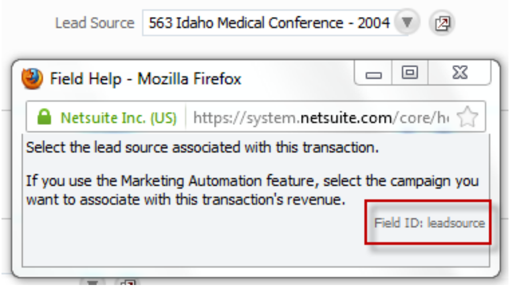
Adding a Field to an Advanced Template

To add a field to an advanced template:

1. Go to Setup > Customization > Advanced PDF/HTML Templates and click the Customize or Edit link for the advanced template.

2. Obtain the field ID for the field you want to add.

   - To make field IDs available, go to Home > Set Preferences and ensure that the Show Internal IDs option is enabled on the General subtab, Defaults area.
   - Find the field in the NetSuite user interface and click the field label to display the field level help text.

   
   Field Help - Mozilla Firefox
Netsuite Inc. (US)
https://system.netsuite.com/core/h

   Lead Source 563 Idaho Medical Conference - 2004

   Select the lead source associated with this transaction.

   If you use the Marketing Automation feature, select the campaign you want to associate with this transaction's revenue.

   Field ID: leadsource

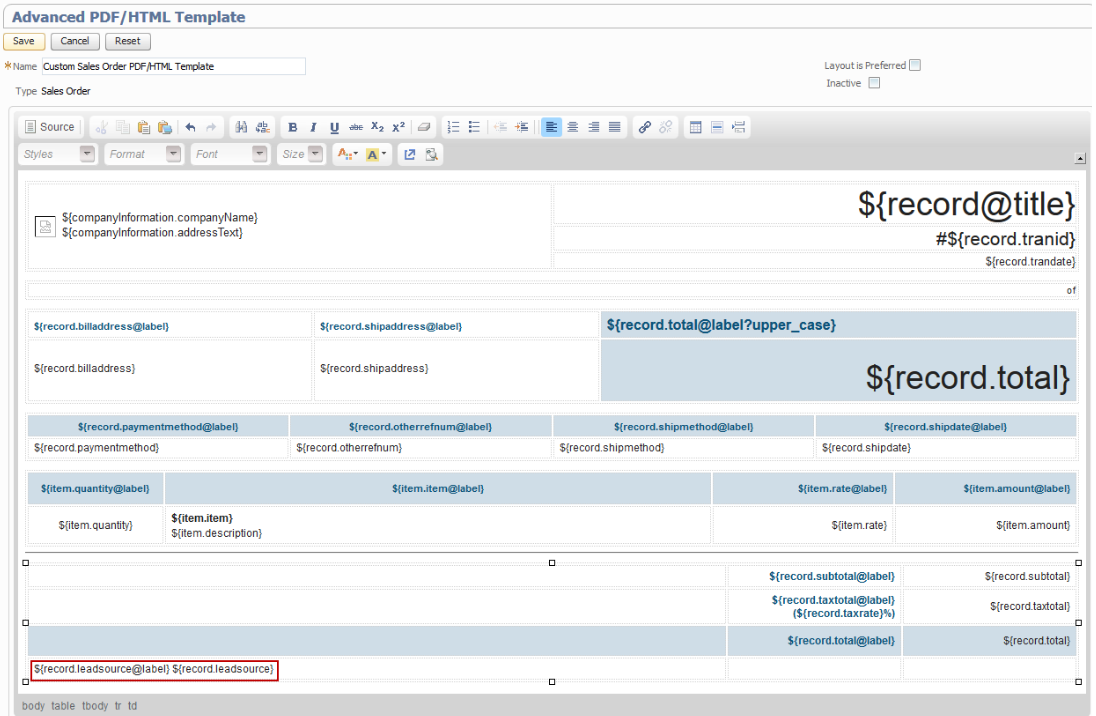
   The field ID is displayed in the popup.

   Note: Field ID information is also available in the SuiteScript Records Browser. See the help topic *Using the SuiteScript Records Browser*.

3. Enter the field, using appropriate syntax, in the desired location of the advanced template.

   The template editor supports a simplified syntax to get values for the fields to be printed. Review Understanding Syntax for Advanced Template Fields for details.

For example, you can add **${record.leadsource}** to a sales order advanced template to print the lead source for the order. You can add **${record.leadsource@label}** to display “Lead Source” in front of the value.



Advanced PDF/HTML Template

*Name Custom Sales Order PDF/HTML Template

Type Sales Order

Layout is Preferred □

Inactive □

${companyInformation.companyName}
${companyInformation.addressText}

${record@title}
#${record.tranid}
${record.trandate}

of

${record.billaddress@label}	${record.shipaddress@label}	${record.total@label?upper_case}
${record.billaddress}	${record.shipaddress}	${record.total}

${record.paymentmethod@label}	${record.otherrefnum@label}	${record.shipmethod@label}	${record.shipdate@label}
${record.paymentmethod}	${record.otherrefnum}	${record.shipmethod}	${record.shipdate}

${item.quantity@label}	${item.item@label}	${item.rate@label}	${item.amount@label}
${item.quantity}	${item.item} ${item.description}	${item.rate}	${item.amount}

	${record.subtotal@label}	${record.subtotal}
	${record.taxtotal@label} (${record.taxrate}%)	${record.taxtotal}
	${record.total@label}	${record.total}
${record.leadsource@label} ${record.leadsource}		

Adding a Currency Symbol to an Advanced Template Field

To add a currency symbol to an advanced template:

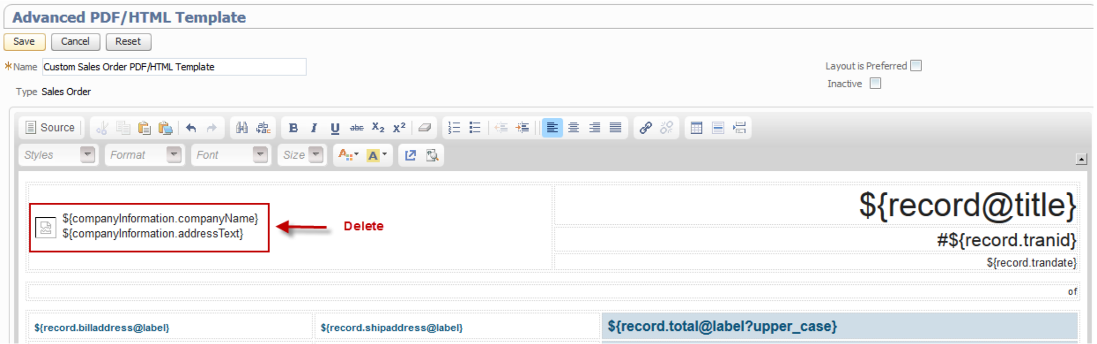
- Add **${record.currencysymbol}** in front of the field that represents a currency amount, for example: **${record.currencysymbol}${record.total}**.

Removing a Field from an Advanced Template

To remove a field from an advanced template:

- To remove a field, simply select it and delete it.

For example, if you do not want sales orders to include the company name and address, you could delete **${companyInformation.companyname}** and **${companyInformation.addressText}**. Note that the example below also results in the deletion of the company logo.



Advanced PDF/HTML Template

*Name Custom Sales Order PDF/HTML Template

Type Sales Order

Layout is Preferred □

Inactive □

${companyInformation.companyName}
${companyInformation.addressText}

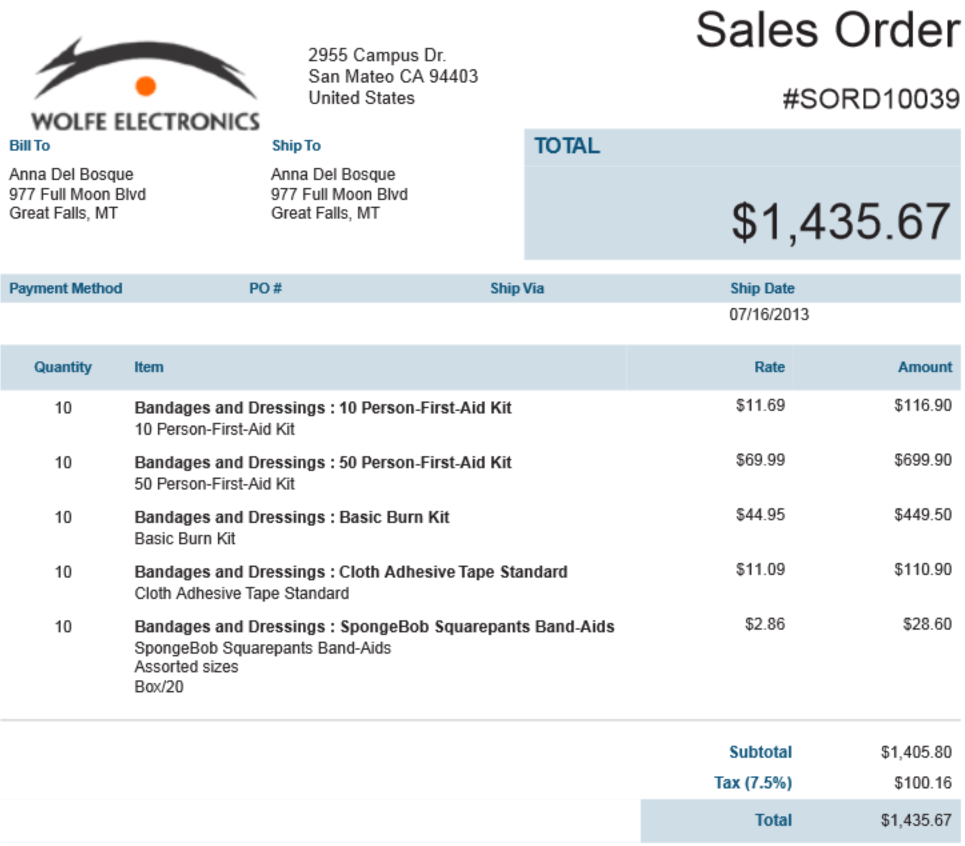
${record@title}
#${record.tranid}
${record.trandate}

of

${record.billaddress@label}	${record.shipaddress@label}	${record.total@label?upper_case}

Printed Sales Order Using Customized Advanced Printing Form

The following sales order uses the Standard Sales Order PDF/HTML Template:



WOLFE ELECTRONICS

2955 Campus Dr.
San Mateo CA 94403
United States

Sales Order

#SORD10039

Bill To	Ship To	TOTAL
Anna Del Bosque 977 Full Moon Blvd Great Falls, MT	Anna Del Bosque 977 Full Moon Blvd Great Falls, MT	$1,435.67

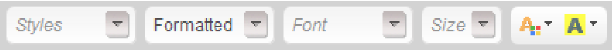
Payment Method	PO #	Ship Via	Ship Date
			07/16/2013

Quantity	Item	Rate	Amount
10	**Bandages and Dressings : 10 Person-First-Aid Kit** 10 Person-First-Aid Kit	$11.69	$116.90
10	**Bandages and Dressings : 50 Person-First-Aid Kit** 50 Person-First-Aid Kit	$69.99	$699.90
10	**Bandages and Dressings : Basic Burn Kit** Basic Burn Kit	$44.95	$449.50
10	**Bandages and Dressings : Cloth Adhesive Tape Standard** Cloth Adhesive Tape Standard	$11.09	$110.90
10	**Bandages and Dressings : SpongeBob Squarepants Band-Aids** SpongeBob Squarepants Band-Aids Assorted sizes Box/20	$2.86	$28.60

Subtotal	$1,405.80
Tax (7.5%)	$100.16
Total	$1,435.67

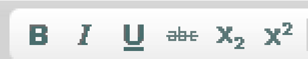
Formatting Text on Advanced Templates

The template editor includes rich text editing buttons that you can use for the following text formatting functions on advanced templates:



- Formatting Styles
- Paragraph Formats
- Font Name
- Font Size
- Text Color
- Background Color

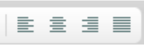
The editor also includes buttons for font styles and effects:



And there is a button that removes formatting from text:

Aligning and Justifying Text on Advanced Templates

The template editor includes rich text editing buttons that you can use for the following text alignment functions on advanced templates:



- Align Left
- Center
- Align Right
- Justify

Adding Tables to Advanced Templates

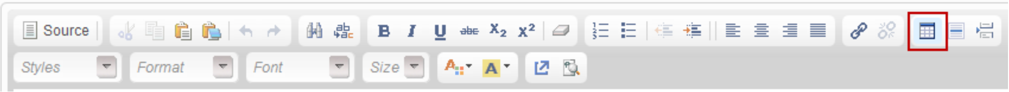
The template editor includes a rich text editing button that you can use to insert tables on advanced PDF/HTML templates.

To insert a table on an advanced template:

1. Open the advanced template in the template editor.

2. Place the cursor in the spot where you want to insert the table.

3. Click the rich text editing button to insert a table:



4. Complete the Table Properties dialog and click OK.

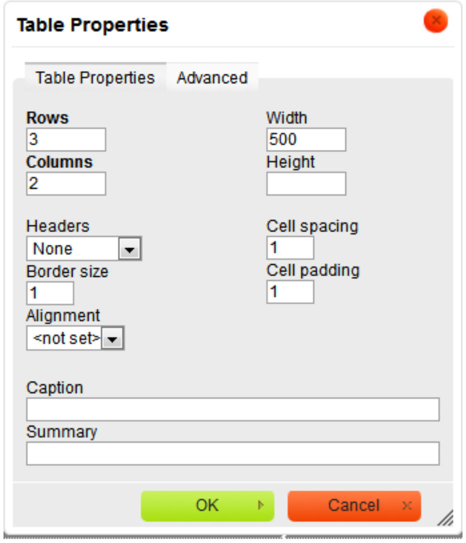
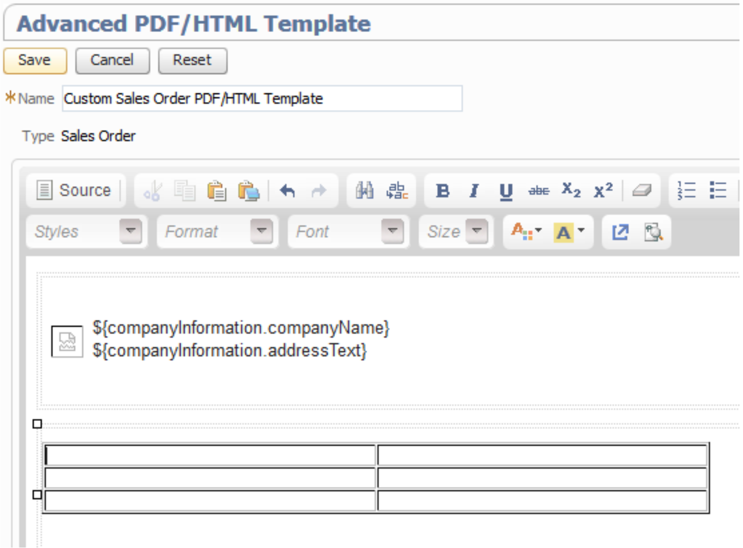


Table Properties

Table Properties | Advanced

Rows: 3
Columns: 2
Width: 500
Height:
Headers: None
Cell spacing: 1
Border size: 1
Cell padding: 1
Alignment: <not set>
Caption:
Summary:

The table is added to the template in the template editor.



Advanced PDF/HTML Template

Save | Cancel | Reset

*Name: Custom Sales Order PDF/HTML Template

Type: Sales Order

${companyInformation.companyName}
${companyInformation.addressText}

Note: You can also use HTML source editing to add a table to an advanced template, but it would be more time-consuming and require knowledge of HTML.

Adding Striping to Advanced Templates

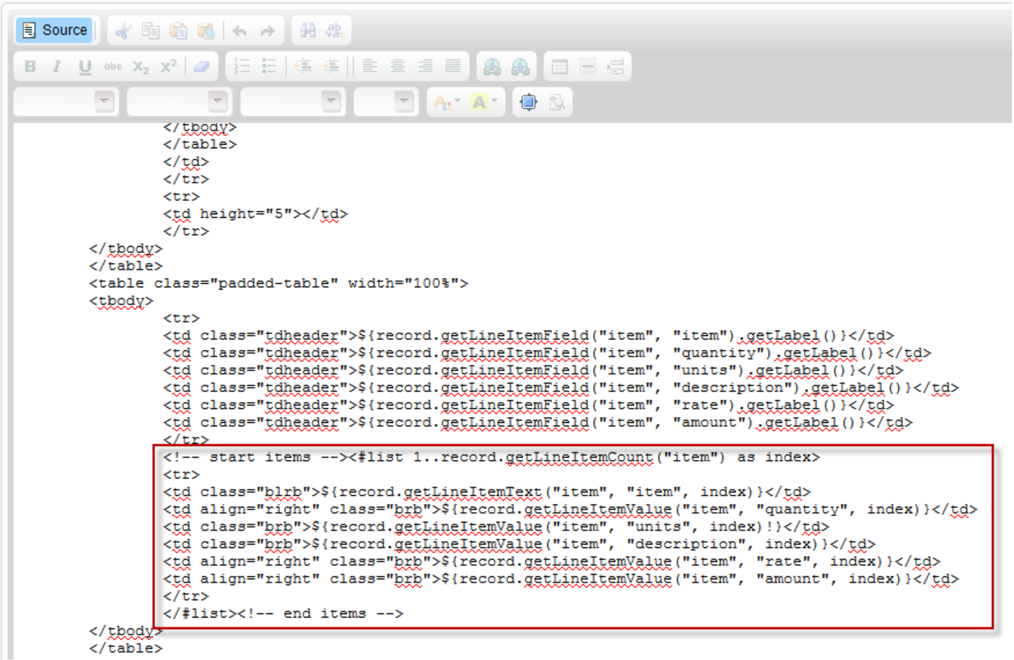
You can edit HTML source in the template editor to add striping to the line items table in an advanced template.

Note: The syntax for adding striping to the line items tables relies on BFO, a set of third party libraries used by NetSuite for generating PDF documents. BFO documentation is available at http://bfo.com/products/report/docs/userguide.pdf.

To add striping to a line items table:

1. Open the advanced template in the template editor and click the Source button.

2. Scroll down to the portion of the HTML source relating to the rows in the line items table:

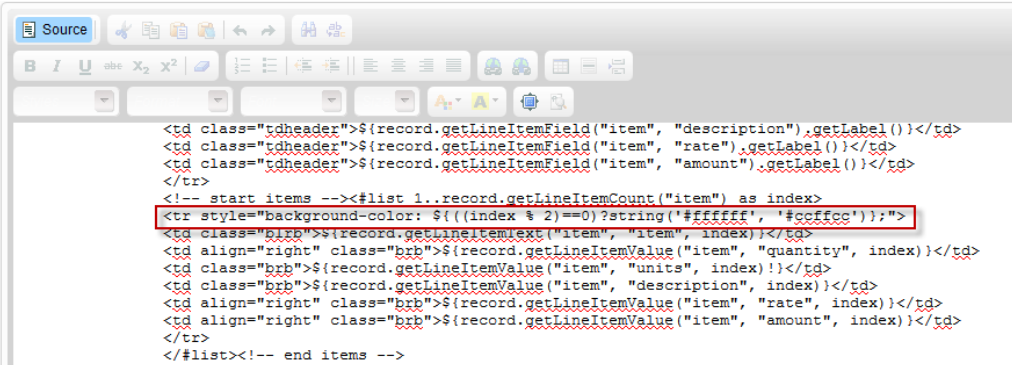


```
                </tbody>
                </table>
                </td>
                </tr>
                <tr>
                <td height="5"></td>
                </tr>
        </tbody>
        </table>
        <table class="padded-table" width="100%">
        <tbody>
                <tr>
                <td class="tdheader">${record.getLineItemField("item", "item").getLabel()}</td>
                <td class="tdheader">${record.getLineItemField("item", "quantity").getLabel()}</td>
                <td class="tdheader">${record.getLineItemField("item", "units").getLabel()}</td>
                <td class="tdheader">${record.getLineItemField("item", "description").getLabel()}</td>
                <td class="tdheader">${record.getLineItemField("item", "rate").getLabel()}</td>
                <td class="tdheader">${record.getLineItemField("item", "amount").getLabel()}</td>
                </tr>
                <!-- start items --><#list 1..record.getLineItemCount("item") as index>
                <tr>
                <td class="blrb">${record.getLineItemText("item", "item", index)}</td>
                <td align="right" class="brb">${record.getLineItemValue("item", "quantity", index)}</td>
                <td class="brb">${record.getLineItemValue("item", "units", index)!}</td>
                <td class="brb">${record.getLineItemValue("item", "description", index)}</td>
                <td align="right" class="brb">${record.getLineItemValue("item", "rate", index)}</td>
                <td align="right" class="brb">${record.getLineItemValue("item", "amount", index)}</td>
                </tr>
                </#list><!-- end items -->
        </tbody>
        </table>
```

Notice that the FreeMarker <#list> directive starts a loop for the line items in a transaction: **<#list 1..record.getLineItemCount("item") as index>**.

3. Edit the first <tr> tag after the <#list> directive, to specify different colors for rows in the line items table, like the following example:

<tr style="background-color: ${((index % 2)==0)?string('#ffffff', '#ccffcc')};">



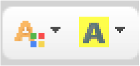
```
<td class="tdheader">${record.getLineItemField("item", "description").getLabel()}</td>
<td class="tdheader">${record.getLineItemField("item", "rate").getLabel()}</td>
<td class="tdheader">${record.getLineItemField("item", "amount").getLabel()}</td>
</tr>
<!-- start items --><#list 1..record.getLineItemCount("item") as index>
<tr style="background-color: ${((index % 2)==0)?string('#ffffff', '#ccffcc')};">
<td class="blrb">${record.getLineItemText("item", "item", index)}</td>
<td align="right" class="brb">${record.getLineItemValue("item", "quantity", index)}</td>
<td class="brb">${record.getLineItemValue("item", "units", index)!}</td>
<td class="brb">${record.getLineItemValue("item", "description", index)}</td>
<td align="right" class="brb">${record.getLineItemValue("item", "rate", index)}</td>
<td align="right" class="brb">${record.getLineItemValue("item", "amount", index)}</td>
</tr>
</#list><!-- end items -->
```

In this example, even rows use the color represented in hexadecimal by #ffffff and odd rows use the color represented by #ccffcc.

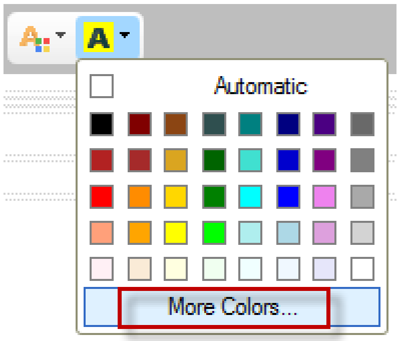
To get hexadecimal codes for striping colors:

You can look up hex codes in the HTML color picker provided in the template editor:

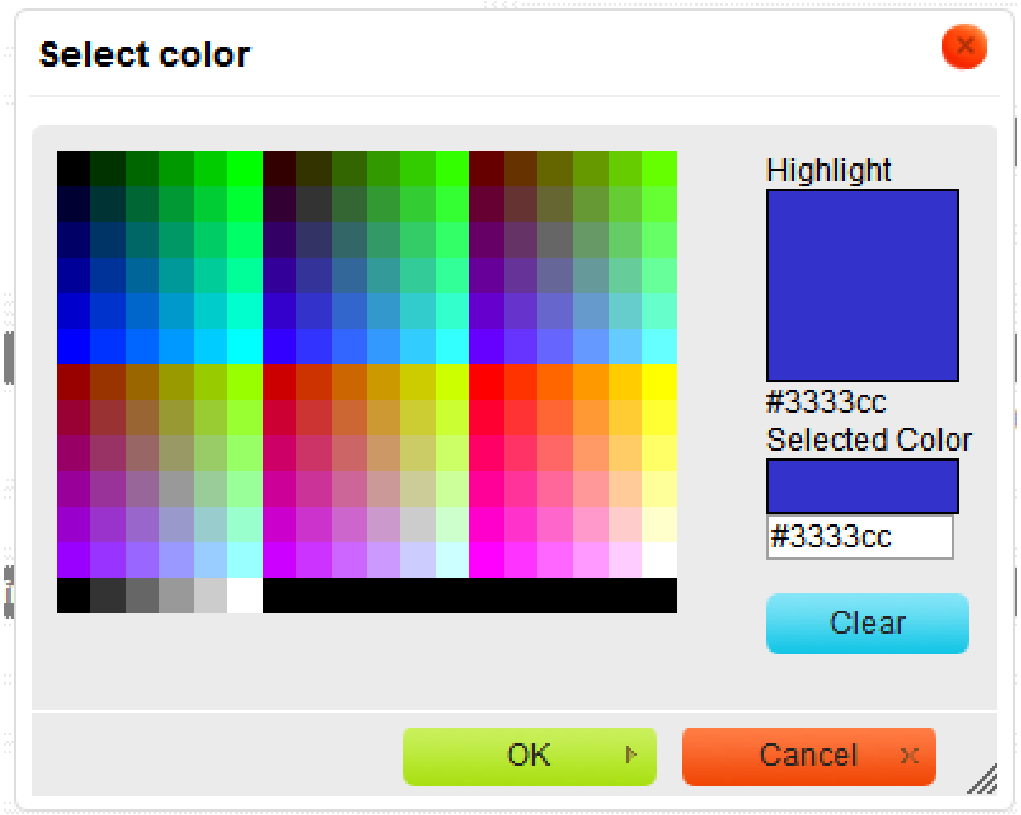
1. In rich text editing mode, click the text color or background color button.



2. Click More Colors.



3. In the Select Color dialog, click a color to see the hex code.



Adding Headers and Footers to Advanced Templates

You can edit HTML source in the template editor to add headers and footers to advanced templates.

Note: The macros for adding headers and footers rely on BFO, a set of third party libraries used by NetSuite for generating PDF documents. BFO documentation is available at http://bfo.com/products/report/docs/userguide.pdf.

The following examples illustrate how to add a header and footer. The added header and footer code is in bold.

To add a header with a page number to each page:

Add text like the following to the HTML source.

```
<pdf>
    <head>
      <macrolist>
        <macro id="header">
           <p align="center">
```

```
             Page <pagenumber/> of <totalpages/>
           </p>
        </macro>
      </macrolist>
    </head>
    <body footer="header" header-height="20mm">
      Document contents here
    </body>
  </pdf>
```

To add a footer to each page:

Add text like the following to the HTML source.

```
<pdf>
    <head>
      <macrolist>
        <macro id="footer">
           <p align="center">
            Copy for Your Files
           </p>
        </macro>
      </macrolist>
    </head>
    <body footer="footer" footer-height="20mm">
      Document contents here
    </body>
  </pdf>
```

The following text adds a disclaimer in the footer:

```
<pdf>
    <head>
      <macrolist>
        <macro id="footer">
           <p align="center">Disclaimer:  All of this software is used at your own risk.</p>
        </macro>
      </macrolist>
    </head>
    <body footer="footer" footer-height="20mm">
```

Adding Bar Codes to Advanced Templates

You can edit HTML source in the template editor to add bar codes to advanced templates.

Note: The syntax for adding bar codes relies on BFO, a set of third party libraries used by NetSuite for generating PDF documents. BFO documentation is available at http://bfo.com/products/report/docs/userguide.pdf.

BFO supports different bar code types. The following example illustrates how to insert a bar code. See the BFO documentation for a list of all code types.

To add a bar code to an advanced template:

Add text like the following to the HTML source.

<barcode codetype="code128" showtext="true" value="My Value"/>

My Value refers to the value of the bar code. In the case of a transaction it would be **${record.tranid}**.

Changing Page Size for Advanced Templates

The default page setup for advanced templates fits American Letter size paper. You can change this setup to fit another page size by specifying the type of paper in the body tag.

To change the page size for an advanced template:

Add text like the following to the HTML source.

<body size=“A4”>

Advanced templates can be set up to fit any page size supported by BFO, a set of third party libraries used by NetSuite for generating PDF documents. BFO documentation is available at http://bfo.com/products/report/docs/userguide.pdf.

Supported sizes include:

- American sizes
  - Letter
  - Legal
  - Executive
  - Ledger
- ISO A series
  - A3
  - A4
  - A5
  - A6
- ISO B series
  - B4
  - B5

- B6

Changing Margins on Advanced Templates

For advanced templates, the margin attribute is a transparent border between a block and its neighbors. This concept comes from BFO, a set of third party libraries used by NetSuite for generating PDF documents. BFO documentation is available at http://bfo.com/products/report/docs/userguide.pdf.

To create a margin of 10 pixels for the entire page:

1. In the style block create a class called "order". Specify the margins.

   .order {margin-left:10; margin-top:10; margin-bottom:10;margin-right:10}

2. In between the body tags create a div as follows:

   ```
   <body>
   <div class=”order”>
   </div>
   </body>
   ```

Current Limitations for the Advanced PDF/HTML Templates Feature

The following limitations apply to the beta version of the Advanced PDF/HTML Templates feature:

- Advanced PDF/HTML templates are not supported for all record types.
- You cannot modify templates in the template editor to include fields from records other than the current transaction record. For example, if you are editing a sales order template, you can only add fields from the current sales order record, not from any related records, to the template. (Fields from related records are available for advanced PDF/HTML templates customized through scripting.)
- The template editor does not provide a prepopulated selector for field names; users must obtain field IDs themselves. (See Adding a Field to an Advanced Template for tips for obtaining field IDs.)
- Advanced PDF/HTML templates cannot be included in bundles.
- The Download PDF Files user preference, set at Home > Preferences on the General subtab's Defaults area, is not supported for transactions using advanced PDF/HTML templates.
- Advanced PDF/HTML templates do not support printing in Chinese, Japanese, or Korean by default. In order to print a transaction using an advanced template in one of these languages, you need to edit the template's HTML source in the template editor, to change

the CSS definition for the table attribute from *font-family: sans-serif;* to *font-family: hygothic, sans-serif;*. Also note that the user who is printing in one of these language needs to have the CJK font pack installed.

- Advanced PDF/HTML templates do not support CRMSDK tags.

Transaction Form PDF Layouts

A transaction form PDF layout defines the arrangement of fields on printed standard and classic (or legacy) PDF transaction documents in NetSuite. You can customize transaction form PDF layouts by customizing borders and content position, repositioning and resizing fields, or by changing the fonts and colors used.

Custom transaction form PDF layouts are only available when you print your forms in PDF. If you print using HTML, a custom layout does not affect the look of your form. To print using PDF, go to Home > Set Preferences > Transactions tab. Clear the Print Using HTML box, and click Save.

To customize a PDF layout, you first set up the color of text and backgrounds on the form and set the page height and width of printed forms. Then, you use the Form Editor to change the size and orientation of elements of the form.

You can choose to enter your measurements in inches or in centimeters by setting the Form Size Measurements preference at Setup > Company > Preferences > Printing, Fax, & Email Preferences on the Printing subtab.

Note: Advanced PDF/HTML templates provide an alternate model for customizing printed and emailed records. They support more customization capabilities than transaction form layouts. The beta version of the Advanced PDF/HTML Templates feature currently supports cash refund, cash sale, credit memo, customer deposit, invoice, packing slip, payment voucher, picking ticket, purchase order, quote (estimate), return authorization, and sales order templates. For details, see Advanced PDF/HTML Templates (Beta).

To customize a transaction form PDF layout:

1. Go to Setup > Customization > Forms > Transaction Form PDF Layouts.
2. Click Customize next to the layout you want to customize.

   Note: A transaction form PDF layout cannot be customized for check vouchers.

3. On the Custom PDF Layout page, enter a name for your custom layout.
4. If desired, edit the width and height properties for your page.
5. In the Top Margin field, enter the distance between the top of the page and the top-most form element.

6. In the Left Margin field, enter the distance between the left edge of the page and the left-most form element.

7. Check the Layout is Preferred box to make this layout your preferred layout for this type of transaction.

8. Set the following colors for your form layout:

   - **Text Color** – Controls the color of the text in the form.
   - **Fill Color** – Sets the background color in the sections of the form.
   - **Label Text Color** – Controls the color of the text at the top of the main section of the form.
   - **Label Fill Color** – Determines the background color of the label for the main section of the form.
   - **Border Color** – Sets the border color for each section of the form.

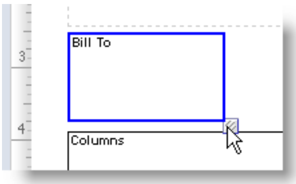
   You can click the color palette to choose a color or enter the hexadecimal code for the color you want to set.

   As you make changes to the colors, the changes are shown in the preview at the bottom of the page.

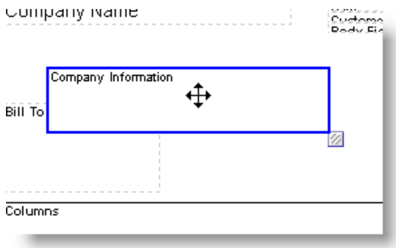
9. Click an element of the form that you want to customize.

   You can do any of the following to each form element:

   - Resize a section by clicking and dragging the handle in the lower right corner.



   - Click and drag an element to change the orientation of the PDF layout.



- Under the Element Label Style heading, determine the font and style of the element's label.

  Check the Use Label Coloring box if you want the color set in the Label Text Color field to be used for the label of a form section.

  Check the Split Horizontally box if you want the text beside the label. Keep this box clear if you want the text beneath the label.

- Under the Element Data Style heading, choose the font and style of the text in the form section.
- Check the Show Border box if you want the section to have a border.

10. Click Preview to see how this PDF layout will look when printed.

    For any element that continues on an additional page, the bottom of the element must be high enough on the page to allow line items to print below it. For example, a logo element on a PDF layout cannot be set to be the whole page.

11. Click Save.

Note: When the PDF opens and you click the Print button, the Print window opens. In the Page Handling section of the Print window, verify that Page Scaling is set to None, and the boxes beneath it are not checked. Automatic scaling can cause difficulties printing checks.

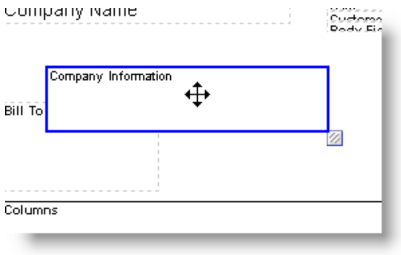
Configuring Borders and Placement

You can add borders and alter the orientation of elements of your PDF forms.

To add a border to a form:

1. Go to Setup > Customization > Forms > Transaction Form PDF Layouts.
2. Click Customize or Edit next to the form you want to change.
3. In the Form Editor click the element you want to add a border to.
4. Under the Additional Options heading, check the Show Border box.
5. Click Preview to see how this PDF layout will look when printed.
6. Click Save.

To change an element's orientation in the PDF layout, click and drag it to where you want it on the form.



Formatting Label Text

You can change the style and alignment of labels for PDF form elements.

To format element labels:

1. Go to Setup > Customization > Forms > Transaction Form PDF Layouts.
2. Click Customize or Edit next to the form you want to change.
3. In the Form Editor, click the element with the label you want to change.
4. Under the Element Label Style heading, set the font style and alignment options for this label.
5. Check the Use Label Coloring box if you want the color set in the Label Text Color field to be used for the label of a form section.
6. Check the Split Horizontally box if you want the text beside the label. Keep this box clear if you want the text beneath the label.
7. Click Preview to see how this PDF layout will look when printed.
8. Click Save.

Formatting Data Text

You can change the style and alignment of the text in the sections of PDF forms.

To format element text:

1. Go to Setup > Customization > Forms > Transaction Form PDF Layouts.
2. Click Customize or Edit next to the form you want to change.
3. In the Form Editor, click the element with the label you want to change.
4. Under the Element Data Style heading, set the font style and alignment options for this label.
5. Click Preview to see how this PDF layout will look when printed.

6. Click Save.

Defining Custom Elements

Custom elements are blocks of static or dynamic text that you can add to and position on PDF transaction layouts. You can define up to ten custom elements per layout.

Custom elements can only be defined for the following types of transaction form PDF layouts:

- packing slip
- picking ticket
- transaction

For example, you can follow these steps to add the Bill To element to a packing slip:

1. Go to Setup > Customization > Forms > Transaction Form PDF Layouts.
2. Click Customize next to your picking ticket form.
3. Click Add Custom Element.

   The Add Element window opens.
4. In the Element Name field, enter a name for the element, such as "Bill To".
5. In the Choose Element(s) list on the left side of the window, scroll down and click Bill To.
6. Click OK.
7. The Bill To box appears in the top left corner of the layout.
8. Click the Bill To element to drag it to the appropriate place on the form.
9. Complete additional changes to the element as necessary.
10. Click Save.

Now when you use this custom form, the form shows the Bill To address.

For each custom element, define the properties as described in the following table. Once an element is added, select Save & Edit. The custom element is then added to the other tabs and can be modified just like any other element on the form.

Property	Description
Element Name	This identifies the custom element on the Border & Placement, Label Text, and Data Text tabs, and may be printed in the label for the element field. This field is mandatory for each custom element where Enable is checked.

Property	Description
	Important: When a Label is created and the Custom Element is saved, the element is automatically available in the Borders and Placement tab with the coordinates defaulted to 0,0,0,0. In order to view the Label, you must edit these coordinates to display the label in the desired position.
Show Label	If checked, the Label for this custom element is printed on transaction forms using this layout, using the same formatting as other element labels as defined on the Label Text tab (see Formatting Label Text). The element does not appear on the Label Text tab if Show Label is not checked.
Element Text	Text entered here is displayed in the element field when printed. The text is formatted the same as other element text as defined on the Data Text tab (see Formatting Data Text).You can use NetSuite tags to dynamically retrieve and display text associated with a specific instance of a form field. To construct a tag enter the field name and enclose it in braces — {entity}. The fields available are the same body fields as those available in custom code. Refer to Custom Code Names for a complete list of available fields. **Note:** These tags are similar to those used on HTML Transaction Layouts, using {} instead of <> and dropping NL from the beginning of the tag. For Example: <NLENTITY> in HTML Layouts would be {entity} in PDF Custom Elements.
Wrap Text	If checked the Text in the element field wraps.

Using a Standard #10 Window Envelope With Transactions

If you use window envelopes, you may need to adjust the printing position of your PDF transaction forms to show the correct address through the window.

To adjust your standard layouts to fit in a standard #10 window envelope:

1. Go to Setup > Customization >Transaction Form PDF Layouts (or Transaction Form HTML Layouts).
2. Click Customize next to the kind of layout you want to change.
3. In the Name field, enter a name for your custom layout.
4. Click the Elements subtab.
5. Click Border & Placement.
6. In the Border & Placement section, locate the Bill To row.
7. In the Left column, enter 1.

   This moves the left edge of the field 1 inch from the left margin of the page.
8. In the Top column, enter 2.25.

   This moves the top edge of the field 2.25 inches from the top margin of the page.
9. In the Right column, enter 4.2.

This moves the right edge of the field 4.2 inches from the left margin.

10. In the Bottom column, enter 3.5.

    This moves the bottom edge of the field 3.5 inches from the top of the page.

11. Locate the Ship To row.

12. In the Left field, enter 4.55.

    This moves the left edge of the field 4.55 inches from the left margin of the page.

13. In the Top column, enter 2.25.

    This moves the top edge of the field 2.25 inches from the top margin of the page.

14. In the Right column, enter 8.

    This moves the right edge of the field 8 inches from the left margin.

15. In the Bottom column, enter 3.5.

    This moves the bottom edge of the field 3.5 inches from the top of the page.

16. When you have finished, click Submit.

    You are returned to the Custom Layouts list.

17. Check the box in the Preferred column next to your custom layout.

    This insures that your layout is applied to all forms of that type.

18. Click Save.

Repeat the steps above to adjust any transaction form layouts that you want to mail in standard #10 window envelopes. You can adjust any of your transaction form layouts for any size window envelope. Simply measure your envelope and the placement of the window to determine where to place the address.

Once you have the measurements for your envelope and determined where to place the address field, you need to convert the fraction measurements into decimal measurements to adjust the layout. The following table is a quick reference guide to converting fraction measurements into decimal measurements.

Fraction	Decimal	Fraction	Decimal
1/2	0.5	1/16	0.0625
1/4	0.25	3/16	0.1875
1/3	0.33	5/16	0.3125

Fraction	Decimal	Fraction	Decimal
3/4	0.75	7/16	0.4375
2/3	0.67	9/16	0.5625
1/8	0.125	11/16	0.6875
3/8	0.375	13/16	0.8125
5/8	0.625	15/16	0.9375
7/8	0.875		

Transaction Form HTML Layouts

Transaction form HTML layouts define the arrangement of fields on HTML transaction documents in NetSuite. Standard and Classic HTML layouts exist for all the standard form types other than shipping label and item label - these are assumed to be printed using PDF (see Transaction Form PDF Layouts).

You can customize transaction form HTML layouts by editing the templates on which they are based. The templates for each layout consist of two blocks of HTML:

- **Style Block** : this block begins with <STYLE> and ends with </STYLE>. The style block defines the styles used by the layout.
- **Body Block** : this block begins with <TABLE> and ends with </TABLE>. The body block defines the actual content of the HTML page.

Totalling is handled in the same way as for Transaction Form PDF Layouts (see Understanding Totalling on Transaction Form Layouts).

Note: Advanced PDF/HTML templates provide an alternate model for customizing printed and emailed records. They support more customization capabilities than transaction form layouts. The beta version of the Advanced PDF/HTML Templates feature currently supports cash refund, cash sale, credit memo, customer deposit, invoice, packing slip, payment voucher, picking ticket, purchase order, quote (estimate), return authorization, and sales order templates. For details, see Advanced PDF/HTML Templates (Beta).

Creating Custom Transaction Form HTML Layouts

To customize a transaction form HTML layout:

1. Click Setup > Customization > Transaction Form HTML Layouts.
2. Click Customize next to the layout you want to customize.

3. In the Label field, enter the name of your customized layout.

4. Check the Layout is Inactive box to remove this layout as an option in dropdown lists. You can always clear this box later and the layout is available again.

5. Check the Layout is Preferred box to make this layout your preferred layout for this type of transaction.

6. In the Templates subtab, edit the Style and Body blocks as desired.

   Embedded in the Body block are NetSuite tags that correspond to individual content elements. The content elements correspond very closely to the layout elements in the existing PDF layouts. Simple customization consists of editing the body or style templates but leaving all of the NetSuite tags in the body block.

   For example, to create a collection layout you could simple add a block of text at the top of the body layout.

7. If desired, click Elements and customize the HTML that corresponds to each element.

   In the elements tab, each element is listed with the corresponding element ID, the HTML code that will be used to display the label that corresponds to the element and the HTML code that will be used to display the data that corresponds to the element.

   When editing the HTML for elements, follow these guidelines:

   - Ensure that the HTML used correlates to any formats as defined in the body block. For example, if the element is included in a <TABLE> tag in the body block, the element itself should begin with a <TR>.
   - Elements always have data and may or may not have a label.
   - The actual label is represented by the tag <NLLABEL> tag which should be included in any customization of the label field.
   - The actual data is represented by the <NLDATA> tag which should be included in any customization of the data field.
   - There is also a <NLATTRIBUTES> tag which is used in the COLUMNS element to indicate where text formatting elements (such as alignment) are inserted.
   - The BODY, COLUMNS, and AGING elements are repeated to produce the HTML of the form.

Understanding Totalling on Transaction Form Layouts

Totalling fields on transaction forms display as follows:

- Discount, shipping, and tax totals are displayed on separate lines of the total.

- Each field is only included if it has a value.
- The discount, shipping item, and tax names are used.
- There are multiple tax lines in CA and VAT-enabled accounts.
- The total area is automatically expanded to fit the necessary information.

Amount Paid and Amount Remaining on Invoices

Amount paid and amount remaining are optionally included on transaction forms. They display as follows:

- The Amount Remaining field is now a footer field instead of a body field for purposes of printing.
- There is an additional Amount Paid footer field. This field is omitted if none of the invoice has been paid.
- If these fields are shown in a customized form, they are displayed in the total section underneath the invoice total.

Defining Preferred Forms

Setting preferred forms for your employees lets you control the entry and transaction forms employees use to enter data. This maintains consistency in your company and allows you to capture the information that is most important to your business.

In NetSuite, there are three places where form preferences can be defined.

- **On the Form:** When you create or edit a custom form, you can check the Form is Preferred box to set the current form as the default form for all users where role form preferences are NOT defined. In the Roles tab, you can also define the form as preferred for specific roles.
- **On the Custom Forms page:** On the custom forms page at Setup > Customization > [Forms], you can check the Preferred box for any form you want to set as the default form for all users where role form preferences are NOT defined.
- **On the Manage Roles page:** Administrators can define form preferences for specific roles at Setup > Users/Roles > Manage Roles > Edit [Role]. In addition to setting preferred forms for a role, you can also restrict access to the preferred form.

For custom records, you can choose a preferred form on the Forms subtab or set preferred forms for specific roles on the Permissions subtab. The preferred form you set on the Permissions subtab takes precedence over the preferred form set on the Forms subtab.

For example, you set **Custom Form A** as the preferred form on the Forms subtab. On the Permissions subtab, you set the default form for the Sales Rep role to **Custom Form B**. When a sales rep creates a new record, **Custom Form B** is selected by default.

To set the preferred form for a specific role:

1. Click Setup > Users/Roles > Manage Roles.
2. Click Edit next to the role you want to set form preferences for.

   Important: Standard roles can NOT be edited. If you wish to define new settings for one of these roles, click Customize next to the role, enter a name for your custom role and then set the desired custom settings. You will then need to assign this custom role to the appropriate users.

3. Click the Forms subtab.
4. Click the subtab for the section you want to set preferences for.
   - **Transaction** : set defaults for cash refund, cash sale, credit memo, invoice, estimate, opportunity, purchase order, return authorization and sales order transaction forms.
   - **Entity** : set defaults for contact, customer, lead, prospect, employee, project, partner and vendor entry forms. When you define preferences for Entities, you can set a separate preferred form for each Customer subtype in the corresponding fields at the top of the Entity subtab — Lead, Prospect and Subcustomer.
   - **CRM** : set defaults for phone call, campaign, case, event, solution and task entry forms.
   - **Item** : set defaults for inventory part, non-inventory part, service, other charge, description, subtotal, discount, markup, group, kit, assembly and payment item entry forms.
5. Set the form defaults. Note that for some roles, you can NOT modify all of these settings.
   - In the Enable column, clear any boxes for forms you do NOT want this role to have access to. (not available for Customer Center roles)
   - Check the box in the Preferred column next to any form you want to be set as the preferred form for this role.
   - If you want this form to be the only form available to this role, check the box in the Restricted column. (not available for Customer Center roles)

     The preferred form settings here override any settings on the Custom Forms page.

   Note the following about marking a transaction or CRM form Preferred for the Customer Center role:

- External forms, meaning forms with names appended with (External), can be marked Preferred for the Customer Center roles, but not for other roles.
- Forms that are not external cannot be marked as Preferred for Customer Center roles, so they are not listed on the Forms tab of Customer Center role records.

Warning: If you set preferred forms and do NOT restrict the forms, your employees can still change the form they use when entering transactions or records.

6. Click Save.

Customizing Address Forms

You can customize your address forms located on entity, location, and, for NetSuite OneWorld, subsidiary, records by changing the labels and hiding certain fields.

To customize your address forms:

1. Go to Setup > Customization > Address Form.
2. For each field, enter a customized label as desired.
3. Set the Show box for each **optional** field as desired.

   Fields that are not optional for an address label do not have this box. If checked, the show box ensures that the field is displayed on your address forms. If cleared, the field is NOT displayed on your address forms.

   Warning: If you use NetSuite OneWorld, be sure to check the Show box for the Country field. If you do not check this box, the Country field is not available on the New Subsidiary page.

4. Click Save.

Now the custom address label is displayed in popup windows when editing addresses and on entity and location records.

Adding Disclaimers to Transaction Forms

You can add a disclaimer to your standard forms on the Set Up Printing & Faxing page.

With a disclaimer added to the bottom of your forms, you can familiarize customers with your company's policies and procedures.

To add a disclaimer to standard transaction forms:

1. Go to Setup> Company > Preferences > Set Up Printing, Faxing & E-mail.

2. Click the Printing subtab.
3. Enter a disclaimer or other message in one or more of the disclaimer fields. Disclaimers can have up to 4000 characters.
    - **Sales Form Disclaimer** – The message entered here appears at the bottom of your standard sales forms.
    - **Purchase Form Disclaimer** – The message entered here appears at the bottom of your standard purchase forms.
    - **Statement Disclaimer** – The message entered here appears at the bottom of your standard statements.
    - **Packing Slip Message** – The message entered here appears at the bottom of your standard packing slips.
    - **Return Policy** – The message entered here appears at the bottom of your return forms.
    - **Remittance Slip Message** – The message entered here appears at the bottom of your remittance slips.
4. When you have finished, click Save.

In order for the disclaimers to appear, you must print your forms using HTML. To set this preference, go to Home > Set Preferences > Transactions tab. Check the box next to Print Using HTML, and click Save.

You can also add disclaimers to specific forms using transaction form customization. For more information on customizing transaction forms, read Custom Forms.

Customizing Multiple Page Transaction Forms

You can create a custom layout to customize how your multiple-page transaction forms look.

You can add page numbers and header information to subsequent pages of your PDF transaction forms.

To add page numbers to your PDF transaction forms:

1. Go to Setup > Customization > Transaction Forms.
2. Click Customize next to the kind of form you want to change.
3. Click Printing Fields.
4. Click Header.
5. Check the box in the Print/E-mail column next to Page Number.

Page number appear beginning on the second page of your transaction.

6. In the Label column you can change the way the numbers appear on your transactions.

   For example, if you wanted the numbers to appear 2/3, you would replace Page {1} of {2} in the label field with {1}/{2}. The entries {1} and {2} act as placeholders for your page number formatting.

7. When you have finished, click Save.

Your printed PDF transaction forms now include page numbers for multiple pages.

To add more header information to multiple page transaction forms:

1. Go to Setup > Customization > Transaction Form PDF Layouts (or Transaction Form HTML Layouts).
2. Click Customize next to the kind of layout you want to change.
3. On the Border & Placement subtab, check the boxes in the Show on Addl. Pages column for each field you want to appear on subsequent transaction pages.
4. When you have finished, click Save.

Your printed PDF transaction forms now include additional header information for multiple pages.

Setting Transaction Forms to Print in Landscape

You can set your transaction forms to print in landscape format on the Printing, Fax and Email Preferences page. When you print in landscape format, the long edge of the page is on top and the short edge is on the side.

By printing in landscape format, you can have wider columns on your transaction forms to include more detailed information for your customers.

To set transaction forms to print in landscape format, go to Setup > Company > Preferences > Printing, Fax and Email Preferences. On the Printing subtab, check the box next to Print Transaction Forms Landscape and click Save.

Note: In order to utilize this preference, you must use a standard transaction PDF layout. Use either a standard transaction form with a standard transaction PDF layout or use a custom transaction form associated with a standard PDF layout.

To print transaction forms, you can go to Transactions > Other > Print Checks and Forms to print transaction forms you have queued, or click Print on any transaction page as you save the transaction.

If you print using HTML, you can override the landscape format by changing your printer settings when you print the form.

Chapter 4 Custom Records

A custom record type is an entry form you create to collect information specific to the needs of your business. You can attach information from custom records to entities, items or transactions using custom fields.

For example, you may want to keep track of your company's computer and electronic equipment. Since a record specific for this purpose does not exist in NetSuite, create a custom record type titled Equipment. In the custom record, create custom fields to store information such as serial number, location, purchase date, date of last service and warranty information. Once the record type is created, data can be entered into individual records.

To enable the Custom Records feature, an administrator can go to Setup > Company > Enable Features > SuiteCloud.

To modify Custom Records, go to Setup > Customization > Record Types. You can do the following:

- Create a new custom record type.
- Edit an existing custom record
- View a list of records that have been created using each custom record type
- Create a new record for a selected custom record type
- Create a new search record for a selected custom record type

Creating Custom Record Types

To create a custom record type, you must first save the record type, and then add fields to the record.

Note: If you want to enable QuickViews for your custom record, this is done through form customization (for the custom record). See Configuring QuickViews for Custom Records for details.

To create a custom record type:

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types > New.
2. In the Name field, enter a name for the record type.
3. If you use custom code, enter a unique ID for this custom record.
4. Select the owner of this custom record type.

   Only the owner can modify this record type.
5. Enter a description of this record type.

6. Check the Include Name Field box to include a required Name field on each record entry.

   If you check this box, the Name field appears as the first field on the record and in the record list. If not, your records are automatically assigned a number based on the order in which they are entered.

7. Customize what to display for each record entry:

   - **Show ID** : displays the record entry ID that is automatically assigned by NetSuite to each new record entry.
   - **Show Creation Date** : Check the On Record box to display the creation date and time on each record entry. Check the On List box to display the creation date and time for each record entry in your list for this record type.
   - **Show Last Modified** : Check the On Record box to display the last modified date and time on each record entry. Check the On List box to display the last modified date and time for each record entry in your list for this record type.
   - **Show Owner** : Check the On Record box to display the record owner on each record entry. The record owner is the person who creates the record entry. Check the On List box to display the record owner for each record entry in your list for this record type. Check the Allow Change box to allow the record entry owner to be changed. If you allow the owner to be changed, you must also show the owner on the record entry. An Owner field is displayed on your record entries as a drop-down list of people with log in access to your NetSuite account. The Owner field on each record entry defaults to the current person entering the record.

8. Choose a permissions model by selecting from the Access Type dropdown:

   - Leave the default of Require Custom Record Entries Permission to indicate that only users logging in with a role with permission granted to the custom record type can access it.
   - Choose Use Permission List to grant access to the custom record type according to the permissions set up on the Permissions subtab of this page.
   - Choose No Permission Required to make access to the custom record type public.
   - For more information, see Setting Permissions for a Custom Record Type.

9. Clear the Allow UI Access check box to indicate that this record can only be accessed through SuiteScript. By default, this box is checked.

   When this box is cleared, users cannot access the custom record type from the NetSuite user interface. Also, the following custom record options are locked as disabled: Allow Quick Search, Allow Quick Add, and Include in Search Menu.

10. Check the Allow Attachments box to use a File Cabinet subtab to attach documents and images to your record entries.

11. Check the Show Notes box to add a Notes subtab to your child record entries.

    Notes are added to this subtab automatically when any change is made to individual records.

12. Check the Enable Mail Merge to enable mail merge capabilities for records of this type.

    For information about using mail merge, see the help topic *Working with Mail Merge*.

13. Check the Records are Ordered box to be able to edit the order your child records appear on each parent record.

    - If you do NOT check this box, child records display in alphabetical order in both View and Edit modes.
    - If you check this box, in View mode child records still display in alphabetical order. In Edit mode, child records initially display in the order in which they were entered and later, in the order in which you have set them through editing.

14. Check the Allow Child Record Editing box to allow records of this type to be edited directly when they display as child records in a sublist on a parent record. Note the following:

    - Enabling this option does not supersede users' role-based permissions. Only users who have permission to edit the record type can do child record editing when this option is enabled.
    - This option was formerly labeled Allow Inline Editing. Its label was changed as of Version 2012 Release 2, because it is not related to the Inline Editing feature or to the Inline Editing switch that displays on list pages when that feature is enabled. The name change also was intended to avoid confusion with the Enable Inline Editing option.

    Warning: Child records are never editable in parent records that have more than 10,000 child record lines, even when the Allow Child Record Editing option is enabled for the record type.

15. If you have checked the Allow Child Record Editing box, you can check the Allow Delete box, to allow users to delete records of this type when they display as child records in a sublist on a parent record. When this option is enabled, the following occurs:

    - When the PARENT record is in Edit mode and users click the Remove button on the CHILD record in the sublist, the entire child record type is deleted from NetSuite.
    - When the PARENT record is in View mode and users click the Remove button on the CHILD record in the sublist, the child record is simply removed from the sublist and the record type still remains in the system.

Warning: Child records cannot be deleted in parent records that have more than 10,000 child record lines, even when the Allow Delete option is enabled for the record type.

16. Check the Available Offline box if you use the Offline Client for Sales and want the custom record type available for download and synching to the offline client.

17. Check the Allow Quick Search box if you want to allow this record type to be searched using the Quick Search portlet on dashboards.

    This option is not available if the Allow UI Access box is cleared.

18. Check the Allow Quick Add box if you want to allow this record type to be added using the Quick Add portlet on dashboards. See the help topic *Using the Quick Add Portlet*.

    This option is not available if the Allow UI Access box is cleared.

19. Clear the Enable System Notes box if you do not want system notes to be created for changes to this record type. By default, system notes are enabled.

20. Clear the Include in Global Search box if you do not want keywords entered in the Search box in the upper right corner of the page to apply to this record type.

21. Clear the Include in Search Menu box if you do not want this record type to be available for searches in the UI.

    If the Allow UI Access box is cleared, this option is disabled and cannot be changed.

22. Review the setting for the Enable Optimistic Locking option.

    Enabling this option causes the system to check for conflicting updates whenever a user or script attempts to save updates to a custom record instance. If another user or script has saved updates to the same custom record instance while the first user or script was entering updates, an error is returned.

    Note: By default, this option is enabled for custom record types created in Version 2012 Release 2 and later, and disabled for custom record types created prior to that release. It is recommended that you enable this option.

23. If available, review the setting for the Enable Inline Editing option.

    - This option is only available if the Inline Editing feature has been enabled at Setup > Company > Enable Features, on the Company subtab, Data Management area.
    - This option is enabled by default.
    - When this option is enabled, an Inline Editing switch on list pages for this custom record type is set to ON, and users can update records quickly by changing data directly in each record's row. For more information, see the help topic *Using Inline Editing*.

- When this option is disabled, the Inline Editing switch is not available on list pages for this custom record type, and users must drill down from the list to each individual record in order to edit it.

24. Note that if you check the Inactive box, this record type no longer appears on the Record Types list unless you check the Show Inactives box at the bottom of the page. Also, you can no longer select this kind of record from any lists on entities, items or transactions.

25. Click Save.

Before you save the custom record type, the following subtabs are displayed for you to further define your custom records:

- **Subtabs** – Create and arrange subtabs for your custom record type. For information, see Adding Subtabs to a Custom Record.

Note: To save time, create and arrange subtabs for your custom records before defining your custom fields.

- **Sublists** – Add search results as sublists on your custom record type.

  For more information, see Applying Custom Sublists to Custom Record Types.

- **Icons** – Select the PNG sprite you want to use to represent this record type in the New Bar, Create New menu, Recent Records menu, Recent Records portlet, and QuickViews. You can choose from built-in icons or create your own custom icon.

  For more information, see Choosing an Icon for a Custom Record.

- **Numbering** – Specify the numbering format for custom records types. For information, see Numbering Custom Record Types.

- **Permissions** – Choose the roles you want to access custom record entry forms, choose a default form and restrict the forms available here. For information, see Setting Up a Permissions List for a Custom Record Type.

Important: You must choose Use Permission List in the Access Type dropdown for these permissions to take affect.

- **Links** – Create links that take you to the list of record entries for this custom record type and choose where to place the links. For information, see Creating Links to Custom Records.

- **Managers** – Define specific employees as managers of the current record type. This allows them to modify the custom record type. When defined as a manager, the employee is automatically granted custom record view permission. This allows them to see the list of custom record types but NOT drill down on them.

Note: If an employee has a role that includes the Custom Record Type permission, they have edit access to ALL custom records types. The Managers subtab allows you to grant permission for an employee to the current Record Type only.

- **Translation** – (when Multi-Language feature is enabled) Define translations for the custom record type name to be used when users change the language preference. See Adding Translations for Custom Records .

After you save a custom record type, the following subtabs are added:

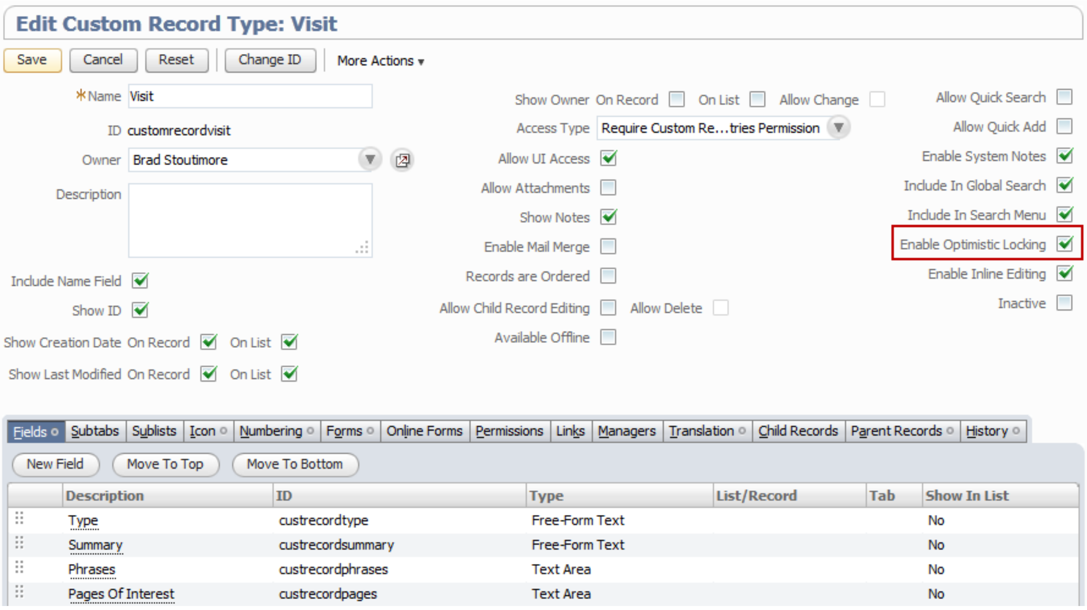
- **Fields** – Create and arrange the fields for your custom record type. For information, see Adding Fields to Custom Record Types.
- **Forms** – Customize and select a preferred entry form for your custom record type. For information, see Adding Custom Forms for a Record.
- **Online Forms** – Create and manage online forms for your custom record types.

  For more information on custom record online forms, see Adding Custom Online Forms for a Record.
- **Child Records** – If this record type is a parent record, its child records are listed here.
- **Parent Records** – If this record type is a child record, its parent records are listed here.

  For information on parent and child records, see Creating Built-in Child Records and Using Child Records.

Enabling Optimistic Locking for Custom Records

Prior to Version 2012 Release 2, NetSuite did not prevent multiple users or scripts from updating a single custom record instance at the same time. As of Version 2012 Release 2 and later, each custom record type has a Enable Optimistic Locking option that can be enabled to protect custom record data integrity.



Edit Custom Record Type: Visit

*Name: Visit
ID: customrecordvisit
Owner: Brad Stoutimore
Description:
Include Name Field ☑
Show ID ☑
Show Creation Date On Record ☑ On List ☑
Show Last Modified On Record ☑ On List ☑

Show Owner On Record □ On List □ Allow Change □
Access Type: Require Custom Re…tries Permission
Allow UI Access ☑
Allow Attachments □
Show Notes ☑
Enable Mail Merge □
Records are Ordered □
Allow Child Record Editing □ Allow Delete □
Available Offline □

Allow Quick Search □
Allow Quick Add □
Enable System Notes ☑
Include In Global Search ☑
Include In Search Menu ☑
Enable Optimistic Locking ☑
Enable Inline Editing ☑
Inactive □

Description	ID	Type	List/Record	Tab	Show In List
Type	custrecordtype	Free-Form Text			No
Summary	custrecordsummary	Free-Form Text			No
Phrases	custrecordphrases	Text Area			No
Pages Of Interest	custrecordpages	Text Area			No

Enabling this option causes the system to check for conflicting updates whenever a user or script attempts to save updates to an instance of this custom record type. If another user or script has saved updates to the same custom record instance while the first user or script was entering updates, the following message is returned:

```
“Unable to save record.  Record was changed by a different user.  Please reload and try again.”
```

The Enable Optimistic Locking option is enabled by default for all custom record types created as of Version 2012 Release 2 and later. For backwards compatibility, this option is disabled by default for custom record types created prior to Version 2012 Release 2. It is recommended that you enable this option, but that you first review any scripts that may be affected by this change and edit them as needed.

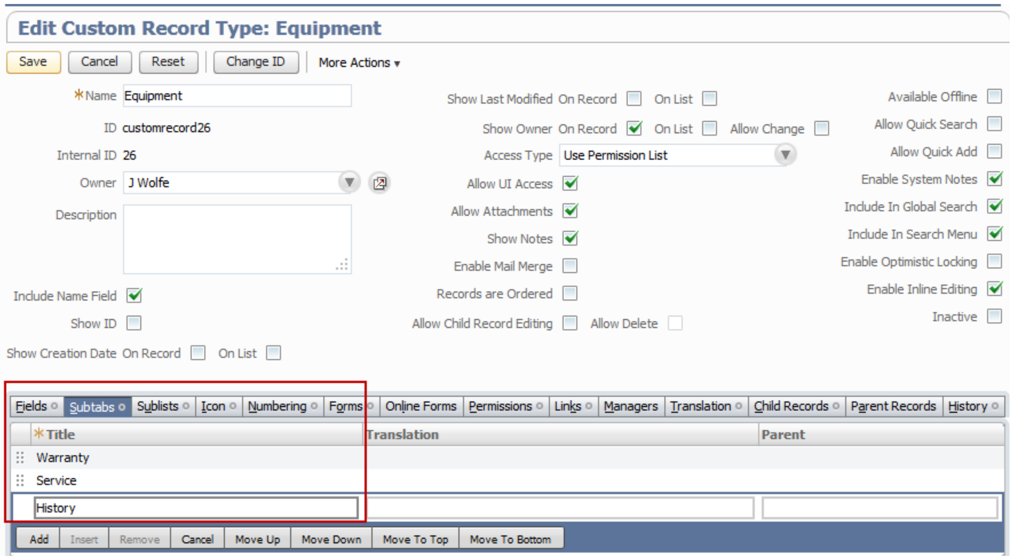
This new support for optimistic locking makes custom records' concurrency control consistent with the optimistic locking used generally for NetSuite standard records. Optimistic locking assumes that multiple concurrent transactions can usually complete without affecting each other, so data resources do not have to be locked while transactions are in process. Instead, a check for conflicts occurs before each transaction is committed. For more information about optimistic locking, you can review a related article athttp://en.wikipedia.org/wiki/Optimistic_concurrency_control.

Adding Subtabs to a Custom Record

The Subtabs subtab of a custom record form allows you to add custom subtabs to your record to better organize fields. To save time when creating custom fields, create any desired custom subtabs first.

To add subtabs:

1. In the title column of the Subtabs subtab, enter the name of your new subtab for this record type.



Edit Custom Record Type: Equipment

*Name Equipment
ID customrecord26
Internal ID 26
Owner J Wolfe
Description
Include Name Field ☑
Show ID □
Show Creation Date On Record □ On List □

Show Last Modified On Record □ On List □
Show Owner On Record ☑ On List □ Allow Change □
Access Type Use Permission List
Allow UI Access ☑
Allow Attachments ☑
Show Notes ☑
Enable Mail Merge □
Records are Ordered □
Allow Child Record Editing □ Allow Delete □

Available Offline □
Allow Quick Search □
Allow Quick Add □
Enable System Notes ☑
Include In Global Search ☑
Include In Search Menu ☑
Enable Optimistic Locking □
Enable Inline Editing ☑
Inactive □

*Title	Translation	Parent
Warranty		
Service		
History		

2. If desired, designate this subtab as a child of an existing subtab.

   In the Parent field, select an existing subtab from the dropdown list. This list consists of any custom subtabs already saved for the current custom record type as well as any custom subtabs that you have defined for that record type.

   Note: Since the parent field is populated with subtabs created for the current custom record type, you can only define the subtab as a child if the custom record has already been saved with pre-defined subtabs.

3. Click Add.
4. Repeat these steps for each subtab you want to add.
5. Click Save.

Important: You must save the record after you have created subtabs in order for the subtabs to be available to place fields on. You can place fields on your new subtabs by editing the field record from the Fields subtab.

6. Rearrange the fields as desired (if you have not yet created fields for your custom record, go on to defining fields. You can assign a subtab while creating custom fields.

   Select a line and drag and drop it to the desired position, or use the Move buttons.

7. To edit the name of a subtab, click the line for that subtab.
8. Click Save.

Numbering Custom Record Types

You can have numbers automatically assigned to your custom records for easier tracking and designation.

Warning: Once you enable auto-numbering, numbers cannot be removed from records. Disabling auto-numbering in the future only prevents future records from being numbered.

To assign automatic numbers to a custom record type:

1. Go to Setup > Customization > Lists, Records, & Fields > Custom Record Types.
2. Click Edit next to the record type you want to have auto-numbered.
3. Click the Numbering subtab.
4. Check the Enable box to turn on automatic numbering.
5. In the Prefix field, enter any numbers or letters you want added before each automatically-generated number.
6. In the Suffix field, enter any numbers or letters you want added after each automatically-generated number.
7. In the Minimum Digits field, enter the number or total digits you want as the minimum for auto-generated numbers.

   This adds placeholder zeros to numbers that do not have the number of minimum digits you set.

   For example, enter **4** to have the first number added as **0001**.

8. In the Initial Number field, enter the number you want to use to start automatic numbering.

   For example, if you enter **20**, the first record created is numbered **20**, and no records are numbered below **20**. The next record numbered would be **21**.

9. Check the Allow Override box if you want to be able to enter a custom number when you edit the record.

10. Check the Update box to number records that have already been created.

    If you have already enabled automatic numbering, checking this box does not change any existing numbers, as those numbers have already been referenced in the system.

11. Click Save.

Important: Be aware that, as of Version 2010 Release 2, handling of the Name criteria for custom records searches has been modified. These changes particulary affect auto-numbered custom record types. Previously, the value specified for the Name criteria was compared to the Name field values of custom records, to IDs containing prefixes, and to IDs with prefixes. Now, the Name criteria value is compared only to Name field values, providing a more clearly delineated set of results. To search for ID values, users can use the ID criteria.

Adding Fields to Custom Record Types

The Fields subtab allows you to add fields to your record. You can add as many fields as necessary to your custom record.

Important: This tab is not available until the record type has been saved.

To add fields to your custom record:

1. Click New Field to create a new field for this record type.
   For step-by-step instructions for creating a new field, refer to Creating a Custom Field.

2. Rearrange the fields as desired.
   Select a line and drag and drop it to the desired position, or click Move To Top or Move To Bottom. If you have placed the fields on subtabs, moving a field here changes its position in relation only to the other fields on the same subtab.

3. To edit a field, click the name of the field in the Description column.

4. Click Save.

Note: If you want this field to be available for data entry in the Quick Add portlet, check the Allow Quick Add box. By default, this option is not enabled. See the help topic *Using the Quick Add Portlet*.

Adding Custom Forms for a Record

The Forms subtab allows you to create custom entry forms for your records. You can create an unlimited number of entry forms for your record type. Once you have created custom entry forms, you can select which form is the preferred entry form.

By default, at least one form is automatically assigned to the custom record type. This custom record form can be customized as desired.

To create custom forms for the record:

1. In the Forms subtab, click Customize or Edit next to the entry form you want to customize.

   Selecting Customize allows you to create a new custom form based on the selected one. Selecting Edit allows you to edit the selected form.
2. Customize the form as desired and then save it.

   For step-by-step instructions on how to customize a form, see Creating Custom Entry and Transaction Forms.
3. In the Preferred column, click the button next to the entry form you want to be set as the preferred form for this record type.

   Note: If you have also set preferred forms on the Permissions subtab, the preferred form set on the Permissions subtab takes precedence over the role set on the Forms subtab.

   For example, you set Custom Form A as the preferred form on the Forms subtab. On the Permissions subtab, you set the default form for the Sales Rep role to Custom Form B. When a sales rep creates a new record, Custom Form B is selected by default.
4. Click Save.

Adding Custom Online Forms for a Record

The Online Forms subtab allows you to create online entry forms to capture information and create new records from outside of your NetSuite account.

To add custom online forms:

In the Online Forms subtab, do one of the following:

- Click New Online Form to create an online form for this record type.
- Click New Online HTML Form to create an online form based on an HTML template.
- To edit an existing online form, click the name of the form.

For a step-by-step description of how to create online forms, see Online Custom Record Forms.

Setting Permissions for a Custom Record Type

To manage access to custom record type data, you can:

- Define the Permissions Model to use permissions on role records, use permissions defined on the custom record itself, or provide public access to the custom record type.
- Prevent Access through the User Interface so that users cannot access custom record type data through the NetSuite user interface.

Define the Permissions Model

You can use the Access Type dropdown on a custom record type page to define a permissions model for a custom record type. This model can be based on: custom record entries permissions defined on role records, permissions defined on the Permissions subtab of a custom record type, or no permissions (meaning access to the custom record type is public).

Note: As of Version 2012 Release 1, the Access Type dropdown replaces the Use Permissions check box that was available in prior releases.

The Access Type dropdown includes the following options:

- **Require Custom Record Entries Permission**
  - This option is the default.
  - Custom record types created prior to Version 2012 Release 1 that did not have the Use Permissions box checked have this option set.
  - This option indicates that only users logging in with a role with permission granted to the custom record type can access it. This permission can be set on the Lists subtab of the Permissions subtab on each Role page. See the help topic *Customizing or Creating NetSuite Roles*.

    Note that this limitation does not apply to the owner of the custom record type; the owner always has full access in any role.
- **Use Permission List**
  - Custom record types created prior to Version 2012 Release 1 that had the Use Permissions box checked have this option set.
  - This option indicates the users logging in with a role with permissions defined on Permissions subtab of the custom record type can access it.

    Note that this limitation does not apply to the owner of the custom record type; the owner always has full access in any role.
  - For details about creating a permission list, see Setting Up a Permissions List for a Custom Record Type.
- **No Permission Required**

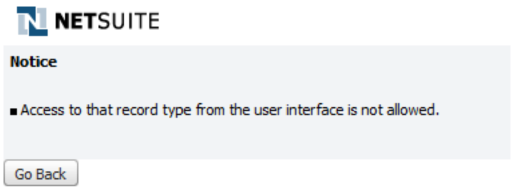
- This option indicates that all users can access this custom record type; it is considered public.
- With this option, any user can modify the record if they get access to its entry form, which they could do through a URL, even if they do not have access to a link to the form.
- You can use this option for records that must be accessible to scripts, but that you do not want users to access. After testing, clear the Allow UI Access box for the record. With this combination of settings, there are no restrictions on programmatic access to the record type, but no access through the user interface. See Prevent Access through the User Interface
- Prior to Version 2012 Release 1, this option was not available for custom record types.

Prevent Access through the User Interface

You can clear the Allow UI Access box for a custom record type, to indicate that it can only be accessed programmatically, for example, through SuiteScript or Web services. By default, the Allow UI Access box is checked.

When this box is cleared:

- Users cannot access the custom record type from the NetSuite user interface.
- The following error is returned when a user attempts to list, search, view, edit, or create a record of this type in the user interface:



Notice

■ Access to that record type from the user interface is not allowed.

- The following custom record options are locked as disabled: Allow Quick Search, Allow Quick Add, and Include in Search Menu.

Important: You need to take additional steps to control access to custom record data through searches. See Limiting Search Access to Custom Records.

Setting Up a Permissions List for a Custom Record Type

You can manage access to a custom record type's data by setting up a permissions list on the Permissions subtab of the record type page.

The role-based restrictions you set on this subtab are also available on the record for each role. Changes made on role records related to this custom record's permissions are reflected here.

Important: For the permission settings in the Permissions subtab to take affect, the Use Permissions option must be selected for Access Type. Be aware that these permission settings are not used to restrict search access to custom record data. You can limit searches' access to custom record data on a per-field or per-search basis. See Limiting Search Access to Custom Records.

To set up a permissions list for a record type:

1. On a custom record type page, click the Permissions subtab.
2. In the Role column, select the role you want to have access to custom record entries of this type.
3. In the Level column, select a level of access for this role. Available options include:
    - **None** : People with this role cannot use custom records of this type at all.
    - **View** : People with this role can view custom records of this type.
    - **Create** : People with this role can view and create custom records of this type.
    - **Edit** : People with this role can view, create and edit custom records of this type.
    - **Full** : People with this role can view, create, edit and delete records of this type.
4. Select a value in the Restrict column to limit the access of users with the selected role to custom records of this type:
    - Select Viewing and Editing to restrict users with this role to viewing or editing only the records of this type that they or their subordinates created.
    - Select Editing Only to restrict users with this role to editing only the records of this type that they or their subordinates created. They can view all records of this type.
    - Leave this column blank to allow users with this role to view and edit all records of this type.
5. In the Default Form column, select a default entry form for this role to use when entering records of this type.

    Note: The default form you set here for a role takes precedence over the preferred form setting on the Forms subtab.

    For example, you set **Custom Record Form A** as the preferred form on the Forms subtab. On the Permissions subtab, you set the default form for the Sales Rep role to **Custom Record Form B**. When a sales rep creates a new record, **Custom Form Record B** is selected by default.
6. Check the box in the Restrict Form column to make the default form the only entry form available to this role.

7. In the Search Form dropdown, select a custom search form to be used for searches of this record type, if one is available.

8. In the Search Results dropdown, select a custom search to be used to limit results for searches of this record type, if one is available.

9. In the List View, Dashboard View, and Sublist View dropdowns, select a custom search to be used to limit displayed results for these lists, if one is available. Check a box in one of the Restricted columns to make the selected view the only one available to this role.

10. Click Add.

11. Repeat these steps for each role you want to give access to.

12. Click Save.

Limiting Search Access to Custom Records

The restrictions to custom record type access set up on the Permissions subtab do not apply to custom record data access by searches. If you want to prevent searches from returning custom record type data, you have the option of restricting search access to specific custom record type fields on a field by field basis, and restricting the audience for custom record type saved searches on a search by search basis.

To limit search and reporting access to a custom field:

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.

2. Click a custom record type.

3. On the Fields subtab for the custom record, click a field.

4. On the Access subtab for the field, set the Default Level for Search/Reporting to None, to prevent any searches or reports from returning data for this field.

   You also can prevent searches or reports run by users with specific roles from returning data for this field, by setting the Default Level for Search/Reporting to None for specific roles on the Role subtab.

   For more information, see Restricting Access to Custom Fields.

To limit access for a custom record saved search:

1. Go to Lists > Search > Saved Searches.

2. Click the saved search that you want to restrict.

3. On the search page, click the Audience subtab and make changes to the users who have access as desired.

For example, you can clear the Public box or the Select All box for roles, and instead select only the specific users or roles that you want to have access to the custom record.

For more information, see the help topic *Defining Audiences for Saved Searches*.

Applying Role-Based Restrictions to Custom Records

On a role record, you can restrict the access of users with that role to standard records, based on these records' values for department, class, location, employee, and subsidiary (OneWorld) fields. For example, you could set an employee-based restriction for the Sales Manager role so that those users only see records where they or their subordinates are the sales rep.

For details about setting these restrictions, see the help topic *Set Employee Restrictions*, *Set Department Restrictions Set Department Restrictions*, *Set Class Restrictions Set Class Restrictions*, *Set Location Restrictions Set Location Restrictions*, and *Restrict Access to Subsidiaries (OneWorld Only) Restrict Access to Subsidiaries in the help topic Customizing or Creating NetSuite Roles Customizing or Creating NetSuite Roles*.

You can apply the restrictions set on role records for a particular category (department, class, location, employee, or subsidiary) to a custom record, by checking the Apply Role Restrictions box for a list/record custom field that stores values in one of these categories. For example, if you check this box for an Employee list/record custom field, the employee-based restriction set on the Sales Manager role record is applied to this custom record, so that those users only see custom records where they or their subordinates are the value for the custom field.

To apply role-based access restrictions to a custom record:

If the class, department, location, or subsidiary field does not yet exist:

1. On the Fields subtab of a custom record definition page, click New Field.
2. On the new field definition page:
    1. Enter a name for the field.
    2. In the Type dropdown, select List/Record.
    3. In the List/Record dropdown, select Class, Department, Location, Employee, or Subsidiary.
    4. Check the Apply Role Restrictions box.
    5. Complete other settings for the custom field as desired, and click Save. See Adding Fields to Custom Record Types.

If the field already exists:

1. On the Fields subtab of a custom record definition page, click the field name.

2. On the field definition page, check the Apply Role Restrictions box and click Save.

Creating Links to Custom Records

The Links subtab allows you to create links throughout your account to access your custom records. When determining where to place links for the records, it is important to consider all roles that will be using the record type and how they will be using it to determine the most logical place for the links.

Important: Even after you create a link to a custom record type, this link does not display for users that do not have permission to access that custom record type.

To add a link to a custom record:

1. In the Center column of the Links subtab, select the center you want the link to be visible in.

   If desired, you can add links to this custom record in each center.

2. In the Section column, select the tab you want the link to display on.

   The tabs available vary depending on the selected center. If a center has NOT been selected, no tabs are listed.

3. In the Category column, select a standard, built-in NetSuite category, or select a custom category you have already created. (See Creating Center Categories for information on creating custom categories.)

4. In the Label column, enter a name for this link.

   If you do not enter a label, the label name defaults to the name of the custom record type.

5. (Optional) In the Translation column, type the translated name for the link.

6. (Optional) In the Insert Before column, choose where you want to place the link. If you do not provide a value in the Insert Before column, the link will appear below the last link in your chosen category.

7. Click Add.

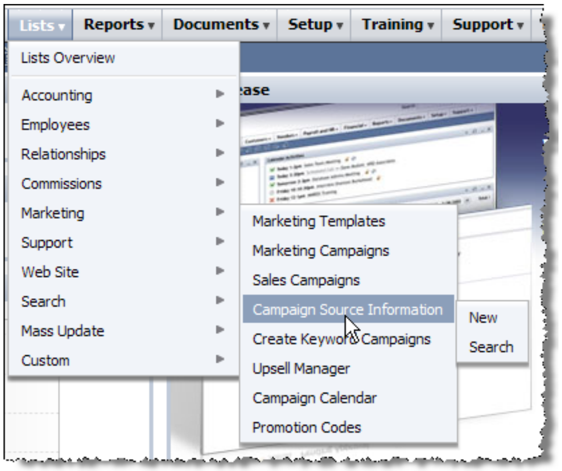
8. Repeat these steps for each link you want to add.

   If you want to edit a link, click the line for that link.

9. Click Save.

The following figure shows the custom **Campaign Source Information** link, which allows users to create new instances of a custom “campaign source” record type. Users can also search all existing records of this type.

This link appears in the standard, built-in Marketing category. The Marketing category appears on the Lists tab.



Choosing an Icon for a Custom Record

You can choose an icon to represent a custom record in the NetSuite user interface. Users can quickly identify the record type through this visual cue. These icons are used in the following places:

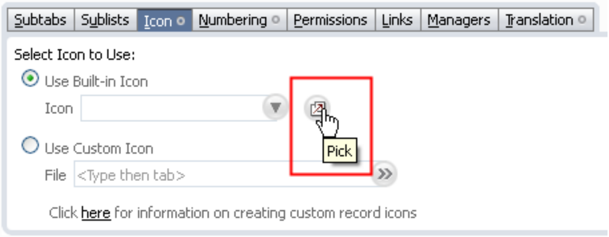
- Create New bar
- Create New menu on records and New column on list pages and list portlets
- Recent Records menu
- Recent Records portlet
- QuickViews

You can choose from 70 prebuilt icons or create your own custom icon.

To associate an icon with a custom record:

1. On the Icons subtab of the custom record, choose the Use Built-In Icon or the Use Custom Icon option.
2. If you choose Use Built-In Icon, you can select an icon from the dropdown.

   You can click the picker to view available icons and click the one you want to use.



3. If you choose Use Custom Icon, select its file. For requirements for custom icons and instructions for creating them, see Creating Icons for Custom Records.

4. Click Save.

Creating Icons for Custom Records

You can create custom icons to represent custom records in the NetSuite user interface.

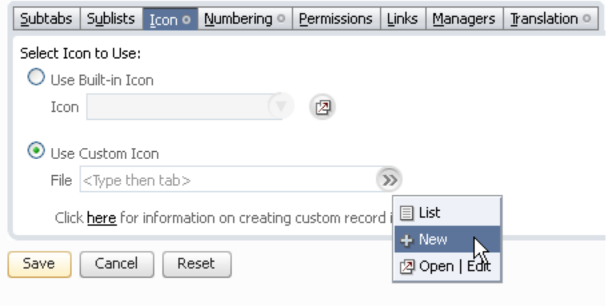
To use a custom record icon:

1. Create an image file for your icon.

2. Open the custom record for which you want to add the icon.

3. On the Icons subtab, choose the Use Custom Icon option.

4. Click New, next to the File field.



5. In the File dialog, choose your icon image.

6. Click Save in the File dialog, then save the custom record.

Important: Custom icons must meet the specific requirements detailed below. It is recommended that you read these requirements thoroughly and give yourself enough time to test your icon. You will need at least a basic understanding of an image-editing application like Adobe Photoshop or another application that allows you to edit artwork and save files as PNGs with transparencies.

The Four Icon Versions

A custom record icon is made up of a set of four slightly different versions of your icon, ranging from grayscale to full-color. These four versions are required so that the icon can be displayed with maximum clarity and contrast on a variety of different backgrounds.

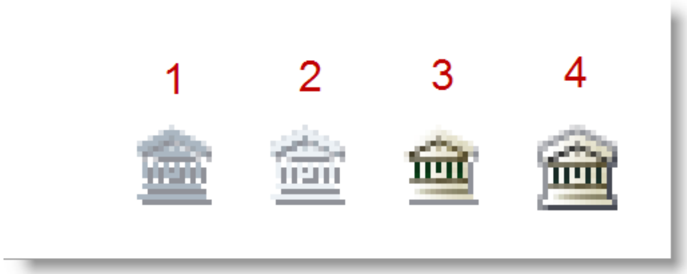


Image 1: Grayscale icon for dark backgrounds. This icon is a simple “ outline ” version of the full-color icon. It is used in the dashboard's New Bar when a dark color theme is used. Since the default color theme is dark-colored, this is the version that is most often seen.

Image 2: Grayscale icon for light backgrounds. This is another simple outline version of the icon, used in the dashboard's New Bar when a light color theme is used. This version is not used as often since few color themes have a light background color for the New Bar.

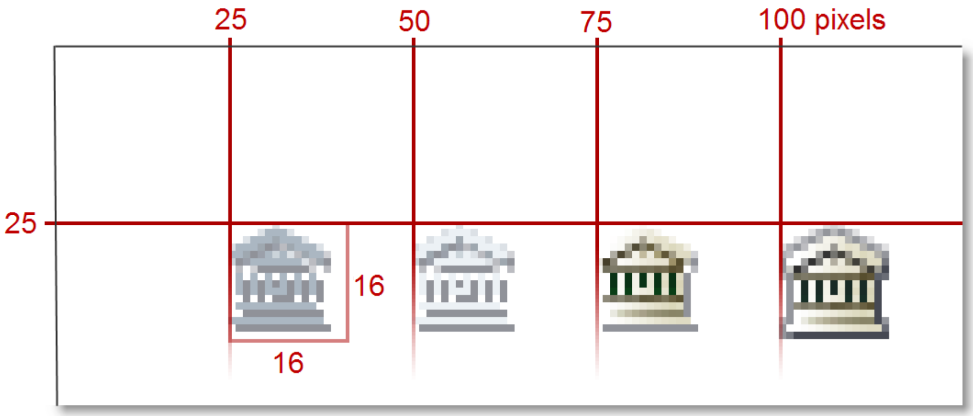
Image 3: Color icon for dark backgrounds. This is a full-color version of the icon, optimized for use on dark backgrounds. It is used in the New Bar as the color icon that replaces the grayscale version when a cursor is nearby.

Image 4: Color icon for light backgrounds. This is another full-color version of the icon, optimized for use on light backgrounds. It is used in menus, which always have a light background — such as the Create New menu and the Recent Records menu.

The Icon Sprite Image File

The four icon versions reside, side-by-side, in a single image file known as a sprite. Using coordinates and other data stored in CSS, NetSuite will display the proper icon version needed while cropping out and hiding the rest of the sprite image. Therefore, in order for your icons to display properly, they must be an exact size and at an exact location within the image file.

Each icon must be no larger than 16 pixels by 16 pixels. Any artwork that goes beyond the 16x16 boundary will not be displayed.



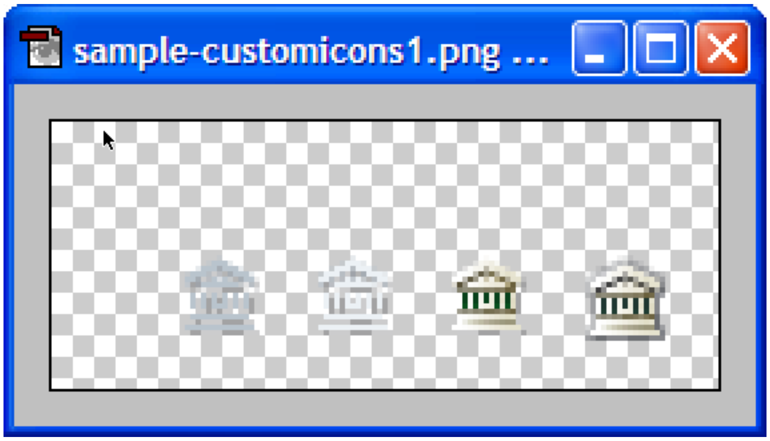
Each icon version must be located at the following locations within the image file, as measured from the upper-left corner of the image:

Icon Version	x Coordinate	y Coordinate
Grayscale icon for dark backgrounds	25	25
Grayscale icon for light backgrounds	50	25
Color icon for dark backgrounds	75	25
Color icon for light backgrounds	100	25

Most image-editing applications have guides that you can set up to help you keep track of these spacing requirements. You can also download the sample icon file pack at the bottom of this topic, which contains different template files that will help keep your icons in order.

File Format

Save the icon image file as a 24-bit PNG file with an 8-bit alpha channel for the transparent background. Your image editing application may refer to this file format as **PNG-24**. We strongly recommend that you always use the PNG-24 format when saving your icon image file. Using GIF or JPEG formats is discouraged due to their limited (or lack of) transparent backgrounds. Other image file formats are not accepted.

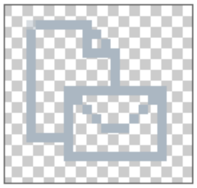


Constructing Your Icon File, Step-by-Step

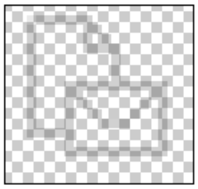
1. Once you have your icon, whether you created it yourself or obtained it from a royalty-free source, the first step in creating your final icon image file is to create an outline of it. This outline drawing will be used to construct the two grayscale icon versions.

The outline version of your icon should be simple, but it should also have sufficient detail so that it can be recognized when it appears on your dashboard's New Bar. Start by using a 1-pixel pencil tool in your image editing application, and trace an outline of your icon. In addition to the outline, trace some of your icon's internal components so that it will be easier to recognize.

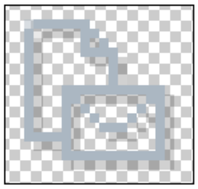
2. Take the outline tracing of your icon and follow these directions to create your grayscale icon for dark backgrounds:



Create copy #1, fill with #B9C4CD



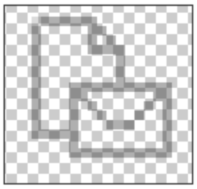
Create copy #2, fill with #000000, set to 30% opacity

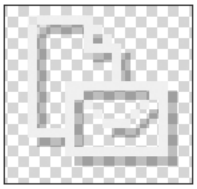


Combine copy #1 & copy #2. Offset copy #2 by 1 pixel down and 1 pixel right

3. Take the outline tracing of your icon and follow these directions to create your grayscale icon for light backgrounds:

Create copy #3, fill with #F0F0F0


Create copy #4, fill with #333333, set to 50% opacity


Combine copy #3 & copy #4. Offset copy #4 by 1 pixel down and 1 pixel right

4. After the two grayscale versions are complete, create the color versions. This involves placing the color icon that you already have on a variety of background colors to check its appearance. Try the following colors:

   - #5A759C - default color for the New Bar
   - #F1F4F9 - default color for the Recent Records menu
   - #EBECEF - default color for the Create New menu

   Try other colors, such as light and dark reds and then test the colors of your NetSuite color theme. Once you have an understanding of how your color icon looks on different background colors, you will be able to create the last version of your icon.

   If you feel that your color icon looks fine regardless of background color, make a copy of it to be placed in the fourth slot of your PNG image.

5. Place the four versions of your icon in a single PNG image, using the proper spacing described above. Save your icon image file as a PNG-24, and upload it to your NetSuite account.

Sample Icons

Click here to download our sample set of custom icons (ZIP archive, 14KB), designed to help you get started.

Adding Translations for Custom Records

If the Multi-Language feature is enabled in your account, you can translate the name of a custom record, its custom subtab titles, and its custom sublist labels, so that they match the language of the NetSuite user interface. You can also translate the names of instances of a custom record. For details, see the following:

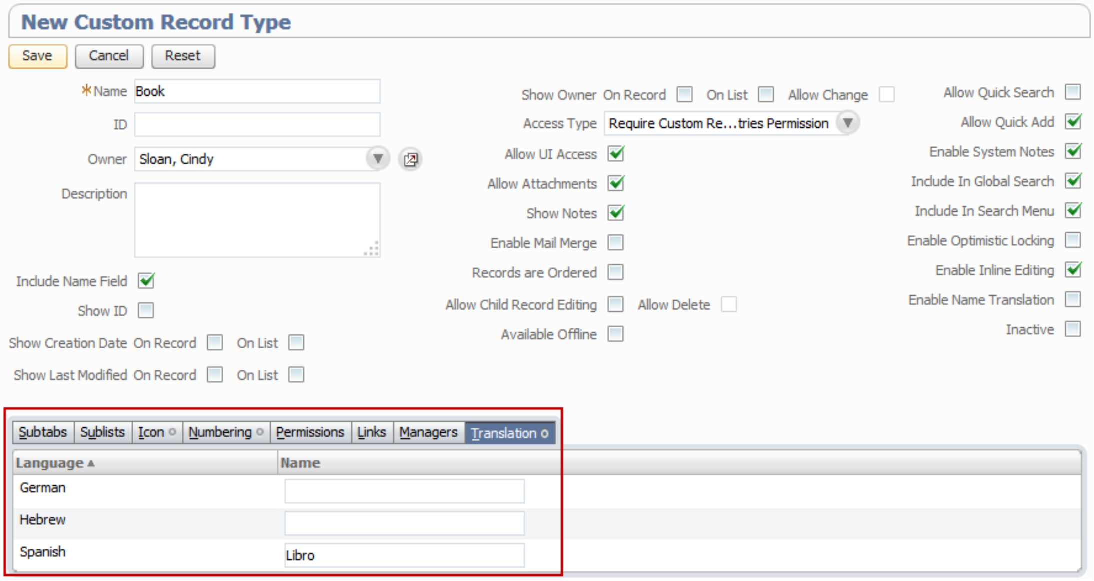
- Translating a Custom Record Name
- Translating Custom Record Subtab Titles

- Translating Custom Record Sublist Labels
- Translating Custom Record Instance Names

Important: Before you can add these translations, you need to select translation languages at Setup > Company > General Preferences, on the Languages subtab. This subtab lists both system-supported languages that can be used for the NetSuite user interface (and are available at Home > Set Preferences), and additional languages that can used for Web site translations only (and are not available at Home > Set Preferences). You should only enter translations for system-supported languages, because these are the only languages that can be displayed in the user interface. For details, see the help topic *Setting Up Multiple Languages*.

Translating a Custom Record Name

You can define translations for a custom record type name on the Translation subtab of the custom record page:



New Custom Record Type

*Name Book
ID
Owner Sloan, Cindy
Description
Include Name Field ☑
Show ID □
Show Creation Date On Record □ On List □
Show Last Modified On Record □ On List □

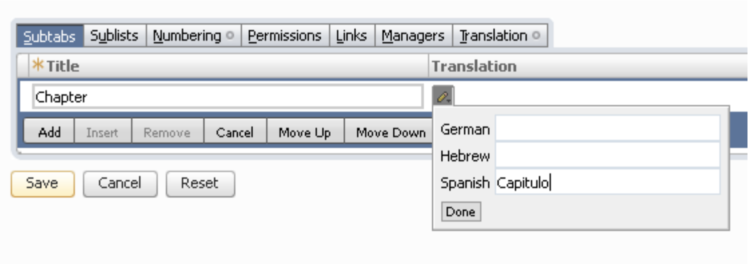
Show Owner On Record □ On List □ Allow Change □
Access Type Require Custom Re...tries Permission
Allow UI Access ☑
Allow Attachments ☑
Show Notes ☑
Enable Mail Merge □
Records are Ordered □
Allow Child Record Editing □ Allow Delete □
Available Offline □

Allow Quick Search □
Allow Quick Add ☑
Enable System Notes ☑
Include In Global Search ☑
Include In Search Menu ☑
Enable Optimistic Locking □
Enable Inline Editing ☑
Enable Name Translation □
Inactive □

Language ▲	Name
German	
Hebrew	
Spanish	Libro

Translating Custom Record Subtab Titles

You can define translations for a custom record type's subtab titles, on the Subtabs subtab of the custom record page:



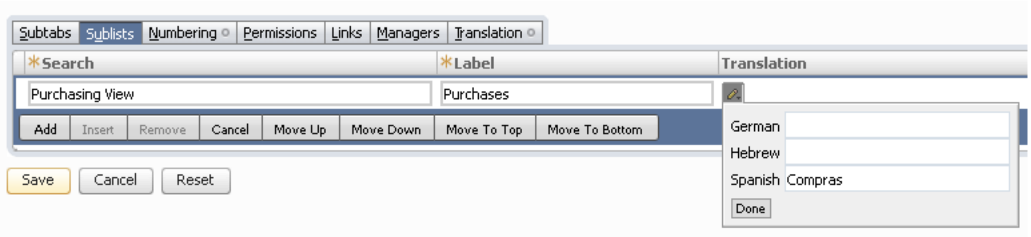
*Title: Chapter

Translation:
German
Hebrew
Spanish: Capitulo

For details about custom record subtabs, see Adding Subtabs to a Custom Record.

Translating Custom Record Sublist Labels

You can define translations for custom record type sublist labels, on the Sublists subtab of the custom record page:



*Search: Purchasing View

*Label: Purchases

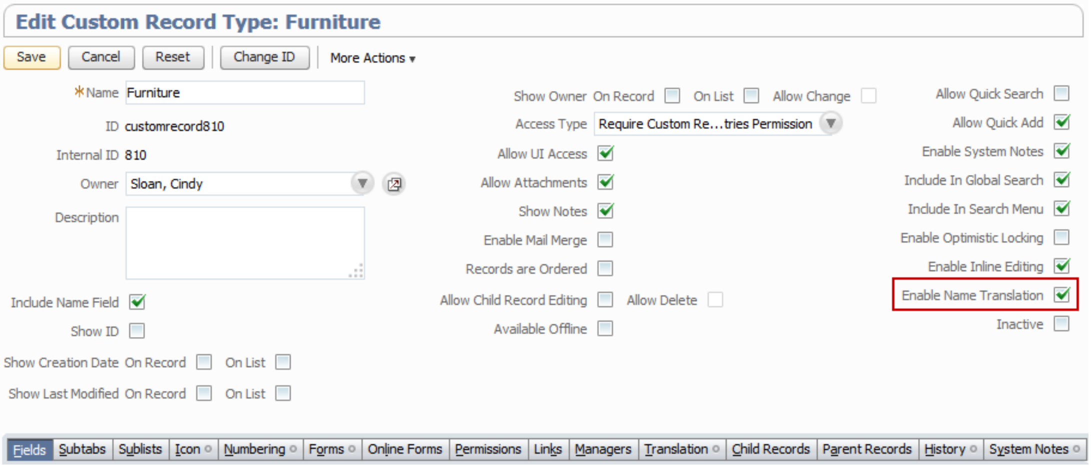
Translation:
German
Hebrew
Spanish: Compras

Custom sublists present information related to the record you are viewing, based on results from a selected saved search of the record type or a related record type. You can apply a custom sublist to a custom record, so it displays on forms for that record. For more details, see Applying Custom Sublists to Custom Record Types.

Translating Custom Record Instance Names

In addition to providing translations for the name of the custom record type itself, you can provide translations for individual instances of that record type. For example, in addition to providing the Spanish translation “Muebles” for a custom record type named Furniture, you also can provide translations for the individual instances of that record type, such as chair, table, bed.

To allow translation of custom record instance names, check the new Enable Name Translation option for the custom record type. This option is disabled by default.



Edit Custom Record Type: Furniture

Save | Cancel | Reset | Change ID | More Actions ▾

*Name Furniture

ID customrecord810

Internal ID 810

Owner Sloan, Cindy

Description

Include Name Field ☑

Show ID □

Show Creation Date On Record □ On List □

Show Last Modified On Record □ On List □

Show Owner On Record □ On List □ Allow Change □

Access Type Require Custom Re…tries Permission

Allow UI Access ☑

Allow Attachments ☑

Show Notes ☑

Enable Mail Merge □

Records are Ordered □

Allow Child Record Editing □ Allow Delete □

Available Offline □

Allow Quick Search □

Allow Quick Add ☑

Enable System Notes ☑

Include In Global Search ☑

Include In Search Menu ☑

Enable Optimistic Locking □

Enable Inline Editing ☑

Enable Name Translation ☑

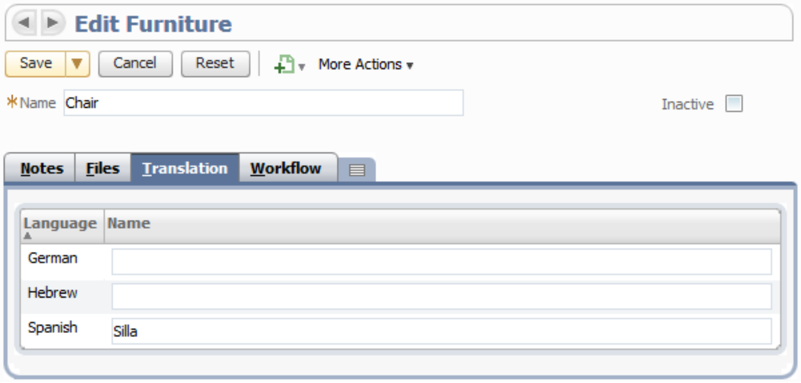
Inactive □

Fields | Subtabs | Sublists | Icon | Numbering | Forms | Online Forms | Permissions | Links | Managers | Translation | Child Records | Parent Records | History | System Notes

You can enable the Enable Name Translation option for a custom record type if ALL of the following are true:

- The Multi-Language feature is enabled for the account.
- The Include Name Field option is enabled for the custom record type.
- Numbering is not enabled for the custom record type. (The Enable box on the Numbering subtab is not checked.)

When the Enable Name Translation box is checked for a custom record type, each custom record instance has a Translation subtab.



Edit Furniture

Save | Cancel | Reset | More Actions ▾

*Name Chair

Inactive □

Language	Name
German	
Hebrew	
Spanish	Silla

Online Custom Record Forms

An online custom record form is used to receive information from customers on your Web site. Information received from Online Forms automatically creates or updates records in your NetSuite account.

For example, you might link to a warranty custom record form from your Web site. You require your customers to include their name, the end date of the warranty, the item purchased, serial number, the start date and the type of warranty.

You can customize the appearance of online forms as well as the information you require from anyone who submits these forms. When creating an Online Form, start with either a default template or your own HTML template and then modify as needed for the current form.

- **Default NetSuite Template** : provides the ability to customize page messages, field labels and properties, etc.
- **HTML Template** : provides the ability to customize HTML templates that have already been created. For details on how to create your own HTML templates, see Creating HTML Templates for Online Custom Record Forms.

Creating Online Custom Record Forms

To create an online custom record form:

1. Go to Setup > Customization > Record Types.
2. In the Edit column, click the name of the record you want to create an online form for.
3. In the Online Forms subtab, click New Online Form or New Online HTML Form.

   Selecting New Online Form allows you to create a form based off of the NetSuite default template. Select New Online HTML Form allows you to create a form based off of a custom HTML template that you have created. You must first create the custom HTML template before you can use it for an online form.

   The steps involved for creating an Online Form versus an Online HTML Form are the same with a couple of exceptions as noted in the following procedure.
4. Enter a title for this form.

   This title is displayed at the top of the form.
5. If creating an HTML template (these options NOT available for Default templates):

   - In the Template field, select the HTML template you want to use for this form.

     You can create online HTML form templates at Lists > Marketing > Marketing Templates.
   - To have NetSuite insert labels for your form fields, check the Include Field Labels box.

     If you leave this box unchecked, you must include field labels in your HTML template file.

6. If creating a Default template (these options NOT available for HTML templates):

   - In the Message field, enter a message that your customers will see at the top of the form.

     This message can include up to 500 HTML characters.

     Note: The message can be formatted using NetSuite's built in rich text formatting tools. If you prefer to edit the message directly with html tags, click the HTML Source Code link.

   - In the Detail Message field, enter a message you would like to display at the bottom of the form.

     This message can include up to 4000 HTML characters. Again, you can use NetSuite's built-in rich text formatting or view as HTML only.

7. Check the Enable Online box if you want to link to this form from a Web site.
8. If desired, check the Form is Inactive box. You can always reactivate it at a later date.
9. In the Select Fields subtab, edit any existing fields or add new fields as desired.

Rearrange the order of fields as desired.

Click on a line and drag and drop it to the desired position or click Move Up, Move Down, Move to Top or Move to Bottom.

1. In the Set Up Workflow subtab:
   - To receive notification when this form is submitted, enter the addresses you want e-mail to be sent to in the Notify by E-Mail field.
   - To specify a page for customers to be sent to after they submit the form, enter the URL for that page in the Redirect to URL field. By default, the user who submits the form is redirected the home page of your primary Web site.
   - In the Handle Duplicate Records field, select how you want NetSuite to handle records that are duplicates of existing records.
2. Click the Set Up Appearance subtab.
   - In the Number of Columns Shown field, select the number of columns for the form.
   - In the Color Theme field, select a color theme for the form.
   - In the Font field, select a font for the form.
   - Check the Default Browser Style box to make the form's appearance match the customer's browser.
   - Check the Unlayered Sections box to expand any subtabs on the form.

- In the Button Alignment field, select where to place the buttons on the form.
- In the Form Logo field, select a logo to place at the top of the form.

  You can upload new logos at Lists > Web Site > Images.

  As you define the appearance options for your form, keep in mind that your end users use various sized browsers and types. For example, if you are designing on a high-resolution large monitor, several columns may display fine but for smaller low-resolution monitors it may be better to design your layout with fewer columns. Also, if customers may need to print the form, ensure that the colors you select will properly display for both color and black and white printers.

3. In the Custom Code subtab:

   The Custom Code tab is only available if you have the Custom Code feature enabled. For detailed information on using custom JavaScript files to perform functions, refer to the Custom Code Developer's Guide.

   - In the Script File field, select the JavaScript file that contains the desired scripts for this form.\

     Important: You must upload your file to the file cabinet before you can select it.

   - In the Page Init Function field, enter the script name to be called from your script file when this entry form is first loaded.
   - In the Save Record Function field, enter the script name to be called from your script file when this record is saved.
   - In the Validate Field Function field, enter the script name to be called from your script file when a field on this entry form is changed.
   - In the Field Changed Function field, enter the script name to be called from your script file when a change made to a field is accepted.

4. Click Save.
5. If desired, preview your new form.

   Select the form from the Online Form subtab and then click Preview Form.

6. Add links to your online form to your website.

   For details on how link to your online form see Creating Links to Custom Records.

Now, customers can enter information on your Web site, and records are automatically created or updated in NetSuite.

Linking Online Custom Record Forms to My Web Site

You can link to an online custom record form from your Web site.

If you have a NetSuite Web site, you can create a link to an online form in one of your information items, category descriptions and from any other HTML description field.

To link to an online custom form from your NetSuite Web site:

1. Click the Setup tab.
2. On the Setup page, under the Customization heading, click Record Types.

   The Custom Record Types list opens.
3. In the Edit column, click the name of the record type you want to edit.
4. Click the Online Forms subtab.
5. Click the name of the form you want to link to.
6. On the Online Custom Record Form page, click the External subtab.
7. Copy the URL from the Publishable Form URL field.

   You can highlight the URL with your mouse, right-click, and then click Copy.
8. Click the Lists tab.
9. On the Lists page under the Web Store heading, click the kind of item you want to paste the link into.
10. On the list, click Edit next to the information item or category you want to link to your online custom record form.
11. Enter or paste the link in the description field you want the link to appear in.

    For example, the administrator of Wolfe Electronics wants to include a line in a Detailed Description field that says, "Click here to register for your warranty." The word "here" links to the custom record form.

    The HTML code entered would be:

    <p>Click <a href='the Online Custom Record Form's URL'>here</a> to register for your warranty.</p>
12. Click Save.

Now your customers can follow the link to your online custom record form on your Web site. Once this form is submitted, a record is created with the customer's information.

You can also link to your online custom record form from an external Web site or from an e-mail message. To do this, simply copy and paste the form's URL into a link in your HTML document.

For more information about entering HTML in your Web site, read Entering HTML in Description Fields.

For more information about online custom record forms, read Creating an Online Custom Record Form.

Creating HTML Templates for Online Custom Record Forms

In order to use an HTML template when creating your Online Forms, you must first create the template and store it within NetSuite. You can create your HTML templates locally on your own machine and then upload them to the NetSuite file cabinet or you can use the built-in NetSuite template creator. See:

- Creating an HTML Template Locally
- Uploading an HTML Template
- Creating an HTML Form Template

Note: Online HTML form templates are especially useful if you do NOT use a NetSuite web site, but you use NetSuite's online forms.

Creating an HTML Template Locally

When creating an online form template on your local machine, you can define how the fields are arranged, which fields to include on the form and the style of the page. Use standard html code to create the template as you would for any other html form and include the following elements:

- <NLFORM> and </form> tags to define the beginning and end of the form.
- Tags for each NetSuite field included on the form. For details, see Using NetSuite Field Tags.
- An input tag that defines the button your customers use to submit the form.

  <input type="submit" value="Button Text">

  You can substitute text in the button by changing the value in the code. For example, if you want the text in the button to read Submit Form, the code would be:

  <input type="submit" value="Submit Form">

For example, the following HTML code is a simple representation of an acceptable form template:

```
<html>
<head>
</head>
<body>
<NLFORM>
<p>Please, enter the name of your company: <NLNAME></p>
<p>Please, enter an e-mail address we can use to contact you: <NLCUSTRECORD1></p>
<p><input type="submit" value="Submit Form"></p>
</form>
</body>
</html>
```

Important: The resulting file must include all tags, including <html>, <head> and <body> tags, in order to ensure that it is a valid HTML file recognizable by NetSuite.

Using NetSuite Field Tags

Field tags in an HTML form are defined as <NLTAG>, where TAG is the ID of the field. Each field in NetSuite has a unique ID and therefore a unique tag definition.

Note: Because field IDs are incorporated into tag definitions for fields, when creating custom fields that will be used in your HTML templates, it is useful if you specify IDs for each field and use consistent naming conventions that make sense in your business environment. If the default NetSuite IDs are accepted when creating your custom fields, the tags may not make sense in your HTML code making it more difficult to know exactly what the field references.

If you have chosen to include a mandatory Name field on your custom record, you must include the tag, <NLNAME>, for that field in your template.

To determine the tags to use for each field on your custom record:

1. Click Setup > Customization > Record Types.
2. In the Edit column, click the name of the record type you want to create a template for.
3. On the Fields subtab, click the name of the field you want to place in your template.
4. In the Address bar of your browser, the URL for this page is displayed. At the end of the URL is the ID for the selected field.

Uploading an HTML Template

Once you have created your own HTML template, you must upload the template to NetSuite in order to make it available when creating Online HTML forms.

Note: HTML templates are saved by default in the Marketing Templates folder in your file cabinet. To change the folder you use to store your templates, an administrator or Sales Administrator can go to Setup > Sales > Preferences > Sales Preferences.

To upload an HTML template:

1. Click > Documents >Files > File Cabinet.
2. In the file cabinet browse to Templates > Marketing Templates.
3. In the File field, click Browse.
4. In the Choose File window locate your HTML template on your hard drive.
5. Select the file.
6. Click Open.
7. On the Folder Contents page, click Add This File.

Creating an HTML Form Template

HTML Form Templates can be creating by selecting an existing template and setting additional properties for it or by designing a new HTML template from within NetSuite.

To create an HTML form template record:

1. Go to Lists > Marketing > Marketing Templates > New.
2. On the Select Type page, click Online Form.
3. In the Name field, enter a name for the template.
4. If desired, enter a title for the template.

   If no title is provided, this field sources the name field.
5. In the description field, enter any additional information for this template.

   This is information that describes the template — it is not a component of the displayed form. It may be useful to describe where and how this template should be used or describe any unusual fields or field relationships.
6. To base the current template record on an existing template, in the Template File field, select the template.

   You can select a template in a number of ways:

   - Enter the first few letters of the name of the template in the Template File field and then click Tab to automatically select a matching template — if multiple templates match the text, a list is displayed in a pop-up where you can select the desired template.

- Click the List icon to display a pop-up where you can select the desired template.
- Click the New icon to upload an HTML template from your hard drive. See Creating an HTML Template Locally for information on creating your own HTML templates.

  Once a template is selected, you can view detailed information about the template by clicking the open icon.

7. To design a new template, enter your text and template tags in the Template field.

   - To add NetSuite fields into your HTML template, select the desired field from the Insert Field drop-down list.
   - To view the HTML code, click the HTML Source Code link.

   If you have selected a template in step 6, skip this step. You do NOT need to enter template tags since the file selected in step 6 contains all of this information.

8. If desired, limit access to this template by selecting a group from the Restrict to Group field.
9. Click Save.Now, when you create an HTML online form, you can select this template.

Understanding Parent - Child Record Relationships

A child record type is a record that is referenced in another record in NetSuite. Child record types can be used to track specific information that requires multiple fields on a record.

For example, a typical Customer record includes a subtab for Notes and Messages. In this case the Customer record is a Parent of the Note and Message records — and therefore the Notes and Messages are child records of the Customer record.

You can also create your own Parent - Child relationships. For example, you could create a custom record type for Service Notes, attach your Service Notes record to an Equipment custom record type. The Service Notes would be a child record of Equipment. On the Service Notes record, you could include fields for service person, price of service, whether it was covered under the warranty and the equipment's next date of service. Employees could then view these service notes while working on Equipment records.

Note: Child records are only available during a record update, not an add, and only in the context of the parent record (you can NOT create a stand-alone child record entry).

Establishing a Parent - Child Relationship

To establish a parent - child relationship between records, you can do one of the following:

- Create a custom child record. To do this, create a custom record as described in Creating Custom Record Types but define at least one custom field as a child record.

  Fields are defined as a child record by checking the Record is Parent box and then selecting a Parent Subtab in the Display subtab of the record.

- Create a custom field, attach a built-in field, and then define the field as a child record. For more information on attaching built-in records, see Creating Built-in Child Records.

Creating Built-in Child Records

Using custom fields, you can attach existing built-in records to other records.

For example, you can attach a list of customer records to your case records. When you view a case record, you can click a subtab to view a list of customers associated with that case.

Important: In order to edit a built-in child record , a user must have sufficient permission levels to access the parent record. Without permission to access the parent, editing of the child record is not allowed, because changes to the child record are effectively changes to the parent record.

To attach a built-in record to another record:

1. Click Setup > Customization and then click New next to the kind of field you want to create.

   The type of field you select depends on the type of record you want to attach. For example, if you want to attach a list of customers to your case records, you would create an entity field for customer records.

2. In the Label field, enter a name for your custom field.

3. In the Type field, select List/Record or Multiple Select.

   Choosing Multiple Select allows you to attach multiple records to another record. For example, you can attach multiple issues to a customer record.

4. In the List/Record field, select the kind of record you want to attach this list to.

   For example, if you want to attach a list of records to your case records, you would select Cases.

5. Check the Record is Parent box.

6. On the Applies To subtab, check the box next to the type of record you want to attach to another record.

   For example, check the Customer box to attach customer records.

7. Click the Display subtab.

8. In the Parent Subtab field, select the subtab where you want this list to display on the parent record.
9. Click Save.

Now, when you view a parent record, a list of built-in records is attached on the specified subtab.

Note: When attaching a list to a transaction, you can only attach to transactions that have customizable forms. For example, you can attach a list of cases to a cash sale but not to a bill. The following transactions are available as parents:

- Cash Refund
- Cash Sale
- Credit Memo
- Estimate
- Invoice
- Opportunity
- Purchase Order
- Return Authorization
- Sales Order

Using Child Records

After you have created your child record types, you can enter data into records from the child record list or parent records.

To enter data in child records from the child record list:

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.
2. Click New Record next to the name of the child record type you want to enter records for.
3. In the field that links to the parent record, choose the record you want this child record to link to.

   For example, in the Equipment field, you would choose the machine you are entering service notes for.
4. Enter data in the remaining fields on the record.
5. When you are finished, choose one of two options to submit the information to NetSuite:
   - Click Submit to submit the information and return to the child record list.

- Click Submit & New to submit the information and add another record.

You can now view this child record from its parent record.

To enter data in child records from the parent record:

- Open the list of records where your parent record appears.
  - If your parent record is a custom record:
    1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.
    2. Click the name of the custom record list you want.

    If you have a link in your center directly to your parent record list, you can click that link instead of the above steps.
  - If your parent record is a standard NetSuite record, on the Lists page, click the name of the list you want.

    You can attach child records to any of the standard NetSuite records.
    1. Click View next to the name of the record you would like to enter data for.
    2. Click the subtab where the child record list appears.
    3. Click New above the list to enter a new child record.
    4. Enter data in the fields on the record.
    5. When you are finished, choose one of two options to submit the information to NetSuite:
  - Click Submit to submit the information and return to the parent record.
  - Click Submit & New to submit the information and add another child record.

To view a child record from the parent record, click the link for the child record in the list of child records. To edit a child record from the parent record, click Edit.

By default, inactive child records are not shown on the parent record. If you customize the sublist and the saved search filters by the Inactive checkbox, inactive records can be shown in the sublist.

Sourcing With Custom Records

You can use sourcing with custom record types for both standard and custom records. You must first create a List/Record field on your custom record type for the kind of record you want to source information from. Once you have created that field, you can then choose it in the Source List field.

For example, you have created an Intern custom record type. You want to include information on your intern records about the employees supervising each intern. You can do this by first creating an Employee field with a list of your employees. Then, you can source information from employee records to other fields on your intern records.

For information on sourcing and custom fields, see Sourcing With Custom Records.

To use sourcing with custom record types:

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.
2. Click Edit next to the record type you want to create a sourced field for.
3. Click the Fields subtab.
4. Click New Field.
5. In the Description field, enter the name of your new field.

   For example, if you are creating a field to list your employees you would enter Employee.
6. In the Type field, select List/Record.
7. In the List/Record field, select the kind of record you want to include information from in other fields on your record type.

   For example, if you want to include information about your employees you would select Employee.
8. Click Save.

   Now, you can create your sourced fields.
9. Click the Fields subtab.
10. Click New Field.
11. In the Description field, enter a description for your field.

    For example, if you wanted to include an employee's e-mail address you would enter Employee E-mail.
12. In the Type field, select a type of custom field.

    The type you select must match the type of field you want to source from.

    For example, if you wanted to include an e-mail address you would choose Email Address in the Type field.

    For a list of standard fields and their types, see Available Standard Fields and Field Types for Custom Record Types and Source Lists.

13. If you are creating a List/Record field, select the appropriate list or record in the List/Record field.

    For example, if you are sourcing from the Sales Rep field on Customer records, choose Employee in the List/Record field because your sales reps are employees.

14. In the Source List field, select the record type you want to source information from.

    This field lists the field you created in steps 1-9.

    You can create multiple Source Lists for your custom records by repeating steps 1-9 for each kind of record you want to be able to source information from.

15. In the Source From field, select the field you want to include information from.

    You must choose a Source List before you can choose a field.

    The selected field's type must match the type selected in the Type field.

16. When you have finished, click Save.

17. To add additional Source List fields, repeat steps 1-9.

    To add additional sourced fields, repeat steps 10-17.

You can now include information from standard and custom fields and records on your custom record types.

Available Standard Fields and Field Types for Custom Record Types and Source Lists

Custom record types and source lists can use sourcing with the standard fields and records listed below. In addition to standard records, you can also use custom fields and record types for sourcing. When using custom fields and record types for sourcing, the field type chosen depends on the field type of the custom field being sourced from.

For example, you have created an custom entity check box field for Quarterly Mailing. If you want to source information from the Quarterly Mailing field to a custom CRM field for cases, you must choose Check Box in the Type field when creating your CRM field.

Customer, Employee, Partner and Vendor Fields

Field	Type	Customer	Employee	Partner	Vendor
Name/ID	Free-Form Text	×	×	×	×
Bill To	Text Area	×	×		×
Ship To	Text Area	×			×
Phone	Phone Number	×	×	×	×

Field	Type	Customer	Employee	Partner	Vendor
Fax	Phone Number	×		×	×
E-mail	E-mail Address	×	×	×	×
City	Free-Form Text	×	×		×
State	Free-Form Text	×	×		×
Zip	Free-Form Text	×	×		×
Country	Free-Form Text	×	×		×
Sales Rep	List/Record*	×			
Expected Close Date	Date	×			
Renewal Date	Date	×			
Contact	Free-Form Text	×			×
Alt. Contact	Free-Form Text	×			×
Alt. Phone	Phone Number	×			×
Balance	Currency	×			×
Credit Limit	Currency	×			×
Account	Free-Form Text	×			×
1099 Eligible	Check Box				×
Tax ID	Free-Form Text				×
Company Name	Free-Form Text			×	×
Supervisor	List/Record*		×		
Soc. Sec. #	Free-Form Text		×		

* In the List/Record field, you must choose the Employee list for sourcing to work properly.

Item

Field	Type
Name	Free-Form Text
Display Name	Free-Form Text
Vendor Name	Free-Form Text
Online Name	Free-Form Text
Available Online	Check Box
Base Price	Currency
Cost	Currency
Preferred Vendor	List/Record*
On Hand	Decimal Number

* In the List/Record Field, you must choose the Vendor list for sourcing to work properly.

Using Custom Records

Custom record entries are the **actual** custom records you create. You can create, edit and search your custom record entries from the list of record types. You can also inactivate or delete a record type that is no longer being used.

For details, see the following:

- Viewing or Editing a Custom Record Type
- Inactivating a Custom Record Type
- Deleting a Custom Record Type
- Viewing a Custom Record Entries List
- Viewing or Editing a Custom Record Entry
- Creating a Custom Record Entry
- Copying a Custom Record Entry
- Searching Custom Record Entries

Viewing or Editing a Custom Record Type

To view or edit a custom record type :

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.
2. In the Edit column, click the name of the custom record type you want to view or edit.
3. Record types have the following subtabs for you to further define them:
   - **Fields** – Create and rearrange the fields for your custom record type. See Adding Fields to Custom Record Types.
   - **Subtabs** – Create and arrange subtabs for your custom record type. For information, see Adding Subtabs to a Custom Record.

     Note: To save time, create and arrange subtabs for your custom records before defining your custom fields.
   - **Sublists** – Add search results as sublists on your custom record type. For more information, see Applying Custom Sublists to Custom Record Types.
   - **Icons** – Select the PNG sprite you want to use to represent this record type in the New Bar, Create New menu, Recent Records menu, Recent Records portlet, and QuickViews. You can choose from built-in icons or create your own custom icon. For more information, see Choosing an Icon for a Custom Record.

- **Numbering** – Specify the numbering format for custom records types. For information, see Numbering Custom Record Types.
- **Forms** – Customize and select a preferred entry form for your custom record type. For more information, see Adding Custom Forms for a Record.
- **Permissions** – Choose the roles you want to access custom record entry forms, choose a default form and restrict the forms available here. For information, see Setting Up a Permissions List for a Custom Record Type.

  Important: You must choose Use Permission List in the Access Type dropdown for these permissions to take affect.

- **Links** – Create links that take you to the list of record entries for this type and choose where to place the links. See Creating Links to Custom Records.
- **Managers** – Define specific employees as managers of the current record type. This allows them to modify the custom record type. When defined as a manager, the employee is automatically granted custom record view permission. This allows them to see the list of custom record types but NOT drill down on them.

  Note: If an employee has a role that includes the Custom Record Type permission, they have edit access to ALL custom records types. The Managers subtab allows you to grant permission for an employee to the current Record Type only.

- **Translation** – (when Multi-Language feature is enabled) Define translations for the custom record type name to be used when users change the language preference. See Adding Translations for Custom Records .
- **Child Records** – If this record type is a parent record, the child records are listed here. See Using Child Records.
- **Parent Records** – If this record type is a child record, the parent records are listed here. See Understanding Parent - Child Record Relationships.

4. When you have finished making changes, click Save.

Inactivating a Custom Record Type

To inactivate a custom record type :

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.
2. At the bottom of the page, check the Show Inactives box.
3. Check the box in the Inactive column next to the record type you want to inactivate.
4. Click Submit.

Deleting a Custom Record Type

To delete a custom record type:

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.
2. In the Edit column, click the name of the custom record type you want to delete.
3. In the More Actions menu, click Delete.

After the deletion, the list of custom records no longer includes the deleted record type.

Important: You cannot delete a custom record type that is used by any custom fields or workflows. When you attempt to delete a custom record type that has this kind of dependency, you receive error messages with lists of custom field names and/or custom workflow names that depend on the custom record. You must delete any dependent custom field(s) and workflow(s) before you can delete the record type.

Viewing a Custom Record Entries List

To view a custom record entries list :

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.

   You can also reach the custom record entries list by placing a link directly to the list in your center. See Creating Links to Custom Records.
2. Click the List link for the custom record entries list you want to see.

   Your list of custom records appears.

   You can also reach the custom record entries list by placing a link directly to the list in your center. See Creating Links to Custom Records.

Viewing or Editing a Custom Record Entry

To view or edit a custom record entry :

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types, and click the List link for the custom record entries list you want to see.

   (You can also reach the custom record entries list by placing a link directly to the list in your center. See Creating Links to Custom Records.)
2. In the records list, click the View or Edit link for the record you want.
3. If you are editing, click Save when you are finished.

Creating a Custom Record Entry

To create a custom record entry :

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types, and click the New Record link.
2. Enter a name and any notes you want to add, and click Save.

Copying a Custom Record Entry

To copy a custom record entry :

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types, and click the List link for the custom record entries list you want to see.
2. Click View next to the record you want to copy.
3. In the More Actions menu, choose Make Copy.
4. Enter a name for your new record, and click Save.

   You cannot create another record with the same name. The information from the original record is copied except the Name.

Searching Custom Record Entries

To search a list of custom record entries :

1. Go to Setup > Customization > Lists, Records, & Fields > Record Types.
2. Click the Search link for the record type you want to search.
3. On the Search page, enter or select criteria to filter search results:

   To enter specific criteria:

   - Select Any to search for all information of that kind.
   - Select Is, Starts With or Contains and enter the information or part of the information you want to find.
   - Select Is Not, Does Not Start With or Does Not Contain and enter the information you want to exclude.
   - Select a choice or choices from a dropdown list or a list of options.
   - Click List next to an empty list field to select from a popup list, or enter the first few letters and press TAB.

- Enter or pick dates in From and To fields if you selected custom in the date field.
- Select Either, Yes or No for the option you want.

Using a combination of these fields is the best way to find what you are looking for.

4. After entering search criteria, choose one of the following options:
   - Click the Submit button to run the search and open a NetSuite page with a list of results.
   - Click the Export button to run the search and save results as a .CSV file that you can save to disk or open on your desktop. For more information, see the help topic *Exporting Search Results*.
   - Click the Reset button to clear the filters you defined.
   - Click the Customize button to open a saved search page with no filters defined, where you can define a custom search form to be your default search form for the record type. See the help topic *Defining a Saved Search as a Preferred Search Form*.
   - Click the Create Saved Search button to open a saved search page that includes the filters you defined. See the help topic *Using Saved Searches*.
5. Once you have submitted a search and a search results page has opened, you can do the following:
   - Export search results as a .CSV or .XLS file. See the help topic *Exporting Search Results*.
   - Email search results to one or more recipients. See the help topic *Emailing Search Results*.
   - Create a saved search with the same definitions as the search. See the help topic *Using Saved Searches*.
6. If the list of results is too large or too small, you can add, remove or expand criteria. To return to the search criteria page, click Return To Previous Search at the bottom of the Search Results page.
   - To set more complex search criteria and define search results display options, enable the Use Advanced Search box on the Search page. For information about using advanced search functionality, see the help topic *Defining an Advanced Search*.
   - Saved searches can be run on demand, include all advanced search functionality and provide additional capabilities, including scheduling, email alerts, audience definition, and highlighting of results. See the help topic *Using Saved Searches*.

Chapter 5 Custom Centers

You can create custom centers that you apply to custom roles. To use the Custom Centers feature, you must enable the Custom Records feature, at Setup > Company > Setup Tasks > Enable Features > SuiteCloud.

NetSuite Centers determine which tabs and links are available for groups of similar user roles. For example, the Sales Center is shared by the Sales Rep, Sales Manager, and Sales Administrator roles and includes tabs such as Leads, Opportunities and Forecast. (For a visual representation of a center, see Custom Centers in the SuiteBuilder Overview section.)

Each tab contains links to transactions, lists and setup pages. The links that appear are based on the user's role and the permissions the role is granted. For example, users assigned to the Sales Rep role would see different links on the Forecast tab than users with the Sales Administrator role because of permissions granted to each role. However, both roles share the Sales Center.

Creating a custom center requires you to create a center record and then to create its custom tabs. When you create custom tabs, you choose the center you want the tab to appear in. This can be either an existing or custom center. You also choose the links and portlets that will appear in the center.

Note: Users can only use links and information that their roles have access to. To customize roles, go to Setup > Users/Roles > Manage Roles. You must create a new role to apply a custom center to it. To do this, click New on the Manage Roles page, select your custom center in the Center Type field and customize your new role.

Creating Custom Centers

To create a custom center, go to Setup > Customization > Centers > New. Enter a name for your center and click Save.

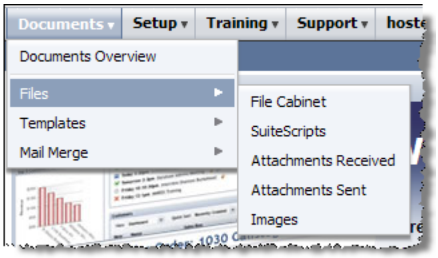
Next, you create the tabs and links you want to appear in your center. To do this, go to Setup > Customization > Center Tabs > New.

Note: If the Multi-Languages feature is enabled in your account, you can define translated center names to be used for different language settings. See Translating Custom Centers, Tabs, Categories, and Links.

Creating Center Tabs

Creating center tabs enables you to add custom tabs to roles. Center tabs can include categories of links to NetSuite pages, custom records, Suitelets, or external web pages. You can also designate which portlets you want to appear on the dashboard when users click the tab.

In the following figure, Documents, Setup, Training, and Support are the **tabs**. Off the Documents tab, Files, Templates, and Mail Merge are the **categories**. Off of the Files category are the **links** to the File Cabinet, SuiteScripts, Attachments Received, and so on.



Note that if you want to create a tab that displays internal information in your account, you should create your own custom intranet tab. For more information, see the help topic *Publishing Information to an Internal Site in the NetSuite Help Center*.

To create a center tab:

1. Go to Setup > Customization > Center Tabs > New.
2. In the Label field, enter a name for the tab.

   Users click this name to view the tab's contents.
3. In the Center field, choose the center you want this tab added to.

   If you want to show this tab in all centers, select -**All**-.

   Note: You cannot include a custom center tab that is shown in all centers in a dashboard that is already published.
4. On the Categories subtab, in the Label column, enter a name for a category of links. After creating your categories, you will later follow the steps in Creating Center Links to assign specific links to each category.

   Warning: Step 4 is for adding custom categories to **custom** tabs. If you want to add custom categories to **standard**, built-in NetSuite tabs, see Creating Center Categories.
5. Click Add.
6. Repeat steps 4 – 5 for each category of links you want to appear on this tab.
7. Click the Portlets subtab.
8. In the Type column, choose Links.

9. In the Column section, choose to place this portlet on the left, right or in the middle of your tab.

10. In the Show column, click the checkbox if you want the link to show.

11. Click Add.

12. Repeat steps 8 – 11 to add additional portlets to this tab.

13. Click the Audience subtab.

14. Using the fields provided, select the users you want to have access to this custom tab.

    Note: If the Multi-Languages feature is enabled in your account, you can define translated tab names to be used for different language settings. See Translating Custom Centers, Tabs, Categories, and Links.

15. Click Save.

16. Next, add links to your categories. See Creating Center Links for details.

Note: If you return to the tab and do not see the changes you have made, clear your cache in your browser. To do so in Internet Explorer, click Tools, and select Internet Options. In the Temporary Internet Files section, click Delete Files, and then click OK.

Creating Center Categories

Use the following steps to add custom categories to standard, built-in NetSuite tabs. After creating custom categories for a standard tabs, you can then add links to each category. The links can go to NetSuite pages, custom records, Suitelets, or external web sites.

Note: To add custom categories to **custom** tabs, see Step *On the Categories subtab, in the Label column, enter a name for a category of links. After creating your categories, you will later follow the steps in to assign specific links to each category.* in Creating Center Tabs.

To add custom categories to standard tabs:

1. Go to Setup > Customization > Center Categories > New.

2. On the Custom Center Categories page:

    1. In the Center Type dropdown, select an existing center.

    2. In the Section dropdown, select one of the standard, built-in NetSuite tabs (also referred to as sections).

    3. In the Category field, provide a UI label for the category.

    4. In the Insert Before dropdown, select where you want to insert your custom category.
    5. In the Link dropdown, select the appropriate link.

       The links that appear in the dropdown are links to other NetSuite pages, custom record, Suitelets, and external web sites. Links to NetSuite pages, custom records, and Suitelets will appear by default.

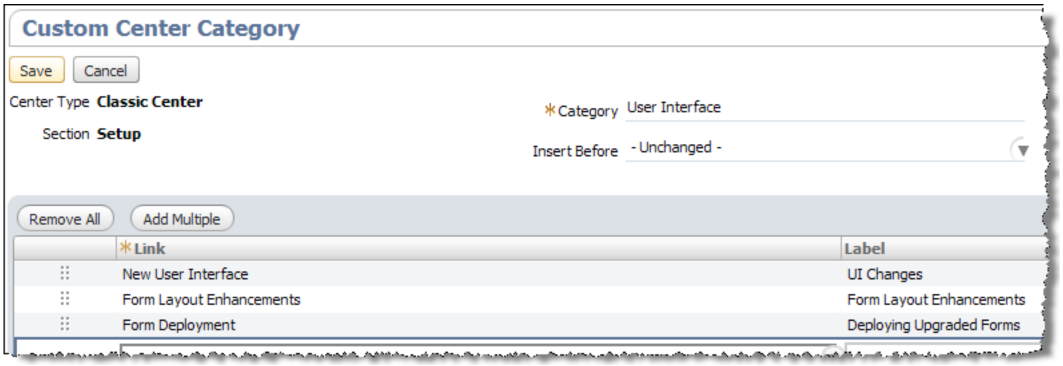
       If you want to select a link to an external web site, you must have already created that link by going to Setup > Customization > Center Links > New. (See Creating Center Links for details.) Once the link has been created, it will be appear in the Link dropdown list.
    6. In the Label field, provide a UI label for the link.
    7. In the Short List checkbox, click the checkbox to indicate whether the link should appear in a portlet when the link is in a narrow column of the dashboard.

Note: If the Multi-Languages feature is enabled in your account, you can define translated category labels to be used for different language settings. See Translating Custom Centers, Tabs, Categories, and Links.

3. Click Add.
4. Click Save.

The following figure shows the Custom Center Category page for a custom category called User Interface. The User Interface category will appear on the standard Setup tab.

The User Interface category will contain three custom links. Because these links are going to web pages in the NetSuite Help Center, these links were created beforehand using the steps in Creating Center Links. Once created, the links then become available in the Link dropdown.



Custom Center Category

Center Type **Classic Center**

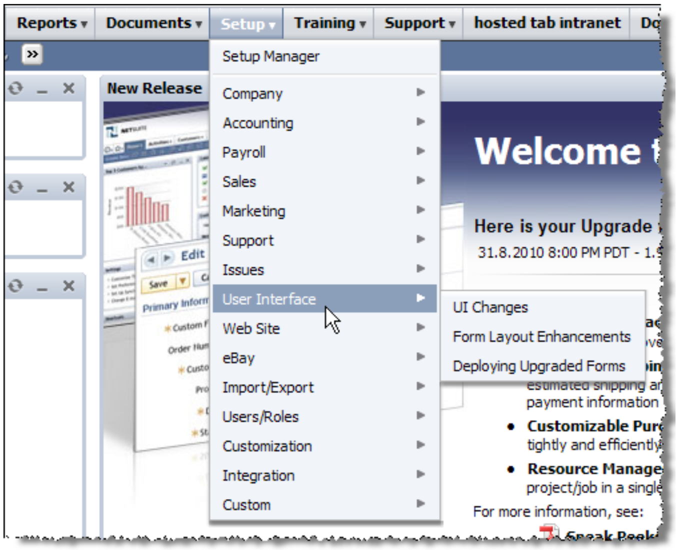
Section **Setup**

*Category User Interface

Insert Before - Unchanged -

	*Link	Label
	New User Interface	UI Changes
	Form Layout Enhancements	Form Layout Enhancements
	Form Deployment	Deploying Upgraded Forms

The next figure shows the User Interface custom category on the Setup tab. Notice three links within the User Interface category.



Note: If you return to the tab and do not see the changes you have made, clear your cache in your browser. To do so in Internet Explorer, click Tools, and select Internet Options. In the Temporary Internet Files section, click Delete Files, and then click OK.

Creating Center Links

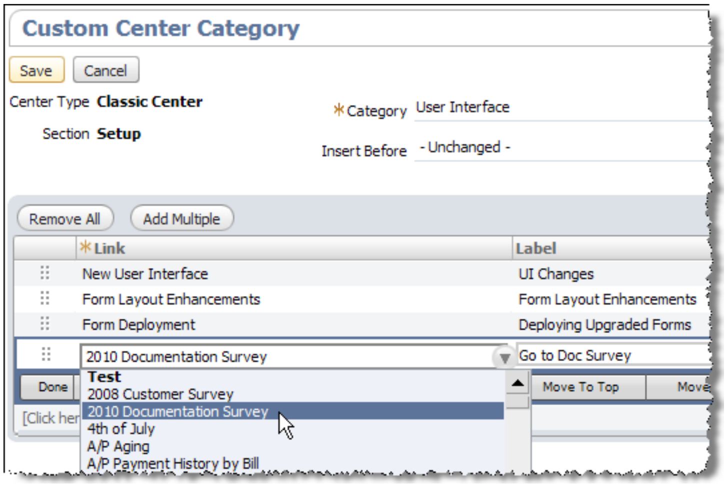
You can create links that appear on your own custom categories as well as on standard, built-in NetSuite categories. Center links can take users to other pages within NetSuite, custom records, Suitelets, or external Web sites.

To create links to NetSuite pages, custom records, and Suitelets:

1. Go to Setup > Customization > Customization > Center Category.
2. On the Custom Center Categories list page, click Edit next to the category you want to add links to.
3. On the Custom Center Category page, in the Link dropdown field, select the NetSuite page, custom record, or Suitelet you want to link to.

   Note: Note that for custom records, you can also create links using the Links subtab on the Custom Record Type definition page. If you want a link to appear on a standard category, you **must** set the link on the Custom Record Type definition page. See Creating Links to Custom Records for details. Also note that a link to a custom record type does not display for users that do not have permission to access that custom record type.

   The following figure shows a link to a custom record type called 2010 Documentation Survey.



Custom Center Category

Center Type **Classic Center**

Section **Setup**

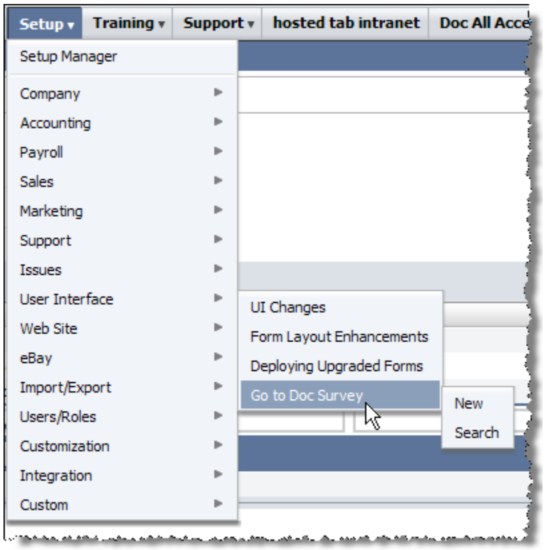
*Category User Interface

Insert Before - Unchanged -

	*Link	Label
	New User Interface	UI Changes
	Form Layout Enhancements	Form Layout Enhancements
	Form Deployment	Deploying Upgraded Forms
	2010 Documentation Survey	Go to Doc Survey

Test
2008 Customer Survey
2010 Documentation Survey
4th of July
A/P Aging
A/P Payment History by Bill

4. Provide a UI label for the link. This is the label that will appear off to the side of the category in the UI. In the previous figure, the name of the link off of the User Interface category will appear as **Go to Doc Survey**. When you hover over the Go to Doc Survey link, you can create new instances of the 2010 Documentation Survey record type, or you can search existing surveys.



Note: If the Multi-Languages feature is enabled in your account, you can define translated link labels to be used for different language settings. See Translating Custom Centers, Tabs, Categories, and Links.

5. Click Add.

6. Repeat steps 3 – 5 to add additional links.

7. Click Save.

To create links to web pages:

1. Go to Setup > Customization > Center Links > New.
2. In the Label column, enter a name for your custom link.
3. In the URL column, enter the URL for the link.
4. Click Add.
5. Repeat steps 2 – 4 to add additional web links.
6. Click Save.
7. Add this link to a custom category by going to Setup > Customization > Center Categories.
8. Click Edit next to the category.
9. In the Link dropdown field, select your custom link.
10. Click Add.
11. Click Save.

Translating Custom Centers, Tabs, Categories, and Links

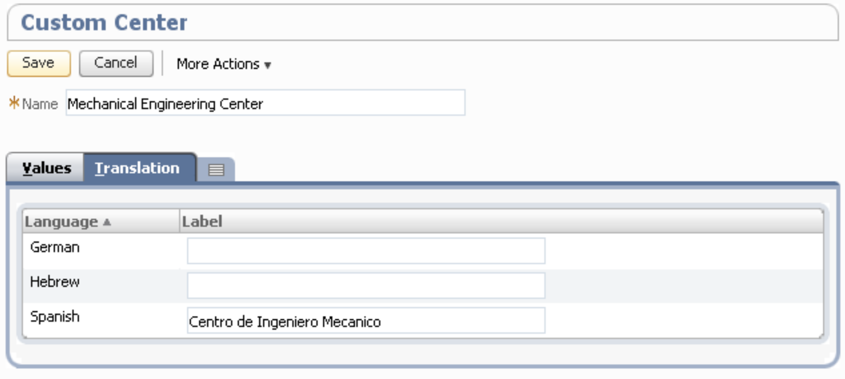
If the Multi-Language feature is enabled in your account, you can translate labels for custom centers, tabs, categories, and links, so that they match the language of the NetSuite user interface, set by each user at Home > Set Preferences. For details, see the following:

- Translating Custom Centers
- Translating Custom Center Tabs
- Translating Custom Center Categories
- Translating Custom Center Links
- Displaying Translated Centers

Important: Before you can translate these labels, you need to select translation languages at Setup > Company > General Preferences, on the Languages subtab. This subtab lists both system-supported languages that can be used for the NetSuite user interface (and are available at Home > Set Preferences), and additional languages that can used for Web site translations only (and are not available at Home > Set Preferences). You should only enter translations for system-supported languages, because these are the only languages that can be displayed in the user interface. For details, see the help topic *Setting Up Multiple Languages*.

Translating Custom Centers

You can define translated labels for a custom center when you edit the center. (You must first create and save it.) After the center has been created, go to Setup > Customization > Centers, click Edit, and enter translated labels on the Translation subtab.



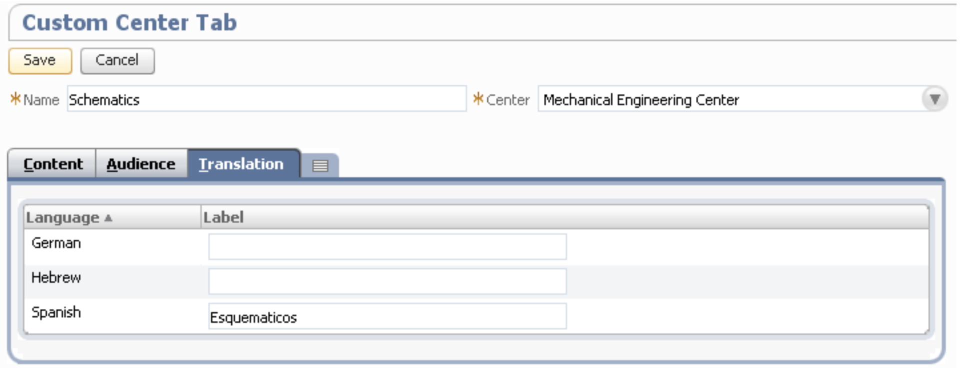
Custom Center

*Name Mechanical Engineering Center

Language ▲	Label
German	
Hebrew	
Spanish	Centro de Ingeniero Mecanico

Translating Custom Center Tabs

You can define translated labels for a custom center tab when you first create it or when you edit it later. Go to Setup > Customization > Center Tabs > New and enter translated labels on the Translation subtab. These labels translate the Name field.

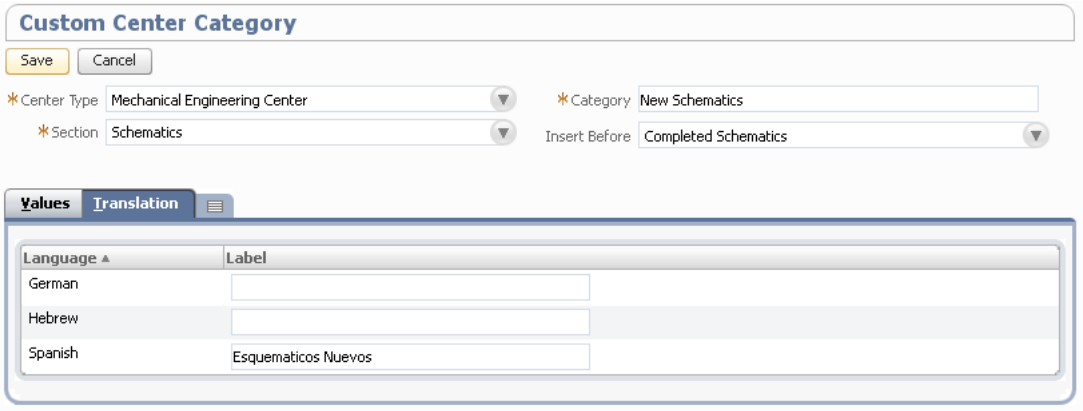

Custom Center Tab

*Name Schematics
*Center Mechanical Engineering Center

Language ▲	Label
German	
Hebrew	
Spanish	Esquematicos

Translating Custom Center Categories

You can define translated labels for a custom center category when you first create it or when you edit it later. Go to Setup > Customization > Center Categories > New and enter translated labels on the Translation subtab. These labels translate the Category field.

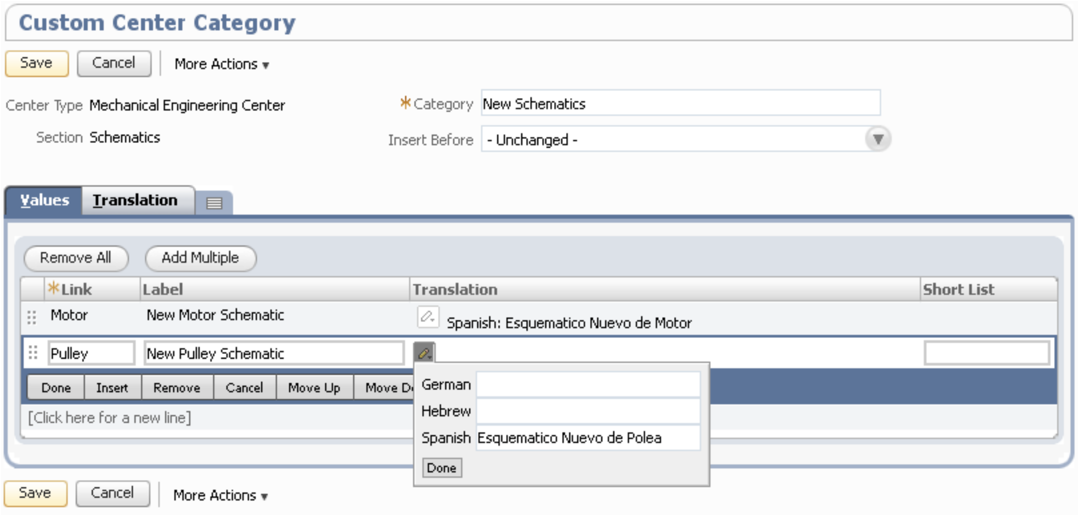

Custom Center Category

*Center Type Mechanical Engineering Center
*Category New Schematics
*Section Schematics
Insert Before Completed Schematics

Language ▲	Label
German	
Hebrew	
Spanish	Esquematicos Nuevos

Translating Custom Center Links

You can define translated labels for custom center links on the Custom Center Category page that lists the links. Go to Setup > Customization > Center Categories and enter link translations on the Values subtab.



Custom Center Category

Center Type Mechanical Engineering Center
Section Schematics
*Category New Schematics
Insert Before - Unchanged -

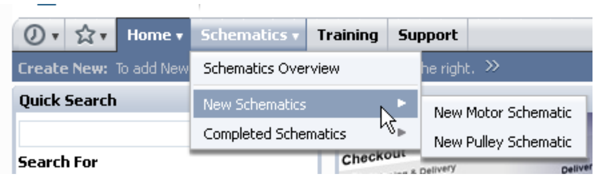
*Link	Label	Translation	Short List
Motor	New Motor Schematic	Spanish: Esquematico Nuevo de Motor	
Pulley	New Pulley Schematic		

[Click here for a new line]

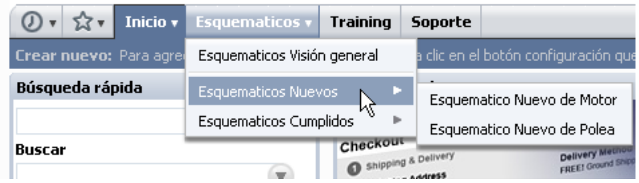
German	
Hebrew	
Spanish	Esquematico Nuevo de Polea

Displaying Translated Centers

The following screenshot shows a custom center, tab, category, and link in U.S. English:



The following screenshot shows the translations for these labels when the user preference is changed to Spanish at Home > Set Preferences:



Chapter 6 Deploying Upgraded Forms

NetSuite administrators can use the Upgrade Checklist to **preview** custom forms with Version 2010 Release 2 *Form Layout Enhancements* applied. These enhancements include *Field Groups* and the standardization of form *Subtabs and Sublists*. After previewing the new layout, administrators can then **deploy** the upgraded custom forms to end users.

To access the Upgrade Checklist, go to Setup > Customization > Entry Forms [or Transaction Forms], and click the link in the message area at the top. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.

Once you deploy an upgraded form, you cannot "rollback" the deployment. When you deploy an upgraded form, you are replacing the existing form.

Important: As of Version 2012 Release 2, all standard forms have been automatically upgraded to use form layout enhancements, so there is no need for administrators to deploy upgraded standard forms. Also, if your NetSuite account was established in Version 2010 Release 2 or later, form layout enhancements are automatically applied to custom forms as well. The custom form deployment process described here is applicable only to NetSuite accounts established prior to Version 2010 Release 2.

Review the following to get an understanding of the process for previewing, upgrading, and deploying custom forms:

- Custom Form Deployment Process (Summary) – Provides a high-level overview of the deployment process.
- Custom Form Deployment Process (In Detail) – Provides a workflow diagram that depicts each step in this process.

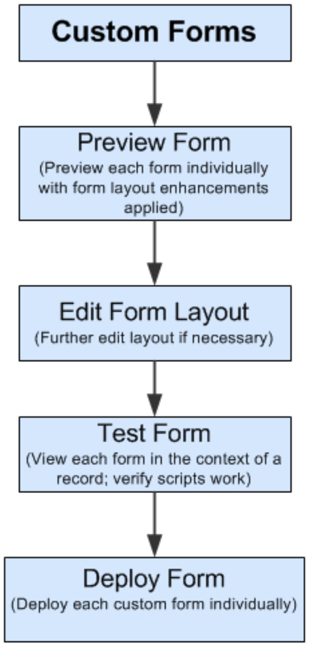
See the following for detailed steps:

- Deploying Upgraded Custom Forms
- Understanding Form Deployment Statuses
- Understanding Form Layout Enhancement Upgrade Logic
- *Form Upgrade Best Practices*

Custom Form Deployment Process (Summary)

The custom form deployment process involves previewing each custom form, further modifying the layout of each form (if necessary), testing each form in the context of a record, and then deploying the form to end users. Custom forms must be deployed individually.

The following diagram summarizes the form deployment process. Use NetSuite's **Upgrade Checklist** to manage this process. To access the Upgrade Checklist, go to Setup > Customization > Entry Forms [or Transaction Forms], and click the link in the message area at the top. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.

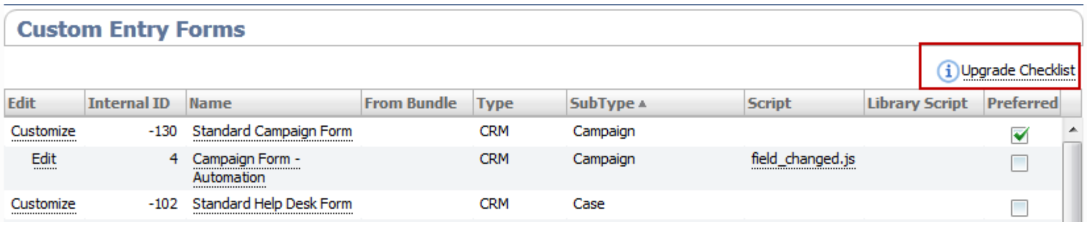

Custom Forms
Preview Form
(Preview each form individually with form layout enhancements applied)
Edit Form Layout
(Further edit layout if necessary)
Test Form
(View each form in the context of a record; verify scripts work)
Deploy Form
(Deploy each custom form individually)

Note: As of Version 2012 Release 2, the upgrade process has been simplified to remove the Skip Upgrade option.

Custom Form Deployment Process (In Detail)

You begin the custom form deployment process by going to the Upgrade Checklist.

To access the Upgrade Checklist for custom **transaction** forms, go to Setup > Customization > Transaction Forms, and click the Upgrade Checklist link. To access the Upgrade Checklist for custom **entry** forms, go to Setup > Customization > Entry Forms, and click the Upgrade Checklist link. If you have already upgraded forms in your account, the link is called Return to Upgrade Checklist.



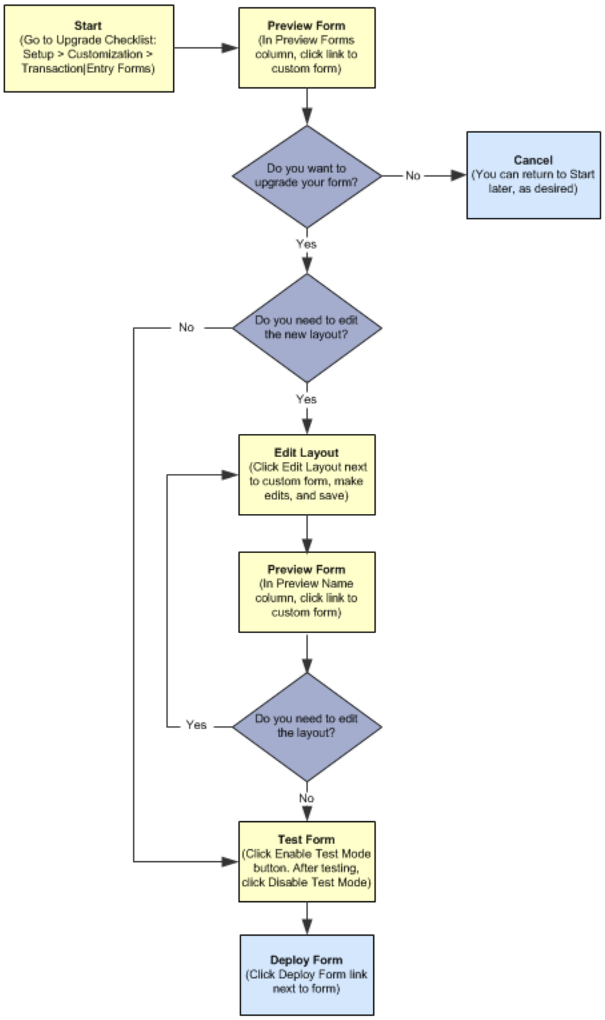
Custom Entry Forms

Edit	Internal ID	Name	From Bundle	Type	SubType ▲	Script	Library Script	Preferred
Customize	-130	Standard Campaign Form		CRM	Campaign			☑
Edit	4	Campaign Form - Automation		CRM	Campaign	field_changed.js		□
Customize	-102	Standard Help Desk Form		CRM	Case			□

Important: If you do not see this link in the upper right of the custom transaction forms or custom entry forms list, it means your account does not include any custom forms that require upgrading.

The following diagram shows the entire workflow for deploying upgraded forms.

Note: If you cannot see the detail in this diagram, you can enlarge your screen by pressing the **Ctrl** key and the plus (**+)** key on your keyboard. Press and hold the **Ctrl** key while pressing the + key multiple times. To reduce your screen to its normal size, you can press and hold the **Ctrl** key while pressing the **–** (minus) key multiple times.



Note: As of Version 2012 Release 2, the upgrade process has been simplified to remove the Skip Upgrade option.

After reviewing the overall form deployment workflow, go to Deploying Upgraded Custom Forms to begin the form upgrade and deployment processes.

Deploying Upgraded Custom Forms

Use the following steps to deploy custom forms that include *Form Layout Enhancements*. When you deploy upgraded custom forms to end users, NetSuite **automatically** applies the layout enhancements to the form. These enhancements include *Field Groups* and the standardization of form *Subtabs and Sublists*.

Note: As of Version 2012 Release 2, all standard forms have been automatically deployed to use form layout enhancements, so there is no need to deploy standard forms. Also, if your NetSuite account was established in Version 2010 Release 2 or later, form layout enhancements are automatically applied to custom forms as well. The custom form deployment process described here is applicable only to NetSuite accounts established prior to Version 2010 Release 2.

To deploy upgraded custom forms, see these sections in the following order:

1. Previewing Undeployed Custom Forms
2. Editing the Layout of Custom Forms Prior to Deployment
3. Avoid Editing Custom Forms in Tandem
4. Testing Undeployed Custom Forms
5. Deploying Custom Forms

Be aware of the following:

- You are NOT required to upgrade any of your custom forms if you prefer that they keep their existing layout. If you choose not to upgrade a custom form, you can take no action at all *Knowing When to Skip Upgrade of Custom Forms*.
- As of Version 2012 Release 2, the form upgrade process has been simplified to remove the Skip Upgrade option. If prior to that release you used this option, and later decide you want Form Layout Enhancements applied to the form, you can perform an Undo Skip operation (see Deploying Skipped Custom Forms for details.) However, if you have added custom field groups to a “skipped” form, you will lose all field group formatting when you deploy the upgraded custom form. The fields in the field groups will remain, however they will be reorganized into auto-generated field groups (which you can later customize).

Previewing Undeployed Custom Forms

Before deploying upgraded custom forms to end users, NetSuite lets you preview what the forms will look like after you deploy them. The preview process for custom forms is **extremely important**, as it lets you consider the following:

- Do you like how NetSuite has applied the *Form Layout Enhancements* to your custom forms?
- Do you see additional layout modifications you need to make to the form so that it suits your business needs?
- Do you want users to start working with the upgraded custom form? (If so, you will deploy the form.)
- Would you prefer to keep the custom form in its existing layout?

To preview undeployed custom forms:

1. Navigate to the Upgrade Checklist:
    1. Go to Setup > Customization > Transaction (*or Entry*) Forms.
    2. On the forms list page, in the message area at the top, click the Upgrade Checklist link. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.
2. In the **Preview Form** column, click the custom form you want to preview.

   The form that appears will have the form layout enhancements applied according to the logic described in Understanding Form Layout Enhancement Upgrade Logic.
3. If you generally approve of the upgraded layout, but see a few areas you need to modify, see Editing the Layout of Custom Forms Prior to Deployment.

   If you decide you want to retain the existing layout that your end users are **currently** working with, click Cancel and take no further action.

Editing the Layout of Custom Forms Prior to Deployment

After previewing a custom form, you will probably notice parts of the layout you want to modify. You will go through the Upgrade Checklist to make these modifications.

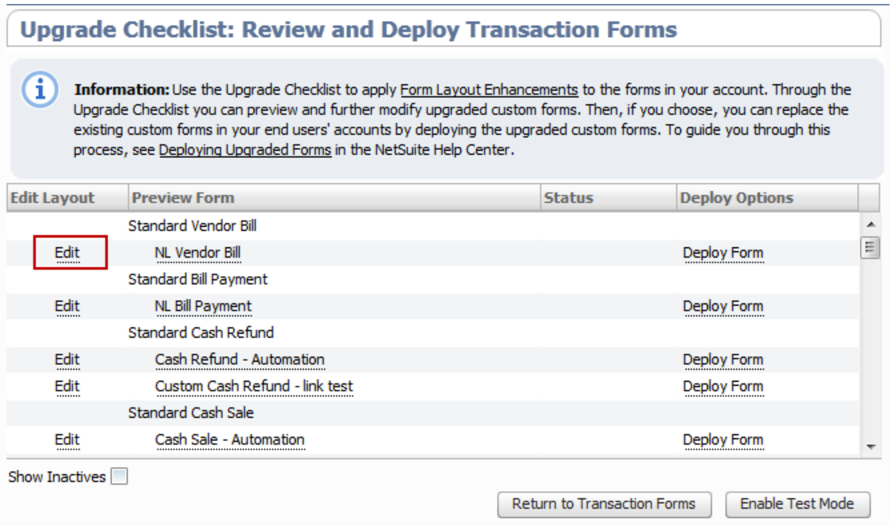
Important: During the form upgrade/deployment process, you must go through the Upgrade Checklist to make form **layout** modifications that apply to the upgraded form. You may continue to make changes to the old form through the forms list pages (Setup > Customization > Transaction (*or Entry*) Forms), however, these layout changes will be lost once the upgraded form is deployed.

To edit the layout of undeployed custom forms:

1. Navigate to the Upgrade Checklist:

1. Go to Setup > Customization > Transaction (*or Entry*) Forms.
2. On the forms list page, in the message area at the top, click the Upgrade Checklist link. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.

2. In the **Edit Layout** column, click Edit next to the custom form you have previewed and now want to modify.



Upgrade Checklist: Review and Deploy Transaction Forms

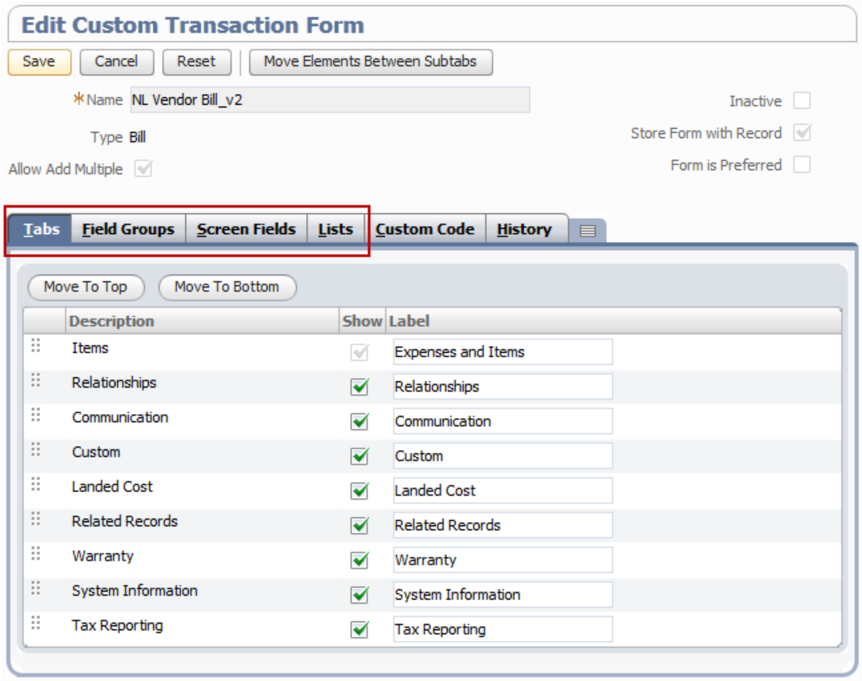
Information: Use the Upgrade Checklist to apply Form Layout Enhancements to the forms in your account. Through the Upgrade Checklist you can preview and further modify upgraded custom forms. Then, if you choose, you can replace the existing custom forms in your end users' accounts by deploying the upgraded custom forms. To guide you through this process, see Deploying Upgraded Forms in the NetSuite Help Center.

Edit Layout	Preview Form	Status	Deploy Options
	Standard Vendor Bill		
Edit	NL Vendor Bill		Deploy Form
	Standard Bill Payment		
Edit	NL Bill Payment		Deploy Form
	Standard Cash Refund		
Edit	Cash Refund - Automation		Deploy Form
Edit	Custom Cash Refund - link test		Deploy Form
	Standard Cash Sale		
Edit	Cash Sale - Automation		Deploy Form

Show Inactives □

3. Use the page that appears to make your additional layout changes (see figure below).

Important: The custom form edit page you access through the Upgrade Checklist should be used to modify ONLY the layout aspects of the form. Use this page to modify tabs, field groups, screen fields, and lists. **Do not use this page to modify any other properties of the form**. All other form properties related to roles, form default settings and permissions are inherited from the custom form currently in use. If you want to change any of these settings, you must do so by accessing the form through the custom forms list page (Setup > Customization > Transaction (*or Entry*) Forms.)



Edit Custom Transaction Form

*Name NL Vendor Bill_v2

Type Bill

Allow Add Multiple ☑

Inactive □

Store Form with Record ☑

Form is Preferred □

Description	Show	Label
Items	☑	Expenses and Items
Relationships	☑	Relationships
Communication	☑	Communication
Custom	☑	Custom
Landed Cost	☑	Landed Cost
Related Records	☑	Related Records
Warranty	☑	Warranty
System Information	☑	System Information
Tax Reporting	☑	Tax Reporting

4. After making your layout changes, click Save.
5. Navigate back to the Upgrade Checklist:
    1. Go to Setup > Customization > Transaction (*or Entry*) Forms.
    2. On the forms list page, in the message area at the top, click Upgrade Checklist. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.
6. Preview the form. The layout changes you made in Step Step 3 will reflect in the preview.
7. If you want to further modify the layout, repeat steps Step 2–Step 6.
8. When you are satisfied with the layout, you can test the form to verify it works as expected. See Testing Undeployed Custom Forms for details.

Avoid Editing Custom Forms in Tandem

NetSuite does not recommend that you simultaneously edit the forms that are currently in use (in your production environment) and their corresponding forms accessed through the Upgrade Checklist. For example, it is not recommended that you edit the layout of **Custom Form A** that is currently used in production, while you also edit the “preview version” of **Custom Form A** accessible through the Upgrade Checklist. Editing both versions will cause the undeployed preview version to get out of synch with the version in production.

Instead, NetSuite recommends you pick one version of the form to modify: either the version currently in production, or the (upgraded) undeployed preview version accessed through the Upgrade Checklist.

If you choose to edit the production version, then when you preview the form through the Upgrade Checklist, the preview will show the *Form Layout Enhancements* – as they apply to **ALL** the modifications you have made to the form. Note that if you continue to modify the production version after you have previewed it, the two versions of the form will be out of sync. *Knowing When to Skip Upgrade of Custom Forms*

If you choose to make all form layout edits to the upgraded/undeployed “preview” version, you will know that when you are ready to deploy the form, none of the modifications you have made will be lost. This cannot be said of the modifications made to the “production” version currently in use.

Testing Undeployed Custom Forms

When you test an undeployed custom form, you are essentially just viewing the form in the context of a record page. When in test mode, you have the opportunity of viewing the form as it appears in New mode and Edit mode.

When it comes to testing SuiteScripts on custom forms, you will be testing any client or user event scripts associated with the record type or the form.

To test undeployed custom forms:

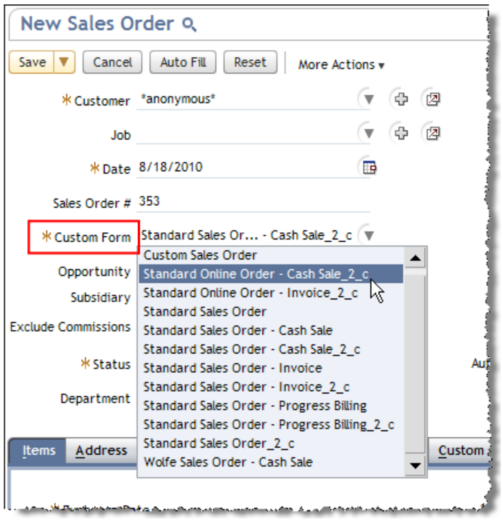
1. Navigate to the Upgrade Checklist:
    1. Go to Setup > Customization > Transaction (*or Entry*) Forms.
    2. On the forms list page, in the message area at the top, click Upgrade Checklist. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.
2. Click the Enable Test Mode button on the bottom of the Upgrade Checklist.
3. Use the navigation menus to go to the form you want to test.

For example, if you want to test a custom sales order form, do one or all of the following:

- Go to Transactions > Sales > Enter Sales Orders > New (to test a form in **New** mode)
- Go to Transactions > Sales > Enter Sales Orders > List > and click Edit next to an existing record (to test a form in **Edit** mode)

4. In the page that appears, from the Custom Form dropdown, select the custom form you want to test (see figure).

Note: When in Test Mode, all custom forms that appear in the Custom Form dropdown will appear with the upgraded layout. This includes both deployed and undeployed custom forms.


New Sales Order

*Customer *anonymous*
Job
*Date 8/18/2010
Sales Order # 353
*Custom Form Standard Sales Or... - Cash Sale_2_c

Save
Cancel
Auto Fill
Reset
More Actions
Opportunity
Subsidiary
Exclude Commissions
*Status
Department
Items
Address
Custom
Au
Custom Sales Order
Standard Online Order - Cash Sale_2_c
Standard Online Order - Invoice_2_c
Standard Sales Order
Standard Sales Order - Cash Sale
Standard Sales Order - Cash Sale_2_c
Standard Sales Order - Invoice
Standard Sales Order - Invoice_2_c
Standard Sales Order - Progress Billing
Standard Sales Order - Progress Billing_2_c
Standard Sales Order_2_c
Wolfe Sales Order - Cash Sale

5. Notice that the page layout changes to include field group and subtab *Form Layout Enhancements*. Verify that the upgraded layout will suit your business needs.

6. If you have any client or user event scripts associated with the form or the record type your are testing, verify that the scripts execute as expected.

7. After testing your custom form, navigate back to the Upgrade Checklist by going to Setup > Customization > Transaction (*or Entry*) Forms, and clicking Upgrade Checklist at the top of the page. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.

8. On the Upgrade Checklist, click the Disable Test Mode button to take the form out of test mode.

9. After testing a custom form, you can now deploy the upgraded form to end users. See Deploying Custom Forms for details.

Deploying Custom Forms

After you have previewed a custom form, edited its layout (if necessary), and tested it thoroughly, you can deploy the upgraded form into the accounts of all users.

Once you deploy a custom form, you will notice that its status on the Upgrade Checklist changes to Deployed. You will also notice that all links to the form are removed from the Upgrade Checklist. After a custom form has been deployed, if you want to access the form to make further changes, you will access the form through the custom forms list page (Setup > Customization > Transactions Forms (*or Entry*) Forms). The Upgrade Checklist is used to facilitate the form deployment process ONLY. It is not meant to be used to access forms that have been deployed.

To deploy custom forms:

1. Navigate to the Upgrade Checklist:
    1. Go to Setup > Customization > Transaction (*or Entry*) Forms.
    2. On the forms list page, in the message area at the top, click the Upgrade Checklist link. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.
2. In the Upgrade Checklist, in the Deploy Options column, click the **Deploy Form** link for the custom form you want to deploy.

    After clicking the link, notice that the status for the form changes to Deployed.

After you have deployed a custom form, the form will be available in the accounts of all of your end users. If the custom form you deploy is currently set as the “Preferred Form” for all users or for users with specific roles, your users will begin using this form immediately once it is deployed.

Deploying Skipped Custom Forms

Use the following steps to deploy a custom form that you had previously “skipped.”

Note that, as of Version 2012 Release 2, the form upgrade process has been simplified to remove the Skip Upgrade option. The following procedure applies to any custom forms for which you skipped upgrade prior to that release.

Important: Field groups created on existing custom forms that were skipped will not be carried over in the new upgraded form. You will notice this when you preview the upgraded custom form.

To deploy skipped custom forms:

1. Navigate to the Upgrade Checklist:

   1. Go to Setup > Customization > Transaction (*or Entry*) Forms.
   2. On the forms list page, in the message area at the top, click Upgrade Checklist. If you have already upgraded forms in your account, the link in the message area is called Return to Upgrade Checklist.

2. Next to the custom form you want to deploy, in the **Deploy Options** column, click Undo Skip and Upgrade Form. After clicking this link, the Edit link reappears next to the form, and the form itself becomes a link.
3. In the **Preview Name** column, click the custom form to preview.

   The form that appears will have the layout enhancements applied according to NetSuite's logic for applying these enhancements. See Understanding Form Layout Enhancement Upgrade Logic for details.

   Notice the following:

   - The new *Field Groups* that automatically render when you preview the form.
   - The *Subtabs and Sublists* that have been added or removed.
   - If you had previously added custom field groups to these forms, when you upgrade the layout and deploy the form, all standard fields you might have placed on a custom field group will automatically be put into the fields groups NetSuite considers appropriate. All custom fields you had placed in a custom field group will appear at the bottom of the main area of the form and will not have a field group title. You will have to manually edit those fields and place them on new field groups.
4. If you decide you approve of most of the changes to your form, but you need to make additional layout modifications before deploying the form to users, see Editing the Layout of Custom Forms Prior to Deployment.
5. Next, test the upgraded form. See Testing Undeployed Custom Forms.
6. After thoroughly previewing and testing the custom form, click the Deploy Form link.

Understanding Form Deployment Statuses

Form deployment statuses appear in the Upgrade Checklist and on the transaction and entry custom form list pages .

Upgrade Checklist Deployment Statuses

In the **Status** column on the Upgrade Checklist you will see these statuses for custom forms:

Status	Description
< No status at all >	The administrator has not performed any action on the form. The form has not been deployed (by clicking the Deploy Form link) .
Deployed	The administrator has clicked the Deploy Form link and deployed the form. The deployed form includes the Version 2010 Release 2 *Form Layout Enhancements* and is now in the accounts of all end users.
Skipped Upgrade	The administrator has clicked the Skip Upgrade link and has explicitly chosen to skip the deployment of the form. The form will retain its existing layout. **Important:** As of Version 2012 Release 2, the Skip Upgrade option is no longer available, but any form that was skipped prior to that release continues to display a status of Skipped Upgrade until the administrator deploys the form.

Understanding Form Layout Enhancement Upgrade Logic

When you preview custom forms, NetSuite executes a backend process that automatically applies *Form Layout Enhancements* to the existing versions of the forms. These enhancements include the arrangement of fields into logical *Field Groups* and the standardization of *Subtabs and Sublists*.

The goal of the backend upgrade process is to create a form layout that is better organized and more consistent. The other goal is to upgrade the layout while maintaining form customizations where ever possible.

See the following sections to learn about the NetSuite upgrade logic as it applies to custom forms, standard and custom fields, subtabs, and custom field record pages used to add or edit custom fields. These topics do not need to be read in order.

- *Upgrade Logic for Standard Forms*
- Upgrade Logic for Custom Forms
- Upgrade Logic for Subtabs
- Upgrade Logic for Fields (Diagram)
- Upgrade Logic for the Custom Field Record Page

For information on field ordering as it applies to form upgrades, see Field Ordering.

Upgrade Logic for Custom Forms

When the backend upgrade process runs against a custom form, the standard fields on the form are rearranged to match the new field location of the standard fields on the upgraded standard form. For example, when a custom Customer form is upgraded, the standard fields on the **custom** Customer form are rearranged to match the layout of the standard fields on the upgraded **standard** Customer form.

Important: The exception to this is standard fields placed on a custom subtab. When a custom form is upgraded, standard fields that have been placed on a custom subtab will remain on the custom subtab.

The location of custom fields on custom forms will also remain unchanged during the upgrade process. Custom fields will only be moved if they are on a subtab that is removed during the upgrade process.

See the following table for details. Also see the Upgrade Logic for Fields (Diagram) for a visual description of these field upgrade changes.

Existing Form ------------------------------>	Upgraded Form
Custom Fields	
Custom fields that are on a valid standard subtab.	Will be kept on the same subtab when the form is upgraded.
Custom fields on a subtab that is removed when the form is upgraded.	Will be moved to a subtab called **Custom**. After previewing the new auto-generated layout, you can manually edit the layout and move your custom fields off of the Custom subtab. See Editing the Layout of Custom Forms Prior to Deployment.
Custom fields that are on a custom subtab.	Will be kept on the same custom subtab.
Custom fields that are in the main area of the form.	Will be placed at the bottom of the main area, below the auto-generated field groups.

After previewing upgraded **custom** forms, if you are not satisfied with NetSuite's auto-generated layout, you can reassign fields to different groups, relabel or remove field groups, and create your own custom fields groups. You can also move fields between the new subtabs that were auto-generated. For more information, see Previewing Undeployed Custom Forms and Editing the Layout of Custom Forms Prior to Deployment.

To create custom layouts, go to Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and click the Customize link next to a layout. Make your changes and click Save. You can choose default layouts to apply to one or more types of forms by checking box(es) in the Preferred column at Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and clicking Submit.

Upgrade Logic for Subtabs

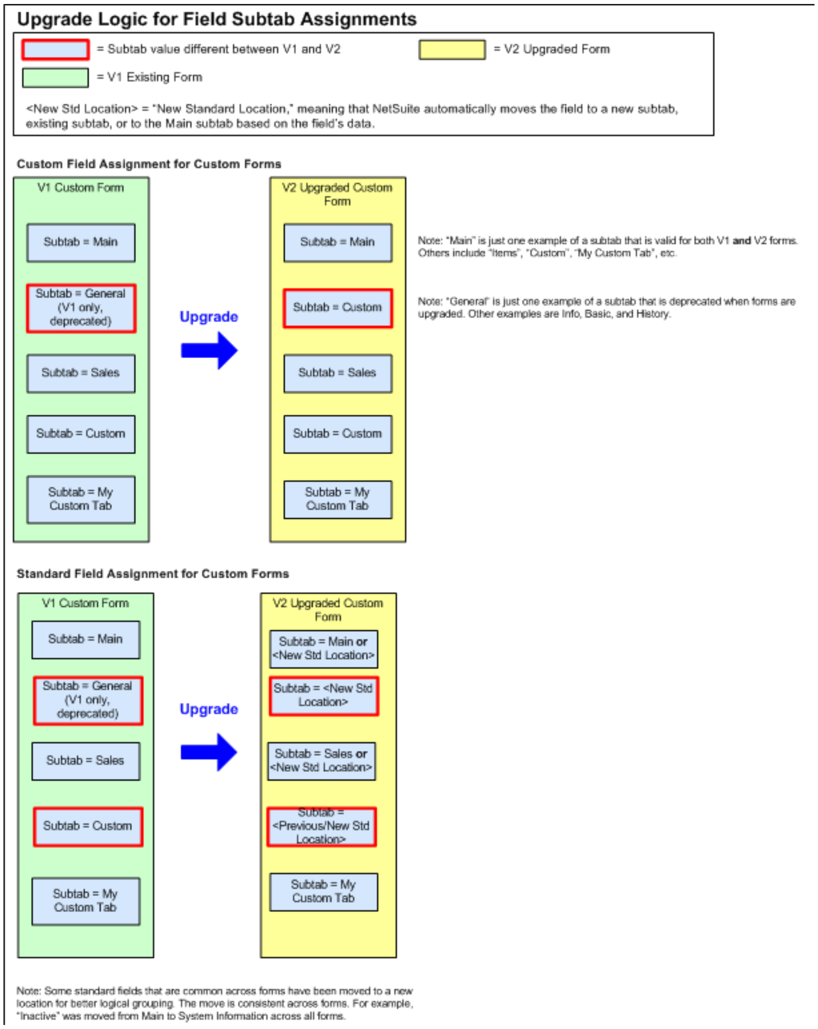
In addition to the creation of field groups on upgraded custom forms, NetSuite's backend upgrade process also reorganizes and relabels certain subtabs and sublists. When forms are upgraded, certain subtabs are added (depending on the form type), and others are removed. Although certain subtabs are removed during the upgrade process, the data on these subtabs are simply moved to another (more logical) area on the form. This new location can be another subtab or it can be under a field group. (See the help topic *Subtabs and Sublists* for more information about the subtabs that are added or removed during the form upgrade process.)

To create custom layouts, go to Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and click the Customize link next to a layout. Make your changes and click Save. You can choose default layouts to apply to one or more types of forms by checking box(es) in the Preferred column at Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and clicking Submit.

Upgrade Logic for Fields (Diagram)

The following diagram visually represents the field arrangement concepts discussed in *Upgrade Logic for Standard Forms* Upgrade Logic for Custom Forms. Technically, all NetSuite fields appear on subtabs, even fields that appear in the main body of the form are (technically speaking) located on a subtab called **Main**.

The following diagram shows the subtab assignment for fields in a “V1” (non-upgraded) environment and a “V2” (upgraded) environment. Boxes outlined in red highlight the scenarios where a field's location (subtab) changes when a form is upgraded.



Upgrade Logic for Field Subtab Assignments

<New Std Location> = "New Standard Location," meaning that NetSuite automatically moves the field to a new subtab, existing subtab, or to the Main subtab based on the field's data.

Note: "Main" is just one example of a subtab that is valid for both V1 **and** V2 forms. Others include "Items", "Custom", "My Custom Tab", etc.

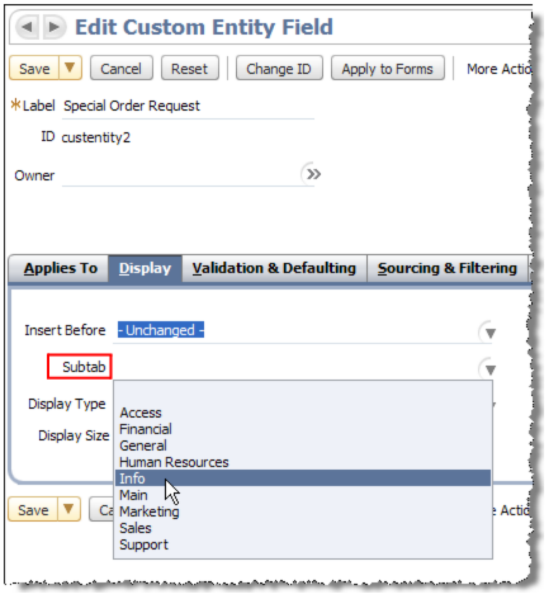
Note: "General" is just one example of a subtab that is deprecated when forms are upgraded. Other examples are Info, Basic, and History.

Note: Some standard fields that are common across forms have been moved to a new location for better logical grouping. The move is consistent across forms. For example, "Inactive" was moved from Main to System Information across all forms.

To create custom layouts, go to Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and click the Customize link next to a layout. Make your changes and click Save. You can choose default layouts to apply to one or more types of forms by checking box(es) in the Preferred column at Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and clicking Submit.

Upgrade Logic for the Custom Field Record Page

The form upgrade logic also affects the custom field record page. The following figure shows the custom entity field record page for a custom field called **Special Order Request**. On the Display subtab, notice that the available subtabs include "V1-only" subtabs. These are subtabs such as **General** and **Info** that are removed in the upgraded environment that includes *Form Layout Enhancements*.



*Label Special Order Request

ID custentity2

Owner

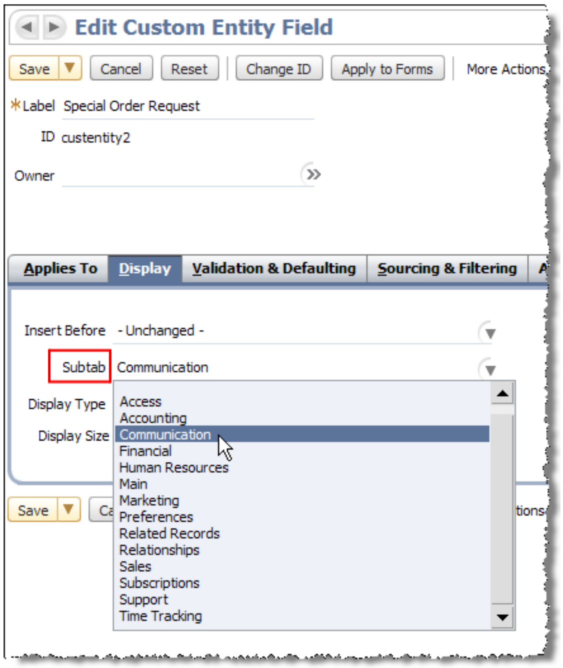
Insert Before - Unchanged -

Subtab

Display Type

Display Size

The next figure shows the same custom field record page, however, the Subtab dropdown now shows all “V2” subtabs. These include subtabs such as Accounting, Communication, Preferences, and so on. *Figure 2* shows the custom field record page in an **upgraded** environment.



Edit Custom Entity Field

*Label Special Order Request

ID custentity2

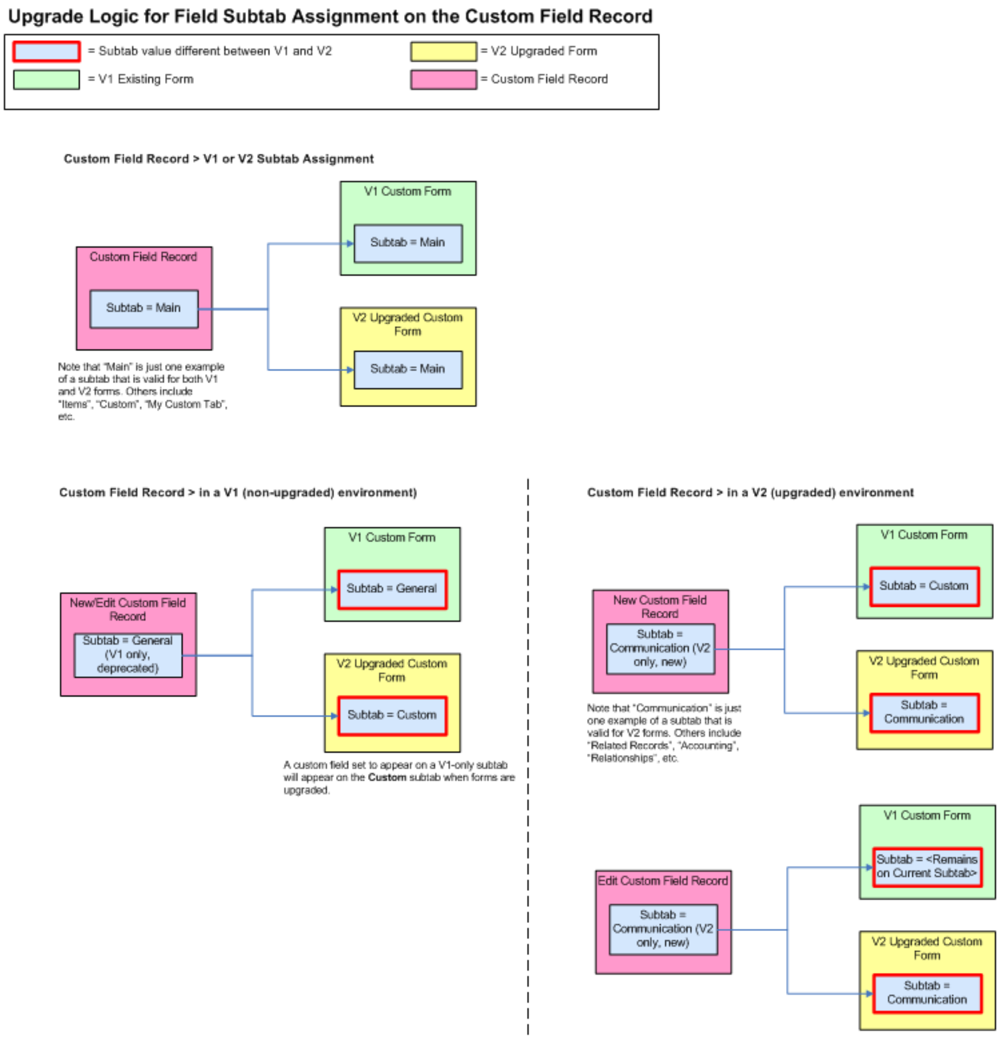
Owner

Insert Before - Unchanged -

Subtab Communication

Once standard forms have been upgraded, the custom field record pages automatically show all of the new subtabs that come with the *Form Layout Enhancements*. Note that as of Version 2012 Release 2, all standard forms have been upgraded to be "V2", so all accounts have a V2 environment.) The subtabs that are available in a non-upgraded environment are removed from the custom field record pages.

The following diagram shows the differences in subtabs currently available to custom fields in the V2 (upgraded) environment versus previously in the V1 (non-upgraded) environment.


Upgrade Logic for Field Subtab Assignment on the Custom Field Record
🟥 = Subtab value different between V1 and V2
🟨 = V2 Upgraded Form
🟩 = V1 Existing Form
🟪 = Custom Field Record
Custom Field Record > V1 or V2 Subtab Assignment
Custom Field Record
Subtab = Main
V1 Custom Form
Subtab = Main
V2 Upgraded Custom Form
Subtab = Main
Custom Field Record > in a V1 (non-upgraded) environment
New/Edit Custom Field Record
Subtab = General (V1 only, deprecated)
V1 Custom Form
Subtab = General
V2 Upgraded Custom Form
Subtab = Custom
Custom Field Record > in a V2 (upgraded) environment
New Custom Field Record
Subtab = Communication (V2 only, new)
V1 Custom Form
Subtab = Custom
V2 Upgraded Custom Form
Subtab = Communication
Edit Custom Field Record
Subtab = Communication (V2 only, new)
V1 Custom Form
Subtab = <Remains on Current Subtab>
V2 Upgraded Custom Form
Subtab = Communication

Note that "Main" is just one example of a subtab that is valid for both V1 and V2 forms. Others include "Items", "Custom", "My Custom Tab", etc.

A custom field set to appear on a V1-only subtab will appear on the **Custom** subtab when forms are upgraded.

Note that "Communication" is just one example of a subtab that is valid for V2 forms. Others include "Related Records", "Accounting", "Relationships", etc.

To create custom layouts, go to Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and click the Customize link next to a layout. Make your changes and click Save. You can choose default layouts to apply to one or more types of forms by checking box(es) in the Preferred column at Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and clicking Submit.

Field Ordering

Customized ordering of fields will not be respected when a form is upgraded. The exception is the order of fields on custom tabs. Their ordering will not change when *Form Layout Enhancements* are applied.

To create custom layouts, go to Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and click the Customize link next to a layout. Make your changes and click Save. You can choose default layouts to apply to one or more types of forms by checking box(es) in the Preferred column at Setup > Customization > Forms > Transaction Form PDF Layouts or Setup > Customization > Forms > Transaction Form HTML Layouts, and clicking Submit.