



---

# Salesforce Billing



© Copyright 2000–2026 salesforce.com, inc. All rights reserved. Salesforce is a registered trademark of salesforce.com, inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

# CONTENTS

<b>Manage Your Billing Processes with Salesforce Billing .....</b>	<b>1</b>
Salesforce Billing Patch Notes .....	4
Salesforce Billing Installation .....	4
Salesforce Billing Overview.....	26
Preparing Your Salesforce Org for Billing .....	58
Managing the Billing Order.....	112
Usage Rating and Processing .....	213
Invoice Generation.....	243
Applying Taxes .....	282
Payments and Credits .....	299
Understanding the Revenue Recognition Process .....	391
Closing and Reopening Finance Periods .....	440
Reporting Essentials.....	443
Finance Logging .....	448
Salesforce Billing Tableau Dashboards .....	462
Understand What Your Customers Have Bought by Using Customer Asset Lifecycle Management.....	468
Printable Tip Sheets & User Guides – For Salesforce Billing .....	487



# Salesforce Billing

Create and automate your invoices, payments, and revenue with Salesforce Billing. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

**!** **Important** The Salesforce Billing managed package continues to be available for existing customers, however, there is no longer any new feature development. We will continue to provide support and services for the duration of your contract. You can also add more billing blocks to your Salesforce org during your current subscription term and renew existing subscriptions. For a more comprehensive and robust Billing solution, we recommend exploring [Billing in Revenue Cloud](#).

### [Salesforce Billing Patch Notes](#)

Salesforce Billing patch notes contain high-level reviews of bug fixes included in each generally available patch for a Billing package. We update patch notes after a new patch becomes generally available for a package. (Salesforce Billing Managed Package)

### [Salesforce Billing Installation](#)

Manage the setup and installation of Salesforce Billing (Salesforce Billing Managed Package)

### [Salesforce Billing Overview](#)

Create and automate your invoices, payments, and revenue with Salesforce Billing. (Salesforce Billing Managed Package)

### [Preparing Your Salesforce Org for Billing](#)

Salesforce Billing allows you to automate the billing process and keep detailed records of your transactions. You can set up several product fields and rules to guide the automation. (Salesforce Billing Managed Package)

### [Managing the Billing Order](#)

The order record defines important fields about when and how Salesforce Billing invoices your order products. While many of these fields are set by default based on package settings, rules, and treatments, you can also edit them based on your business needs. (Salesforce Billing Managed Package)

### [Usage Rating and Processing](#)

Usage products are billed based on a consumed amount of service, such as an electricity bill. Vendors will not know how much of the service was used, and therefore how much to bill the customer, until the usage period has elapsed. Salesforce Billing helps you organize usage-based products and invoice them based on the total amount of usage. (Salesforce Billing Managed Package)

### [Invoice Generation](#)

Invoices display a list of purchased items and services alongside the total amount a customer must pay. The invoice record itself contains important details such as the balance, due date, and payment status. You can control several date fields in the Salesforce CPQ and Salesforce Billing packages to manage the number of invoices generated from an order, their billing dates, and the order products converted to invoice lines. (Salesforce Billing Managed Package)

### Applying Taxes

Salesforce Billing can use internal or external tax engines to provide automatic tax calculations on your order products and invoice lines. The tax integration object acts as a bridge between your org and the external service. When you set up your org, you'll need to configure a few basic tax settings and then create your tax integration. (Salesforce Billing Managed Package)

### Payments and Credits

Collect payments against posted invoices. Salesforce Billing lets you manually collect and allocate payments or automate the payment process. You can then post the payment to keep your books up-to-date. (Salesforce Billing Managed Package)

### Understanding the Revenue Recognition Process

Salesforce Billing uses several objects to manage the revenue recognition system. Once you define these objects and their relationships, you can automate most of the revenue recognition process. (Salesforce Billing Managed Package)

### Closing and Reopening Finance Periods

Finance departments close accounting and revenue finance periods at the end of each accounting period. After a period closes, the finance department can prepare statements and users can review their financial status. Finance periods need to close at least once a year, though most businesses close them monthly for easier bank statement reconciliation, sales tax report submission, and paying and sending out invoices. Salesforce Billing allows for convenient creation, management, and closing of your business's finance periods. (Salesforce Billing Managed Package)

### Reporting Essentials

Salesforce Billing provides a collection of customizable options for reporting on stages within the billing process. (Salesforce Billing Managed Package)

### Finance Logging

Finance transactions show details about a financial action performed against one of your financial records. Finance balance snapshots show details on the state of a header-level financial record's financially significant values following an action. Use Salesforce and Tableau reports to track important financial information and address aging or bad debt circumstances across all the records affected by a financial action. (Salesforce Billing Managed Package)

### Salesforce Billing Tableau Dashboards

Salesforce Billing provides a collection of Tableau dashboards that help you get business insights from your Salesforce Billing data. Each dashboard comes with a default layout that you can customize and extend based on your data tracking needs. (Salesforce Billing Managed Package)

### Understand What Your Customers Have Bought by Using Customer Asset Lifecycle Management

Customer Asset Lifecycle Management gives you visibility into products your customers have bought, from initial sale through the end date of a subscription or service. As an account, sales, or service rep, you see an asset's quantity, amount, and monthly recurring revenue at any point during an asset's lifecycle. You can also see related invoice lines, the source of a change, and other information. Your

business consolidates purchases and changes in one system, making subscriptions and other complex products easier to manage, and showing trends in a dashboard and reports. Developers or integrators automate creation of, changes to, and cancellation of lifecycle-managed assets using objects and fields that enhance the Asset object. (Salesforce Billing Managed Package)

**Printable Tip Sheets & User Guides – For Salesforce Billing**

Printable implementation guides for Salesforce Billing admins. (Salesforce Billing Managed Package)

# Salesforce Billing Patch Notes

---

Salesforce Billing patch notes contain high-level reviews of bug fixes included in each generally available patch for a Billing package. We update patch notes after a new patch becomes generally available for a package. (Salesforce Billing Managed Package)

- [Salesforce Billing Summer '23 Patch Notes](#)
- [Salesforce Billing Spring '23 Patch Notes](#)
- [Salesforce Billing Winter '23 Patch Notes](#)
- [Salesforce Billing Summer '22 Patch Notes](#)
- [Salesforce Billing Spring '22 Patch Notes](#)
- [Salesforce Billing Winter '22 Patch Notes](#)
- [Salesforce Billing Summer '21 Patch Notes](#)
- [Salesforce Billing Spring '21 Patch Notes](#)
- [Salesforce Billing Winter '21 Patch Notes](#)
- [Salesforce Billing Summer '20 Patch Notes](#)
- [Salesforce Billing Spring '20 Patch Notes](#)
- [Salesforce Billing Winter '20 Patch Notes](#)
- [Salesforce Billing Summer '19 Patch Notes](#)
- [Salesforce Billing Spring '19 Patch Notes](#)
- [Salesforce Billing Winter '19 Patch Notes](#)
- [Salesforce Billing Summer '18 Patch Notes](#)
- [Salesforce Billing Spring '18 Patch Notes](#)
- [Salesforce Billing Winter '18 Patch Notes](#)

# Salesforce Billing Installation

---

Manage the setup and installation of Salesforce Billing (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### [Install Salesforce Billing](#)

Install Salesforce Billing from your package installation link. (Salesforce Billing Managed Package)

### [Avoid Translation Problems](#)

To avoid language translation problems when users change languages, override the translated labels. (Salesforce Billing Managed Package)

### [Configure Installed Packages](#)

Use Salesforce Billing to install a collection of basic rules, treatments, and finance books. These records are useful for testing in sandbox orgs. (Salesforce Billing Managed Package)

### Set Up Page Layouts for Salesforce Billing

After installing Salesforce Billing, change key page layouts to their Billing versions and edit your page layouts to show important billing fields. (Salesforce Billing Managed Package)

### Configure Products for Salesforce Billing

After you've configured your products for CPQ, configure them for billing by providing values on several key fields. (Salesforce Billing Managed Package)

### Salesforce Billing Permission Requirements

To support admin and user processes for Salesforce Billing, customize profile permissions. Each profile permission contains read, create, edit, or delete access to different objects. (Salesforce Billing Managed Package)

### Mapping Custom Salesforce Billing Fields Between Objects

Certain pairs of Salesforce Billing objects pass custom field values from the first object to the second object when the second object is created. The values pass if the custom fields are editable, have matching field types, and have matching API names. We call these field pairs "twin fields." (Salesforce Billing Managed Package)

### Billing Package Settings

Package settings control the properties of feature areas within Salesforce Billing. (Salesforce Billing Managed Package)

## Install Salesforce Billing

Install Salesforce Billing from your package installation link. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Salesforce Billing requires Salesforce CPQ. We recommend that you install the same release of Salesforce Billing as Salesforce CPQ. For example, install Billing Spring '19 with CPQ Spring '19.

Salesforce Billing requires several Salesforce CPQ package settings.

- In the Subscriptions and Renewals settings, if your org uses Percent of Total (PoT) products that cover assets, you must deselect the PoT Renewals (Contracting from Orders) setting.
- In the Subscriptions and Renewals settings, the Subscription Term Unit field must have a value of month.

1. From your installation link, click **Continue**.

2. Select the security level to assign to users during installation.

We recommend that you grant access to all users to avoid spending extensive time later to adjust permissions.

3. Click **Install**.

Salesforce sends you an email after your installation is complete.

## Avoid Translation Problems

To avoid language translation problems when users change languages, override the translated labels. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. Log in to your org as a user whose language is to English.
2. From Setup, in the Quick Find box, search for and select **Custom Labels**.
3. Sort the custom labels by name.
4. Use the alphabet list to go to the custom labels that start with C.
5. Select **Calculation Proration Type Calendar Days** and override translations for all languages to calendar days.
6. Select **Calculation Proration Type CPQ Month** and override translations for all languages to monthly (CPQ formula).
7. In the Quick Find box, search for and select **Installed Packages**.
8. Configure the Salesforce Billing installed package.
9. From the Invoice tab, select a proration type value.
10. Save the changes.
11. Validate your setting for other updated languages.

Upgrading a package can interrupt active APEX jobs. After you upgrade Salesforce Billing, check for failures in your active invoice runs, payment runs, and balance snapshots, and then restart the jobs where necessary.

Based on your company's Salesforce Billing implementation, you must set up tax integration and payment gateway integrations.

### See Also

[Tax Integrations](#)

[Processing Payments with Payment Gateways](#)

## Configure Installed Packages

Use Salesforce Billing to install a collection of basic rules, treatments, and finance books. These records are useful for testing in sandbox orgs. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

You can install a collection of basic rules, treatments, and finance books from the Salesforce Billing package settings page.

1. From Setup, in the Quick Find box, search for and select **Installed Packages**.
2. Find the Salesforce Billing package and click **Configure**.
3. Click the **Additional Settings** tab.
4. Click **Insert Sampledata**.

As of Winter '21, Salesforce Billing manages user licensing with Permission Set Licenses instead of Managed Package Licenses. Fields and settings related to Managed Package Licenses in Salesforce Billing are deprecated. On the Installed Packages page, the Allowed Licenses and Expiration Date fields refer to managed packages. As a result of the deprecation, they now show Unlimited and Does not Expire for Salesforce Billing.

## Set Up Page Layouts for Salesforce Billing

After installing Salesforce Billing, change key page layouts to their Billing versions and edit your page layouts to show important billing fields. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. Change your page layout to the Billing layout version for the following objects.
    - a. Account
    - b. Order
    - c. Order Product
    - d. Product
    - e. Consumption Schedule
    - f. Order Product Consumption Schedule
  2. On your account page layout, add the Bill To Contact field.
  3. On the credit note and debit note page layouts, remove the Cancel Tax button and Cancel Tax lightning action.
-  **Note** Salesforce Billing doesn't currently support the Cancel Tax button. We'll let you know if that changes.

## Configure Products for Salesforce Billing

After you've configured your products for CPQ, configure them for billing by providing values on several key fields. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. Choose a billing rule.

The billing rule controls how Salesforce Billing handles billing triggers and amended orders, and whether it creates an order for an invoice product. It also contains billing treatments, which let you further customize how Salesforce Billing invoices order products.

## 2. Choose a tax rule.

The tax rule controls whether Salesforce Billing applies tax to an order product. If the Taxable (Yes/No) field is No, the rule won't have tax treatments. Otherwise, the rule's tax treatments define the tax integration and tax code for related order products.

## 3. Choose a revenue recognition rule.

The revenue recognition rule controls whether Salesforce Billing creates a revenue schedule for an order product. It also contains revenue treatments, which let you further customize revenue recognition reporting through your revenue schedule.

## 4. Choose a charge type.

Charge types define whether your products bills once, as a recurring subscription, or based on usage. Salesforce Billing evaluates the charge type when calculating billing date for an order product.

## 5. Choose a billing frequency.

A product's billing frequency determines how often Salesforce Billing bills an order product.

## 6. Choose a billing type.

Advance billing invoices a product or service before you provide it, while arrears billing invoices a product or service after you provide it. Salesforce Billing evaluates billing type when calculating an order product's next billing date.



**Note** The Picklist Value Sets for billing frequency, charge type, and billing type don't support edited values or custom values. However, default values in the sets can be deactivated.

## Salesforce Billing Permission Requirements

To support admin and user processes for Salesforce Billing, customize profile permissions. Each profile permission contains read, create, edit, or delete access to different objects. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

**!** **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

---

Available in: All Salesforce Billing Editions

---

Salesforce Billing comes with a permission set that grants permissions for the package's custom objects. However, some Billing actions involve or are related to standard objects. As a managed package, Salesforce Billing can't give permissions for standard objects, so you must provide those permissions to users manually.

- Some Salesforce Billing objects, such as invoices and payments, have a master-detail relationship to the Account object. When you assign permissions for objects with a master-detail Account relationship, include access to the Account object as well.

- The Order and Order Product objects are standard objects with standard fields, Salesforce CPQ fields, and Salesforce Billing fields. When you assign permissions for objects related to orders and order products, include access to the Order or Order Product as needed.

 **Note** In Salesforce Billing, you can adjust picklist value labels, but adjusting picklist value API names isn't supported. While it's possible to adjust picklist value API names in Salesforce, doing so causes any instance of the picklist to malfunction.

## Invoice Actions

Action	Read	Create	Edit	Delete
Create an invoice using Bill Now	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Sub Invoice Line</li> </ul>	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Sub Invoice Line</li> </ul>	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Sub Invoice Line</li> </ul>	None
Run an invoice scheduler	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Sub Invoice Line</li> </ul>	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Sub Invoice Line</li> </ul>	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Sub Invoice Line</li> </ul>	None
Post an invoice	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Sub Invoice Line</li> <li>• Price Book</li> </ul>	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Sub Invoice Line</li> </ul>	<b>Starting Winter '20 for non-admins</b> <ul style="list-style-type: none"> <li>• Billing Transaction</li> <li>• Credit Note</li> <li>• Credit Note line</li> <li>• Credit Note's Status field</li> <li>• Credit Note Line's Status field</li> <li>• Order</li> <li>• Order Product</li> </ul>	None

Action	Read	Create	Edit	Delete
			<ul style="list-style-type: none"> <li>Usage Summary</li> </ul>	
Cancel an invoice using the Cancel & Rebill button	<ul style="list-style-type: none"> <li>Credit Note</li> <li>Credit Note Line</li> <li>Credit Note Allocation</li> <li>Debit Note</li> <li>Debit Note Allocations (Invoice Line)</li> <li>Debit Note Lines</li> <li>Invoice</li> <li>Invoice Line</li> <li>Payment Allocations (Debit Note Line)</li> <li>Payment Allocations (Invoice)</li> <li>Payment Allocations (Invoice Line)</li> <li>Payments</li> <li>Sub Invoice Line</li> </ul>	<ul style="list-style-type: none"> <li>Credit Note</li> <li>Credit Note Line</li> <li>Credit Note Allocation</li> </ul>	<ul style="list-style-type: none"> <li>Invoice</li> <li>Invoice Line</li> <li>Credit Note</li> <li>Credit Note Line</li> <li>Credit Note Allocation</li> </ul>	None
Apply taxes through a tax integration such as Avalara	<ul style="list-style-type: none"> <li>Invoice</li> <li>Invoice Line</li> <li>Credit Note</li> <li>Credit Note Line</li> <li>Debit Note</li> <li>Debit Note Line</li> </ul>	<ul style="list-style-type: none"> <li>Invoice</li> <li>Invoice Line</li> <li>Credit Note</li> <li>Credit Note Line</li> <li>Debit Note</li> <li>Debit Note Line</li> </ul>	None	None
Apply standard tax	<ul style="list-style-type: none"> <li>Invoice</li> <li>Invoice Line</li> <li>Credit Note</li> <li>Credit Note Line</li> <li>Debit Note</li> </ul>	<ul style="list-style-type: none"> <li>Invoice</li> <li>Invoice Line</li> <li>Credit Note</li> <li>Credit Note Line</li> <li>Debit Note</li> </ul>	None	None

Action	Read	Create	Edit	Delete
	<ul style="list-style-type: none"> <li>• Debit Note Line</li> </ul>	<ul style="list-style-type: none"> <li>• Debit Note Line</li> </ul>		

## Payment Actions

Action	Read	Create	Edit	Delete
Allocate a Payment	<ul style="list-style-type: none"> <li>• Debit Note Allocation (Invoice Line)</li> <li>• Debit Note Lines</li> <li>• Invoice</li> <li>• Invoice Lines</li> <li>• Payment</li> <li>• Payment Allocation (Invoice)</li> <li>• Payment Allocation (Invoice Lines)</li> </ul>	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Payment</li> </ul>	<ul style="list-style-type: none"> <li>• Debit Note Lines</li> <li>• Invoice</li> <li>• Invoice Lines</li> <li>• Payment</li> <li>• Payment Allocation (Invoice)</li> <li>• Payment Allocation (Invoice Lines)</li> </ul>	None
Allocate a Credit Note	<ul style="list-style-type: none"> <li>• Credit Note</li> <li>• Credit Note Allocation</li> <li>• Credit Note Lines</li> <li>• Invoice Lines</li> </ul>	<ul style="list-style-type: none"> <li>• Credit Note</li> </ul>	<ul style="list-style-type: none"> <li>• Credit Note</li> <li>• Credit Note Allocation</li> <li>• Credit Note Lines</li> <li>• Invoice Lines</li> </ul>	None
Allocate a Debit Note	<ul style="list-style-type: none"> <li>• Credit Note Lines</li> <li>• Debit Note</li> <li>• Debit Note Allocation (Credit Note Line)</li> <li>• Debit Note Allocation (Invoice Line)</li> </ul>	<ul style="list-style-type: none"> <li>• Debit Note</li> <li>• Debit Note Allocation (Invoice Line)</li> </ul>	<ul style="list-style-type: none"> <li>• Credit Note Lines</li> <li>• Debit Note</li> <li>• Debit Note Allocation (Credit Note Line)</li> <li>• Debit Note Allocation (Invoice Line)</li> </ul>	None

Action	Read	Create	Edit	Delete
	<ul style="list-style-type: none"> <li>• Debit Note Lines</li> <li>• Invoice Lines</li> </ul>		<ul style="list-style-type: none"> <li>• Debit Note Lines</li> <li>• Invoice Lines</li> </ul>	
Create a credit note	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Credit Note</li> <li>• Credit Note Line</li> <li>• Credit Note Allocation</li> </ul>	<ul style="list-style-type: none"> <li>• Credit Note</li> <li>• Credit Note Line</li> <li>• Credit Note Allocation</li> </ul>	<ul style="list-style-type: none"> <li>• Credit Note</li> <li>• Credit Note Line</li> <li>• Credit Note Allocation</li> </ul>	None
Post a credit note	<ul style="list-style-type: none"> <li>• Credit Note</li> <li>• Credit Note Line</li> </ul>		<p><b>Starting Winter '20 for non-admins</b></p> <ul style="list-style-type: none"> <li>• Credit Note</li> <li>• Credit Note line</li> <li>• Credit Note's Status field</li> <li>• Credit Note Line's Status field</li> </ul>	None
Create a debit note	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Debit Note</li> <li>• Debit Note Line</li> <li>• Debit Note Allocation (Invoice Line)</li> <li>• Debit Note Allocation (Invoice Line)</li> <li>• Debit Note Allocation (Credit Note Line)</li> </ul>	<ul style="list-style-type: none"> <li>• Debit Note</li> <li>• Debit Note Line</li> <li>• Debit Note Allocation (Invoice Line)</li> <li>• Debit Note Allocation (Credit Note Line)</li> </ul>	<ul style="list-style-type: none"> <li>• Debit Note</li> <li>• Debit Note Line</li> <li>• Debit Note Allocation (Invoice Line)</li> <li>• Debit Note Allocation (Credit Note Line)</li> </ul>	None

Action	Read	Create	Edit	Delete
Post a debit note	<ul style="list-style-type: none"> <li>Credit Note</li> <li>Credit Note Line</li> </ul>		<b>Starting Winter '20 for non-admins</b> <ul style="list-style-type: none"> <li>Debit Note</li> <li>Debit Note line</li> <li>Debit Note's Status field</li> <li>Debit Note Line's Status field</li> </ul>	None
Create a payment	<ul style="list-style-type: none"> <li>Invoice</li> <li>Invoice Line</li> <li>Debit Note</li> <li>Debit Note Line</li> <li>Payment</li> <li>Payment Method</li> </ul>	<ul style="list-style-type: none"> <li>Payment</li> <li>Payment Allocation (Invoice)</li> <li>Payment Allocation (Invoice Line)</li> </ul>	<ul style="list-style-type: none"> <li>Payment</li> <li>Payment Allocation (Invoice)</li> <li>Payment Allocation (Invoice Line)</li> </ul>	None
Use the Payment Center	<ul style="list-style-type: none"> <li>Invoice</li> <li>Invoice Line</li> <li>Debit Note</li> <li>Debit Note Line</li> <li>Payment Method</li> <li>Payment Allocation (Invoice)</li> <li>Payment Allocation (Invoice Line)</li> <li>Payment Allocation (Debit Note Line)</li> </ul>	<ul style="list-style-type: none"> <li>Payment</li> <li>Payment Allocation (Invoice)</li> <li>Payment Allocation (Invoice Line)</li> <li>Payment Allocation (Debit Note Line)</li> <li>Payment Method</li> <li>Payment Transaction</li> </ul>	<ul style="list-style-type: none"> <li>Payment</li> <li>Payment Allocation (Invoice)</li> <li>Payment Allocation (Invoice Line)</li> <li>Payment Allocation (Debit Note Line)</li> <li>Payment Transaction</li> </ul>	None

Action	Read	Create	Edit	Delete
	<ul style="list-style-type: none"> <li>• Payment Transaction</li> </ul>			
Refund	<ul style="list-style-type: none"> <li>• Refund</li> <li>• Refund Line (Payment)</li> <li>• Refund Line (Credit Note Line)</li> <li>• Payment Allocation (Invoice)</li> <li>• Payment Allocation (Invoice Line)</li> <li>• Payment Allocation (Debit Note Line)</li> <li>• Payment</li> <li>• Payment Method</li> </ul>	<ul style="list-style-type: none"> <li>• Refund</li> <li>• Refund Line (Payment)</li> <li>• Refund Line (Credit Note Line)</li> </ul>	<ul style="list-style-type: none"> <li>• Refund</li> <li>• Refund Line (Payment)</li> <li>• Refund Line (Credit Note Line)</li> <li>• Payment Allocation (Invoice)</li> <li>• Payment Allocation (Invoice Line)</li> <li>• Payment Allocation (Debit Note Line)</li> <li>• Payment</li> </ul>	None

## Usage Actions

Action	Read	Create	Edit	Delete
Create a usage record	<ul style="list-style-type: none"> <li>• Consumption Schedule</li> <li>• Order</li> <li>• Order Product</li> <li>• Order Product Consumption Rate</li> <li>• Order Product Consumption Schedule</li> <li>• Price Book</li> </ul>	<ul style="list-style-type: none"> <li>• Usage Summary</li> <li>• Usage</li> </ul>	<ul style="list-style-type: none"> <li>• Usage Summary</li> <li>• Usage</li> </ul>	None

Action	Read	Create	Edit	Delete
	<ul style="list-style-type: none"> <li>• Price Schedule</li> </ul>			
Upload usage to a usage summary	<ul style="list-style-type: none"> <li>• Consumption Schedule</li> <li>• Order</li> <li>• Order Product</li> <li>• Order Product Consumption Rate</li> <li>• Order Product Consumption Schedule</li> <li>• Price Book</li> <li>• Price Schedule</li> </ul>	<ul style="list-style-type: none"> <li>• Usage Summary</li> <li>• Usage</li> </ul>	<ul style="list-style-type: none"> <li>• Usage Summary</li> <li>• Usage</li> </ul>	None

## Customer Asset Lifecycle Management

Action	Read	Create	Edit	Delete
View lifecycle-managed assets	<ul style="list-style-type: none"> <li>• Asset</li> <li>• Asset Action</li> <li>• Asset Action Source</li> <li>• Asset State Period</li> </ul>	None	None	None

In addition to the access in the table, give these permissions.

- Sales reps: Field-level security access to objects and fields that reps update as part of asset lifecycle management
- Developers using Customer Asset Lifecycle Management APIs: **Access Customer Asset Lifecycle Management APIs** and **API Enabled** permissions

## Mapping Custom Salesforce Billing Fields Between Objects

Certain pairs of Salesforce Billing objects pass custom field values from the first object to the second object when the second object is created. The values pass if the custom fields are editable, have matching field types, and have matching API names. We call these field pairs “twin fields.” (Salesforce

Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

### Important

- To map one field to another, you need Read access on the source field and Edit access on the target field.
- Standard fields and package fields can't map to or from any other fields.
- Salesforce Billing can't map values from a formula field that returns text to a text field.

Objects in Column 1 pass field values to custom fields with matching types and API names from the objects in Column 2.

Column 1	Column 2
Credit Note Line	Revenue Schedule
Debit Note Line	Revenue Schedule
Invoice	Credit Note <p> <b>Note</b> Invoice fields map to credit note fields automatically when you click <b>Cancel and Rebill</b>. To manually create a credit note on your invoice and map the invoice fields, provide a lookup to the invoice record on the credit note's Source Invoice field.</p>
Invoice Line	Credit Note Line
Invoice Line	Payment Allocation Invoice Line
Payment Allocation Invoice Line (Custom Field Type: Allocation)	Payment Allocation Invoice Line (Custom Field Type: Unallocation) <p> <b>Note</b> When a payment is unallocated from an invoice line, the custom fields on the payment allocation invoice line of type Allocation are transferred to a payment allocation invoice line of type Unallocation.</p>
Invoice Line	Revenue Schedule

Column 1	Column 2
Order Product	Invoice Line
Order Product	Revenue Schedule

## Billing Package Settings

Package settings control the properties of feature areas within Salesforce Billing. (Salesforce Billing Managed Package)

### Configure Salesforce Billing Package Settings

Access the Salesforce Billing Settings Editor to change settings across the entire package. (Salesforce Billing Managed Package)

### General Package Settings

The general package settings control the tax, date, and rounding options for Salesforce Billing. (Salesforce Billing Managed Package)

### Invoice Package Settings

Invoice package settings control the access to the invoice features of Salesforce Billing. (Salesforce Billing Managed Package)

### Payment Package Settings

Learn how payment settings control payment scheduler processes, credit card detail management, and other key payment features for Salesforce Billing. (Salesforce Billing Managed Package)

### Additional Package Settings

Learn how other package settings in Salesforce Billing let you re-run installation scripts or add sample data to sandbox orgs. (Salesforce Billing Managed Package)

## Configure Salesforce Billing Package Settings

Access the Salesforce Billing Settings Editor to change settings across the entire package. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce CPQ Editions

---

1. From Setup, enter *Installed Packages*, and then click **Installed Packages**.
2. Find the Salesforce Billing package and click **Configure**.
3. Navigate between the setting area tabs to change your settings.
4. Click **Save** when you're done.

## General Package Settings

The general package settings control the tax, date, and rounding options for Salesforce Billing.  
(Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

Setting	Definition
Tax calculation is based on?	Standard tax integrations use this address during tax calculation. Different addresses can cause higher or lower tax values based on your tax rates.
Align Billing day of month to Order start date	<p>By default, an order product's Billing Day of Month field inherits the order's start date. If this setting is disabled, admins must set the order product's billing day of month, either manually or via automation.</p> <p>Also, when this setting is disabled, and there's no automation to set it, sales reps can't select the Ordered field to create orders from opportunities or quotes. They can manually add the billing day of month by using the Create Order button before they save the order.</p>
Billable Unit Price Rounding	The Billable Unit Price field rounds to six decimal places. Use this field to control whether Salesforce Billing rounds this value upward or downward when it ends in 5.
Usage Rating Process based on	<p>Choose how to rate usage and usage summaries.</p> <p><b>By Process</b></p> <p>When you load usage to a usage summary, Salesforce Billing doesn't automatically rate the new usage or recalculate the usage summary. Use the <code>usageSchedulable</code> Apex class instead to run rating and calculations on the usage summary. When you run <code>usageSchedulable</code>, Salesforce Billing</p>

Setting	Definition
	<p>rates only the usage summary based on its pricing and total usage quantity, it doesn't calculate prices on the summary's individual usage records.</p> <p><b>On Trigger</b></p> <p>When you load usage to a usage summary, Salesforce Billing automatically rates the new usage and recalculates the usage summary. If your usage summary uses range pricing, Salesforce Billing also rates all the summary's other usage to determine whether the new usage moves the summary into a new pricing tier. If your usage summary uses slab pricing, Salesforce Billing evaluates the summary's total quantity to determine a pricing tier for the new usage, then rates all the new usage that falls in that tier.</p> <p>If your summary uses range pricing, On Trigger rating sometimes causes long load times due to Salesforce Billing running the rating process every time a new usage is uploaded, especially for summaries with large quantities of usage. We recommend customers with range-priced usage summaries and large quantities to use By Process rating. We advise testing in your org to determine the acceptable load times and the usage rating method that meets your needs.</p> <p> <b>Note</b> When you use the On Trigger rating, you must load the usage records by using a single thread and a batch size of 1 to ensure correct processing of the record.</p> <p>See <a href="#">Usage Rating</a>.</p>
Disable triggers	Disables all triggers in the Salesforce Billing package. We recommend disabling triggers during data migrations or when deleting data from a sandbox org.

Setting	Definition
Enable Debugs	Show debug messages from the invoice batch process.
Enable Billing Order Validations	<p>When this setting is enabled, these fields are required and validated.</p> <p>Required Fields for all Billing Order Products</p> <ul style="list-style-type: none"> <li>• SBQQ__ChargeType__c</li> <li>• BillingRule__c</li> <li>• RevenueRecognitionRule__c</li> <li>• TaxRule__c</li> <li>• ServiceDate</li> <li>• Quantity</li> <li>• UnitPrice</li> </ul> <p>Additional Fields Required for Subscription Products (Charge Type is Recurring)</p> <ul style="list-style-type: none"> <li>• SBQQ__BillingType__c</li> <li>• SBQQ__BillingFrequency__c</li> </ul> <p>Additional Fields Required for Cancellation Orders</p> <ul style="list-style-type: none"> <li>• SBQQ__RevisedOrderProduct__c</li> <li>• SBQQ__TerminatedDate__c</li> </ul>

## Invoice Package Settings

Invoice package settings control the access to the invoice features of Salesforce Billing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Classic and Lightning Experience

---

Available in: **Professional, Enterprise, Unlimited, and Developer** Editions

## Feature Settings

Invoice package settings control payment allocation, balance allocation of cancel order products, proration, scheduler batch sizes, balance remainders, and other Salesforce Billing invoice features.

Setting	Definition
Advanced AR Application	Allocate payments to invoice lines by enabling this feature. You can allocate payments to the invoice's total balance when this feature is disabled. We recommend that you enable this feature.
Default Cancel Order Rule	<p>Select the allocation of <a href="#">cancel order products</a>' balance to reduce the pending balances of prior order products to zero.</p> <ul style="list-style-type: none"> <li>• To cancel the original order products and subtract the sum of their pending balances from the pending balance of cancel order products, select <b>Legacy</b>.</li> <li>• To cancel pending billings on the newest order product when your amending order products can't fully cancel the pending balances of the original order products, select <b>LIFO by Order Product Creation Date</b>. See <a href="#">LIFO Order Product Cancellation</a>.</li> </ul> <p>You can override the default cancel order rule on a billing treatment by selecting a different cancel order rule.</p>
Proration Type	<p>Select the calculation type for the proration multiplier for prorated billing periods.</p> <ul style="list-style-type: none"> <li>• To divide the number of invoiced days in the month by the total number of days in the month, select <b>Calendar Days</b>.</li> <li>• To divide the number of invoiced days in the month by 30, select <b>30 Days</b>.</li> <li>• To divide the number of invoiced days in the month by (365/12), select <b>Monthly (CPQ Formula)</b>.</li> </ul>
Partial Proration Type	Select the proration type for the results of partial billing periods.
Billing Remainder Process	Select the period in which you want to include the remaining invoice balances that don't evenly divide over your order product's billing period.
Revenue Remainder Process	Select the period in which you want to include the

Setting	Definition
	remaining revenue balances that don't evenly divide over your order product's billing period.
Invoice Run Clean Up	When you enable this feature, the invoice clean-up run changes the status of invoices as well as invoice lines to canceled. The run also reverts the changes to order products and usage summaries so that they're pending billing and can be reprocessed. You can also configure the batch size of the clean-up. The error logs help you differentiate between invoice run and clean-up errors.
Consecutive Invoice Post Batch Jobs	When you enable this feature, draft invoices that are associated with invoice runs are posted one after the other.
Support Billing Period Calculation when Quarterly Billing Day of Month is During End of Month	<p>Enable this feature for products that have quarterly billing, and the numeric value of the billing day of the month is greater than the numeric value of the last day of the month in which the next billing date falls. For example, the billing day of the month is 29 and the next billing date is 28 February.</p> <p>When you enable this feature, the billing for such products works as expected, that is, they're charged for the duration between the order product's start date and the billing day of the month.</p>
Support Billing Period Calculation when Monthly Billing Day of Month is 31st	Enable this feature for products that have monthly billing, and the billing day of the month is 31. When you enable this feature, the billing for such products works as expected, that is, there are no errors when calculating the billing periods and partial periods are billed correctly.

## Batch Size Settings

Invoice runs can encounter errors when you run batch Apex jobs that create or update large numbers of invoice lines. These errors often occur when you have about 2,000 invoice lines. You can reduce the likelihood of these errors occurring by setting your own batch size. For example, specify 300 as the draft invoice batch size. If your invoice run evaluates 2,100 order products that can be invoiced, the run uses

seven batch processes to convert the order products into invoice lines. You're less likely to encounter Apex errors because each job processes only 300 invoice lines. A reduction in the batch size reduces the chance of errors but slows the invoice run speeds.

If your customer's invoice run encounters Apex errors when they create invoices, calculate invoice tax, post invoices, or clean-up invoices, then decrease the batch size. You can update the batch size to the maximum value when the run isn't processing a batch of large invoices.

Setting	Definition
Draft Invoice Batch Size	The maximum number of invoice lines that each Apex batch job evaluates for creating invoices. You can specify up to 2,000 as the batch size. For faster processing, specify at least 200 as the batch size.
Posted Invoice Batch Size	<p>The maximum number of invoices that each Apex job posts. You can specify up to 70 as the batch size.</p> <p>When the Consecutive Invoice Post Batch Jobs toggle is enabled, the optimal batch size for the successful posting of invoices is 1. The time taken to post invoices depends on the number of invoices to be posted.</p>
Tax Processing Batch Size	The maximum number of invoices that each Apex calculates tax for. You can specify up to 70 as the batch size.
Invoice Run Clean Up Batch Size	<p>The maximum number of invoices that can be part of the clean-up run. The default value is 50. You can reduce the batch size if the clean-up run fails. An increment in the batch size carries performance risks.</p> <p>The optimal batch size for the best performance of the clean-up run is 30 invoices, but can be lowered to improve performance and to reduce errors.</p>

## See Also

[Clean Up Invoice Runs](#)

## Payment Package Settings

Learn how payment settings control payment scheduler processes, credit card detail management, and other key payment features for Salesforce Billing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

Setting	Definition
Card Mapping	<p>When you use a payment gateway package, the Card Mapping value defines the payment method field that Salesforce Billing references to retrieve the gateway token. If Card Mapping is null, Salesforce Billing uses the payment method's Payment Gateway Token Field.</p> <p>When you enter a Card Mapping value, provide only the field name. For example, use <code>CustomTokenField__c</code> instead of <code>blng__PaymentMethod__c.CustomTokenField__c</code></p>
Payment Scheduler Pickup Date(on or after)	Allows users to choose whether the payment scheduler picks up the invoice for payment on the invoice date or due date.
Payment Method Display	Choose how credit card payment method options appear to users in the Payment Center.
Payment creation batch size	Payment runs can encounter errors when running batch APEX jobs that create large numbers of payment lines. These errors often occur around 2000 payment lines, but the exact number varies based on org settings. You can reduce the likelihood of these errors by setting your payment creation batch size. For example, if you set a posted invoice batch size of 300 and your payment run must create 2100 payment lines, Salesforce Billing runs 7 APEX batch jobs that each create 300 payments. Reducing the batch sizes will slow payment run speeds but reduces the chance of errors from batches with multiple large

Setting	Definition
	<p>invoices.</p> <p>If your customer's payment run encounters APEX errors, decrease the payment creation batch size.. Then, return it to the maximum value when their payment runs are no longer evaluating a large volume of invoice lines.</p> <p>Salesforce Billing supports a minimum value of 1 and a maximum value of 70.</p>
Save credit card details	<p>This setting applies to Experience Cloud and Guest User payments only. When enabled, this setting exposes both of the following checkboxes, which allows saving the credit card payment token as a payment method record and allows the customer to choose whether to allow payment runs to use this payment method to pay future invoices.</p> <ul style="list-style-type: none"> <li>• Save information for easy payment later?</li> <li>• Sign up for the automatic payment option?</li> </ul> <p>The default values are <code>false</code> and can't be customized. We don't support setting any to <code>true</code> by default or exposing only one of the options.</p>

## Additional Package Settings

Learn how other package settings in Salesforce Billing let you re-run installation scripts or add sample data to sandbox orgs. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Setting	Definition
Post-Install Steps	If scripts failed during installation or upgrade, re-run them by selecting Execute Scripts.
Sample Organization Data	Add a collection of objects such as billing rules

Setting	Definition
	and invoice schedulers to your sandbox to test Salesforce Billing features. We recommend adding sample data to sandbox orgs only.

## Salesforce Billing Overview

---

Create and automate your invoices, payments, and revenue with Salesforce Billing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

#### **Integrating Salesforce CPQ and Salesforce Billing**

Salesforce Billing is an add-on package that inherits key records and information from Salesforce CPQ. After a sales rep finalizes a quote and orders it within Salesforce CPQ, Salesforce Billing picks up the order record for invoicing, payment, and revenue recognition. Before you integrate Salesforce Billing with Salesforce CPQ, review important features and differences between the packages. (Salesforce Billing Managed Package)

#### **Configure Salesforce Billing to Coexist with Revenue Cloud**

Transition to Revenue Cloud while maintaining the integrity of your existing Salesforce Billing workflows. Create a custom field and specific rules in Salesforce Billing to properly process Revenue Cloud orders. Then, create orders in Revenue Cloud by bypassing standard Salesforce Billing processes.

#### **Order Management in Salesforce Billing**

The order object lives in the CPQ package and acts as the intersection between Salesforce CPQ and Salesforce Billing. Before you get started with Salesforce Billing, review key guidelines about the order's use in CPQ and billing workflows. (Salesforce Billing Managed Package)

#### **Revenue Recognition Reporting**

Create and manage reports for how and when your customers recognize revenue for a product or service. (Salesforce Billing Managed Package)

#### **Bookings, Billings, Cash, and Revenue**

Salesforce Billing operates within the financial domain and contains unique processes for tracking and reporting on your company's bookings, billings, cash, and revenue. Before you get started with Salesforce Billing, review the processes to understand their differences and how Salesforce Billing manages the processes and flows for each of them. (Salesforce Billing Managed Package)

#### **ERP Integration: Complementing Accounting Systems**

Salesforce Billing complements enterprise resource planning (ERP) platforms by converting Salesforce CPQ's lead-to-order data into transactional data. The conversion allows ERP systems to inherit matching data, which they can use for accounting functions like general ledger and financial reporting.

This process lets Salesforce CPQ and Salesforce Billing manage financial customer touchpoints and upholds the ERP as the system of record for GL and financial reporting. (Salesforce Billing Managed Package)

## Integrating Salesforce CPQ and Salesforce Billing

Salesforce Billing is an add-on package that inherits key records and information from Salesforce CPQ. After a sales rep finalizes a quote and orders it within Salesforce CPQ, Salesforce Billing picks up the order record for invoicing, payment, and revenue recognition. Before you integrate Salesforce Billing with Salesforce CPQ, review important features and differences between the packages. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Salesforce CPQ Features

Salesforce CPQ lets sales reps configure, price, and prorate items on a quote. They can then create a contract to record subscriptions for future sales transactions, amendments, and renewals. They can order the quote, which produces an order record in Salesforce CPQ. Each order contains order products, which represent quote lines that a customer has ordered. When you're ready to bill for an order, Salesforce Billing evaluates order and order product field to develop the invoice and related transactional records.

#### Salesforce Billing Features

Salesforce Billing lets you invoice an order and manage its balances through credit and debit notes. You can automate customer payments and process them manually or through a third-party payment gateway. You can also record all your transactional data into finance books and finance periods, then map that data into an external general ledger or ERP. We've also provided a revenue recognition reporting system that lets you account for revenue for a product you provide to a customer.

Salesforce Billing contains several rules that define when and how it creates transactional records. You define lookups to these fields on your product records, which pass their rules to the resulting order products when a sales rep orders a quote. Many order products can look up to the same rule, which lets you consolidate groups of order products under one rule based on how you want Salesforce Billing to handle them during invoicing events. Each rule also contains treatment records, which allow Salesforce Billing to apply more specific actions based on shared values between an order product and the treatment. Let's look at a few important actions that rules control.

- Whether to create an invoice line for an order product, then how the invoice line's pricing and date values drive billing periods, billing dates, and billing cycles.
- Whether to create a revenue schedule and revenue transactions for an invoice line.
- Whether to calculate tax for an invoice line and how Salesforce Billing calculates tax based on

tax integrations and tax data.

This flowchart walks through a complete quote-to-cash process that begins with uses the CPQ platform for quote development, contracting, and booking, then moves to Salesforce Billing for invoicing, adjustments, cash flow, and finance books.



## Use Cases

Let's look at a few common scenarios where Salesforce CPQ users can integrate Salesforce Billing.

### Quote-to-Invoice Development

Universal Containers is familiar with CPQ functionality and are now looking to automate invoice creation and revenue schedules all on a single platform. This integration will decrease time spent on invoice review and validation compared to their current process. Revenue Schedule generation, coupled with time saved on administrative tasks, will allow for improved forecasting. Universal Containers will then integrate with their Finance System of Record for all other functionality, like payments.

### Quote-to-Cash with Automated Payment Deployment

Universal Containers is familiar with CPQ functionality and are now looking to automate invoice creation and payments within Salesforce. Salesforce Billing offers various integrations with payment processing vendors. Applying Salesforce Billing's built-in features and integrations will speed up payment collection, and improve cash flow. Universal Containers will then integrate to their Finance System of Record for any additional functionality, like Accounts Receivables.

## Considerations

- Salesforce CPQ and Billing have independent proration settings. When implementing Salesforce Billing, make sure that you align proration settings in both packages to ensure accurate order product and invoice line totals. For more information, check out [Proration Methods in CPQ and Billing](#).
- Rounding isn't available for the Japanese yen (JPY) and other nondecimal currencies in Salesforce Billing.

## Configure Salesforce Billing to Coexist with Revenue Cloud

Transition to Revenue Cloud while maintaining the integrity of your existing Salesforce Billing workflows. Create a custom field and specific rules in Salesforce Billing to properly process Revenue Cloud orders. Then, create orders in Revenue Cloud by bypassing standard Salesforce Billing processes.

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

### USER PERMISSIONS NEEDED

To configure coexistence with Revenue Cloud in System Admin profile  
Salesforce Billing:

To configure Revenue Cloud and Salesforce Billing to operate in coexistence, upgrade Salesforce Billing to Summer '25 and then complete these steps.

## Create a Custom Field on Orders

Create a custom picklist field on the Order object.

1. From the [object management settings for orders](#), go to Fields & Relationships.
2. Click **New**, select **Picklist** as the data type, and click **Next**.
3. Enter *Salesforce Billing Order Processing* as the field label.
4. Select **Enter values, with each value separated by a new line** and enter *skip* as the picklist value.
5. Enter *SalesforceBillingOrderProc* as the field name and click **Next**.
6. Define the necessary field-level security for the appropriate user profiles and click **Next**.
7. Add the field to the appropriate page layouts.
8. Save your changes.

The screenshot shows the 'Custom Field Definition Detail' page for the 'Salesforce Billing Order Processing' field. The field label is 'Salesforce Billing Order Processing'. The field name is 'SalesforceBillingOrderProc' and the API name is 'SalesforceBillingOrderProc\_c'. The data type is 'Picklist'. The object name is 'Order'. The field is defined on the 'Order' object. The 'Field Name' and 'API Name' fields are highlighted with an orange box.

Field Label	Salesforce Billing Order Processing	Object Name	Order
Field Name	SalesforceBillingOrderProc	Data Type	Picklist
API Name	SalesforceBillingOrderProc_c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Admin User, 5/27/2025, 8:18 AM	Modified By	Admin User, 5/27/2025, 8:18 AM

## Create a Billing Rule

1. Create a [billing rule](#).
2. Enter the name as *Do Not Invoice*.
3. Select *No* as the Generate Invoice value.

The screenshot shows the 'Billing Rule' creation page for a rule named 'Do Not Invoice'. The 'Details' tab is selected. The 'Generate Invoice' field is set to 'No' and is highlighted with an orange border. Other fields include 'Initial Billing Trigger', 'Order Product Activation Date', 'Partial Period Treatment' (set to 'Separate'), 'Period Treatment for BTDO' (with a help icon), 'Exclude Billing Day of Month', and 'Notes'.

Field	Value	Action
Name	Do Not Invoice	Active <input checked="" type="checkbox"/>
Initial Billing Trigger		
Order Product Activation Date		
Partial Period Treatment	Separate	
Period Treatment for BTDO	(i)	
Exclude Billing Day of Month		
Notes		
Generate Invoice	No	

## Create a Tax Rule

1. Create a [tax rule](#).
2. Enter the name as *Do Not Tax*.
3. Select *No* as the Taxable (Yes/No) value.

**Related** **Details**

Tax Rule Name Do Not Tax	Owner Admin User
Active <input checked="" type="checkbox"/>	
Notes	
<b>Taxable (Yes/No)</b> No	
Created By Admin User, 10/28/2022, 12:07 PM	Last Modified By Admin User, 4/4/2025, 6:24 AM

## Create Revenue Recognition Rule

1. Create a [revenue recognition rule](#).
2. Enter the name as *Do Not Recognize*.
3. Select *No* as the Create Revenue Schedule? value.

**Related** **Details**

Name Do Not Recognize	Active <input checked="" type="checkbox"/>
Notes	
<b>Create Revenue Schedule?</b> No	
<b>System Information</b>	
Created By Admin User, 8/10/2021, 2:26 AM	Owner Admin User
Last Modified By Admin User, 4/4/2025, 6:26 AM	

## Create Orders in Revenue Cloud

Create Order records in Revenue Cloud by using Place Order API.

1. Create a Revenue Cloud order by using [Place Order API](#).
2. In your API request, specify these values.

<b>Order.SalesforceBillingOrderProcessing__c</b>	Skip
<b>OrderItem.blnG__BillingRule__c</b>	do_not_invoice_billing_rule_record_id
<b>OrderItem.blnG__TaxRule__c</b>	do_not_tax_rule_record_id
<b>OrderItem.blnG__RevenueRecognitionRule__c</b>	do_not_recognize_revenue_recognition_rule_record_id

The API payload will structure these details within the graph and record arrays for the Order, App Usage Assignment, Order Action, and Order Item objects.

## Example

```
POST /commerce/sales-orders/actions/place

{
  "pricingPref": "Skip",
  "configurationInput": "Skip",
  "graph": [
    {
      "graphId": "1",
      "records": [
        {
          "referenceId": "refOrder",
          "record": {
            "attributes": {
              "type": "Order",
              "method": "POST"
            },
            "Status": "Draft",
            "AccountId": "001xx000003GgZH",
            "EffectiveDate": "2025-02-01",
            "Pricebook2Id": "01sxx0000005tFu",
            "SalesforceBillingOrderProc__c": "Skip"
          }
        },
        {
          "referenceId": "refAppTag",
          "record": {
            "attributes": {
              "type": "AppUsageAssignment",
              "method": "POST"
            },
            "AppUsageType": "RevenueLifecycleManagement",
            "RecordId": "@{refOrder.id}"
          }
        },
        {
          "referenceId": "refAction"
        }
      ]
    }
  ]
}
```

```

    "referenceId": "refOrderAction",
    "record": {
        "attributes": {
            "type": "OrderAction",
            "method": "POST"
        },
        "OrderId": "@{refOrder.id}",
        "Type": "Add"
    }
},
{
    "referenceId": "refOrderItem",
    "record": {
        "attributes": {
            "type": "OrderItem",
            "method": "POST"
        },
        "OrderId": "@{refOrder.id}",
        "OrderActionId": "@{refOrderAction.id}",
        "PricebookEntryId": "01uxx0000008zeM",
        "ServiceDate": "2025-02-01",
        "EndDate": "2026-01-31",
        "PricingTermCount" : 12.0,
        "Quantity": 2.0,
        "UnitPrice": 100.0,
        "NetUnitPrice": 100.0,
        "TotalLineAmount": 2400.0,
        "PeriodBoundary": "AlignToCalendar",
        "BillingFrequency2": "Monthly",
        "blng__BillingRule__c": "a1Txx00000052Fm",
        "blng__TaxRule__c": "a2Lxx0000004D1s",
        "blng__RevenueRecognitionRule__c": "a2Dxx000000ybRl"
    }
}
]
}
}

```

Alternatively, you can automate this process by creating a Record Type picklist default for orders in Revenue Cloud and setting the SalesforceBillingOrderProc\_\_c field's default value to Skip for this record type.

## See Also

- [Coexistence of Revenue Cloud with Salesforce CPQ and Billing Managed Packages](#)
- [Create a Custom Picklist Field](#)

## Order Management in Salesforce Billing

The order object lives in the CPQ package and acts as the intersection between Salesforce CPQ and Salesforce Billing. Before you get started with Salesforce Billing, review key guidelines about the order's use in CPQ and billing workflows. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

After quoting in Salesforce CPQ, sales reps can create an order and a contract. The contract acts as the source of truth for future sales transactions, allowing users to create amendments and renewals for their subscription-based products. The order acts as the source of truth for future billing transactions, allowing Salesforce Billing users to convert lead-to-order data into transactional data.

An order record contains order products, which represent quote lines that a customer has ordered. When a sales rep creates an order from a quote, its order products have an unactivated status. Only activated orders can be billed, so many businesses use unactivated order products for provisioning and activate them when they're ready for billing. Each order product can be billed separately or as a group. The Salesforce Billing invoice scheduler or Bill Now process pulls order information into an invoice, which lives in the Salesforce Billing package. The invoice record's invoice lines represent billed order products.

Orders also look up usage summaries, which store information for usage-based order products. For more information, check out [Usage Rating and Processing](#).

Sales reps can create both contracts and orders from a quote. We recommend creating an order from the quote, and then creating a contract from the activated order. This method allows for a standardized process within your business and lets you avoid duplicate data. Customers are also less likely to amend a contracted order than a quote.

### Considerations

- Salesforce Billing evaluates several fields and settings to determine when an order product becomes eligible for invoicing. For more information, see [Understanding Next Billing Date](#).
- Enable orders and customize order settings in Salesforce CPQ package settings.
- We recommend automating invoice generation with an invoice scheduler. Creating invoices manually using Bill Now is recommended only for testing.

## Revenue Recognition Reporting

Create and manage reports for how and when your customers recognize revenue for a product or service. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing 7.0 and later

Revenue recognition reporting is the process of accounting for revenue for a product you provide to a customer. For example, when a customer pays for and receives a shirt in a clothing store, the store owner makes a record of the revenue received. How and when the store owner recognizes the revenue depends on many factors.

Let's look at a few common ways of reporting revenue.

### Service delivered over time

Sometimes you want to spread revenue over a set period. For example, on an annual order you might recognize revenue during each of 12 months even though you invoice it all at once.

### Bundled products and services

Bundles include components that you deliver all at once or over time. For example, an IT infrastructure bundle could contain a component for a server cooling system and a server repair service agreement. Since the cooling system is a physical product, you provide its value when your customer receives it. However, the repair service has a variable amount of value based on how many times the customer needs repairs during a billing cycle.

### Discounts and credits

You can apply discounts and credits even after you've begun recognizing revenue for a product by aligning your existing revenue schedules to reflect the difference.

### Add-on orders

Add-ons follow the contract terms of the original quote or order and their revenue recognized accordingly.

### Contract amendments

When you change a contract, such as upgrading or downgrading a service, you also have to adjust the revenue associated to these changes over time.

### Customer acceptance and service deliverables

Products and services can be delivered and used while also having delayed acceptance rights that impact when you can recognize revenue.

### Customers that stop paying invoices

Revenue is recognized based on when the customer pays and not upon delivery of the product or service. This process is useful if there's a significant risk of the customer defaulting.

### Selling a product to a new region or customer base

When you sell a product or service to a new customer segment, you may require different

revenue recognition methods for the new customers.

Revenue recognition requires organizations to track what their customers have purchased, company obligations, and how they deliver their services. Salesforce Billing helps users understand types of revenue recognition reporting, configuration, pricing, and payment obligations, creating a straightforward revenue calculation process.

## Bookings, Billings, Cash, and Revenue

Salesforce Billing operates within the financial domain and contains unique processes for tracking and reporting on your company's bookings, billings, cash, and revenue. Before you get started with Salesforce Billing, review the processes to understand their differences and how Salesforce Billing manages the processes and flows for each of them. (Salesforce Billing Managed Package)

### Quote-Level Revenue Flow

Review how bookings flow to billings, cash, and revenue over the quote-to-invoice process. (Salesforce Billing Managed Package)

### Quote-Level Revenue Flow

Review how bookings flow to billings, cash, and revenue over the quote-to-invoice process. (Salesforce Billing Managed Package)

Before we begin, let's look at a table of subscription contracts for context. Let's say your company sells bronze, silver, and platinum mobile device management plans monthly and annually. Bronze plans are \$650 per month, Silver plans are \$950 per month, and Platinum plans are \$1,400 per month. Contracts are for one year and can be purchased for monthly billing or annual billing. One of your accounts has the following plan contracts.

Customer	Subscription	Starting Month	Billing Type	Monthly Fee	Contract Total
A	Platinum	March	Monthly	\$1,400	\$16,800
B	Bronze	March	Annually	\$650	\$7,800
C	Bronze	April	Annually	\$650	\$7,800
D	Platinum	May	Monthly	\$1,400	\$16,800
E	Bronze	May	Monthly	\$650	\$7,800
F	Silver	June	Annually	\$950	\$11,400

## Bookings

Bookings refers to a prospective contract's value over a time period. It represents the amount you're expecting to make from the contract. For example, when Customer A orders their \$1400-a-month contract for 12 months, the contract has a \$16,800 bookings value. You often see references to bookings in proposals, statements of work, and other presales documents.

At the end of a billing period, such as one month, your bookings for that period represent the total of closed deals in that period in terms of the full contract duration.

Month	Bookings	What's Included
March	\$24,600	Contract total for Customer A and Customer B
April	\$7,800	Contract total for Customer C
May	\$24,600	Contract total for Customer D and Customer E
June	\$11,400	Contract total for Customer F

Companies use bookings to evaluate and plan for future revenue growth. For example, if you notice that your bookings are consistently low in March but high in June, you can evaluate June sales patterns and see if you can apply them during March or other low-bookings months.

## Billings

Billings refer to the actual billed amount that a customer receives on their invoice for the invoice for a given billing period. You can think about billings as the cash you get when a service has been invoiced, while bookings is the revenue you expect to get after a service has been delivered. When you calculate Billings for a given period, include all the accounts that are active for that period, not just the new ones – for example, when you calculate billings for March, include both accounts that started in March and any accounts that started in January and February that are still billing.

Let's look at the original accounts table again, where your customers have annual and monthly subscription plans. When you calculate billings each month, remember that the annual customers pay their total contract cost up-front, while monthly customers pay every month.

Month	Bookings	Billings	What's Included
March	\$24,600	\$9,200	Customer A's \$1,400 (monthly) and Customer B's \$7,800 (annual)
April	\$7,800	\$7,800	Customer C's \$7,800

Month	Bookings	Billings	What's Included
			(annual)
May	\$24,600	\$2,050	Customer D's \$1,400 (monthly) and Customer E's \$650 (monthly)
June	\$11,400	\$11,400	Customer F's \$11,400 (annual)

Because bookings aren't evaluated on financial reports or income statements, billings are the key metric for letting your company accurately track the money it's owed. Your financial managers report your billings on your company's balance sheets and income statements.

## Revenue

Revenue refers to the income you earn after the customer is invoiced. Revenue is different from billings in this sense, as billings are calculated at the end of the billing period but before service is provided. For example, most businesses charge customers for a contract before providing that billing period's worth of services. After the service is delivered, the business can recognize the revenue. This flow ensures that businesses follow Generally Accepted Accounting Principles rules, which state that you can recognize revenue only after you've "earned" it.

Your revenue generally looks different from your billings because it tracks only the services delivered at the end of the billing period. Even though Customer B paid their entire \$7,800 up front, we track only \$650 for them in March, given the services they received at the end of March ( $\$7,800 \div 12 = \$650$ ). Similar to billings, revenue also includes contracts that started in earlier months but are still being invoiced.

A company can recognize revenue only if it meets the ASC 606 standards for recognizing revenue.

Month	Bookings	Billing	Revenue	What's Included
March	\$24,600	\$9,200	\$2,050	Customer A's \$1,400 and Customer B's \$650.
April	\$7,800	\$7,800	\$650	Customer C's \$650.
May	\$24,600	\$2050	\$2,050	Customer D's \$1,400 and Customer E's \$650.

Month	Bookings	Billings	Revenue	What's Included
June	\$11,400	\$11,400	\$950	Customer F's \$950.

If you billed money but haven't delivered the service (and thus can't recognize it), you can track it as deferred revenue. You can calculate deferred revenue as Billings–Total Revenue.

Month	Bookings	Billings	Revenue	Deferred Revenue
March	\$24,600	\$9,200	\$2,050	\$7,150
April	\$7,800	\$7,800	\$650	\$7,150
May	\$24,600	\$2,050	\$2,050	\$0
June	\$11,400	\$11,400	\$950	\$10,450

## Cash Flow

After you invoice a customer, the amount owed is considered revenue. Cash flow refers to the net amount of cash moving in and out of a company—in other words, the amount of money that a customer has paid. You can use revenue to track your company's sales and marketing effectiveness, while cash flow represents your company's actual liquidity.

Your cash flow likely varies relative to your revenue over a series of billing periods. For example, some customers don't pay an invoice immediately after receiving it. This condition causes the cash flow for invoices sent out within a given billing period to be lower than revenue accumulated in that specific period. However, if customers pay several invoices from an earlier period at the same time, that influx can cause the current period's cash flow to be higher than the period's revenue.

Month	Bookings	Billings	Revenue	Cash Flow	What's Included
March	\$24,600	\$9,200	\$2,050	\$7,800	Customer B paid their invoice during this period, but Customer A did not.
April	\$7,800	\$7,800	\$650	\$10,600	Customers A, B, C, and D paid their invoices during this period. Customer A

Month	Bookings	Billings	Revenue	Cash Flow	What's Included
					also paid their late January invoice.
May	\$24,600	\$2,050	\$2,050	\$10,600	All customers paid their invoices during this period.
June	\$11,400	\$11,400	\$950	\$14,200	All customers paid their invoices during this period.

## Putting It Together

Now that you've reviewed the basics, here's how the bookings-to-revenue pipeline aligns with the Salesforce CPQ and Salesforce Billing workflow.



## ERP Integration: Complementing Accounting Systems

Salesforce Billing complements enterprise resource planning (ERP) platforms by converting Salesforce CPQ's lead-to-order data into transactional data. The conversion allows ERP systems to inherit matching data, which they can use for accounting functions like general ledger and financial reporting. This process lets Salesforce CPQ and Salesforce Billing manage financial customer touchpoints and upholds the ERP as the system of record for GL and financial reporting. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Lead to Invoice

Create invoice data out of Salesforce CPQ's quote and order data. Customer account data pairs with a single product catalog to help optimize and simplify the invoicing process. A clean integration can then link to the ERP, which handles accounts receivable, payments, and GL functions.

#### Lead to Cash

Add payments, adjustments, and revenue reporting to the Salesforce Billing Lead to Invoice process. When you perform these functions in Salesforce Billing, you can send extensive

customer data to the ERP for general ledger logging and financial reporting.

### Revenue Recognition Reporting Integration

Revenue recognition processes require organizations to track how they deliver their services, company obligations, and what their customers have purchased, company obligations.

Salesforce Billing lets businesses configure different types of revenue recognition reporting scenarios to help create a straightforward revenue calculation process.

### Partner Architectural Considerations

Consider using partners to further expand your Salesforce Billing integration's capabilities.

- Third-party tax integrations can connect to quotes to provide tax estimations. They can also connect to the invoice calculation process to generate tax line items.
- eSignature integrations connect with proposals to provide automatic signature conversions to contracts and orders.
- Usage mediation providers enable mediation, pre-rating, and conversion of usage records to Salesforce Billing-supported formats for invoice generation.
- Payment gateways provide PCI-compliant tokenized integrations for automatic captures of bulk recurring credit card payments.

### Identify Key Stakeholders

If you're planning to integrate Salesforce CPQ and Billing with an ERP system, make sure that you identify key roles and departments during your design and requirement gathering process.

Stakeholder Group	Function
Finance Organization	Outlines billing and invoice processes, balance management, payment processing, subscription management, collections, tax processing, and revenue recognition.
Marketing	Organizes lead and campaign management, marketing, and monetization strategies.
Sales & Sales Ops	Finalizes selling processes, including opportunity management, partner management, quoting, discounting, up-selling and cross-selling, approvals, renewals, and deal amendments.
Product	Handles new product introduction, product bundling, entitlement processing, promotions and discounting, and trial processing.
Customer Service	Handles customer service, inquiry and dispute resolution, payments, adjustments, product and service maintenance, and usage

Stakeholder Group	Function
	monitoring.
IT and Business Platforms	Understands integration framework and data flow considerations as they relate to front-end and back-end selling and billing processes.
Legal	Manages contract lifecycle management, compliance needs, and selling performance obligations.

### Lead-to-Invoice

When you implement Salesforce Billing, consider the differences between Salesforce and an ERP within the lead to invoice approach. These differences will help you identify whether lead-to-invoice is the correct deployment option and strategy for your Salesforce Billing implementation. (Salesforce Billing Managed Package)

### Lead to Cash

When you implement Salesforce Billing, consider the differences between Salesforce and an ERP within the lead-to-cash approach. These differences help you identify whether lead-to-cash is the correct deployment option and strategy for your Salesforce Billing implementation. (Salesforce Billing Managed Package)

### Revenue Recognition Reporting Integration

When you implement Salesforce Billing, consider the accounting policies needed to accurately create revenue recognition reports. These requirements help you figure out whether a revenue recognition reporting integration out of Salesforce Billing is the correct strategy for your business. (Salesforce Billing Managed Package)

## Lead-to-Invoice

When you implement Salesforce Billing, consider the differences between Salesforce and an ERP within the lead to invoice approach. These differences will help you identify whether lead-to-invoice is the correct deployment option and strategy for your Salesforce Billing implementation. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The Salesforce platform enables companies to drive internal technologies with the best interest of customers in mind. Salesforce Billing embodies this purpose by completing the end-to-end processes of the customer engagement lifecycle. The platform starts with a lead, which is priced through Salesforce CPQ and intersects with Salesforce Billing during the transition from order to invoice. This architecture allows Salesforce CPQ and Billing to function as a single system and product catalog for carrying

customer data from lead to invoice. Keeping your data in a single system results in a cleaner handoff to an ERP for receivables, collections, and financial reporting.



### Solving Common Business Challenges with Lead-to-Invoice

Implementing Salesforce Billing for lead-to-cash allows you to solve several common business challenges. (Salesforce Billing Managed Package)

### Key Lead-to-Invoice Functions in Salesforce Billing

Salesforce Billing provides the capability for important lead-to-invoice functions. (Salesforce Billing Managed Package)

### ERP Integration Architectural Considerations for Lead to Invoice

When you evaluate Salesforce Billing as part of a lead-to-invoice solution, consider the other systems necessary to perform billing functions such as payments and revenue recognition reporting. You'll also need to define the scope in which you use Salesforce Billing to integrate with an ERP. (Salesforce Billing Managed Package)

### Other Lead-to-Invoice Integration Considerations

Look at other important factors to consider when implementing Salesforce Billing as a lead-to-invoice solution. (Salesforce Billing Managed Package)

## Solving Common Business Challenges with Lead-to-Invoice

Implementing Salesforce Billing for lead-to-cash allows you to solve several common business challenges. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Launching a New Product or Business Model Transition

Lead-to-invoice helps solve scaling challenges such as launching new products and managing usage-based product offerings. Salesforce Billing can also help businesses that transition to a recurring or consumption sales model and want to maintain their A/R in the ERP system.

### Invoicing Process is Laborious and Manual

Some businesses with ERP systems face challenges handling complex scenarios like milestone billing, usage rating, and prorated partial periods. You can solve many of these scenarios with multiple systems and manual intervention, but this approach can create disparate systems with minimal source of truth for what the customer owes and owns. A lead-to-invoice implementation solves this problem by allowing Salesforce Billing to generate complex charges on the invoice with clean and accurate from Salesforce CPQ. Salesforce Billing then passes that information downstream for ERP consumption.

## Maintaining a Product Catalog in Salesforce

One of Salesforce Billing's primary advantages is a consolidated selling and billing product catalog. This catalog helps the product launch process scale, even when introducing complex subscription and usage pricing models. Invoice, order, quote, and opportunity data is traceable back to the product catalog, giving the customer a single view of what they are buying throughout the sales and order process.

### See Also

- [Key Lead-to-Invoice Functions in Salesforce Billing](#)
- [ERP Integration Architectural Considerations for Lead to Invoice](#)
- [Other Lead-to-Invoice Integration Considerations](#)

## Key Lead-to-Invoice Functions in Salesforce Billing

Salesforce Billing provides the capability for important lead-to-invoice functions. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Customer Billing Profile

An accurate, up-to-date customer profile is critical to closing the sales to finance gap. These departments are often siloed, creating discrepancies between what sales and finance view as an account and customer. A lead-to-cash approach fills this gap by using sales's view of the account when staging bill-to-account and contact information on an invoice. Rich account and billing data can be pushed to an ERP, promoting a single source of truth and accurate view of customers and accounts.

### Expansions to the Standard CPQ Product Catalog

Salesforce Billing adds billing, tax, GL, and revenue recognition rules to the existing Salesforce product catalog. This is the core value of Salesforce CPQ used with Salesforce Billing, as owning and managing multiple product catalogs create significant operational overhead and reduces the ability to launch new products and pricing.

One of Salesforce Billing's most powerful features is the ability to define different billing, revenue recognition, and tax treatments by product. This way, you don't need to create a different SKU when you have a different billing, revenue, or tax treatment.

Many companies have several products catalogs, such as one for selling and another for billing. The billing product catalog typically works with selling and provisioning to ensure that products and services are sold properly, activated correctly, and billed accordingly. A separate sales product catalog would relate to quoting. Since Salesforce CPQ and Billing are both on the Salesforce platform, all product catalog information is created and managed in one system.

## Billing One Time & Recurring Charges

The consolidation of CPQ and Billing functions on the Salesforce Platform allows for convenient operation of subscription timing functions. Start and end dates carry automatically from the quote to the order, which then stages subscriptions for invoicing. Order proration calculations can be reconciled based on provisioning and then invoiced manually or through a scheduled job.

Provide a convenient customer invoicing experience by using Salesforce's contract and order staging to align add-on subscriptions to existing orders. This data flow puts less weight on finance teams by passing information directly to the ERP with little consolidation, conversion, or recalculation.

## Billing Usage-Based Products

Salesforce CPQ and Billing handle usage rating and invoicing with the same consolidated view subscriptions and one-time products. Usage stages at the order and can be loaded, rated, and invoiced as part of standard batches. This data flow allows Salesforce to consolidate invoice information into one view, passing a full picture of provisioned and invoiced products down to an ERP for presentation and collections.

## Billing by Milestone

Variable invoicing allows teams to control billing scope and timing. Override processes let users meet exception use cases such as milestone billing on ship dates or project completion. At the same time, admins can automate process changes based on billing criteria.

## Taxation

Taxation is optional for a lead-to-cash implementation. Salesforce Billing can apply taxes using integration to external tax engines. Your ERP can also apply taxation.

## Invoice Creation & Presentation

Customers expect a consistent experience between accepting the quote and viewing the final charges on the invoice. Salesforce Billing stages invoice data in one location, enabling a simple document generation experience regardless of whether you're using the ERP or a third-party application to handle document presentation and delivery.

## See Also

- [Solving Common Business Challenges with Lead-to-Invoice](#)
- [ERP Integration Architectural Considerations for Lead to Invoice](#)
- [Other Lead-to-Invoice Integration Considerations](#)

## ERP Integration Architectural Considerations for Lead to Invoice

When you evaluate Salesforce Billing as part of a lead-to-invoice solution, consider the other systems necessary to perform billing functions such as payments and revenue recognition reporting. You'll also need to define the scope in which you use Salesforce Billing to integrate with an ERP. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

### Invoice and Invoice Line Record Data Generation

The most critical lead-to-invoice integration point falls on the invoice and invoice line objects.

These objects integrate with an ERP system, which then performs payments, adjustments, revenue recognition reporting, A/R, and financial reporting.

The invoice and its related Salesforce records contain all the information needed to determine what the customer is charged, when payment is due, and what to recognize for revenue. The invoice also drives A/R and supports financial reporting. Salesforce Billing generates all relevant invoice data including dates, quantities, and totals for calculating tax and customer presentation. Salesforce stores this data and then passes it to the ERP system for actual data.

For example, you could use Salesforce Billing for the creation of invoice and invoice line data, then send that information downstream to an ERP system for the next billing processes and balance management. In this scenario, Salesforce is the source of invoice charge information while the ERP system is the master for payments, adjustments, and the invoice and account balance.

### Tax Management

Lead-to-Invoice solutions don't require tax calculation in Salesforce. Salesforce Billing provides flexible options for calculating tax via Salesforce native tax, third-party tax integrations, or in the downstream ERP system.

Salesforce's native tax engine and third-party integrations calculate and apply tax during invoice posting. Salesforce Billing then sends the invoice lines downstream to the ERP for digital or printed presentation and distribution. Invoice lines are also sent to the general ledger.

If you don't calculate tax in Salesforce, Salesforce Billing sends invoice data downstream to the ERP for tax calculation. Tax information will not appear in Salesforce unless the integration sends it back.

### Invoice Presentation & Distribution

The invoice presentation process takes Salesforce-generated invoice data and formats it into a digital or printed format for the customer. Like taxation, invoice presentation and distribution doesn't need to happen in Salesforce. However, Salesforce can still support it using a third-party tool.

### See Also

[Solving Common Business Challenges with Lead-to-Invoice](#)

[Key Lead-to-Invoice Functions in Salesforce Billing](#)

[Other Lead-to-Invoice Integration Considerations](#)

## Other Lead-to-Invoice Integration Considerations

Look at other important factors to consider when implementing Salesforce Billing as a lead-to-invoice solution. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Fulfillment and Provisioning

Synchronized fulfillment and provisioning processes ensure that customers are invoiced at the appropriate time. Salesforce Billing handles this information on the order and order product objects, which store information for how much to invoice the customer, how frequently to invoice, and when to invoice. Before passing this information to the invoice in Salesforce Billing, consider the other fulfillment and provisioning activities that your implementation requires. For example, you may need other integrations such as shipping carriers or custom software systems that provide license key generation that tell Salesforce Billing when to start invoicing. This data flow ensures that invoicing occurs at the proper time relative to when the customer receives their good or service.

#### Subscription and Asset Management

Salesforce CPQ and Billing let you manage subscriptions and assets over time and oversee how this information converts into an order and invoice. Some customers may use the contract, subscription, and asset objects to integrate with other systems such as a BI or ERP. The information on these objects can help businesses understand their customers better and provide reporting data such as customer churn, upsell and cross-sell scenarios, and product popularity.

#### Invoice Adjustments

Human error, customer concerns, or fee changes often require invoice balance adjustments, cancellations, or recreations. Since the invoice is a legal record, users can't change its balances directly. However, you can use Salesforce Billing's credit note and debit note features to apply balance changes and keep records of these changes available for bookkeeping.

When a user makes a credit or debit adjustment transaction in Salesforce Billing, the ERP system must receive a record of this transaction so it can adjust the balance for payment collection. Communicating this information downstream is also important for financial reporting and general ledger management.

#### See Also

- [Solving Common Business Challenges with Lead-to-Invoice](#)
- [Key Lead-to-Invoice Functions in Salesforce Billing](#)
- [ERP Integration Architectural Considerations for Lead to Invoice](#)

## Lead to Cash

When you implement Salesforce Billing, consider the differences between Salesforce and an ERP within the lead-to-cash approach. These differences help you identify whether lead-to-cash is the correct deployment option and strategy for your Salesforce Billing implementation. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The Salesforce platform enables companies to drive internal technologies with the best interest of customers in mind. Salesforce Billing embodies this purpose by completing the end-to-end processes of the customer engagement lifecycle. The platform starts with a lead, which is priced through Salesforce CPQ and intersects with Salesforce Billing during the transition from order to invoice and the collection of payment. This architecture allows Salesforce CPQ and Billing to function as a single system and product catalog for carrying customer data from lead to payment (cash). Keeping your data in a single system results in a cleaner handoff to an ERP for the general ledger and financial reporting.

A pain point for many companies today is what we call the Sales-to-Finance gap. Sales and finance are too often siloed from one another, which result in missed selling opportunities, poor selling and quoting experiences, bad invoicing that result in poor forecasting, and ultimately an incomplete picture of your customers. As the subscription economy matures, cutting-edge sales and finance strategies require increased flexibility in quoting, selling, and delivery of goods and services, as well as improved customer billing experience and payment capture.

Subscription selling models are increasing in popularity, so you need the ability to quickly update products and pricing, automate deal renewals, easily update existing subscriptions, get invoices to customers, and report on recurring revenue. Finance departments running legacy billing and ERP systems are forced to dedicate more time and resources to correcting invoices, manually updating forecasts and revenue reports, and attempting collection.

These are costly human solutions compensating for shortcomings in technology. Salesforce CPQ and Billing address these issues with an improved end-to-end Lead-to-Cash experience. This can involve Lead to Contact or Lead to Opportunity conversion, quoting with Salesforce, ordering within Salesforce, account invoicing, collections, and managing payments and credits. Furthermore, finance teams can keep pace with Sales by using complete selling and billing data for a complete view of your customers. That allows easier transactional data integrations with ERP systems for dunning, General Ledger (GL), financial reporting, and month end closing processes.

- ❗ **Important** Accounts receivables (AR) is defined as the subledger. Invoice generation, payment creation, and other financial processes impact AR. Some customers define AR as the invoice, payment, and credit, while others define AR as the financial close functions needed to close the AR subledger. Regardless of your definition, Salesforce Billing creates and manages payments and credits. This scenario focuses on managing AR in Salesforce Billing and feeding the details to the

ERP. If this scenario doesn't work for you, consider a Lead-to-Invoice implementation.



### Solving Common Business Challenges with Lead-to-Cash

Implementing Salesforce Billing for lead-to-cash allows you to solve several common business challenges. (Salesforce Billing Managed Package)

#### Key Lead-to-Cash Billing Functions

Salesforce Billing provides the capability for important lead-to-invoice functions. (Salesforce Billing Managed Package)

#### ERP Integration Architectural Considerations for Lead to Cash

When you evaluate Salesforce Billing as part of a lead-to-cash solution, consider the other systems necessary to perform billing functions such as payments and revenue recognition reporting. You'll also need to define the scope in which you use Salesforce Billing to integrate with an ERP. (Salesforce Billing Managed Package)

### Solving Common Business Challenges with Lead-to-Cash

Implementing Salesforce Billing for lead-to-cash allows you to solve several common business challenges. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Current AR and Financial Processes are Limited in ERP

Salesforce Billing provides a streamlined customer experience from lead to cash while supplementing downstream ERP with valuable transactional data for general ledgers, financial reporting, and month-end processes. If you're looking to use Salesforce Billing for lead-to-cash, make sure that you understand how your current business processes fit into the Salesforce and Salesforce Billing work flow. Businesses using Salesforce Billing for lead-to-cash functions generally have a customer focused view on A/R activities. Salesforce Billing is not intended to replace an existing ERP system.

### AR Managed in Salesforce and Details Loaded into ERP

A lead-to-cash solution is best for businesses that want a customer-focused view on A/R and currently have lightweight A/R and collections processes. Businesses may not benefit from Salesforce Billing if they already have robust A/R with complex dunning notifications in an existing ERP system. Salesforce is a customizable platform, but it's best practice to perform a full analysis of the business requirements to determine the best design and solution.

### Payment Process is Laborious and Manual

Finance users often have to switch between different processes or systems, which is time consuming and error prone. Siloed systems often lead to unallocated payments. Lead-to-cash

helps solve existing challenges with payment collections and allocations processes. Businesses that currently process payments manually will benefit from using a preferred payment gateway with Salesforce Billing. This workflow allows customers to sign up for auto-pay, where invoices are paid in full automatically. All products and services are invoiced exactly as they were quoted and ordered.

## See Also

[Key Lead-to-Cash Billing Functions](#)

[ERP Integration Architectural Considerations for Lead to Cash](#)

## Key Lead-to-Cash Billing Functions

Salesforce Billing provides the capability for important lead-to-invoice functions. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Lead-to-cash includes all of the key billing functions that lead-to-invoice provides. To review those functions, check out [Key Lead-to-Invoice Functions in Salesforce Billing](#).

## Lead-to-Cash Functions

### Payments & Payment Allocation Against Invoices

The primary difference between the lead-to-invoice and lead-to-cash deployment methods is the process of payment collections within Salesforce Billing. Rich customer account information helps businesses track payment methods and different payment types, allowing flexibility whether the customer is paying through credit, debit, ACH, or check. Native integrations with leading payment processors allow finance teams to schedule recurring payment runs to pay off invoice balances in full and automatically. Advanced AR provides flexible line-level allocation, which awards further precision to the collection process. Users can accomplish all these tasks through a simple point and click interface, all without leaving Salesforce Billing.

### Invoice Adjustments and Refunds

Adjustments are changes to a customer's accounts receivable or balance. Salesforce Billing offers flexibility for managing adjustments, such as the addition of a fee or applying a credit against an invoice. Adjustments also come from service cancellation and error correction. All of these adjustments change invoice balances, and managing them in multiple systems is often a highly manual and error-prone process. Within lead-to-cash, Salesforce Billing users can apply credits and debits and allocate them appropriately to individual invoice lines to increase or decrease the balance. Salesforce automates the crediting process when canceling an invoice and gives finance teams the ability to manually create and allocate credit and debit notes for one-off scenarios. Applying these adjustments to existing Salesforce Billing invoice data helps curb errors and

package AR data for handoff to the ERP system.

### Accounts Receivable (AR)

Accounts Receivables represents the cash owed to a business for goods and services they provided their customers. AR increases as goods and services are delivered and decreases after payment for those services. Accurate AR relies on billing and financial management systems to closely align, since bills generated from the billing engine directly impact receivables. Without alignment, companies risk overstating or understating how much their customers owe jeopardizing the relationship with the customer and creating challenges for forecasting future costs and revenue.

AR in Salesforce Billing includes the aggregation of posted open invoice balances against posted unapplied payments, credit notes, and debit notes. Salesforce Billing provides custom reporting functionality that provides a point-in-time view of an AR balance on an account. Salesforce Billing also provides flexible custom report functionality that can be configured to meet specific business requirements. When you use payments and adjustments, Salesforce Billing becomes the source record for all invoice balance information, then passes it down to the ERP for GL and financial reporting. As a best practice, we recommend integrating Salesforce Billing with a more robust ERP system for in-depth AR reporting general ledger posting, and custom subledger posting.

### Collections & Dunning Notifications

Collection activities occur when a customer's invoice balance is past-due. Businesses need to define thresholds to begin collection activities and also define the different metrics to trigger collection attempts. Dunning is one of the possible processes within collections for communicating demands for collecting payment on past-due invoice balances. Salesforce Billing allows for two types of collection processes.

- **Light Collections:** Processes aimed at decreasing past-due balances by collecting payments and decreasing bad debt. Dunning includes lightweight automation and configuration together with using balance snapshots and custom reports.
- **Heavy Collections:** Includes the same objectives as light collections, but also replaces robust ERP and FMS. Heavy collections include customizations required for purpose-built screens for complex financial operations processes like write-off, collections, treatments, and late fee management.

Use the combination of invoice data and powerful automation tools to build custom dunning processes within Salesforce Billing. These processes help financial departments streamline customer interactions and track outstanding past-due invoice balances for further analysis. Salesforce Billing's lead-to-cash flow easily rolls this information back up to the customer account, ensuring a complete view of customer touch-points.

### See Also

- [Solving Common Business Challenges with Lead-to-Cash](#)
- [ERP Integration Architectural Considerations for Lead to Cash](#)

## ERP Integration Architectural Considerations for Lead to Cash

When you evaluate Salesforce Billing as part of a lead-to-cash solution, consider the other systems necessary to perform billing functions such as payments and revenue recognition reporting. You'll also need to define the scope in which you use Salesforce Billing to integrate with an ERP. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

 **Important** Evaluate object integration points and ERP-required data on a business-by-business basis.

#### Invoice and Invoice Line Record Data Generation

The most critical lead-to-invoice integration point falls on the invoice and invoice line objects. These objects integrate with an ERP system, which then performs payments, adjustments, revenue recognition reporting, A/R, and financial reporting.

The invoice and its related Salesforce records contain all the information needed to determine what the customer is charged, when payment is due, and what to recognize for revenue. The invoice also drives A/R and supports financial reporting. Salesforce Billing generates all relevant invoice data including dates, quantities, and totals for calculating tax and customer presentation. Salesforce stores this data and then passes it to the ERP system for actual data.

For example, you could use Salesforce Billing for the creation of invoice and invoice line data, then send that information downstream to an ERP system for the next billing processes and balance management. In this scenario, Salesforce is the source of invoice charge information while the ERP system is the master for payments, adjustments, and the invoice and account balance.

#### Tax Management

Lead-to-Invoice solutions don't require tax calculation in Salesforce. Salesforce Billing provides flexible options for calculating tax via Salesforce native tax, third-party tax integrations, or in the downstream ERP system.

Salesforce's native tax engine and third-party integrations calculate and apply tax during invoice posting. Salesforce Billing then sends the invoice lines downstream to the ERP for digital or printed presentation and distribution. Invoice lines are also sent to the general ledger.

If you don't calculate tax in Salesforce, Salesforce Billing sends invoice data downstream to the ERP for tax calculation. Tax information will not appear in Salesforce unless the integration sends it back.

## Invoice Presentation & Distribution

The invoice presentation process takes Salesforce-generated invoice data and formats it into a digital or printed format for the customer. Like taxation, invoice presentation and distribution doesn't need to happen in Salesforce. However, Salesforce can still support it using a third-party tool.

## Applied and Unapplied Payment Information

Salesforce Billing provides flexible functionality for collecting payments and enriching that data for downstream systems. Payment records track important payment information such as payment type, amount of payment, date of payment, and payment account.

When you record payments in Salesforce Billing, you'll need to send the transaction information downstream to the ERP/FMS system for GL and financial reporting. Salesforce Billing contains four objects to consider for this integration.

### Payment

The payment header record contains important information including payment status, balance, and financial details. Payment record information feeds the downstream GL.

### Payment Allocation (Invoice) and Payment Allocation (Invoice Line)

Salesforce Billing creates a payment allocation record after a payment is allocated toward an invoice or invoice line. The package's Advanced AR setting determines whether the allocation goes toward the invoice or invoice line records. Payment allocations show how much of the payment header was allocated. You can use allocations to reconcile customer balance information.



**Note** Advanced AR is a global setting, so only the payment allocation (invoice) or payment allocation (invoice line) will be used for a possible integration point.

### Payment Allocation (Debit Note Lines)

Salesforce Billing also makes allocation records for payment allocations performed on debit note lines. This information is not necessary for downstream integration, but you can use it to improve downstream processes and reporting.

## Invoice Adjustments - Credit Notes

During the Salesforce Billing invoicing process, businesses often need to reduce invoice balances or cancel an invoice and recreate it. These changes can result from human error, customer concerns, or early-payment discounts. You'll need to make these adjustments in Salesforce Billing to ensure accurate invoice balances and payment collections. You can use credit notes or Cancel & Rebill functionality for your invoice balance correction needs.

When you make a credit note adjustment, you'll need to send the ERP system necessary information for financial reporting and GL purposes. This information can include data on the credit note, credit note line, and credit note allocation objects.

### Invoice Adjustments - Debit Notes

When you make a debit note adjustment, you'll also need to send the ERP system necessary information for financial reporting and GL purposes. This information can include data on the debit note, debit note line, and debit note allocation objects.

### Invoice Adjustments - Refunds

Salesforce Billing allows businesses to refund customers for previously collected payments or credit notes. With refunds, businesses can manage adjustments all the way through the billing process.

After performing a refund adjustment, send the ERP system necessary information for financial reporting and GL purposes. This information can include data on the refund, refund line, and refund line (credit note line), and refund line (payment) objects.

#### See Also

[Solving Common Business Challenges with Lead-to-Cash](#)

[Key Lead-to-Cash Billing Functions](#)

## Revenue Recognition Reporting Integration

When you implement Salesforce Billing, consider the accounting policies needed to accurately create revenue recognition reports. These requirements help you figure out whether a revenue recognition reporting integration out of Salesforce Billing is the correct strategy for your business. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

In Salesforce Billing, revenue recognition reporting represents the source of all revenue data that passes down to the ERP system as the general ledger receives the source data. Salesforce Billing's revenue recognition functionality can support or enhance existing revenue recognition solutions and improve downstream revenue processes by providing accurate revenue data.

When you integrate with an ERP system, the generation of the following records has an impact on the ERP's deferred and recognized revenue reporting.

- Revenue schedules and revenue transactions related to order products
- Invoice lines
- Credit note lines
- Debit note lines

Regardless of your deployment option, Salesforce Billing is flexible and allows a business to utilize revenue recognition capabilities. It's important to review your GAAP and IFRS guidance, accounting policies, and revenue triggers along with current business processes and ERP system limitations to

determine the best solution.

### Solving Challenges with Revenue Recognition Reporting Integration

Businesses may gain value from implementing a revenue recognition reporting integration from Salesforce Billing (Salesforce Billing Managed Package)

### Key Revenue Recognition Reporting Functions in Salesforce Billing

When you're working with revenue recognition reporting in Salesforce Billing, review important functions. (Salesforce Billing Managed Package)

### Considerations for Integrating Revenue Recognition Reporting to an ERP

When you're integrating Salesforce Billing's revenue recognition reporting processes with an ERP, review important considerations. (Salesforce Billing Managed Package)

## Solving Challenges with Revenue Recognition Reporting Integration

Businesses may gain value from implementing a revenue recognition reporting integration from Salesforce Billing (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Improving Revenue Recognition Capabilities

Salesforce Billing provides extensive revenue recognition reporting capabilities for one-time, subscription, and usage-based products. You can track standard pro-rata recognition for subscriptions and full recognition for one-time and usage charges. You can also manage complex scenarios by splitting revenue amounts across several performance obligations and define custom recognition periods for each product in a sale.

### Recurring Revenue Forecasting

Many businesses with existing ERP systems face challenges with insufficient or disorganized data across multiple systems. They may require manual intervention to determine what was sold to the customer and how to recognize the revenue based on the products and services delivered. These challenges are also common when moving to a recurring business model. Legacy software suites are not built to manage recurring revenue schedules, especially under changing International Financial Reporting Standards (IFRS) and Generally Accepted Accounting Principles (GAAP) revenue recognition standards. Salesforce Billing solves this problem by generating revenue schedules for each service included in a sale, based on the contract's revenue recognition policy.

### Meet Accounting Standards Codification (ASC) 606 Revenue Recognition Requirements

Many organizations doing B2B commerce enter into complex contract arrangements with their customers and must comply with ASC-606 and IFRS-15 revenue reporting regulations. Salesforce Billing brings together customer, order, contract, and invoice data on one platform to help handle complex revenue recognition requirements.

Each line item in a sale looks up to a revenue schedule. Salesforce Billing creates revenue schedules upon either order activation, invoice posting, or credit note/debit note posting. Each schedule's duration is based on the service period of the performance obligation. This period can be the standard start and end dates for an order or invoice line, or a custom date range. Salesforce Billing also supports assigning different revenue recognition policies and GL accounts to a sale based on the legal entity responsible for the transaction.

## Key Revenue Recognition Reporting Functions in Salesforce Billing

When you're working with revenue recognition reporting in Salesforce Billing, review important functions. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Revenue Extensions to the Standard CPQ Product Catalog

Salesforce Billing adds revenue recognition rules, treatments, and distribution methods to support your Salesforce CPQ product catalog.

#### Order-Based Revenue Recognition for Forecasting

Order-based revenue recognition reporting lets you track and earn revenue throughout an order product's lifecycle, regardless of its invoicing activity. Order-based revenue recognition reporting policies are often used when service delivery begins before or after invoicing.

When you generate revenue upon order product activation, you get high visibility into a company's future revenue streams. Creating revenue schedules based on order products forecasts your future deferred revenue liability for the life of a performance obligation. This forecasting allows businesses to view accurate revenue forecasts up-front for one-time and recurring products.

#### Billing-Based Revenue Recognition

Recognizing revenue on the invoice line is ideal for basic pro-rata recognition of subscription revenue. Invoice-based recognition creates revenue schedules that mirror the service period or periods invoiced and take revenue evenly throughout the period of performance. If the amount of revenue earned isn't equal to the amount billed, you can also use custom revenue amounts. In addition to pro-rata recognition, you can use invoice plans to create custom invoice schedules. Custom invoice schedules are often used for milestone billing and revenue recognition.

Recognizing revenue on credit note lines is necessary for adjustment scenarios including credit and rebilling, write-offs, and SLA credits. When you use the Cancel & Rebill button to cancel an invoice, Salesforce Billing creates credit notes to reduce the invoice line's balance to zero. To recognize revenue information accurately in this case, you'll need to offset previously recognized revenue using revenue recognition treatments and distribution methods.

## Order and Invoice Revenue Recognition

Salesforce Billing lets you configure a product with a revenue recognition rule that tracks revenue on both the order product and its related invoice line. This setup helps split revenue pipeline forecasting from the actual revenue reporting process. You can configure this setup by creating a rule with two unique revenue recognition treatments, each associated to its own revenue distribution method. Salesforce Billing then creates one revenue schedule for forecasting when an order product is activated, and another schedule for reporting when the related invoice line posts.

## Considerations for Integrating Revenue Recognition Reporting to an ERP

When you're integrating Salesforce Billing's revenue recognition reporting processes with an ERP, review important considerations. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Order-Based Revenue Integration Timing

When recognizing revenue on the order product, the push from Salesforce Billing to your back-end system can happen at any time after order activation. However, consider whether you need to factor amendments or cancellations into the integration architecture. An amendment always creates an order and order product for the delta change in quantity. This new order product in turn creates its own revenue schedule and transaction or transactions with the delta calculated. You'll need to determine whether your reconcile this revenue information in Salesforce Billing, middleware, or the ERP.

#### Invoice-Based Revenue Integration Timing

When you set up your ERP integration, evaluate how you want to recognize revenue. When you use invoice-based revenue recognition rules, you can push information to the back-end system at any point after invoice posting. Make sure that you also consider whether your integration requires real-time or scheduled integration. If the integration is real-time, you'll also need to evaluate scenarios such as cancel and rebill.

#### Difference Between Revenue Schedules and Revenue Transactions

Many ERP integrations pass down both revenue schedules and revenue transaction data. Revenue transactions provide more detailed revenue information than revenue schedules, including the specific amount per financial period. We recommend using revenue schedules when you have a performance obligation that recognizes revenue over time.

Revenue schedules contain header-level information including totals and revenue dates. Schedules are related to other records specific to the general ledger. Use revenue schedules when you recognize revenue only in full. The objects and information you'll have to integrate will vary based on business financial reporting requirements. You'll have to format these objects to

meet ERP system requirements as well.

## Preparing Your Salesforce Org for Billing

---

Salesforce Billing allows you to automate the billing process and keep detailed records of your transactions. You can set up several product fields and rules to guide the automation. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The billing process begins when you invoice your order. When this action occurs, Salesforce Billing creates an invoice record that contains invoice lines for each of your products. We call the act of creating an invoice from an order a “transaction.” The invoice lines and revenue schedules created from an invoicing action are called “transactional records.”

When you’re preparing your org for billing, consider key billing processes.

### Manage the Creation of Transactional Records

Salesforce Billing contains several rules that define when and how it creates your transactional records. You define lookups to these fields on your product, which pass their values to your order products when you order a quote. Many order products can look up to the same rule. This process allows you to group order products under one rule based on how you want Salesforce Billing to handle them. Rules answer the following questions.

#### Billing Rule

Should Salesforce Billing create an invoice line for this order product? If so, how should the invoice line’s pricing and date values reflect billing periods, billing dates and billing cycles?

#### Tax Rule

Should Salesforce Billing apply tax to this invoice line?

#### Revenue Recognition Rule

Should Salesforce Billing create a revenue schedule for this invoice line?

### Keep Records of Your Transactions

Modern legal standards also require that vendors keep detailed records of their transactions. Salesforce Billing provides two recordkeeping objects to help you with these processes.

## Finance Books

Store lookups to all the transactional records that occurred during one or more user-defined periods.

## GL Accounts

All your transactional records have lookup paths that end on the GL Account. This lets the GL Account act as an “endpoint” object that contains relationships to certain types of transactional records. You can export your Salesforce Billing GL Account data into an external general ledger management system for easy general ledger data organization.

The GL Rule creates the relationships between your transactional record and a GL account.

## Apply More Actions to Your Transactional Records

Rules also contain treatments. A treatment is an object containing a set of actions that you can apply to or lookups you can define for the order products that your rule evaluates. Salesforce Billing applies a treatment to an order product if the treatment and the order product have a matching legal entity.

Your rule can contain multiple treatments. This setup lets you group order products under one rule, then apply different treatments to subsets of that group. All treatments define the finance book and GL account that reference the transactional record the treatment is evaluating. The tax GL, and revenue recognition treatments define a few other actions.

### Tax Treatment

Which tax integration receives Salesforce Billing data for tax calculation?

### Revenue Recognition Treatment

When does Salesforce Billing make the revenue schedule, and how does Salesforce Billing spread revenue over a set time period?

### GL Treatment

Which GL Accounts have relationships to the tax, billing, and revenue information for this transaction?

## Aligning Your Org to Accounting and ERP Systems

Salesforce Billing lets you pass transactional data such as invoice lines, payments, and adjustments to an ERP system for reporting and revenue recognition. You can follow several guidelines for treating Salesforce Billing data to ensure a clean handoff to your accounting and ERP systems. (Salesforce Billing Managed Package)

## Defining Rules and Treatments

Salesforce Billing uses rules and treatments to group records for evaluation and take action on specific records within that group based on matching values. After Salesforce Billing performs certain actions, it evaluates all records that share the same rule and applies general changes to those records based on the rule’s settings. The rule also contains treatment records, which allow Salesforce Billing to apply

more specific changes. Salesforce Billing applies a treatment to one of the group's records only if the record and the treatment have the same legal entity. (Salesforce Billing Managed Package)

#### **Dynamic Invoice Plans**

Define an invoicing plan with custom amounts and billing frequencies. (Salesforce Billing Managed Package)

#### **Setting Up Usage Products**

Salesforce Billing provides several options for customizing pricing for your usage products. You can define minimum-usage fees, included amounts and overages, and price usage based on volume or tiered price levels. (Salesforce Billing Managed Package)

#### **Guidelines for Using Salesforce Billing with Multicurrency Enabled**

Review important guidelines for working with Salesforce Billing with multicurrency enabled. (Salesforce Billing Managed Package)

#### **Billing Account Fields**

Salesforce Billing provides custom managed fields for the account object. (Salesforce Billing Managed Package)

#### **Automation in Salesforce Billing**

Automating Salesforce Billing processes, such as activating orders and posting invoices, helps streamline cash collection for customers. It also lowers the potential for errors from users making a mistake during manual process configuration. When you work with automated processes, we recommend reviewing when your managed package automation starts and finishes, so that it doesn't interfere with custom implemented record updates. (Salesforce Billing Managed Package)

## Aligning Your Org to Accounting and ERP Systems

Salesforce Billing lets you pass transactional data such as invoice lines, payments, and adjustments to an ERP system for reporting and revenue recognition. You can follow several guidelines for treating Salesforce Billing data to ensure a clean handoff to your accounting and ERP systems. (Salesforce Billing Managed Package)

#### **Identifying Legal Entities**

Legal entities are Salesforce Billing objects that represent how your organization is structured. Each legal entity record relates to a billing, tax, revenue recognition, and general ledger treatment for an order. For example, if your business is split into multiple legal entities that sell different products or services to different regional customers, you can provide each entity with a different treatment that determines how data is mapped for handoff to an ERP. Based on these treatments, a legal entity can specify actions such as how to report on recognized revenue or how to credit and debit different general ledger accounts within the ERP system.

Every order product requires a lookup to a legal entity. You can control several treatment setups by creating a legal entity for each setup, then assigning your legal entities to order products as needed. For example, when you apply tax to an order product, Salesforce Billing checks the order product's legal entity, checks for a tax treatment record with a lookup to that legal entity, then applies the tax treatment accordingly.

By default, order products don't have a value for their legal entity. Use a process builder, workflow rule, or manual assignment to provide legal entities to your order products.

### Translating the Chart of Accounts

An organization's chart of accounts defines how to categorize billing transactions within the company for reporting purposes. We refer to these charts as GL Accounts in Salesforce Billing. Common account types include assets, liability, revenue/income, expense, and contra-revenue.

Salesforce Billing allows businesses to dynamically choose the correct account for a product based on legal entities and selling scenarios. You can also recognize a product or service's revenue within one bucket or split revenue into multiple buckets and accounts. This flexibility lets you minimize the number of SKUs you need while providing accurate information for accounting in an ERP ledger.

Salesforce Billing accommodates these scenarios through billing, revenue recognition, and tax rules, and their related treatments. You can set up an organization's chart of accounts as a series of records within Salesforce Billing, then associate them with rules and treatments through GL rules and GL treatments.

### Determining Revenue Distribution Methods

Subscription economies have caused complex rules around recognizing recurring revenue. Compliance requirements for ASC-606 and IFRS-15 revenue reporting regulations have added further challenges for enterprise B2B scenarios.

Salesforce Billing lets you recognize revenue for one-time, subscription, and usage-based products. Organizations can determine when in the billing process that your organization will recognize revenue from a good or service based on performance obligations. We've provided revenue distribution methods, which let you customize how you recognize revenue across different products. You can even split the revenue allocation of a product into multiple revenue transactions based on how you're charging the customer and recognizing revenue behind the scenes. You can then summarize Salesforce Billing's transactional revenue data and send it to an ERP system to complete revenue recognition, financial reporting, and month-end processes. Since Salesforce CPQ and Billing manage the product catalog, the ERP can then leverage active revenue processes.

### Integration Options

When you're integrating Salesforce Billing and an ERP system, consider the volume of data you're working with, timing and frequency requirements, and the ERP system's input formats and methods. At a high level, you have three options for sending data to the ERP.

- If your data volume and complexity is low, you can use a swivel chair process for manual data entry using Salesforce Billing transaction reports.
- Use a middleware tool to get Salesforce data via APIs, then transform it into a format that the ERP can load and consume. This approach is useful if you need an automated process to transform data due to its volume or complexity.

- Build a custom point-to-point integration to move transaction data in real-time or in batches. This approach is useful if your data requires a high degree of transformation.

### Store Transaction Records in Finance Books

Store financial transaction records in a finance book. These records include invoice lines, payment allocations, credit note lines, and revenue schedules. Salesforce Billing organizes these records by date and legal entity into finance periods. (Salesforce Billing Managed Package)

### Chart of Accounts

An organization's chart of accounts, known as GL Accounts in Salesforce Billing, defines how the company categorizes billing transactions for reporting purposes. You can set up a chart of accounts in Salesforce Billing and associate it with transactions by linking billing, tax, and revenue recognition treatments to GL rules and treatments. (Salesforce Billing Managed Package)

### Setting Up Taxation

Salesforce Billing uses tax rates and default tax addresses alongside your chosen tax engine. (Salesforce Billing Managed Package)

## Store Transaction Records in Finance Books

Store financial transaction records in a finance book. These records include invoice lines, payment allocations, credit note lines, and revenue schedules. Salesforce Billing organizes these records by date and legal entity into finance periods. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Finance books are useful for organizing your transaction records into several groups for reporting and recordkeeping purposes. For example, you could make a revenue finance book to track all your ASC606 revenue for each month of the year. You could also make an accounting finance book to track invoice lines created from your organization's quarterly UK sales.

The finance book Period Type field lets you define the types of transactions that Salesforce Billing records in the book.

- Choose Accounting to create lookups in this finance book to all invoice lines, credit note allocations, and debit note allocations from a transaction.
- Choose Revenue to create lookups in this finance book to all revenue transaction records from a transaction.

Your treatment's finance book lookup defines the finance book that receives records of the transactions that your treatment evaluates. Let's review each type of treatment and the transactional information it can pass to a finance book.

- Billing Treatment: When your billing rule creates an invoice line from your transaction, Salesforce Billing creates a lookup to that invoice line in an accounting finance book.

- Revenue Recognition Treatment: When your revenue recognition rule creates a revenue schedule from your transaction, Salesforce Billing creates lookups to the schedule's revenue transaction records in a revenue finance book.

Salesforce Billing groups a finance book's transaction records by date and legal entity into finance periods. The finance period record contains related lists with lookups to the invoice lines, credit note allocations, debit note allocations, or revenue transactions that your finance book is tracking. This process lets you organize your transaction records by a certain time frame. For example, you could create 12 month-long finance periods in an accounting finance book so you can organize all your accounting transactional record by month. Once your treatment assigns a transaction to a finance book, Salesforce Billing associates your transaction with one of the book's finance periods based on the following values:

- First, Salesforce Billing checks for a matching legal entity between the treatment and the finance period.
- Next, Salesforce Billing checks for a finance period where the transaction date falls within the period's Period Start Date and Period End Date.

Finance books and finance periods are useful for organizing your transaction records into several groups for reporting and recordkeeping purposes. Here are some examples of ways to organize your finance books.

### Finance Periods

A finance period is a reference object that stores lookups to all the accounting or revenue transactions that occurred within a period of time. (Salesforce Billing Managed Package)

## Finance Periods

A finance period is a reference object that stores lookups to all the accounting or revenue transactions that occurred within a period of time. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Versions

---

Revenue finance periods contain lookups to all Revenue Transaction records from a transaction. Accounting finance periods contain lookups to all invoice line, credit note allocation, and debit note allocation records from a transaction. All finance periods inherit Period Type from their parent finance book.

When Salesforce Billing creates a transaction, it evaluates finance periods that have the following criteria.

- Open period status
- Period type must be Revenue for revenue transactions, or Accounting for all other transactions
- Legal entity matches the transaction's legal entity
- System period start date falls within the period's start and end dates
- Period currency matches the transaction's currency (for multicurrency orgs only)

When a finance period meets all these requirements, Salesforce Billing assigns the transaction to the period. If multiple periods have the same matching values, Salesforce Billing assigns the transaction to the first period that the search process returns. You can view assigned revenue transactions in a period's Revenue Transactions related list, and all other assigned transactions in a period's Invoice Line, Debit Note Allocations, and Credit Note Allocations related lists.

Only open periods receive transactions. To prevent a finance period from receiving more transactions, set its period status to Closed.

When you change a finance period's status from Open to Closed and save the period record, Salesforce Billing then sets the status to Pending Closed while it runs validations on the finance period and its related objects. If any of the validations returns an error, Salesforce Billing changes the period's status to Error and creates an error log in the period's Error Logs related list. Review the error log and take corrective action, then change the period status from Error to Closed.

If you must reopen a closed finance period, change its period status from Closed to Open. Salesforce Billing then sets the status to Pending Open while it runs validations, and follows the same error logging process for any validation errors that occur. After you've made corrections, change the status from Error to Pending Open.

**!** **Important** Change period statuses only from Open to Closed and from Closed to Open. If you set a pending value manually, Salesforce Billing doesn't run validations on the period's related entities. If you accidentally set a period to a pending value manually, change it back to the original value before trying to open or close it again.

You can assign a different legal entity to various finance periods within a finance book. This process lets you make a finance book that tracks the transactions for several legal entities. If you don't provide a legal entity, Salesforce Billing evaluates other finance periods with null legal entities during transaction assignment.

Standard finance periods must begin on the first of a month and end on the last day of the same month. When you create a finance book, Salesforce Billing lets you automate the creation of monthly finance periods for each month within the finance book's date range. You can also create finance periods for custom date ranges.

#### [Batch Create Monthly Finance Periods](#)

Finance periods store records of all your transactions within a certain timeframe. Use the finance book's Create Finance Periods button to create batches of up to 49 monthly finance periods at a time. (Salesforce Billing Managed Package)

#### [Manually Create Finance Periods](#)

Finance periods store records of all your transactions within a certain timeframe. You can manually create finance periods of any date range by using the finance book's New Finance Period button. You can also create multiple finance periods at once by uploading them through a CSV file. (Salesforce Billing Managed Package)

#### [Create Finance Periods for Custom Date Ranges](#)

Create finance periods for non-calendar month timelines. (Salesforce Billing Managed Package)

## Batch Create Monthly Finance Periods

Finance periods store records of all your transactions within a certain timeframe. Use the finance book's Create Finance Periods button to create batches of up to 49 monthly finance periods at a time. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Revenue finance books support up to 180 finance periods. Accounting finance books don't have a finance period limit. Closed finance periods don't count toward a revenue finance book's period limit.

When you create finance periods using the Create Finance Periods button, you can create up to 49 finance periods at a time.

1. Click **Create Finance Periods** on your finance book.
2. Set your finance period start date.  
The start date should be the first day of a month.
3. Enter your number of finance periods. Salesforce Billing can create between 1 and 49 periods at a time.  
Salesforce Billing creates a finance period for one month starting on your finance period start date, then adds months based on the number of finance periods.
4. Set your legal entity.
5. If your org is multicurrency, set the finance period's currency.
6. Click **Submit**.



**Note** If you want to create more than 49 periods for your finance book, repeat these steps as needed.



### Example

You want a finance book that tracks monthly transactions from January 1, 2018, through August 31, 2019. Click **Create Finance Periods** and fill out the values as follows.

- Finance Period Start Date: Jan 1, 2018
- Number of Finance Periods: 20

## Manually Create Finance Periods

Finance periods store records of all your transactions within a certain timeframe. You can manually create finance periods of any date range by using the finance book's New Finance Period button. You can also create multiple finance periods at once by uploading them through a CSV file. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

Revenue finance books support up to 180 finance periods. Accounting finance books don't have a finance period limit.

1. From your finance book, go the Finance Periods related list and click **New Finance Period**.
2. Give your finance period a name.
3. Enter a period start date on the first day of the month that you want your finance period to cover.
4. Enter a period end date on the last day of the same month as your start date.
5. Set your period status.
6. Optional: Set your finance period legal entity.

A legal entity provides another way to filter transactions into your finance period. After filtering transactions that fall within your finance period's dates, Salesforce Billing includes only transactions with legal entities that match your finance period's legal entity. If your finance period doesn't have a legal entity, Salesforce Billing filters by period date and then includes only the transactions that don't have legal entities.

## Create Finance Periods for Custom Date Ranges

Create finance periods for non-calendar month timelines. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Spring '20 and later

-  **Note** Salesforce Billing allows up to 180 open finance periods for revenue finance periods that share a finance book, legal entity, and currency ISO code. This limit can't be increased. Accounting finance books don't have finance period record limits.

1. On your finance book, choose your period type, and then set your finance period duration to Custom. Save your changes.
2. On your finance book, click **New Finance Period**.  
If you click **Create Finance Periods** while finance period duration is Custom, Salesforce Billing opens the Create Finance Periods Visualforce page instead. This page lets you make only monthly finance periods, even if you're working in a custom period finance book.
3. Enter a start date and end date for your period, and then save your changes.  
A finance period's dates can't overlap with other finance periods that use the same legal entity in the same finance book.
4. Create more finance periods as needed.  
Salesforce Billing doesn't validate for gaps between finance periods. If your finance book contains a gap between periods, revenue schedules won't create transactions for dates in that gap. For example, you could create one finance period for 01/01/20 through 02/29/20, and another for 12/01/20 through 12/31/20. If you create a revenue schedule for 01/01/20 through 12/31/20, Salesforce

Billing will create revenue transactions only for transactions that fall in 01/01/20 through 02/29/20 or 12/01/20 through 12/31/20.

 **Example** Your company follows the 4-4-5 calendar for accounting periods, where each year contains four 13-week quarters. Each quarter contains a 4-week period, another 4-week period, and a 5-week period. This way, your financial reporting covers an equal number of days over all four quarters in a year.

Set up a finance book for the first quarter by using a Custom finance period duration and the following finance periods.

#### **Finance Period 1**

Start Date: 01/01/2020

End Date: 01/28/2020

#### **Finance Period 2**

Start Date: 01/29/2020

End Date: 02/26/2020

#### **Finance Period**

Start Date: 02/27/2020

End Date: 04/01/2020

## Chart of Accounts

An organization's chart of accounts, known as GL Accounts in Salesforce Billing, defines how the company categorizes billing transactions for reporting purposes. You can set up a chart of accounts in Salesforce Billing and associate it with transactions by linking billing, tax, and revenue recognition treatments to GL rules and treatments. (Salesforce Billing Managed Package)

Salesforce Billing uses three objects to manage GL functionality.

#### **GL Rule**

The parent record related to one or many GL treatments

#### **GL Treatment**

Child record of GL rules. Contains lookups to GL account through either the Credit Account and Debit Account fields.

#### **GL Account**

All records on the GL account represent the business chart of accounts for Salesforce Billing

transactions.

Let's look at how you could use a billing rule and a revenue recognition rule to direct data from a covered product into two different GL accounts. One account records your invoice lines while another records your revenue transactions.

### **Billings Information**

- You have a GL account that you use to track billing information from your vendor-resold products. The account is named Vendor Reselling GL account - Billings.
- You have a GL rule called GL - Vendor Reselling. The rule contains a GL treatment that defines your Vendor Reselling GL Account - Billings as the treatment's credit GL account.
- You have a billing rule called Billing - Vendor Reselling, which defines the invoice line generation and billing details for your vendor-resold products. This rule contains a billing treatment with a Billing GL Rule field that looks up to GL - Vendor Reselling.
- When you invoice your order product, Salesforce Billing uses your billing rule to create invoice lines on your invoice. Each invoice line shows the GL rule and GL treatment that your billing treatment applied. You can find these lookups in the Billing GL Rule and Billing GL Treatment fields. This series of lookups lets you define a path from the invoice line to your GL treatment, and then to Vendor Reselling GL Account - Billings.

### **Revenue Information**

- You have a GL account that you use to track revenue information from your vendor-resold products. The account is named Vendor Reselling GL Account - Revenue.
- You have a GL rule called GL - Revenue for Vendor Reselling. The rule contains a treatment that defines your Vendor Reselling GL Account - Revenue as the treatment's credit GL account.
- You have a revenue recognition rule called Revenue - Vendor Reselling, which defines the revenue schedule creation details for your vendor-resold products. This rule contains a revenue recognition treatment. The treatment's Revenue Recognition GL Rule field looks up to GL - Revenue for Vendor Reselling.
- When you invoice your order product, Salesforce Billing uses your revenue recognition rule to create a revenue schedule and revenue transactions. The revenue transactions contain lookups to the GL rule and GL treatment that your revenue recognition treatment applied. This series of lookups lets you define a path from the revenue transaction to your GL treatment, and then to Vendor Reselling GL Account - Revenue.



**Example** Let's also look at a sample chart of accounts. Most companies have a name and numbering system related to their accounts so that they support journal entries. Common chart of account types include asset, liability, equity, revenue/income, expense, and contra-revenue.

Chart of Accounts

Balance Sheet Account Type	Name	Account Number
Asset	Cash	Cash 100
Asset	Accounts Receivable	Accounts Receivable 100

Balance Sheet Account Type	Name	Account Number
Liability	Deferred Revenue	Deferred Revenue 300
Income Statement	Earned Revenue	Earned Revenue 500

### GL Accounts

Store your accounting journal entries in a general ledger account. Salesforce Billing allows you to define a credit GL account and a debit GL account as endpoints for your entries. Later, you can export your GL accounts and all associated records into an external general ledger bookkeeping system. (Salesforce Billing Managed Package)

### GL Accounts

Store your accounting journal entries in a general ledger account. Salesforce Billing allows you to define a credit GL account and a debit GL account as endpoints for your entries. Later, you can export your GL accounts and all associated records into an external general ledger bookkeeping system. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The general ledger account record itself does not store any data outside a GL account number and GL account description. Rather, it acts as an endpoint of lookup relationships that you establish when you create a transactional record. You use these relationships to create a “map” of data that begins with the transaction record and ends with the GL account itself.

Your GL rule and GL treatment define how Salesforce CPQ creates these relationships. The GL treatment contains two fields that look up to GL accounts: The Credit GL Account field and the Debit GL Account field.

Your billing treatments, tax treatments, and revenue recognition treatments all have lookups to a GL rule. This allows you to assign different GL rules, and thus GL accounts, among several treatments on a rule. You can also assign different GL accounts based on the type of rule. This way, you could have different GL accounts for tracking your invoice line generation, tax application, and revenue transaction creation.



#### Example

Let's look at how you could use a billing rule and a revenue recognition rule to direct data from a covered product into two different GL accounts. You'll have one for your invoice lines and one for your revenue transactions.

- You create a GL account to track billing information from your vendor-resold products. The

account is named “Vendor Reselling GL Account - Billings.”

- You have a GL rule called “GL - Vendor Reselling.” The rule contains a treatment that defines your Vendor Reselling GL Account - Billings as the treatment’s credit GL account.
- You have a billing rule called “Billing - Vendor Reselling,” which defines the invoice line generation and billing details for your vendor-resold products. This rule contains a billing treatment. The treatment’s Billing GL Rule field looks up to GL - Vendor Reselling.
- When you invoice your order product, Salesforce Billing uses its billing rule to create invoice lines on the invoice record. The invoice lines display the GL rule and GL treatment that your billing treatment applied. You can find these lookups in the Billing GL Rule and the Billing GL Treatment field. This series of lookups lets you define a path from the invoice line to your GL treatment, and then to Vendor Reselling GL Account - Billings.
- You create a GL account to track revenue information from your vendor-resold products. The account is named “Vendor Reselling GL Account - Revenue.”
- You have a GL rule called “GL - Revenue for Vendor Reselling.” The rule contains a treatment that defines your Vendor Reselling GL Account - Revenue as the treatment’s credit GL account.
- You have a revenue recognition rule called “Revenue - Vendor Reselling,” which defines the revenue schedule creation details for your vendor-resold products. This rule contains a revenue recognition treatment. The treatment’s Revenue Recognition GL Rule field looks up to GL - Revenue for Vendor Reselling.
- When you invoice your order product, Salesforce Billing uses its revenue recognition rule to create a revenue schedule and revenue transactions. The revenue transactions contains lookups to the GL rule and GL treatment that your revenue recognition treatment applied. This series of lookups lets you define a path from the revenue transaction to your GL treatment, and then to Vendor Reselling GL Account - Revenue.

## Setting Up Taxation

Salesforce Billing uses tax rates and default tax addresses alongside your chosen tax engine. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### [Configuring Default Tax Addresses](#)

Salesforce Billing lets you define a default tax calculation address in the package settings. By default, all your org’s tax calculations reference this setting. You can also define tax address override values on individual order product records. (Salesforce Billing Managed Package)

#### [Set Up Tax Rates for Standard Tax Integrations](#)

Salesforce Billing uses the tax rate object to store tax rates for standard tax integrations. To accurately perform tax calculations for a standard tax integration, Salesforce Billing requires a tax rate for every

tax jurisdiction or address. (Salesforce Billing Managed Package)

## Configuring Default Tax Addresses

Salesforce Billing lets you define a default tax calculation address in the package settings. By default, all your org's tax calculations reference this setting. You can also define tax address override values on individual order product records. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The Salesforce Billing package settings General tab contains the **Tax calculation is based on?** field, which lets you pick from several types of addresses. Make sure you select an address here before calculating tax, regardless of your tax engine. If you don't select an address, your org's tax calculations will return unexpected rates.

Default Tax Calculation Address Types

Field Value	Description
Order: Account Billing Address	Billing address on the account record defined by your order's Account field.
Order: Account Shipping Address	Shipping address on the account record defined by your order's Account field.
Order: Bill To Contact	Mailing address on the contact record defined by your order's Bill to Contact field.
Order: Ship To Contact	Mailing address on the contact record defined by your order's Ship to Contact field.
Billing Account: Billing Address	Billing address on the account record defined by your order product's Billing Account field.
Billing Account: Shipping Address	Shipping address on the account record defined by your order product's Billing Account field.
Billing Account: Bill To Contact	Mailing address on the contact record defined by your billing account's Bill to Contact field.

-  **Note** If you choose a billing account field and your order product doesn't have a billing account, Salesforce Billing uses your order's account instead.

Order products also contain several fields for overriding all or part of the default tax address. We recommend using these fields if any of your products have different tax addresses than your org's default tax address.

### Order Product Override Fields

Field Label	API
Tax Street 1 (Override)	TaxStreet__c
Tax Street 2 (Override)	TaxStreet2__c
Tax City (Override)	TaxCity__c
Tax State (Override)	TaxState__c
Tax County (Override)	TaxCounty__c
Tax Postal Code (Override)	TaxZipCode__c
Tax Country (Override)	TaxCountry__c

 **Important** Salesforce Billing requires a package-level default tax address, even if all of your order products use address override fields.

 **Example** Global Containers uses Salesforce CPQ and Salesforce Billing. Within their business model, one contact always relates to several orders, each of which have different billing and shipping addresses. Instead of spending a lot of time remaking contact records to account for the different billing and shipping addresses, Global Containers can use tax override fields on each order.

## Set Up Tax Rates for Standard Tax Integrations

Salesforce Billing uses the tax rate object to store tax rates for standard tax integrations. To accurately perform tax calculations for a standard tax integration, Salesforce Billing requires a tax rate for every tax jurisdiction or address. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

For standard tax integrations, a tax rate defines a percentage value to apply to an order product or invoice line during tax calculation. Upon record creation, Salesforce Billing evaluates the order product or invoice line's tax treatment. If the tax treatment and tax rate have matching legal entities, Salesforce uses that tax rate to calculate tax for the order product or invoice line.

1. Create your tax rate.
  - a. From the tax rate page, click **New**.
  - b. Choose a tax identifier. This represents the name of your record and can be anything you want.
  - c. Add a state, city, zip code, and/or country. You must provide all the information above the lowest address level that you populate. For example, if your integration applies tax rates at the zip code level, your tax rate requires a zip code, city, state and country. If you apply tax at the state level, your tax rate requires only the state and country.
  - d. Provide a priority number. When multiple rates apply to a transaction with matching priorities,

Salesforce Billing aggregates the rates and applies them against the taxable amount. When priorities differ, Salesforce applies the lowest priority first. The taxable amount will add to the first tax amount according to priority level. The sum of the taxable amount and the first tax rate is used as the taxable amount for the second tax rate applied, and so on.

For example, let's say you had a \$5000 order amount with a 15% priority 0 tax rate and a 20% priority one tax rate. Salesforce Billing applies these calculations:

$$\$5000 * .15 = \$750. \$5000 + \$750 = \$5750$$

$$\$5750 * .2 = \text{Total estimated tax amount of } \$1900$$

- e. Choose a tax code. Tax codes are unique labels that group items together, like products, services, or charges. You can use tax codes with tax rules to create custom taxation setups.
  - f. Choose a tax rate value (as a percentage) to apply for this tax jurisdiction.
  - g. Choose a legal entity. Salesforce Billing applies taxes to order products and invoice lines if the tax rate's legal entity matches the legal entity on the order product/invoice line's tax treatment.
2. Configure tax treatment records.
    - a. To calculate tax accurately on order products and invoice lines, a tax rate and tax treatment must have matching tax codes.
  3. Configure product records.
    - a. Configure your tax rules on your product records. Your order products inherit their parent product's tax rule, but you can change the tax rule if necessary. Remember, Salesforce Billing applies a tax treatment to an order product if the records have matching legal entities.



**Example** Your company needs a standard tax rate for taxing software services in Germany.

Remember, this rate applies to all of Germany, so you need to define only the country in the tax rate's address fields.

- Tax Identifier: Software Services – Germany
- Country: Germany
- Priority: 0
- Rate: 20%
- Legal Entity: Germany Legal Entity

After changing a tax record, Salesforce Billing doesn't store that record's historical data. Administrators must manage changes to tax rates over time. Make sure that you have an historical data storage plan in place if this information is critical to business operations and overhead.

## Defining Rules and Treatments

Salesforce Billing uses rules and treatments to group records for evaluation and take action on specific records within that group based on matching values. After Salesforce Billing performs certain actions, it evaluates all records that share the same rule and applies general changes to those records based on the rule's settings. The rule also contains treatment records, which allow Salesforce Billing to apply more

specific changes. Salesforce Billing applies a treatment to one of the group's records only if the record and the treatment have the same legal entity. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Rules are initially defined on the product record and on legal entities.



### **GL Rules and GL Treatments**

General ledger rules define how Salesforce Billing records transactional data to your finance books. (Salesforce Billing Managed Package)

### **Define Your Invoicing Setup with Billing Rules**

Billing rules define how your order product produces an invoice line during an invoicing process. (Salesforce Billing Managed Package)

### **Tax Rules and Treatments**

Tax rules and their treatments define how Salesforce Billing processes tax for your transactions. (Salesforce Billing Managed Package)

### **Configuring Your Revenue Recognition Rules**

Revenue recognition rules define whether Salesforce Billing creates a revenue schedule for an invoiced order product or invoice line. (Salesforce Billing Managed Package)

## GL Rules and GL Treatments

General ledger rules define how Salesforce Billing records transactional data to your finance books. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

GL rules are a type of rule within the Salesforce Billing rules engine. All the products that you plan on billing require a lookup to a GL rule. When you order an invoice, your order products inherit this lookup.

A GL rule record contains the rule itself and a related list that contains GL treatments. The rule itself allows you to group order products for evaluation under a certain GL rule when you perform your transaction. The GL treatments contain lookups to a credit GL account and a debit GL account. These lookups establish a relationship between your transaction and your GL accounts. When a GL rule evaluates an order product, Salesforce Billing applies one of the rule's treatments if the order product and treatment have a matching legal entity.

The relationship between rules and treatments allow you to organize GL rules by the types of treatments they contain. For example, you could create one GL rule containing all the treatments for tracking your

billing transaction records. You could then create another GL rule containing the treatments for your revenue transaction records, and a third GL rule for your tax transaction records.

Your billing, tax, and revenue treatments all contain lookups to GL rules. This way, when you perform a transaction, the resulting transactional records all contain lookups to the rule defined in each treatment.

- Invoice lines contain a lookup to the GL rule you defined on the billing treatment that evaluated your transaction. Salesforce Billing lists this lookup on the invoice line record as the Billing GL Rule.
- Invoice lines contain a lookup to the GL rule you defined on the tax treatment that evaluated your transaction. Salesforce Billing lists this lookup on the invoice line record as the Tax GL Rule.
- Revenue transactions contain a lookup to the GL rule you defined on the revenue recognition treatment that evaluated your transaction. Salesforce Billing lists this lookup on the revenue transaction record as the Revenue Recognition GL Rule.

This series of lookups allows you to establish a relationship between a transactional object and a GL Account every time you perform a transaction. You can use these relationships for bookkeeping records in external general ledger systems.

 **Example** Let's look at a basic general ledger setup. This setup uses GL accounts to track your billing, revenue, and tax transactional data for both credit and debit transactions in the US and the UK.

GL Rule name	GL Treatment Name	Credit GL Account Name	Debit GL Account Name
Standard Billing GL Rule	Billing GL Treatment - US	Billing Credit Account - US	Billing Debit Account - US
	Billing GL Treatment - UK	Billing Credit Account - UK	Billing Debit Account - UK
Standard Revenue GL Rule	Revenue GL Treatment - US	Revenue Credit Account - US	Revenue Debit Account - US
	Revenue GL Treatment - UK	Revenue Credit Account - UK	Revenue Debit Account - UK
Standard Tax GL Rule	Tax GL Treatment - US	Tax Credit Account - US	Tax Debit Account - US
	Tax GL Treatment - UK	Tax Credit Account - UK	Tax Debit Account - UK

## Define Your Invoicing Setup with Billing Rules

Billing rules define how your order product produces an invoice line during an invoicing process. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

Billing rules are a type of rule within the Salesforce Billing rules engine. All the products that you plan on billing require a lookup to a billing rule. When you order an invoice, your order products inherit this lookup.

A billing rule record contains the rule itself and a related list that contains billing treatments. The rule determines whether Salesforce Billing performs an action on the order product when you invoice an order. If Salesforce Billing determines it must perform that action, the treatments specify how Salesforce Billing performs that action and stores a record of the action in a general ledger. When a billing rule evaluates an order product, Salesforce Billing applies one of the rule's treatments if the order product and treatment have a matching legal entity.

Let's review some important billing rule fields.

### Generate Invoice

- Yes: When Salesforce Billing creates an invoice, it creates an invoice line for this order product
- No: When Salesforce Billing creates an invoice, it doesn't create an invoice line for this order product

### Partial Period Treatment

Define how Salesforce Billing creates invoice lines for order products that bill in advance of the order start date.

- Separate: Salesforce Billing creates two invoice lines on your invoice. The first line covers the period between the billing start date and the order start date, while the second line covers the entire next billing period. In the example above, the first invoice line has a start date of March 15 and an end date of March 31. The second line has a start date of April 1 and an end date of April 30.
- Combine: Salesforce Billing creates a single invoice line that covers the partial billing period and the first full billing period. In the example above, the invoice line has a start date of March 15 and an end date of April 30. All subsequent invoice lines follow a monthly billing period.

### Amendment Bill Cycle Date Alignment

Choose how Salesforce Billing determines the billing date of a new order product on an order made from an amended quote. This field applies only to the order products with a subscription charge type.

- Do not align amended order product: Default functionality. The amended order product bases its Next Charge Day and Next Bill Day fields from the amended order's Billing Day of Month field.
- Align to original order product: Optional functionality. If the Next Billing Date falls before the Last Charge To Date and the Billing Day of Month isn't set, the amended order product's Bill Through Date Override is set to the Last Charge To Date of the revised Order Product. See [Aligning Billing Dates for Amended Order Products](#).

## Period Treatment for BTDO

Control how Salesforce Billing creates invoice lines when an order product's Bill Through Date Override value creates a partial billing period. For more information, review [Managing Partial Periods When Using Bill Through Date Override](#).

### Billing Treatments

Billing treatments define how Salesforce Billing records billing rule actions in your finance books and general ledgers. You can use billing treatments to specify an invoice plan and cancellation invoice plan for dynamic invoice plans. You can also set up the cancellation rule for amendment orders to override the package setting for the default cancel order rule. (Salesforce Billing Managed Package)

## Billing Treatments

Billing treatments define how Salesforce Billing records billing rule actions in your finance books and general ledgers. You can use billing treatments to specify an invoice plan and cancellation invoice plan for dynamic invoice plans. You can also set up the cancellation rule for amendment orders to override the package setting for the default cancel order rule. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Your billing treatment contains lookups to the following objects.

- A finance book
- A GL rule
- A legal entity

When your billing rule evaluates an order product, Salesforce Billing applies one of the rule's treatments if the order product and treatment have matching legal entities. The treatment defines the finance book that receives finance period lookups to all invoice lines that your billing rule creates from this order product. For example, consider the following process.

1. You sell an enterprise resource planning order product in the UK.
2. The billing treatment associates your order product with the finance book you use to track your ERP sales.
3. Your billing rule creates an invoice line for that order product.
4. The treatment creates a lookup to the invoice line in one of the finance book's finance periods. Salesforce Billing chooses the finance period based on the date of the invoice line's creation and matching legal entities between the billing treatment and finance period.

Your billing treatment also associates the order product with a GL rule. Use this association to create general ledger records for each transaction that occurs from the order product your billing rule is evaluating.

### See Also

[Dynamic Invoice Plans](#)  
[Invoice Package Settings](#)

## Tax Rules and Treatments

Tax rules and their treatments define how Salesforce Billing processes tax for your transactions.  
(Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and later

---

Tax rules are a type of rule within the Salesforce Billing rules engine. All the products that you plan on billing require a lookup to a tax rule. When you order an invoice, your order products inherit this lookup.

A tax rule record contains the rule itself and a related list that contains the rule's treatments. The rule determines whether Salesforce Billing performs an action on the order product when you create an invoice from an order. If Salesforce Billing performs that action, the treatments specify how that action is performed. They also define how a record of that action is stored in a general ledger account.

You can have multiple treatments on a rule, and each treatment can have a unique legal entity lookup. Salesforce Billing applies a treatment to an order product if the two records have matching legal entities. This process considers a broad group of order products under a single tax rule while specifying how smaller sets of order products in that group are treated based on their legal entities.

The tax rule contains a Name, Active, and Taxable (Yes/No) fields.

- If you select **Yes** for Taxable (Yes/No), your tax rule has to contain a tax treatment. Salesforce Billing calculates and applies tax to the invoice lines that are created from the order products that the tax rule is evaluating. Your treatments define whether Salesforce Billing uses its own tax integration or an external tax platform during tax calculation.
- If you select **No** for Taxable (Yes/No), your tax rule can't contain a tax treatment. Use of a tax treatment for non-taxable products causes errors and unexpected behaviors.

Updates to the tax treatment on the order's products isn't supported. If a tax rule requires multiple tax treatments to satisfy business needs, we recommend multiple tax rules, each with a unique treatment for each affected legal entity. Rather than updating the treatment, update the order product tax rule by using a flow. Salesforce Billing updates the treatment. To change the rule or legal entity, make sure that the order is in the Draft status.

 **Important** You're limited to one tax treatment per legal entity.

Fields for Tax Treatments

Field	Description
Name	Provide a name for your tax treatment.

Field	Description
Tax Integration	Define the tax integration that evaluates all the transactions related to this tax treatment.
Tax Code	If your tax integration uses tax codes to calculate taxes, provide a tax code value.
Tax GL Rule	Choose the GL rule that evaluates this tax calculation. The GL rule and its treatment define the record-making process for all tax calculations related to this tax treatment.
Tax Legal Entity	Define a legal entity for this treatment.

## Configuring Your Revenue Recognition Rules

Revenue recognition rules define whether Salesforce Billing creates a revenue schedule for an invoiced order product or invoice line. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: **All** Salesforce Billing Editions

Revenue recognition rules are a type of rule within the Salesforce Billing rules engine. All the products for which you want to recognize revenue require a lookup to a revenue recognition rule. When you order a quote, your order products inherit this lookup. When you invoice an order, your invoice lines inherit the same lookup. Salesforce Billing begins the revenue transaction development process at this time.

A revenue recognition rule record contains the rule itself and a related list that contains revenue recognition treatments. The rule determines whether Salesforce Billing performs creates a revenue schedule for the invoice line when you invoice an order product. If Salesforce Billing determines it must perform that action, the treatments specify how Salesforce Billing performs that action and stores a record of the action in the general ledger and finance book. When a revenue recognition rule evaluates an invoice line, Salesforce Billing applies one of the rule's treatments if the order product and treatment have a matching legal entity.

If you want Salesforce Billing to create a revenue schedule for your invoice line, set your revenue recognition rule's **Create Revenue Schedule?** field to Yes.

Once you define which products should have a revenue schedule, your revenue recognition treatments let you specify how Salesforce Billing splits up and processes that product's revenue. Each treatment references a revenue distribution method, which controls how Salesforce Billing spreads revenue over a set time period.

### Revenue Recognition Treatments

Revenue recognition treatments let you control how and where Salesforce Billing recognizes a product's revenue. (Salesforce Billing Managed Package)

## Configuring Total Revenue Amount

The revenue schedule's Total Revenue Amount field shows the amount of revenue that Salesforce Billing can distribute across revenue transactions. The total revenue amount inherits its value from one of several order product fields based on how you configure the Revenue Schedule Amount field. You can find the Revenue Schedule Amount field on your revenue recognition treatment. (Salesforce Billing Managed Package)

## Revenue Recognition Treatments

Revenue recognition treatments let you control how and where Salesforce Billing recognizes a product's revenue. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Your revenue recognition treatments define how Salesforce creates revenue schedules and revenue transactions for a transaction. They also define how Salesforce Billing records revenue transaction data in finance books and general ledgers. Let's review the treatment fields that you can use to customize this process.

#### Processing Order

If you have multiple revenue recognition treatments targeting a single order product, you can give each of them a value here. Salesforce Billing applies your treatments in order starting with the lowest number.

#### Type

Defines the amount of the order product's revenue that Salesforce Billing attributes to this treatment. If you select Percentage or Flat Amount, you can define their values in the Percentage and Flat Amount fields. Salesforce Billing divides your Flat Amount value by a time period that you define in the revenue distribution method. If you select Remainder, Salesforce Billing applies any of your order product's leftover revenue to this treatment.

#### Percentage

Defines the percentage of a transaction that this treatment associates with a revenue schedule. For example, if you choose 75% of a \$1000 transaction, your revenue schedule accounts for \$750.

#### Revenue Schedule Creation Action

Defines the action that causes Salesforce Billing to create a revenue schedule and populate it with revenue transactions.

#### Revenue Distribution Method

Associate a revenue distribution method with this revenue treatment.

### Revenue GL Rule

All transactions processed through this treatment record journal entries based on this GL rule.

### Revenue Finance Book

All transactions processed through this treatment record entries in this finance book record.

### Revenue Legal Entity

This treatment applies to all order products that were considered under the parent revenue recognition rule and have a matching legal entity.

 **Example** You want to recognize half the revenue of your UK enterprise software sales in a standard finance book and the other half in a deferred finance book. You could make a revenue recognition rule with the following treatments. Since you want both treatments to apply to the same order products, your treatments have the same legal entity.

Name	Processing Order	Type	Percentage	Revenue Finance Book	Revenue Schedule Creation Action	Legal Entity
50% Treatment - Standard	1	Percentage	50	ES Standard Revenue	Order Activation	UK ES
50% Treatment - Deferred	2	Percentage	50	ES Deferred Revenue	Order Activation	UK ES.

### Configuring Total Revenue Amount

The revenue schedule's Total Revenue Amount field shows the amount of revenue that Salesforce Billing can distribute across revenue transactions. The total revenue amount inherits its value from one of several order product fields based on how you configure the Revenue Schedule Amount field. You can find the Revenue Schedule Amount field on your revenue recognition treatment. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

The Revenue Schedule Amount field has five picklist values. Each value corresponds to a field on the order product. When your revenue recognition treatment evaluates an order product, Salesforce Billing evaluates the corresponding order product field. It then sends the value of that field the revenue schedule's Total Revenue Amount field.

When you configure your revenue recognition treatments, choose the Revenue Schedule Amount value based on your revenue recognition needs for order products that fall under the treatment. For example, if you want to recognize revenue based on bookings amounts, set your revenue schedule amount to Bookings Amount. If your revenue recognition treatment evaluates an order product with a bookings amount of \$5000, the resulting revenue schedule has a total revenue amount of \$5000.

Let's review how each of these fields functions on the order product.

### **Bookings Amount**

Represents the total amount that Salesforce Billing will invoice the order product for across all its billing periods. Many organizations that recognize revenue from the order product use this field for forecasting purposes.

On standard subscription order products, the bookings amount and total price are always the same. However, an evergreen subscription has no predefined end date, so its total amount can differ from the actual amount you bill for. To prevent inaccurate data, Salesforce Billing always sets the bookings amount for an evergreen subscription order product to 0.

### **Revenue Allocation Amount**

Represents the revenue that has been allocated from the revenue element to revenue distributions. Different organizations have different ways of calculating revenue allocation, so the field doesn't have any value on the order product by default. You can populate it manually or use custom automation to provide a value based on your organization's requirements for calculating revenue allocation amounts.

### **Revenue Expected Amount**

Represents the revenue that finance operations users expect to come from the revenue source. Different organizations have different ways of calculating expected revenue, so the field doesn't have a default value. You can populate it manually or use custom automation to provide a value based on your organization's requirements for calculating expected revenue amounts.

### **Revenue Liability Amount**

Represents the amount of revenue liable for delivery for payments made in advance of delivering a service. Different organizations have different ways of calculating expected revenue, so the field doesn't have a default value. You can populate it manually or use custom automation to provide a value based on your organization's requirements for calculating revenue allocation amounts.

### **Transaction Amount**

Represents the total amount of the sales transaction, and inherits its value from the order product's Total Price field. This field is useful for organizations that recognize revenue based on the full amount of the sales transaction.

## Dynamic Invoice Plans

Define an invoicing plan with custom amounts and billing frequencies. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

Many customers have invoicing needs based on milestones or project acceptance. These workflows require customized invoicing dates that don't follow standard billing frequencies such as monthly or quarterly invoicing. You may also need to define amounts to invoice that vary based on certain criteria. For example, you could bill 10% of an order product's amount for the first billing transaction. You could then bill 25% when the first delivery occurs, and the remainder after the final delivery.

Salesforce Billing lets you define a customized invoicing process using the Invoice Plan object, which contains invoice plan lines. You can define a lookup to an invoice plan on a billing treatment. When you invoice an order product with a Billing Frequency field set to Invoice Plan, Salesforce Billing checks the invoice plan referenced on your order product's billing treatment. It then associates that invoice plan with your order product.

Your invoice plan contains invoice plan lines, which specify the custom behavior for one invoice transaction. You can define whether you want to bill an order product based on a percentage or remainder of its total amount. You can also define when Salesforce Billing can invoice the order product and provide a custom service period when applicable. The Processing Order field defines the order in which Salesforce Billing applies the lines of an invoice plan to the targeted order product.

### Important

- Dynamic invoice plans don't support amendments.
- Dynamic invoice plans don't support flat amount invoice plan lines.
- Dynamic invoice plans don't support changing the [billing frequency](#) from Invoice Plan to a recurring value, such as Monthly, after order product activation. Recurring billing frequencies can't be changed to Invoice Plan after order product activation. If you want to make such a change, deactivate the order product first, update the billing frequency, and then activate the order product.
- Invoice plan order products don't populate their Pending Billing Amount (without tax) field.
- Invoice lines created from invoice plan lines don't calculate unit price. Unit price is assigned from the Billing Transaction total.

## Billing Schedules and Billing Transactions

Billing schedules and transactions define how Salesforce Billing invoices an order product. When you invoice an order product with an invoice plan-driven billing frequency, Salesforce Billing creates a billing

schedule that contains billing transaction records for each invoice plan line. A billing transaction displays how much you've billed based on the relationship between the invoice plan line's commencement date and your order product's start and end dates. The billing schedule record displays the total amount that has been billed for that order product based on all the related invoice plan lines.

 **Note** When you're invoicing on a billing transaction, Salesforce Billing copies data from the billing transaction to the invoice line. Nothing is calculated. Amount is copied to the subtotal and unit price. Quantity is copied to the quantity field on the invoice line. The billing transaction must have a quantity.

For example, let's say you have a recurring order product with a value of \$1000. The order product is tied to an invoice plan that bills 20% one month after order product activation and the remainder two months after activation. When you invoice after one month, your invoice plan produces a billing schedule with two billing transactions. The first billing transaction has an amount of \$200 and the second billing transaction has an amount of \$800. However, since you haven't hit the second month of billing, your second invoice transaction has a status of Pending and an Amount To Be Billed of \$800. Your billing schedule reflects this data with a Billed Amount field of \$200. When you invoice again after two months, your second billing transaction accounts for the remaining \$800.

Your invoice contains an invoice line for each billing transaction with a status of Billed.

 **Example** Let's look at a visual representation of the dynamic invoicing process. This example shows an order product invoiced for 50% on activation and for 50% one month after activation. 

 **Example** Your organization provides server mainframe installation and management. You sold these services to a customer with an agreement to split the bill as follows.

- A 20% invoice after the first 10 days
- A 50% invoice after the first month
- An invoice covering the remainder of your fees after six months.

The order for this sale contains the following order products. All order products are recurring and billed in advance, with a billing frequency set to Invoice Plan.

Order Product				
Name	Total Price	Start Date	End Date	Next Billing Date
Project Customization	\$1200	01/01/2018	12/31/2018	05/20/2018

You can create an invoice plan to manage this invoicing process. Start off with an active invoice plan that creates billing schedules upon order product activation. Next, make your invoice lines. All your lines should commence and begin their service periods upon order product creation, and have their Service Period End fields set to Create Without Date.

### Invoice Plan Lines

Name	Type	Percentage	Commencement Date Offset	Commencement Date Offset Units	Processing Order
20 Percent	Percent	20%	10	Days	1
50 Percent	Percent	50%	1	Months	2
Remainder	none	none	6	Months	3

You invoice your order on 03/31/2018. Your invoice plan produces a billing schedule with three billing transactions. Since you haven't passed the sixth month yet, only the first two billing transactions in your schedule are billed.

### Billing Transactions

Name	Status	Amount	Amount To Be Billed	Invoice Plan Line	Invoice Line
20 Percent	Billed	\$240	\$240	20 Percent	Invoice line is created
50 Percent	Billed	\$600	\$600	50 Percent	Invoice line is created
Remainder	Pending	\$360	\$360	Remainder	Invoice line isn't created

## **Create an Invoice Plan and Billing Treatment Setup**

Create an invoice plan and associate it with a billing treatment. (Salesforce Billing Managed Package)

### **Milestone Billing**

Use invoice plans to invoice your customers in response to milestones. A milestone is a scheduled event that includes the completion of a major deliverable. (Salesforce Billing Managed Package)

### **Weekly Billing with Invoice Plans**

Use invoice plans to invoice your customers based on weekly schedules. (Salesforce Billing Managed Package)

### **Invoice Plans for Custom Billing Frequencies**

Use invoice plans to create billing schedules that don't follow standard billing frequencies such as monthly or quarterly. For example, you could make your first transaction ten days after the sale, and your second transaction two months after that. (Salesforce Billing Managed Package)

### **Override Billing Transactions**

You can override the target date and amount on your billing transactions. This feature is useful in milestone billing if a sales rep renegotiated terms and changes an invoicing setup that you already created. (Salesforce Billing Managed Package)

### **Canceling Order Products with Invoice Plans**

If your billing treatment looks up to a new order invoice plan, it also requires a cancellation invoice plan. When you cancel an order product covered by a billing treatment with these two lookups, Salesforce Billing makes a billing schedule and billing transactions based on your cancellation invoice plan. (Salesforce Billing Managed Package)

## Create an Invoice Plan and Billing Treatment Setup

Create an invoice plan and associate it with a billing treatment. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: Salesforce Billing Spring '18 and later.

Salesforce Billing uses an invoice plan to invoice order products under the following conditions.

- The order product's Billing Frequency field has a value of Invoice Plan.
- The order product's billing treatment has a populated Invoice Plan field.

Invoice plans have a one-to-many relationship with billing treatments. In this example, we'll create an invoice plan and associate it to one billing treatment that covers several order products.

1. Go to the Invoice Plan object and click **New**.
2. Give your invoice plan a name, and then pick a value for the Billing Schedule Creation field.
  - Order Product Activation: When Salesforce CPQ activates an order product, Salesforce Billing immediately creates a billing schedule for it.
  - Manual: Admins have to create the billing schedule on their own and associate it with this invoice plan.
3. From your invoice plan, click **New Invoice Plan Line**. Enter your invoice plan line's name, processing order, commencement information, and type.  
Invoice plan lines for flat amounts aren't supported as of Salesforce Billing Spring '18.
4. Enter values for the service period start action and service period end action.  
If Service Period Start and Service Period End are left blank or set to Create without Date, the resulting invoice line won't have start or end dates, which may prevent Salesforce Billing from recognizing revenue.
5. Repeat Steps 3 and 4 for as many invoice plan lines as you need.
6. Create a billing treatment and set its New Order Invoice Plan value to the invoice plan you created in steps 1 through 3.
7. Find or create a billing rule that you want to associate with your invoice plan. Set the rule's billing treatment to the treatment you created in Step 4.
8. Assign your billing rule with each product that you want to invoice based on your invoice plan. You can also assign the billing rule to an unactivated order product.



**Note** If an order product has a billing frequency of Invoice Plan, but the order product's billing treatment doesn't specify an invoice plan, Salesforce Billing still allow activation of the order product. However, the activated order product won't have a next billing date, so it won't be picked up for invoicing.

## Milestone Billing

Use invoice plans to invoice your customers in response to milestones. A milestone is a scheduled event that includes the completion of a major deliverable. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later.

 **Example 50/50 Milestone** Your sales rep sold a system performance management subscription on 03/01/18 for \$20,000. As part of the deal, your rep and the customer agreed to the following milestones.

- Milestone 1: Invoice for 50% immediately
- Milestone 2: Invoice for 50% after two months

Create an active invoice plan with billing schedule creation upon order product activation. Then create two invoice plan lines. Invoice Plan Line 1

- Name: Milestone 1
- Type: Percent
- Percentage: 50
- Commencement Date: Order Product Activation
- Commencement Date Offset Amount: 0
- Commencement Date Offset Units: Months

Invoice Plan Line 2

- Name: Milestone 2
- Type: Remainder
- Commencement Date: Order Product Activation
- Commencement Date Offset Amount: 2
- Commencement Date Offset Units: Months

This invoice plan creates a billing schedule containing two billing transactions. The first transaction has a billing target date of 03/01/18 and an amount of \$10,000. The second billing transaction has a billing target date of 05/01/18 and an amount of \$10,000.

 **Example Milestone Billing with Manual Billing Transactions** Your sales rep sold a service desk management subscription on 04/01/2018 for \$10,000. As part of the deal, your rep and the customer agreed to the following milestones.

- Milestone 1: Invoice for \$4000 immediately
- Milestone 2: Invoice for \$3000 upon UAT completion
- Milestone 3: Invoice for \$3000 upon deployment

Create an active invoice plan with billing schedule creation upon order product activation. Since you

want \$4000 invoiced right away, you need only one line for your invoice plan. You'll create the remaining billing transactions after your customer meets their milestones.

- Invoice Plan Line Name: Milestone 1
- Type: Percent
- Percentage: 40
- Commencement Date: Order Product Activation
- Commencement Date Offset Units: Days
- Commencement Date Offset: 0
- Processing Order: 1

Your customer meets Milestone 2 on 06/17/18. Go to your billing schedule and create the following billing transaction.

- Billing Transaction 1
- Billing Transaction Name: Milestone 2 Met
  - Status: Pending
  - Billing Target Date: 06/17/18
  - Amount: \$3000

Your customer meets Milestone 3 on 10/14/18, so you can create one more billing transaction.

- Billing Transaction 2
- Billing Transaction Name: Milestone 3 Met
  - Status: Pending
  - Billing Target Date: 10/14/18
  - Amount: \$3000

## Weekly Billing with Invoice Plans

Use invoice plans to invoice your customers based on weekly schedules. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later.

 **Example** A sales rep sells a media advertising service that begins on 01/01/2018. The service tracks searches for your company's name and moves it to the top of search results at \$1 per search hit. The service has a two-month subscription billed biweekly. You can use invoice plans to set up a series of biweekly invoices.

- Weeks 1-2: 300 Charges
- Week 3-4: 500 Charges
- Week 5-6: 100 Charge

- Week 7-8: 700 Charges

Since you won't know how many hits your service received until the end of each two-week period, create an invoice plan that generates a billing schedule on order product activation, but do not create any invoice plan lines. Instead, you'll create billing transactions manually.

#### **Billing Transaction 1**

Name: Week 1

Billing Target Date: 01/14/18

Status: Pending

Amount: \$1

Quantity: 300

#### **Billing Transaction 2**

Name: Week 2

Billing Target Date: 01/28/18

Status: Pending

Amount: \$1

Quantity: 500

#### **Billing Transaction 3**

Name: Week 3

Billing Target Date: 02/11/18

Status: Pending

Amount: \$1

Quantity: 100

#### **Billing Transaction 4**

Name: Week 4

Billing Target Date: 02/25/18

Status: Pending

Amount: \$1

Quantity: 700

## Invoice Plans for Custom Billing Frequencies

Use invoice plans to create billing schedules that don't follow standard billing frequencies such as monthly or quarterly. For example, you could make your first transaction ten days after the sale, and your second transaction two months after that. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later.

---

 **Example** Your organization provides server mainframe installation and management. Your sales rep sold these services for \$25,000 to a customer on 01/01/18, with an agreement to invoice the sale as follows.

- 40% up front
- 25% after 25 days
- 15% after 70 days
- The remaining balance after 3 months

Create an active invoice plan that bills on order product activation. The invoice plan contains the following invoice plan lines.

Invoice Plan Lines

Name	Type	Percentage	Commencement Date Offset	Commencement Date Offset Units	Processing Order
Initial Payment	Percent	40	0	Days	1
Second Invoice	Percent	25	25	Days	2
Third Invoice	Percent	15	70	Days	3
Remainder	Remainder	None	3	Months	4

## Override Billing Transactions

You can override the target date and amount on your billing transactions. This feature is useful in milestone billing if a sales rep renegotiated terms and changes an invoicing setup that you already

created. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later.

 **Example** Your sales rep sold an enterprise IT performance management package on 06/02/18 for \$30,000, with an initial billing date of 07/01/18. Your rep and the customer agreed to invoicing \$10,000 every 45 days, so you made an invoice plan with invoice plan lines that reflected the division. When you activated the order product, your invoice plan created a billing schedule with 3 billing transactions. However, on 06/09/18, your sales rep renegotiated to the following terms.

- Milestone 1: 60% on 07/15/18
- Milestone 2: 40% upon UAT completion

Use override fields on your billing transactions to reflect the changes. Field values that the related invoice plan line originally created are represented with italics.

### Billing Transaction 1

Status: *Pending*

Billing Target Date: *07/01/18*

Override Billing Target Date: 07/15/18

Amount: *10000*

Override Amount: 18000

Since you don't know when your customer will hit their Milestone 2 goal, find your second and third billing transactions and set their Override Status field to Cancelled. This ensures you won't accidentally invoice based on the original terms. Your customer meets Milestone 2 on 09/30/18, so you can add the following billing transaction to your billing schedule.

### Billing Transaction 2

Status: Pending

Billing Target Date: 09/30/18

Amount: 12000

## Canceling Order Products with Invoice Plans

If your billing treatment looks up to a new order invoice plan, it also requires a cancellation invoice plan. When you cancel an order product covered by a billing treatment with these two lookups, Salesforce Billing makes a billing schedule and billing transactions based on your cancellation invoice plan.

(Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Winter '19 and later

Your cancellation invoice plan lines don't have to match the setup of your new order invoice plan lines. For example, let's say you had the following values for your new order invoice plan.

New Order Invoice Plan

Type	Percentage	Commencement Date	Commencement Date Offset	Commencement Date Offset Units
Percent	40	Order Product Activation	1	Months
Percent	40	Order Product Activation	2	Months
Remainder	null	Order Product Activation	3	Months

You could use different values for your cancellation order plan, such as the following.

Cancellation Invoice Plan

Name	Type	Percentage	Commencement Date	Commencement Date Offset	Commencement Date Offset Units
60%	Percent	60	Order Product Activation	1	Months
Remainder	Remainder	null	Order Product Activation	6	Weeks

If your cancellation order product had a value of -\$2000 and used this cancellation invoice plan, your billing transactions would have the following values.

Billing Transactions

Name	Quantity	Amount To Be Billed
60%	-1	-\$1200
Remainder	-1	-\$800

**!** **Important** Amending invoice plans with cancel order products is not supported.

## Setting Up Usage Products

Salesforce Billing provides several options for customizing pricing for your usage products. You can define minimum-usage fees, included amounts and overages, and price usage based on volume or tiered price levels. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

#### Usage Floors

Order products can define a minimum quantity of usage that overrides the actual quantity of usage uploaded to a usage summary. This field is useful when you want to ensure that usage order products don't invoice below a predefined amount. (Salesforce Billing Managed Package)

#### Included Usage Quantities

You can include a quantity of usage to provide for free before you begin charging for usage. (Salesforce Billing Managed Package)

## Usage Floors

Order products can define a minimum quantity of usage that overrides the actual quantity of usage uploaded to a usage summary. This field is useful when you want to ensure that usage order products don't invoice below a predefined amount. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

The order product field Usage Floor Quantity defines the minimum usage quantity to charge for, even in the event of little or no usage. For example, if your order product has a usage floor quantity of 10, and usages sum to 7, the actual invoiced quantity will be 10. If you still want your invoice line to contain an effective unit price equal the summary's actual quantity, create a custom field.

Your usage summary's total quantity field reflects the quantity of related usage records. However, Salesforce Billing uses the usage floor quantity to calculate the summary's unbilled subtotal and unit price from the related price schedule.

Usage floors are available only for legacy usage. However, if you're using consumption schedule usage, you can create a consumption schedule that emulates the effects of usage floors. In this case, your first consumption rate should have an upper bound equal to the usage floor and a price equal to the upper bound multiplied by the cost per unit.

-  **Example** Create a consumption schedule for a product that costs \$1.50 per unit and has a usage floor of 50.

**Consumption Schedule**

Type: Range

**Consumption Rate 1**

Lower Bound: 0

Upper Bound: 50

Pricing Method: Flat Fee

Price: \$75

**Consumption Rate 2**

Lower Bound: 51

Upper Bound: Null

Pricing Method: Per Unit

Price: \$1.50

## Included Usage Quantities

You can include a quantity of usage to provide for free before you begin charging for usage. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

The order product's Included Usage field defines a quantity that you provide for free before you begin charging for usage. If your order product has an included quantity of 10 and usages sum to 12, the actual invoiced quantity will only be 2.

Your usage summary's Total Quantity field reflects the quantity of related usage records. However, Salesforce Billing uses the value of Included Usage to calculate the summary's unbilled subtotal and unit price from the related price schedule.

## Guidelines for Using Salesforce Billing with Multicurrency Enabled

Review important guidelines for working with Salesforce Billing with multicurrency enabled. (Salesforce Billing Managed Package)

- The following objects records in currencies other than their own.
  - Billing Treatments
  - Revenue Recognition Treatments
  - Tax Treatments
  - GL Treatments
  - Tax Rates
  - Legal Entities
- Salesforce Billing performs revenue recognition reporting using the evaluated transaction's currency.
- Finance books in one currency can have finance periods in different currencies. This setup is useful for recording transactions from several currencies in the same finance book. You can use this setup with revenue recognition rules, treatments, and finance books in the corporate currency. These features, however, must have finance periods in all currencies used for the finance book.
- General ledger (GL) treatments must be in each currency used. The GL rule can be in the corporate currency.

## Billing Account Fields

Salesforce Billing provides custom managed fields for the account object. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

Field	API Name	Data Type	Definition
Bill To Contact	<code>bIng__BillToContact__c</code>	Lookup(Contact)	Primary contact for the billed account.
Customer Profile ID	<code>bIng__CustomerProfileId__c</code>	Text(50)	<p>Authorize.Net payment gateways require this value to communicate with Salesforce Billing.</p> <p>For accounts with Authorize.Net payment gateways, Salesforce Billing creates a customer profile ID when a user selects <b>Save for future use</b> in the Payment Center. It also create a profile ID when a user creates a</p>

Field	API Name	Data Type	Definition
			credit card payment method on the account. Salesforce Billing then passes the ID to Authorize.Net when it communicates with the payment gateway.
Default Payment Type	<b>bIng__DefaultPaymentType__c</b>	Text(50)	<p>Used in payment runs. A payment run picks up an invoice if the run and the invoice have matching Default Payment Type values. Salesforce Billing provides two ways to set Default Payment Type on the invoice:</p> <p><b>Relate a payment method to the account and select the payment method's Autopay field</b></p> <p>The account's default payment type inherits the value of the payment method's Type field. All the account's invoices inherit the account's default payment type. This setup is useful for quickly setting a default payment type for all of the account's invoices.</p> <p><b>Enter a payment method in the invoice's Override Autopay Payment Method field.</b></p> <p>The invoice's default</p>

Field	API Name	Data Type	Definition
			payment type ignores the account's payment method and inherits the Type value from the Override Autopay Payment Method instead. Use this setup when you need to temporarily change the payment method used for a specific invoice.

## Automation in Salesforce Billing

Automating Salesforce Billing processes, such as activating orders and posting invoices, helps streamline cash collection for customers. It also lowers the potential for errors from users making a mistake during manual process configuration. When you work with automated processes, we recommend reviewing when your managed package automation starts and finishes, so that it doesn't interfere with custom implemented record updates. (Salesforce Billing Managed Package)

### **Roll-Up Summary Fields and Salesforce Billing Automation**

When you update a roll-up summary field, Salesforce triggers automation that runs off update triggers on the field's object. Because Salesforce Billing uses many automated processes, we recommend thoroughly testing custom roll-up summaries and custom automation. That way, you reduce the risk of triggering a large number of automated processes, which can impact performance. (Salesforce Billing Managed Package)

### **Automation Guidelines for Orders and Order Products**

Review use cases and guidelines for building automation on orders and order products. (Salesforce Billing Managed Package)

### **Automation Guidelines for Invoices and Invoice Lines**

Review use cases and guidelines for building automation on invoices and invoice lines. (Salesforce Billing Managed Package)

### **Trigger Context for Salesforce Billing Actions**

Certain Salesforce Billing actions cause Salesforce to fire automation trigger contexts on related objects. The trigger context determines whether Salesforce launches automated processes on related objects. When you create or edit custom automation in Salesforce Billing, we recommend reviewing these cascading update paths to ensure that you understand the results of changes made in custom automation. (Salesforce Billing Managed Package)

### **Further Billing Automation Guidelines**

When you work with automation in Salesforce Billing, consider these guidelines. (Salesforce Billing Managed Package)

## Roll-Up Summary Fields and Salesforce Billing Automation

When you update a roll-up summary field, Salesforce triggers automation that runs off update triggers on the field's object. Because Salesforce Billing uses many automated processes, we recommend thoroughly testing custom roll-up summaries and custom automation. That way, you reduce the risk of triggering a large number of automated processes, which can impact performance. (Salesforce Billing Managed Package)

 **Example** Your accounts have a custom roll-up summary field called Invoice Balances (1). It sums the Balance fields of all invoices on the account (2). For example, if your account has one invoice with a balance of \$70 and one invoice with a balance of \$100, your account would have an Invoice Balances value of \$170.  When a customer makes a payment against an invoice, Salesforce Billing updates the invoice's balance. The balance update triggers an update on the account's Invoice Balances field. For example, if you make a \$30 payment against your second invoice, the update causes the accounts Invoice Balances field to recalculate to \$140. Whenever a rollup summary field updates, Salesforce triggers all automation that runs off updates on the same object. These types of automation include workflow rules, process builders, flows, validation rules, approval processes, and filtered lookups.  If you use an automated process to pay many invoices at the same time, such as a payment run, your account-level automation triggers would add up quickly. Let's say your account uses three custom update-based workflow rules and three custom update-based validation rules. If a payment run evaluates and pays 50 invoices, the resulting Invoice Balance updates trigger 300 instances of your custom account automation. Depending on the complexity of your automation, you can encounter performance impacts as Salesforce Billing resolves each automated process. To reduce the likelihood of performance impacts, we recommend testing custom rollup summaries and custom automation in a sandbox. Note any significant impact to CPU, row lock errors, and instances of trigger recursion.

### See Also

[Summary Field](#)

[Knowledge Article: Order of execution in Row lock seen in Master-Detail relationship](#)

[Knowledge Article: Unable to lock row - Record currently unavailable errors](#)

[Knowledge Article: Proactive Alert Monitoring: Rowlock Timeout Errors](#)

## Automation Guidelines for Orders and Order Products

Review use cases and guidelines for building automation on orders and order products. (Salesforce Billing Managed Package)

## Asynchronous Order and Order Product Jobs

Salesforce Billing runs asynchronous jobs in response to several order and order product actions. Review these asynchronous jobs to ensure they don't interfere with any actions in your custom automation.

Job Name	Job Type	Job Occurs	Considerations
OrderPriceCalculatorService	Queueable	Checking Ordered on the opportunity or quote.	This process has completed when the order's Price Calculation Status field has a value of Completed, Not Needed, or Failed.
OrderPriceCalculatorService.populatePriceTiersFuture	Future	Checking Ordered on the opportunity or quote.	This process runs only for products with a charge type of Usage.
QueueableContractOrdersClient	Queueable	Checking Contracted on the order.	none
QueueableRevenueSchedule	Queueable	Order activation.	This process runs only for order products related to revenue recognition treatments with a Revenue Schedule Creation Action field set to Order Activation.
QueueableRevenueTransaction	Queueable	Order activation.	This process runs only for order products related to revenue recognition treatments with a Revenue Schedule Creation Action field set to Order Activation.
QueueableTaxCalculator	Queueable	Order activation and updating tax override fields.	none

## Order Use Cases

Let's review some common use cases for building automation on orders.

Use Case	Description	Considerations
Order creation	Create an order from the quote or opportunity when a user or process selects the Ordered	Order creation is a synchronous process that occurs in Salesforce CPQ. However, Salesforce Billing

Use Case	Description	Considerations
	field.	uses the <code>OrderPriceCalculatorService</code> and <code>OrderPriceCalculatorService.populatePriceTiersFuture</code> triggers to populate Billing fields on the order.
Order activation	<p>Activate an order based on certain criteria when a user or process changes the order's activation status to Activated.</p> <p> <b>Note</b> Activating an order activates all its child order products.</p>	<p>We recommend ensuring that CPQ and Billing triggers related to the order object have finished before your automation activates the order. This way, you can ensure that price and tax calculation callouts have finished for pricing on all of the order's order products.</p> <ul style="list-style-type: none"> <li>• Price calculation triggers have finished when the order product's price calculation status is Completed or Not Needed.</li> <li>• Tax calculation triggers have finished when the order's tax calculation status is Completed.</li> </ul>
Create a contract from the order	<p>Create a contract from the order by selecting the order's Contracted field.</p> <p> <b>Note</b> Contracting an order contracts all its child order products.</p>	<p>This process is synchronous by default. However, if you have many order products to contract, you can use Salesforce CPQ's asynchronous contracting process instead. To enable asynchronous contracting, deselect the Contract in Foreground field in Salesforce CPQ's Subscriptions and Renewals package setting.</p> <p>Depending on your org configurations, the</p>

Use Case	Description	Considerations
		asynchronous contracting process can take less time to complete than the synchronous contracting process.
Provisioning and fulfillment integrations	Orders and order products are common integration points for customers to turn on services for a customer. In many cases, the fulfillment controls when the invoicing process can begin.	Evaluate your integration patterns and consider when the integrations must run. Most integrations have process considerations for order activation and contract generation. Review your end-to-end process requirements when designing integration points.

## Order Product Use Cases

You can also set automation to occur on order products. Follow the above standards for order product creation, activation, and contracting. You also have several options specific to order products.

Use Case	Description	Considerations
Update CPQ or Billing fields on the order product	Update any order product-managed package field. For example, moving the invoice frequency from monthly to quarterly.	When you use automation to update managed package fields on the order product, make sure that your automation doesn't affect any order-level processes. Evaluate and adjust the timing and criteria for triggering automation events if necessary. For example, you can update billing frequency during order creation. To avoid timeouts based on the number of order products created, space out when you trigger billing frequencies.
Override date fields	Update the Override Next Billing Date field based on your business's invoicing needs.	Due to the complexity of managing order product dates for billing, we recommend using Apex to build date logic.

Use Case	Description	Considerations
Legal entity mapping	Map legal entities to order products based on factors such as contacts and addresses.	<p>By default, legal entity mapping occurs upon record creation in the Process Builder, or in a Before Insert trigger for Salesforce Billing Summer '20 and later. We recommend using a Before Insert trigger if you're working with many order products.</p> <p>You can also trigger a Flow upon record creation and run it before saving the record.</p>

 **Note** We recommend consolidating automation on an object into one trigger, process builder, or flow automation.

## Automation Guidelines for Invoices and Invoice Lines

Review use cases and guidelines for building automation on invoices and invoice lines. (Salesforce Billing Managed Package)

### Asynchronous Invoice and Invoice Line Jobs

Salesforce Billing runs asynchronous jobs in response to several invoice and invoice line actions. Review these asynchronous jobs to ensure they don't interfere with any actions in your custom automation.

Job Name	Job Type	Job Occurs	Considerations
<code>QueueableRevenueSchedule</code>	Queueable	Invoice posting	Salesforce Billing fires this job even if you're not using invoice-based revenue schedule generation.
<code>QueueableRevenueTransaction</code>	Queueable	Invoice posting	Salesforce Billing fires this job only if you use invoice-based revenue schedule generation.
<code>QueueableTaxCalculator</code>	Queueable	Invoice batch execution and invoice posting	During invoice runs, Salesforce Billing fires

Job Name	Job Type	Job Occurs	Considerations
			this job first when the invoice is created and again when the invoice is posted.

## Invoice Use Cases

Review some common use cases for building automation on invoices.

Use Case	Description	Considerations
Invoice Posting	<p>Automatically post an invoice one key fields have been finalized and the invoice has been sent to the customer.</p> <p> <b>Note</b> Posting an invoice posts all its invoice lines.</p>	<p>You have three ways to post an invoice.</p> <ul style="list-style-type: none"> <li>• Change the invoice status to Posted.</li> <li>• Automate updating the invoice status based on certain criteria.</li> <li>• Configure your invoice scheduler to post invoices upon invoice creation.</li> <li>• Use the invoice run's Post All Invoices button.</li> </ul> <p>For custom invoice automation, Salesforce Billing requires the automation to trigger after the invoice's tax status is Complete and while the invoice is still in draft status. Due to the asynchronous nature of tax calculation, the invoice may show tax calculation as completed before the invoice line tax shows as completed.</p>
Invoice Dates	Update the invoice date or due date based on custom values that aren't available in Salesforce Billing by default, such as payment term conditions.	All changes to date fields on the invoice must occur before the invoice is posted.

Use Case	Description	Considerations
Managed Package Field Updates	General updates to Salesforce Billing fields, such as overriding the tax address.	All changes to Salesforce Billing fields on the invoice must occur before the invoice is posted.
Tax Integration	Calculate the billable tax to charge on an invoice.	Salesforce Billing applies tax based on your custom tax integration. Review your tax integration vendor's guidelines for more information.

## Invoice Line Use Cases

Invoice line automation is useful for updating Salesforce Billing fields and sending information to external ERP systems after posting an invoice line. When you automate populating Salesforce Billing fields on the invoice line, remember that invoice line data can change based on invoice processes and automation when you post the parent invoice. Changes to the invoice line can also affect its parent order product.

## Trigger Context for Salesforce Billing Actions

Certain Salesforce Billing actions cause Salesforce to fire automation trigger contexts on related objects. The trigger context determines whether Salesforce launches automated processes on related objects. When you create or edit custom automation in Salesforce Billing, we recommend reviewing these cascading update paths to ensure that you understand the results of changes made in custom automation. (Salesforce Billing Managed Package)

Each table lists a Salesforce Billing action in the first row, and Salesforce Billing processes related to that action in the rows below. The Trigger Context Fired column shows the Salesforce Billing update triggers that fire for each object related to the action or process.

Actions	Trigger Context Fired	Notes
Create an invoice from an invoice scheduler or the Bill Now field	<b>Order Item</b> Before Update  After Update  <b>Invoice</b> Before Insert  After Insert	None

Actions	Trigger Context Fired	Notes
	<p>Before Update</p> <p>After Update</p> <p><b>Invoice Line</b></p> <p>Before Insert</p> <p>After Insert</p> <p>Before Update</p> <p>After Update</p> <p><b>Account</b></p> <p>Before Update</p> <p>After Update</p>	
<code>QueueableRevenueSchedule</code>	<p><b>Invoice Line</b></p> <p>Before Update</p> <p>After Update</p> <p><b>Revenue Schedule</b></p> <p>Before Insert</p> <p>After Insert</p>	Triggers fire only for Salesforce Billing orgs using invoice-based revenue recognition.
<code>QueueableRevenueTransaction</code>	<p><b>Revenue Schedule</b></p> <p>Before Update</p> <p>After Update</p> <p><b>Revenue Transaction</b></p> <p>Before Insert</p> <p>After Insert</p>	Triggers fire only for Salesforce Billing orgs using invoice-based revenue recognition.

Actions	Trigger Context Fired	Notes
Select Ordered on the quote	<b>Quote</b> Before Update  After Update  <b>Order</b> Before Insert  After Insert  <b>Order Item</b> Before Insert  After Insert  Before Update  After Update	None
<code>OrderPriceCalculationQueueable</code>	<b>Price Schedule</b> Before Insert  After Insert  <b>Price Tier</b> Before Insert  After Insert  <b>Order Item</b> Before Update  After Update	<code>OrderPriceCalculatorService</code> adds a job to the queue with the listed price schedule, price tier, and order item actions.
<code>OrderPriceCalculationFuture</code>	<b>Price Tier</b> Before Update  After Update	None

Actions	Trigger Context Fired	Notes
Select Ordered on the quote	<b>Quote</b> Before Update  After Update  <b>Order</b> Before Insert  After Insert  <b>Order Item</b> Before Insert  After Insert  Before Update  After Update	None
<code>OrderPriceCalculationQueueable</code>	<b>Price Schedule</b> Before Insert  After Insert  <b>Price Tier</b> Before Insert  After Insert  <b>Order Item</b> Before Update  After Update	<code>OrderPriceCalculatorService</code> adds a job to the queue with the listed price schedule, price tier, and order item actions.

Actions	Trigger Context Fired	Notes
Order Activation	<b>Order Item</b> Before Update  After Update	None

Actions	Trigger Context Fired	Notes
	<b>Order</b> Before Update  After Update	
<code>QueueableRevenueSchedule</code>	<b>Revenue Schedule</b> Before Insert  After Insert	Record updates occur only when using order-based revenue recognition.
<code>QueueableRevenueTransaction</code>	<b>Revenue Schedule</b> Before Update  After Update	Record updates occur only when using order-based revenue recognition.

Actions	Trigger Context Fired	Notes
Select Contracted on the order  <code>QueueableContractOrdersClient</code>	<b>Contract</b> Before Insert  After Insert  Before Update  After Update  <b>Subscription</b> Before Insert  After Insert  Before Update  After Update  <b>Asset</b> Before Insert  After Insert	Asset updates occur only for products that generate assets during the contracting process.

Actions	Trigger Context Fired	Notes
	Before Update After Update <b>Order</b> Before Update After Update <b>Order Item</b> Before Update After Update	

Actions	Trigger Context Fired	Notes
Create an invoice from an invoice scheduler or the Bill Now field	<b>Order Item</b> Before Update After Update  <b>Invoice</b> Before Insert  After Insert  Before Update  After Update  <b>Invoice Line</b> Before Insert  After Insert  Before Update  After Update	None

Actions	Trigger Context Fired	Notes
	<b>Account</b> Before Update  After Update	
<code>QueueableRevenueSchedule</code>	<b>Invoice Line</b> Before Update  After Update  <b>Revenue Schedule</b> Before Insert  After Insert	Triggers fire only for Salesforce Billing orgs using invoice-based revenue recognition.
<code>QueueableRevenueTransaction</code>	<b>Revenue Schedule</b> Before Update  After Update  <b>Revenue Transaction</b> Before Insert  After Insert	Triggers fire only for Salesforce Billing orgs using invoice-based revenue recognition.

Actions	Trigger Context Fired	Notes
Post invoices outside of a batch process	<b>Invoice</b> Before Update  After Update  <b>Invoice Line</b> Before Update  After Update  <b>Order Item</b> Before Update	None

Actions	Trigger Context Fired	Notes
	After Update	
<b>QueueableRevenueSchedule</b>	<b>Invoice Line</b> Before Update After Update  <b>Revenue Schedule</b> Before Insert After Insert	Revenue schedule triggers fire only for Salesforce Billing orgs using invoice-based revenue recognition.
<b>QueueableRevenueTransaction</b>	<b>Revenue Schedule</b> Before Update After Update  <b>Revenue Transaction</b> Before Insert After Insert	Triggers fire only for Salesforce Billing orgs using invoice-based revenue recognition.

## Further Billing Automation Guidelines

When you work with automation in Salesforce Billing, consider these guidelines. (Salesforce Billing Managed Package)

When you build automation or custom fields on the account object, consider how these customizations behave downstream in the CPQ and Billing workflow. For example, let's say you create rollup fields on your account for posted invoices with open balances. Whenever you post an invoice, evaluate whether your posting action causes any of your custom account automations to fire. In some cases, you may also have invoice automation triggered by account updates.

When you automate invoice processes such as posting, evaluate how invoice runs affect your automation. For example, if you use invoice runs to automatically post your invoices, ensure competing rules don't change the invoice status at the same time.

When you post invoices, Salesforce Billing updates invoice-related fields on order products related to the invoice's invoice lines. It also updates invoice-related fields on orders.

When you combine multiple instances of custom automation together, you may encounter the

`System.LimitException: Too many queueable jobs added to the queue: 2.` error. To avoid this error, we recommend breaking up synchronous processing by using either platform events or future callouts in Apex.

## Managing the Billing Order

---

The order record defines important fields about when and how Salesforce Billing invoices your order products. While many of these fields are set by default based on package settings, rules, and treatments, you can also edit them based on your business needs. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### **Order Setup**

Salesforce CPQ lets you configure and customize important billing fields on the order, allowing for a wide range of control over the invoicing process. (Salesforce Billing Managed Package)

#### **Overriding Billing Dates**

Salesforce Billing allows you to temporarily change important billing dates on your order products. These features help you control the timing and duration of invoice generation for your customers. (Salesforce Billing Managed Package)

#### **Order Creation and Activation**

Salesforce CPQ runs several validations during the order creation and also following order activation. You have different options for adjustments based on your order's activated status. (Salesforce Billing Managed Package)

#### **Billing Status Fields**

Review key status fields to track the position of order products and usage summaries in their billing lifecycles. They're useful if you're unsure whether an order product or usage summary has been invoiced or is available for invoicing. (Salesforce Billing Managed Package)

#### **Proration with Invoices**

Salesforce Billing uses proration to calculate balances for invoice lines that cover partial billing periods. The invoice line field Calculated Quantity performs a similar function to the quote line and order product's Prorate Multiplier fields. (Salesforce Billing Managed Package)

#### **Billing and Invoice Cancellation**

Salesforce Billing provides several ways to manage canceled invoices or order products that you cancel during the billing process. (Salesforce Billing Managed Package)

#### **Aligning Proration Between CPQ and Billing**

When you use CPQ and Billing together, we recommend aligning CPQ's Subscription Proration Precision with Billing's Proration Type. Otherwise, the proration methods can cause unwanted differences between an invoice line's balance and the customer's expected billings based on the product's price. (Salesforce Billing Managed Package)

#### **Align CPQ and Billing Cancellation Based on Billing Periods**

Salesforce CPQ and Salesforce Billing use different formulas to calculate proration periods. Depending on your order product's billing fields, Salesforce Billing sometimes splits your order product's overall term into more or fewer proration periods than were used in CPQ. When you cancel an order product when CPQ and Billing use different proration periods, a pending balance sometimes remains. If you want to avoid pending balances when you cancel an order product, align CPQ and Billing to use the same proration periods. (Salesforce Billing Managed Package)

### Troubleshooting Proration Issues

When you're working with proration in CPQ and Billing, review some important guidelines to ensure that your proration values align and deliver expected results. (Salesforce Billing Managed Package)

### CPQ Billing Field Mapping

Salesforce CPQ includes several important billing fields that map between products, quotes, orders, quote lines, and order products. Review these fields to understand data flow and the implications of overriding fields. (Salesforce Billing Managed Package)

### Billing Order Fields

The Order object contains fields for both the CPQ and the Billing packages. API names for Billing package fields use the `bIIng__` prefix. With certain page layout and field-level security settings, some fields aren't visible or editable. (Salesforce Billing Managed Package)

## Order Setup

Salesforce CPQ lets you configure and customize important billing fields on the order, allowing for a wide range of control over the invoicing process. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

**!** **Important** Standalone order products and order products made in Salesforce CPQ can't be on the same order.

### Managing the Billing Day of Month

Salesforce Billing uses the order's billing day of month to calculate the next billing date for each of the order's recurring order products. (Salesforce Billing Managed Package)

### Legal Entities

Legal entities represent a way a structure is organized. For example, a company could create two legal entities to represent their American and APAC branches. Each legal entity record relates to a billing, tax, revenue recognition, and general ledger treatment for an order product or order product consumption schedule. (Salesforce Billing Managed Package)

### Billing Accounts

Billing accounts represent the organization that you're charging for products and services sold. Sometimes, businesses may use one account for quoting and ordering, and a different account to receive and pay invoices. The order product field Billing Account lets you override the default account inherited from the quote. (Salesforce Billing Managed Package)

## Recurring Billing for Subscription Products

Salesforce Billing invoices subscription products repeatedly over time. You can control when and how frequently they're invoiced. (Salesforce Billing Managed Package)

## Grouping Order Products into Invoices

Salesforce Billing creates invoices for different groups of order products based on several order and order product fields. This process is useful for invoicing certain types of order products separately from your other order products. It's also useful if your account managers want one invoice record for all their orders. (Salesforce Billing Managed Package)

## Billing Frequency

A product's billing frequency determines how often Salesforce Billing bills an order product. When you order a quote, your order products inherit the billing frequency of their parent products. You can also leave a product's billing frequency blank and set the order product's billing frequency manually or through automation. (Salesforce Billing Managed Package)

## Advance and Arrears Billing

When configuring your products for billing, consider when you want to invoice the customer for the product and the specific date range and time period for that invoice. The product's Billing Type field lets you define when Salesforce Billing bills a product or service relative to when you provide it to the customer. Advance billing invoices a product or service before you provide it, while arrears billing invoices a product or service after you provide it. Salesforce Billing evaluates billing type when calculating an order product's next billing date. (Salesforce Billing Managed Package)

## Payment Terms

A payment term represents the period of time customers have to pay for an invoiced order product. You can select a payment term on a quote, which flow to the resulting order and all its order products. You can also select an order's payment term on your own before invoicing the order. When you invoice the order, Salesforce Billing uses the order's payment term to calculate the invoice's due date. (Salesforce Billing Managed Package)

## Billing Dates

Salesforce Billing uses dates to control order product eligibility to invoice, date range to invoice and aligning amendment orders to create single invoices. Each date field fulfills a distinct role throughout the ordering and invoicing process. Many of these date fields also impact the values held in other fields on one or more objects. Understanding these dates and the automation behind them lets you control when an invoice is created, which lines it will generate, and what period of these lines should be billed. (Salesforce Billing Managed Package)

## Proration in CPQ Order Products

Salesforce Billing uses a prorate multiplier and other billing fields to calculate the order product's billable unit price. Order products inherit the prorate multiplier of their parent quote lines. The order product's prorate multiplier stays unchanged throughout a standard order. (Salesforce Billing Managed Package)

## Set Up Standalone Orders

Manage your orders and order products in Salesforce and Salesforce Billing without first creating a CPQ quote. We call this type of order a standalone order. They're useful if you manage your quotes and opportunities in an external platform but plan to use Salesforce Billing for order management and billing. (Salesforce Billing Managed Package)

## Managing the Billing Day of Month

Salesforce Billing uses the order's billing day of month to calculate the next billing date for each of the order's recurring order products. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

-  **Note** If the billing day of month is set to a value higher than the number of days in the starting month of the order, and the order products are billed in advance, use Bill Now, or set Override Next Billing Date to the start date of the order product to create the first invoice.

### Using Billing Day of Month in Next Billing Date Calculation

Recurring order products inherit the start date of their parent order by default. Salesforce CPQ then evaluates whether the order product is billed in advance or arrears.

- If the order product is billed in advance, Salesforce CPQ evaluates the order's billing day of month to choose the nearest date on or before the order product's start date. For example, if the order product's start date is January 1, and the order's billing day of month is 15, the next billing date is December 15.
- If the order product is billed in arrears, Salesforce CPQ evaluates the order's billing day of month to choose the nearest date after the order product's start date. It then evaluates the order product's billing frequency. For example, if the order product's start date is January 1, the billing day of month is 1, and the billing frequency is quarterly, the next billing date is April 1.

Sales reps can also edit an order product record to change its start date.

 **Example**

Order Product	Quote Start Date	Quote Line Start Date	Charge Type	Billing Type	Billing Frequency	Billing Day of Month	Order Start Date	Next Billing Date
A	03/15	null	Recurring	Advance	Monthly	10	04/05	03/10
B	03/15	null	Recurring	Advance	Quarterly	10	03/15	03/10
C	03/15	null	Recurring	Arrears	Monthly	10	04/05	04/10
D	03/15	null	Recurring	Arrears	Monthly	31	04/05	04/30
E	01/15	null	Recurring	Arrears	Quarterly	10	01/15	02/10

### Changing the Billing Day of Month

Salesforce Billing uses the order's billing day of month to calculate the billing period and next billing date for each of its order products. The billing day of month also affects usage summary periods for

usage products. When you change the billing day of month on an order product, consider how it can affect your billing timelines. (Salesforce Billing Managed Package)

## Changing the Billing Day of Month

Salesforce Billing uses the order's billing day of month to calculate the billing period and next billing date for each of its order products. The billing day of month also affects usage summary periods for usage products. When you change the billing day of month on an order product, consider how it can affect your billing timelines. (Salesforce Billing Managed Package)

**Important** If you change the billing day of month of an activated order, the order product's next billing date is removed. So, before you change the billing day of month, change the activated order's status to Draft, delete the billing schedule of the order product, and then reactivate the order. After the order is reactivated, a billing schedule is created and the next billing date of the order product is recalculated. You can change the billing day of month only for the order products that have a Bill Through Date Override value.

### Changing the Billing Day of Month for Standard and Recurring Products

A one-time order product's next billing date inherits the order product's start date. Recurring order products use the order's billing day of month differently if the order product is billed in advance or billed in arrears. (Salesforce Billing Managed Package)

### Changing the Billing Day of the Month for Usage Products

When you change the billing day of the month for a usage order product, the order product's first usage summary covers a partial billing period. This way, Salesforce Billing can align the second usage summary to the new billing day of the month. If you change the order or order product's start date so that its day value is different from the order product's billing day of the month, the first usage summary covers a partial billing period for the same reason. To avoid partial period usage summaries, set your billing day of the month to the same day value as your order product's start date. (Salesforce Billing Managed Package)

## Changing the Billing Day of Month for Standard and Recurring Products

A one-time order product's next billing date inherits the order product's start date. Recurring order products use the order's billing day of month differently if the order product is billed in advance or billed in arrears. (Salesforce Billing Managed Package)

For order products billed in advance, Salesforce Billing evaluates the order's billing day of month and chooses the nearest date with that number on or before the order product's start date. For example, if the order product's start date is January 1, and the order's billing day of month is 15, your order product's next billing date is December 15.

For order products billed in arrears, Salesforce Billing evaluates the order's billing day of month and chooses the nearest date with that number after the order product's start date. It then evaluates the product's billing frequency. For example, if the order product's start date is January 1, the billing day of month is 14, and the billing frequency is quarterly, the next billing date becomes April 14.

 **Example** Your order has a billing day of month set to 20. It contains the following order products.

Name	Charge Type	Billing Type	Billing Frequency	Order Product Start Date	Next Billing Date
Cloud threat detection	Recurring	Advance	Monthly	04/01/18	03/20/18
Security suite	Recurring	Arrears	Monthly	04/01/18	04/20/18
Server tower	One-time	None	None	04/01/18	04/01/18

Your invoice scheduler launches an invoice run with a target date of 03/15/18, which produces an invoice with the same target date. Remember, invoices include order products if the invoice's target date falls on or before the order product's next billing date. Your resulting invoice has invoice lines for cloud threat detection and the server tower. However, let's say you changed your order's billing day of month to 30 before the scheduler launched its invoice run. In this case, your scheduler picks up only the server tower for an invoice line on the resulting invoice.

Name	Charge Type	Billing Type	Billing Frequency	Order Product Start Date	Next Billing Date
Cloud threat detection	Recurring	Advance	Monthly	04/01/18	03/30/18
Security suite	Recurring	Arrears	Monthly	04/01/18	04/30/18
Server tower	One-time	None	None	03/01/18	03/01/18

Customers can have concerns if they expect an invoice that contains lines for the server tower and the cloud threat detection subscription. To avoid these issues, always make sure you're aware of how changing an order's billing day of month affects the next billing date of the related order products.

## Changing the Billing Day of the Month for Usage Products

When you change the billing day of the month for a usage order product, the order product's first usage summary covers a partial billing period. This way, Salesforce Billing can align the second usage summary to the new billing day of the month. If you change the order or order product's start date so that its day value is different from the order product's billing day of the month, the first usage summary covers a partial billing period for the same reason. To avoid partial period usage summaries, set your billing day of the month to the same day value as your order product's start date. (Salesforce Billing Managed Package)

 **Note** These processes apply to both usage order products and order products related to consumption schedules. You can change the billing day of the month and order product start date only when the status of all associated usage summary records is new.

Let's say you have a usage product whose start date is 01/15/21 and the billing day of the month is 15. By default, your usage summaries cover full periods.

Usage Summary	Monthly	Quarterly	Annual
First usage summary	01/15/21-2/14/21	01/15/21-04/14/21	01/15/21-01/14/22

Next, let's look at a few examples of how changing the billing day of month, changing the start date, and changing both fields affects your order product for different billing frequencies.

## Changing the Billing Day of the Month

Changes to the Billing Day of Month field value affects your order in different ways for the various billing frequencies.

- If you change the billing day of month on an Order record successfully, the usage summaries of all the associated order products will be deleted and new usage summaries are generated based on the billing day of month.
- If you change the Billing Day of Month field value to 1. In this case, the first usage summary covers a partial period so that the second usage summary can align with the new billing day of month.

Usage Summary	Monthly	Quarterly	Annual
First usage summary	01/15/21-01/31/21	01/15/21-01/31/21	01/15/21-01/31/21
Second Usage Summary	02/01/21-02/28/21	02/01/21-04/30/21	02/15/21-02/14/21

## Changing the Start Date

Changes to the start date affects your order products in different ways for the various billing frequencies.

- If you change the start date of an order product, all usage summaries for that specific order product are deleted, and new usage summaries are generated based on the new start date.
- If you move the order product start date to 01/25/21, your first usage summary also covers a partial period so that your second usage summary can align with the billing day of month and cover a full period.

Usage Summary	Monthly	Quarterly	Annual
First usage summary	01/25/21-02/14/21	01/25/21-02/14/21	01/25/21-02/14/21
Second Usage Summary	02/15/21-3/14/21	02/15/21-05/14/21	02/15/21-02/14/22

- If you move the order and order product start dates to 01/01/21, your first usage summary also covers a partial period.

Usage Summary	Monthly	Quarterly	Annual
First usage summary	01/01/21-01/14/21	01/01/21-01/14/21	01/01/21-01/14/21

Usage Summary	Monthly	Quarterly	Annual
Second Usage Summary	01/15/21-2/14/21	01/15/21-04/14/21	01/15/21-01/14/22

## Changing the Start Date and Billing Day of the Month

If you change the billing day of month and order product start date, all existing usage summary records are deleted and new usage summaries are generated based on the billing day of month and the start date.

Now let's see how you can align changes to the Start Date and Billing Day of Month fields to produce usage summaries for full periods. In this case, you're changing the billing day of month to 1 and the order product start date to 02/01/21. This configuration causes the first usage summary to cover a full period, so all the subsequent usage summaries cover full periods as well.

Usage Summary	Monthly	Quarterly	Annual
First usage summary	02/01/21-02/28/21	02/01/21-04/30/21	02/01/21-01/31/22

## Legal Entities

Legal entities represent a way a structure is organized. For example, a company could create two legal entities to represent their American and APAC branches. Each legal entity record relates to a billing, tax, revenue recognition, and general ledger treatment for an order product or order product consumption schedule. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Every order product and order product consumption schedule has a Legal Entity field. You can control several treatment setups by creating a legal entity for each setup, then assigning your legal entities to order products as needed.

When an order product is created, it inherits the billing rule, tax rule, and revenue recognition rule from the parent product. Salesforce Billing then looks for a match between the order product's legal entity and the legal entity on one of the billing rule's billing treatments, and one of the tax rule's tax treatments. Design your implementation so that the order product's legal entity matches at most one treatment on the billing rule and on the tax rule. If multiple treatments are required for an order product, create matching rules such that the rule can be updated on the order product to achieve the result you want. Updating the rule or legal entity triggers the Billing package to update the treatment. Updating only the treatment can result in unanticipated outcomes.

Salesforce Billing uses the same process to assign a revenue recognition treatment to a revenue

schedule. However, the lookup to the revenue recognition treatment appears only on the revenue schedule.

 **Tip** By default, order products and order product consumption schedules don't have a value for their legal entity. We recommend using a process builder, Flow, or manual assignment to assign legal entities to your order products and your order product consumption schedules. A useful process builder for this situation would run on Create, while a Flow would run upon record creation and before the record is saved. You could also use after insert triggers.

Legal entities and rules can't be updated on activated order products or for order product consumption schedules that look up to an activated order product.

Salesforce Billing accepts null values for the Legal Entity lookup field. However, this configuration can cause data validation issues, especially in orgs with a combination of null and populated Legal Entity lookup fields across different objects. We recommend always using legal entities in your Legal Entity lookup field.

When you invoice your order, the invoice line inherits the legal entity from its originating order product. However, you can also set the legal entity manually on invoice lines that haven't been posted. A revenue schedule inherits its legal entity from the parent order product or invoice, but you can set legal entity manually on the revenue schedule too. In both cases, you must manually change the associated treatment, because it doesn't happen automatically on these objects. You can also have different legal entities between an order product and a related Bill to Account record or Ship to Account record.

In multicurrency orgs, legal entities of one currency can evaluate records in other currencies.

 **Note** Order products and order product consumption schedules have independent legal entities. Each legal entity must be set individually.

## Billing Accounts

Billing accounts represent the organization that you're charging for products and services sold. Sometimes, businesses may use one account for quoting and ordering, and a different account to receive and pay invoices. The order product field Billing Account lets you override the default account inherited from the quote. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Order-level billing accounts are useful when you're selling to larger organizations, which often have a billing account separate from the account they use for developing quotes. We recommend associating a process or workflow rule with the quoting account so that the account's orders automatically populate their Billing Account fields with the correct account record.

 **Tip** By default, invoice runs automatically group order products into invoices based on matching billing account fields.

## Recurring Billing for Subscription Products

Salesforce Billing invoices subscription products repeatedly over time. You can control when and how frequently they're invoiced. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing versions

---

Salesforce Billing evaluates several factors when invoicing for a recurring order product, such as how frequently to invoice. For example, billing each month is less than billing quarterly. We use a formula to account for these factors and calculate how much Salesforce Billing bills a recurring order product on any given invoice. The result of this calculation is known as the billable unit price. Salesforce Billing calculates it for recurring products using the following formula. The “unit” in Billable Unit Price represents a unit of time, not a per unit value.

```
Billable Unit Price = [(Total Amount * Billing Frequency) ÷ (Prorate Multiplier * Subscription Term)]
```

 **Note** If the charge type is recurring, the product subscription type isn't evergreen, and the order product is created from a quote line, then the prorate multiplier value is equal to the quote line value.

Here are a few exceptions to the formula:

- If the charge type is one time, then the billable unit price is determined by the total amount of the order product.
- If the charge type is recurring and the subscription type is evergreen, then `Billable Unit Price = Billing Frequency * Total Amount`.
- If the charge type is recurring and the billing frequency is invoice plan, then the billable unit price is null. In this case, the billable unit price is ignored for invoice line subtotal calculation.

 **Note** The billing frequency can be monthly, quarterly, semiannual, or annual.

We now have a value for the amount we expect to bill a recurring order product on any given invoice. When you invoice this order product, its balance by default is equal to the billable unit price. If your order had partial periods or proration, Salesforce Billing then prorates the invoice line's balance accordingly.

Users can also manually set the billable unit price. If it's lower than the original calculated amount, Salesforce Billing accrues the difference over time and puts the remaining amount in the final invoice.

This feature is useful if a sales rep renegotiates terms after the initial sale.

 **Note** We recommend against changing a non-evergreen recurring order product's billable unit price from the value that Salesforce CPQ calculated, as the change causes errors. If you're using an uninvoiced evergreen order product, you can use the order product's [Override Billable Unit Price](#) field to set a new billable unit price.

 **Example** A \$100 product sold in yearly increments is quoted for 10 months, to be invoiced quarterly. Salesforce CPQ is using monthly proration precision, so our proration multiplier is  $(10/12)$ . This gives us the following billable unit price formula.  $[(\$100 * 3) \div ((10/12) * 12)] = \$30$  With a billable unit price of \$30, a typical invoice in this scenario would be billed at \$30. The first three invoices, which cover 9 of the 10 months, would cover \$30 each. The last invoice has a different balance than the billable unit price, since there's only \$10 left to bill.

 **Example** Next, let's look at how subscription prorate precision can affect recurring billings. A product sold in monthly increments is quoted for January 1 through March 5 for \$21.64, to be invoiced monthly. In this scenario, we'll have a small portion to bill for the five-day period in March. Depending on the CPQ package setting Subscription Prorate Precision, you can charge either for the full month, or prorate based off the five extra days. If your proration precision is Month + Day, you'll bill a total amount of \$21.64: \$10 in January, \$10 in February, and \$1.64 in March. However, dividing the total amount of \$21.64 across three invoices evenly comes out to a price per month of \$7.21, notably different than the \$10/\$10/\$1.64 we originally expected. It's important to keep these differences in mind for internal book keeping and revenue recognition reporting. So, to bill for the right amount at the right time, you have to use the known prorate multiplier to calculate the price per month. In this case, the prorate multiplier for Month + Day precision is 2.164. This leads to a billable unit price of \$10.

## Grouping Order Products into Invoices

Salesforce Billing creates invoices for different groups of order products based on several order and order product fields. This process is useful for invoicing certain types of order products separately from your other order products. It's also useful if your account managers want one invoice record for all their orders. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and later

---

When your invoice run evaluates an order, it checks several fields when deciding how to group order products as invoice lines on an invoice.

- By default, the invoice run groups order products by their matching Billing Account fields.
- If the order product billing account is null, the invoice run uses the order's Billing Account field.
- If the order's billing account is null, the invoice run uses the order's Account field.

If the billing account has order products with different payment terms, it then groups the order products by matching payment terms.

You can define more levels of grouping with the order product's Invoice Group field. After your invoice run groups order products by their billing account and payment terms, it then considers the order's invoice grouping.

### **Contract Number**

Order products are grouped by matching contract numbers. Salesforce Billing references the contract number on the order product's originating subscription record.

### **Order**

Salesforce Billing places all of this order's order products on the same invoice.

### **Order PO Number**

Order products are grouped by matching order PO numbers.

### **Legal Entity**

Order products are grouped by matching legal entities.

### **Separate Invoice**

Every order product on this order is placed into a separate invoice.

### **Invoice Group ID**

Users can create a custom Invoice Group ID field. Salesforce Billing groups order products by matching invoice group ID values.

Finally, if any order products in your group have different currencies, each combination of the invoice group field and a currency type receives a separate invoice.

#### **Create Invoice Group IDs**

Salesforce Billing offers a way to customize invoice groups so that customers can bill order products matching certain criteria. This process is useful for separating invoices that don't share the same billing or payment terms. It's also helpful for combining several order products into one invoice. By defining an Invoice Group ID, you get to control how and when you invoice an order product. (Salesforce Billing Managed Package)

#### **Billing Rule Amendment Settings**

Use billing rules to adjust billing dates following amendments to recurring service plans. (Salesforce Billing Managed Package)

#### **Consolidating Order Products to One Invoice Line**

Salesforce Billing can combine the data from an order product and a revised order product into one invoice line. This feature is useful if you invoice a recurring order product, amend it, and must account for the amended order product on future invoices. (Salesforce Billing Managed Package)

## Create Invoice Group IDs

Salesforce Billing offers a way to customize invoice groups so that customers can bill order products matching certain criteria. This process is useful for separating invoices that don't share the same billing or payment terms. It's also helpful for combining several order products into one invoice. By defining an Invoice Group ID, you get to control how and when you invoice an order product. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later

---

There are several ways to group invoices—by order, account, contract, PO number, legal entity, and so on. For example, if you have one order that is associated with two different billing accounts, you could separate the order into two invoice groups: Invoice Group 1 (Billing Account 1) and Invoice Group 2 (Billing Account 2). If you'd like to manually group invoices according to your own criteria, create an Invoice Group ID on the Order Product Detail page.

Let's look at how this process works. Say that you have five order products for a single account and you want to invoice them by tax status to make accounting easier. Two of your order products are tax exempt, two use standard tax, and one uses an external tax engine.



The tax exempt invoices go out in invoice group 1, invoices using standard tax in group 2, and one using an external tax engine in group 3.

To create a customized invoice group, in Salesforce Classic:

- Go to one of your orders and select an associated order product.
- On the Order Product Detail list under Billing Frequency Information, make sure that you have both fields, Invoice Group and Invoice Group ID. If you don't see them, select Edit Layout at the top right of the page (in Salesforce Classic). Under the Fields section, find the field(s) (1) and drag to the Billing Frequency Information list. Save your work.
- Return to the Order Product Detail page. Click to edit the Invoice Group field and select Invoice Group ID as the Invoice Group (2).
- Enter a unique name for the Invoice Group (3) and save your work.



In Lightning Experience:

- Select an order product.

- Under Billing Frequency Information on the Order Product Details page, click to edit the Invoice Group field .
- Select Invoice Group ID (1).
- In the Invoice Group ID field, enter a name for your invoice group (2).
- Save your work.



 **Note** You can create an invoice group based on the order number. On the Invoice Group ID field on the Order Product Detail page (1), enter the number. This action creates one invoice per order.



## Billing Frequency

A product's billing frequency determines how often Salesforce Billing bills an order product. When you order a quote, your order products inherit the billing frequency of their parent products. You can also leave a product's billing frequency blank and set the order product's billing frequency manually or through automation. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The Billing Frequency field has these options.

- Monthly
- Quarterly
- Semiannual
- Annual
- Invoice Plan



**Warning** The billing frequency picklist set supports only the preceding values. Changing them or adding custom values can result in unpredictable outcomes and errors.

The product is the first record to hold a billing frequency. Quote lines inherit billing frequency from their parent product, and order products inherit billing frequency from their parent quote line. However, you can also use process builders or workflow rules to group or change billing frequencies on any record.

The billing frequency on its own doesn't control when Salesforce Billing invoices an order product. However, we use it along with several other fields to calculate each order product's next billing date. When a sales rep orders a quote line, Salesforce Billing calculates the order product's initial next billing date based on its charge type, billing type, and billing day of month fields. Then, when a user, workflow, or invoice run first invoices an order, Salesforce Billing compares each order product's next billing date with the invoice's target date. If the next billing date falls on or before the target date, Salesforce Billing

makes an invoice line for that order product.

After an order product has been invoiced once, Salesforce Billing uses the order product's billing frequency to calculate the upcoming next billing date. After each invoice, Salesforce calculates a new next billing date using the billing frequency until the order product has been invoiced for its final billing period. Let's look at a few examples.

Next Billing Date (Before First Invoice)	Billing Frequency	Next Billing Date (After First Invoice)
March 10	Monthly	April 10
March 10	Quarterly	June 10
March 10	Semiannual	September 10

 **Note** If you change the billing frequency of an activated order product, it can result in unpredictable outcomes and errors. Instead, deactivate the order product, update the billing frequency, and then reactivate the order product. If there are invoice lines for the order product, the invoice must be canceled and rebilled before you deactivate the order product. If you can't cancel and rebill the invoice, amend the contract and replace the order product. When you change the billing frequency from Invoice Plan to a periodic frequency, set all billing transactions to Canceled or delete the billing schedule. Also, if the next billing date is incorrect, correct the override next billing date. When you change the billing frequency from a periodic frequency to Invoice Plan and the next billing date is incorrect, correct the override billing target date of the first billing transaction. This change automatically updates the next billing date.

## Advance and Arrears Billing

When configuring your products for billing, consider when you want to invoice the customer for the product and the specific date range and time period for that invoice. The product's Billing Type field lets you define when Salesforce Billing bills a product or service relative to when you provide it to the customer. Advance billing invoices a product or service before you provide it, while arrears billing invoices a product or service after you provide it. Salesforce Billing evaluates billing type when calculating an order product's next billing date. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Advance Billing

If the order product is billed in advance, Salesforce CPQ evaluates the order's billing day of month to choose the nearest date **on or before** the order product's start date. For example, if a monthly order product's start date is January 1, and the order's billing day of month is 15, the next billing date is December 15.

## Arrears Billing

If the order product is billed in arrears, Salesforce CPQ evaluates the order's billing day of month to choose the nearest date **after** the order product's start date. For example, if a monthly order product's start date is January 1 and the order's billing day of month is 15, the order product's next billing date is January 15.

## Payment Terms

A payment term represents the period of time customers have to pay for an invoiced order product. You can select a payment term on a quote, which flow to the resulting order and all its order products. You can also select an order's payment term on your own before invoicing the order. When you invoice the order, Salesforce Billing uses the order's payment term to calculate the invoice's due date. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

An invoice's due date equals the invoice's Invoice Date field plus the order's payment terms. For example, if your invoice has an invoice date of March 5 and your order has Net 45 payment terms, your invoice's due date is April 19. Salesforce Billing begins the AR aging process for invoices that haven't been paid by their due date.

Salesforce Billing doesn't calculate invoice due dates from custom values added to the Payment Term field.

 **Important** By default, the invoice date is the date a user, workflow rule, process, or invoice created the invoice record.

## Billing Dates

Salesforce Billing uses dates to control order product eligibility to invoice, date range to invoice and aligning amendment orders to create single invoices. Each date field fulfills a distinct role throughout the ordering and invoicing process. Many of these date fields also impact the values held in other fields on one or more objects. Understanding these dates and the automation behind them lets you control when an invoice is created, which lines it will generate, and what period of these lines should be billed. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

## Order and Order Product Dates

If the CPQ package setting Default Order State Date has a value of Today, the Order Start Date defaults to the day the order is created. If it has a value of Quote Start Date, it defaults to the start date of the primary quote that created the order. Sales reps can also override the default value by entering an order start date manually after clicking Create Order.

An order product's start date depends on its charge type.

- **One-Time:** Salesforce CPQ counts the days between the quote's start date and the quote line's start date, then adds them to the order's start date. If the quote line's start date is before the quote's start date, the order product inherits the quote's start date.
- **Recurring and Usage:** Recurring order products inherit the start date of their parent order by default.

Order Product	Quote Start Date	Quote Line Start Date	Charge Type	Order Start Date	Order Product Start Date	Billing Day of Month
A	07/01	null	One-Time	07/10	07/10	10
B	07/01	07/03	One-Time	07/10	07/12	10
C	07/01	06/25	One-Time	07/10	07/10	10
D	07/01	null	Recurring or Usage	07/10	07/10	10
E	07/01	07/03	Recurring or Usage	07/10	07/12	10

### Invoice Dates

Users can create invoices from orders using either the order's Bill Now checkbox, or an invoice scheduler to automate invoice creation. These two methods have slightly different effects on how dates convert from the order and order products to an invoice and invoice lines. (Salesforce Billing Managed Package)

### System Changes to Order Dates

Salesforce Billing updates order billing dates in response to system actions such as invoicing and amending. (Salesforce Billing Managed Package)

### Manual Changes to Order Dates

You can update certain order and order product dates during the bill cycle. However, changes to the order product's start, end, or next charge dates can impact order and order product price calculations. Before manually changing any order or order product dates, make sure that you understand the results of your change. (Salesforce Billing Managed Package)

### Understanding Next Billing Date

Salesforce Billing evaluates two key date fields during invoice line generation: Next Billing Date on the order product, and Invoice Target Date on the invoice. When a user, workflow, or invoice scheduler invoices an order, active order products with a next billing date on or before the invoice target date are

included on the invoice. (Salesforce Billing Managed Package)

### Aligning Billing Dates for Amended Order Products

When you order an amendment quote, the next billing dates and charge dates of the amendment order products can be different from the dates of the original order products. In some cases, these differences can cause the amended order products to invoice separately from the original order products. However, Salesforce Billing provides several date alignment options if you want to send out only one invoice for the original order products and amended order products. (Salesforce Billing Managed Package)

## Invoice Dates

Users can create invoices from orders using either the order's Bill Now checkbox, or an invoice scheduler to automate invoice creation. These two methods have slightly different effects on how dates convert from the order and order products to an invoice and invoice lines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Using Bill Now

#### Invoice

- Invoice Date: Date when Bill Now was selected
- Due Date: Invoice date + order's payment terms. For example, an invoice date of 03/01/18 and Net 45 payment terms would create a due date of 04/15/18.
- Target Date: Inherited from order start date

#### Invoice Lines

- Start Date: Inherited from order product's next charge date
- Charge Date: Date when Bill Now was selected
- End Date: Based on the charge type and order's start and end date. If the order product's terms go beyond the end date of the parent order, then each invoice line's term aligns to the order's end date. Salesforce Billing also adds a prorated invoice line to the end of the term for the remaining period.

### Using Invoice Scheduler

#### Invoice

- Invoice Date: Inherited from the invoice date on the invoice run that created the invoice
- Due Date: Invoice date + order's payment terms
- Target Date: Inherited from the invoice scheduler's target date.

#### Invoice Lines

- Start Date: Inherited from order product's next charge date
- Charge date: The day the scheduler ran and created the invoice
- End Date: Based on the charge type and order's start and end date. If the order product's terms

go beyond the end date of the parent order, then each invoice line's term aligns to the order's end date, and an extra prorated invoice line is added to the end of the term for the remaining period.



**Note** If you want to set a custom due date, add a custom value to your order's Payment Terms field, then set the new value as your term and invoice the order. The resulting draft invoice will have a null due date. You can then use automation to calculate the invoice's due date based on your custom term. For example, if you wanted a Net-10 payment term, add a value of Net-10 to your payment term field. Then, create automation the sets the draft invoice's due date to its invoice date + 10 days for.

## System Changes to Order Dates

Salesforce Billing updates order billing dates in response to system actions such as invoicing and amending. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Invoicing the Order

Invoicing an order doesn't change any of the order or order product's dates. However, posting the invoice does change a few order product dates

- Last Charge to Date: Updates to the most recently posted invoice line's end date.
- Next Charge Date: Updates to the day after the order product's last charge to date.
- Next Billing Date: Updates based on the order's billing day of month, billing frequency, and billing type. If the Override Next Billing Date field has a value, Salesforce Billing uses that date instead. For more information, check out Understanding Next Billing Date.

#### Amending the Order

Amendment orders evaluate CPQ package settings and the order's Billing Day of Month field to determine order product start and end dates. For example, if your CPQ package setting for Order Start Date is Today, then any orders created from your amendment quote use the date of creation as their start date.

When you cancel a quote line on an amendment, the resulting order product inherits its terminated date from the canceled subscription record's terminated date. Salesforce Billing doesn't create invoice lines or invoice charges for an order product from its terminated date onward. If the order product bills in advance, Salesforce Billing creates a credit invoice for the period from the termination date through the day before the next charge date.

You can further customize amendment dates by using the billing rule's Amendment Bill Cycle Date Alignment field. This field lets you configure amended subscription order products to set their next charge date and next billing date to their original order product's next billing date. For

more information check out Aligning Charge Dates for Amended Order Products.

## Manual Changes to Order Dates

You can update certain order and order product dates during the bill cycle. However, changes to the order product's start, end, or next charge dates can impact order and order product price calculations. Before manually changing any order or order product dates, make sure that you understand the results of your change. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Mid-Cycle Updates to Orders and Order Products

Changing Order Field	Updates Order Product Field
Billing Day of Month	Next Billing Date
Start Date	<ul style="list-style-type: none"> <li>• Start Date</li> <li>• End Date</li> <li>• Next Charge Date</li> </ul>

Changing Order Product Field	Updates Order Product Field
Start Date	<ul style="list-style-type: none"> <li>• End Date</li> <li>• Next Charge Date</li> </ul>
End Date	<ul style="list-style-type: none"> <li>• Start Date</li> <li>• Next Charge Date</li> </ul>

### Using Process Automation to Update Billing Dates

You can use process builders or workflow rules at the order product level to change invoicing dates before invoice generation. We recommend using automation to impact the order product's Override Next Billing Date or Bill Through Date Override fields. If you want to update the order product start, end, or next charge date, we recommend doing so before or after price calculation to avoid errors.

- **Override Next Billing Date:** When this field has a value, invoice schedulers use this date rather than the next billing date. After the resulting invoice posts, Salesforce Billing clears this field. This field allows users to create invoices based on custom billing dates. For more information, check out [Override Next Billing Date](#).
- **Bill Through Date Override:** When Bill Now or an invoice scheduler picks up the order product for an invoice, Salesforce Billing creates one invoice line for the entire billing period up to and including the value of Bill Through Date Override. For more information, check out [Bill Through](#)

### Date Override.

Using processes to directly change invoice and invoice line dates isn't best practice – doing so results in an Error status on the related order product's Invoice Run Processing Status field, and Salesforce Billing won't create invoice lines for other lines on the invoice. Instead, we recommend controlling all invoice and invoice line dates at the order and order product level.

### Understanding Next Billing Date

Salesforce Billing evaluates two key date fields during invoice line generation: Next Billing Date on the order product, and Invoice Target Date on the invoice. When a user, workflow, or invoice scheduler invoices an order, active order products with a next billing date on or before the invoice target date are included on the invoice. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Salesforce Billing evaluates several fields to calculate an order product's next billing date. It's important to understand this calculation process so that you can configure your order products to invoice based on organizational needs.

 **Important** Next Billing Date is a system field that users and admins shouldn't change.

#### Order Start Date

If the CPQ package setting Default Order State Date has a value of Today, the Order Start Date defaults to the day the order is created. If it has a value of Quote Start Date, it defaults to the start date of the primary quote that created the order. Sales reps can also override the default value by entering an order start date manually after clicking Create Order.

#### Order Product Start Date

An order product's start date depends on its charge type. For One-Time order products, Salesforce CPQ counts the days between the quote start date and quote line start date, then adds them to the order's start date.

Recurring order products inherit the start date of their parent order by default. Salesforce CPQ then evaluates whether the order product is billed in advance or arrears.

- If the order product is billed in advance, Salesforce CPQ evaluates the order's billing day of month to choose the nearest date on or before the order product's start date. For example, if the order product's start date is January 1, and the order's billing day of month is 15, the next billing date is December 15.
- If the order product is billed in arrears, Salesforce CPQ evaluates the order's billing day of month to choose the nearest date after the order product's start date. It then evaluates the order product's billing frequency. For example, if the order product's start date is January 1, the billing

day of month is 1, and the billing frequency is quarterly, the next billing date is April 1.

Sales reps can also edit an order product record to change its start date.



### Example

Order Product	Quote Start Date	Quote Line Start Date	Charge Type	Billing Type	Billing Frequency	Billing Day of Month	Order Start Date	Next Billing Date
A	03/15		One-Time			10	03/15	03/15
B	03/15	03/21	One-Time			10	03/15	03/21
C	03/15	03/09	One-Time			10	03/15	03/15
D	03/15		Recurring	Advance	Monthly	10	04/05	03/10
E	03/15		Recurring	Advance	Quarterly	10	03/15	03/10
F	03/15		Recurring	Arrears	Monthly	10	04/05	04/10
G	03/15		Recurring	Arrears	Monthly	31	04/05	04/30
H	01/15		Recurring	Arrears	Quarterly	10	01/15	02/10

### Aligning Billing Dates for Amended Order Products

When you order an amendment quote, the next billing dates and charge dates of the amendment order products can be different from the dates of the original order products. In some cases, these differences can cause the amended order products to invoice separately from the original order products. However, Salesforce Billing provides several date alignment options if you want to send out only one invoice for the original order products and amended order products. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and later

---

Amendment orders follow package-level settings for determining the start dates and end dates of their order product, as well as the order's billing day of month. For example, if your CPQ package-level setting for Order Start Date has a value of Today, then all your amendment orders use their creation date as their start date.

Unless your original and amendment orders have the same start dates, the amended order products will have different next billing dates and next charge dates than the original order products. These differences can cause separate invoices for both sets of order products based on your invoice date.

For example, you sell a \$20 monthly subscription that bills on the first of the month. Your customer orders 20 subscriptions on June 1, then decides to add another five subscriptions on June 15. After you make the amendment on June 15, you'll have the original order product, which bills for \$200 on the first of the month, and the amended order product, which bills for \$75 starting June 15. If you use a monthly invoice scheduler that runs on the first of the month, you get an invoice for \$200. The scheduler doesn't pick up your amended order products until July 1, when it creates a \$275 invoice for both the amended order products and the original order products of the next billing period.

If you make the amendment on a different day of the month and use a daily invoice scheduler, you get two invoices per month unless you align the order product billing dates.

The Amendment Bill Cycle Date Alignment billing rule field controls how Salesforce Billing handles the billing dates on amended order products. If this field has a value of null or Do Not Align Amended Order Product, your amendment order products use their default next billing date and next charge date. If this field has a value of Align to original Order Product, the Next Billing Date falls before the Last Charge To Date, and the Billing Day of Month isn't set, the amendment order product's Bill Through Date Override is set to the Last Charge To Date of the revised Order Product.

If you still want to include your original and amendment order products on the same invoice, you can do these steps.

- In the CPQ Order package settings, set the Default Order Start Date setting to Quote Start Date. Then, during amendment quote creation, set the amendment quote's start date to match the original order's billing day of the month. When a sales rep orders the amendment quote, the order inherits the quote's revised start date, and the order's billing day of the month inherits the day value of the order start date. The amendment order now has the same start date and billing day of the month as the original order, so the amendment order products have the same next billing date as the original order products.
- During order creation, set the amendment order's billing day of the month to match the original order's billing day of the month.
- Set the amendment quote's proration day of the month to the same value as the original order's billing day of the month. When a sales rep orders the quote, the order's billing day of the month inherits the quote's proration day of the month.

 **Note** Changing the billing day of the month can result in a next billing date earlier than the order product's start date. For example, an order product bills in advance with a start date of 08/01. If you change the order's billing day of the month to 15, the order product's next billing date becomes 07/15. Salesforce Billing invoices it for a prorated amount (from 08/01 to the end of the billing period) on its first invoice.

## Proration in CPQ Order Products

Salesforce Billing uses a prorate multiplier and other billing fields to calculate the order product's billable unit price. Order products inherit the prorate multiplier of their parent quote lines. The order product's prorate multiplier stays unchanged throughout a standard order. (Salesforce Billing Managed Package)

### Billable Unit Price

Each quote line has a billable unit price, which represents the unprorated amount that bills in each billing period. When a customer orders a quote, the order product inherits the parent quote line's Billable Unit Price value.

If the quote line's charge type has a value of One-Time, the quote line's Billable Unit Price matches its Total Price. If the quote line has a charge type of Recurring, Salesforce CPQ calculates its billable unit price as follows. All fields in this calculation are from the quote line.

```
Billable Unit Price = [(Total Price * Billing Frequency) ÷ (Prorate Multiplier * Default Subscription Term)]
```

When the quote line is ordered, the resulting order product inherits its billable unit price value from the quote line.

 **Important** Salesforce Billing doesn't support calculating billable unit prices for order products based on quote lines created while Salesforce CPQ used daily subscription term units.

We use these numbers for each billing frequency value.

Billing Frequency	Equivalent Value
Monthly	1
Quarterly	3
Semiannual	6
Annual	12

 **Example** Let's start out with a simple example. A subscription product has a \$100 price, subscription term of 12 months, and a quarterly billing frequency. A sales rep quotes it for 10 months in a CPQ org that uses monthly prorate precision (10/12). Our total price is  $(\$100 / 12 * 10) = \$83.33$ , so we use this billable unit price formula.  $[(\$83.33 * 3) \div ((10/12) * 12)] = \$25$  With quarterly billing frequency, Salesforce Billing invoices the order product 4 times over its 10-month term. The first three invoices are each for the full billable unit price of \$25. In the final quarter, Salesforce Billing invoices the remaining \$8.33 to account for the 10th month. The billing day of month also impacts proration, because it helps determine the order product's next billing date.

## Advance Billing

When you bill in advance, the Billing Day of Month field determines the next billing date on or before the order product's start date. For an order product that starts on May 23 with an 01 billing day of month, the next billing date is May 1. If the billing day of month is 12, the next billing date is May 12.

## Arrears Billing

If the order product is billed in arrears, Salesforce CPQ evaluates the order's billing day of month to choose the nearest date after the order product's start date. It then evaluates the order product's billing frequency. Since our order product is billed monthly, its next billing date would be 06/01/19. If it was billed quarterly, its next billing date would be 09/01/19.

# Set Up Standalone Orders

Manage your orders and order products in Salesforce and Salesforce Billing without first creating a CPQ quote. We call this type of order a standalone order. They're useful if you manage your quotes and opportunities in an external platform but plan to use Salesforce Billing for order management and billing. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

Our standalone orders documentation contains required fields and setup instructions unique to standalone orders and order products. For concepts shared between standard orders and standalone orders, we provide links to the original CPQ and Billing topics.

You still need the Salesforce CPQ package installed to create standalone orders and order products. While you won't use any Salesforce CPQ features or objects, Salesforce Billing runs internal validations against the CPQ package for several features and workflows.

**!** **Important** Standalone order products and order products made in Salesforce CPQ can't be on the same order.

### Enable Standalone Orders

Orders and order products created in Salesforce CPQ have several field values inherited from parent products and quote lines. When you create an order without CPQ, you must provide these values on your own or through API. Enable a set of validations that confirm that your standalone orders and order products have the correct values for Salesforce Billing features. (Salesforce Billing Managed Package)

### Creation and Activation Guidelines for Standalone Orders

Standalone orders and order products have several unique rules for activation. (Salesforce Billing

Managed Package)

#### Revenue Recognition for Standalone Orders

Standalone orders and order products require additional configuration for revenue recognition in Salesforce Billing. (Salesforce Billing Managed Package)

#### Limitations for Standalone Orders

Orders and order products created without CPQ have unique guidelines and validations to ensure that Salesforce Billing can manage them. Review these guidelines when you create your orders and order products. (Salesforce Billing Managed Package)

#### Standalone Order Product Pricing

Standalone order products require you to calculate several pricing and pricing-related fields that would otherwise come from a CPQ quote. (Salesforce Billing Managed Package)

#### Required Configurations for Types of Standalone Order Products

Standalone orders require you to fill out all the fields that would otherwise come from a CPQ quote. We Certain fields require a specific value, while some are required but (Salesforce Billing Managed Package)

## Enable Standalone Orders

Orders and order products created in Salesforce CPQ have several field values inherited from parent products and quote lines. When you create an order without CPQ, you must provide these values on your own or through API. Enable a set of validations that confirm that your standalone orders and order products have the correct values for Salesforce Billing features. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

1. From Setup, in the Quick Find box, enter *Installed Packages*, and then select **Installed Packages**.
2. Go to Salesforce Billing, and click **Configure**.
3. Select **Enable Billing Order Validations**.

 **Note** Enable Billing Order Validations is active by default only for new Salesforce Billing Winter '21 customers. It's inactive for customers who upgraded from an earlier version.

4. Provide Read access to these objects:

Consumption Schedules

Consumption Rates

Order Item Consumption Schedules

Order Item Consumption Rates

Salesforce Billing runs the validations whenever a customer or process activates an order or order product. If you have Billing Order Validations enabled, the order product must have a billing rule to be

activated.

-  **Important** Activating standalone orders and order products without Enable Billing Order Validations enabled can cause errors.

## Creation and Activation Guidelines for Standalone Orders

Standalone orders and order products have several unique rules for activation. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

To activate a standalone order or standalone order product, change its Activation Status field from Draft to Activated. Activating a standalone order activates all of its child standalone order products. You can activate standalone orders and order products on your own or through automation such as triggers and flows.

Salesforce Billing doesn't use the Activated checkbox for standalone order products to activate them. We recommend not setting this field to True when loading order products or activating them.

A standalone order product refers to an order product that doesn't have a parent quote line. If you activate a standalone order with child order products that have parent quote lines, Salesforce Billing doesn't activate them. Only the standalone order and its standalone order products will be activated.

When you create a standalone order, its child order products inherit the product's charge type, billing frequency, and billing type. If any order products were related to a product with a product consumption schedule and rates, the order products also receive order product consumption schedules and rates.

 **Important**

- Create your standalone orders and order products in a draft status, and then activate them after creation. Many Salesforce Billing features fire only when an order product's status changes from Draft to Activated. If you create your order product with an Activated status, change it back to Draft, then activate it again.
- In Salesforce Billing Winter '21 only, activating a standalone order doesn't activate its child standalone order products. In this case, you must activate the standalone order products on your own or through automation.
- In Salesforce Billing Winter '21 only, standalone order products don't inherit the order's charge type, billing frequency, and billing type by default. They also don't receive order product consumption schedules and rates by default if they were related to a product with a product consumption schedule and rates.

## Revenue Recognition for Standalone Orders

Standalone orders and order products require additional configuration for revenue recognition in Salesforce Billing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

By default, Salesforce Billing creates revenue schedules for standalone order products only when you change their status from Draft to Activated. Remember that activating a standalone order doesn't activate its order products.

Standalone order products allow you to create an order product in Draft or Activated status. If you plan to create your standalone order products with an Activated status, you have two options for creating revenue schedules. For both cases, we recommend using automation such as a process builder or workflow rule.

- Change the order product's status to Draft, then change it back to Activated.
- Update the order product's revenue schedule status to Queued.

### See Also

[Configuring Your Revenue Recognition Rules](#)

[Understanding the Revenue Recognition Process](#)

[Organizing Your Revenue Transactions in a Revenue Schedule](#)

## Limitations for Standalone Orders

Orders and order products created without CPQ have unique guidelines and validations to ensure that Salesforce Billing can manage them. Review these guidelines when you create your orders and order products. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

- You can't group standalone orders into invoices based on contracts.
- You can't edit the unit price and quantity of an activated order product.
- To avoid sending invalid data to invoices, don't change CPQ fields and standard fields on an activated standalone order or order product.
- Standalone order products and order products made in Salesforce CPQ can't be on the same order.
- You can't create contracts for standalone orders or order products.
- You can't use standalone evergreen subscription order products with amendment order products. To create an evergreen subscription order product for a new quantity, cancel the existing order product.

## Standalone Order Product Pricing

Standalone order products require you to calculate several pricing and pricing-related fields that would otherwise come from a CPQ quote. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

#### [Pricing and Quantity Fields for Standalone Order Products](#)

Standalone order products contain several price fields that you must populate on your own. Review our pricing formulas to ensure that your prices are accurate when you pass your standalone order products to Salesforce Billing. (Salesforce Billing Managed Package)

#### [Proration Multipliers for Standalone Order Products](#)

An order product's prorate multiplier tells us how many pricing periods are in the order product's term. Salesforce Billing uses a prorate multiplier and other billing fields to calculate the order product's billable unit price. By default, Salesforce CPQ calculates a prorate multiplier for a quote line and passes it to the resulting order product. Because you're creating an order product without CPQ, you must calculate your order product's prorate multiplier on your own. (Salesforce Billing Managed Package)

## Pricing and Quantity Fields for Standalone Order Products

Standalone order products contain several price fields that you must populate on your own. Review our pricing formulas to ensure that your prices are accurate when you pass your standalone order products to Salesforce Billing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

#### **Unit Price**

The base unit price per full billing period for your product. You can provide this value based on the unit price from your pricing catalog.

#### **Total Price**

Quantity \* Unit Price

#### **Billable Unit Price**

For one-time standalone order products, set the billable unit price to the total price.

Billable unit prices aren't required for invoice plan standalone order products.

For subscription standalone order products, Salesforce Billing can calculate a billable unit price, or you can provide your own.

If you want Salesforce Billing to calculate a billable unit price, provide a total amount, billing frequency, prorate multiplier, and subscription term on your standalone order product. Salesforce Billing then calculates a billable unit price using the following formula.

$$[(\text{Total Amount} * \text{Billing Frequency}) \div (\text{Prorate Multiplier} * \text{Subscription Term})]$$

Salesforce Billing recalculates billable unit price upon order product activation. To provide your own billable unit price, enter it after order product activation.

If you provide your own billable unit price, then go back and provide a prorate multiplier and subscription term, Salesforce Billing overrides your billable unit price with its own calculation. However, the change in billable unit price can cause errors within other Salesforce Billing features. If you provide your own billable unit price, we strongly recommend that you don't cause Salesforce Billing to calculate a new billable unit price.

For more information, review [Recurring Billing for Subscription Products](#).

### Prorate Multiplier

Prorate multipliers must be null for one-time standalone order products and have a value of one for standalone evergreen order products.

To calculate a prorate multiplier for all other standalone order products, review [Proration Multipliers for Standalone Order Products](#).



**Note** While you can expect most order product configurations to have the same quantity and ordered quantity, Salesforce Billing still supports different values for these fields. For example, volume-based pricing setups often have a quantity of 1 and a larger ordered quantity. If you create an order product without an ordered quantity, it inherits the value of the Quantity field.

### Proration Multipliers for Standalone Order Products

An order product's prorate multiplier tells us how many pricing periods are in the order product's term. Salesforce Billing uses a prorate multiplier and other billing fields to calculate the order product's billable unit price. By default, Salesforce CPQ calculates a prorate multiplier for a quote line and passes it to the resulting order product. Because you're creating an order product without CPQ, you must calculate your order product's prorate multiplier on your own. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: Salesforce Billing Winter '21 and later

Before starting, note the length of your order product terms, and whether they're in days or months. For

For our examples, we use a monthly order product with a term of 5/23/19 through 09/30/19.

First, let's review the types of proration precisions that CPQ supports and how each option calculates prorate multipliers for the order product. You can then use the formulas to calculate a prorate multiplier on your own.

Proration Precision Type	Calculation	Example
Day	<p><b>Daily Subscriptions</b></p> $\frac{(\text{Number of days in order product's term} \div \text{Number of days in one full subscription term})}{\text{product's default subscription term}}$ <p><b>Monthly Subscriptions</b></p> $\frac{(\text{Number of days in order product's term} \div \text{Number of days in one full subscription term from the order product's start date})}{\text{product's default subscription term}}$ <p>If the full subscription term includes a leap day, use 366 in proration multiplier calculation, even if the quote's term doesn't contain the leap day.</p>	<p><b>Term Units = Day</b></p> <p>05/23/19 through 09/30/19 totals 131 days. Our prorate multiplier is <math>(131 \div 365) = 0.3589</math>. Our prorated list price is <math>\\$12000 * (131 \div 365) = \\$4306.85</math>.</p> <p><b>Term Units = Months</b></p> <p>The default subscription term of 12 months from our start date of 05/23/19 puts the default end date at 05/22/20. Because 2020 is a leap year, we use 366 days when calculating the prorate multiplier. <math>(131 \div 366)</math> gives us a prorate multiplier of 0.3579 and a prorated list price of <math>\\$12000 * (131 \div 366) = \\$4295.08</math>.</p> <p>If the full subscription term includes a leap day, Salesforce CPQ uses 366 in proration multiplier calculation, even if the quote's term doesn't contain the leap day.</p>
Day with Calendar Month Weighted	This value works similarly to Day. The only difference occurs when	<p><b>Term Unit = Month</b></p> <p>The default subscription</p>

Proration Precision Type	Calculation	Example
	<p>your org uses monthly subscription term units and calculates one full subscription term. If your quote's term doesn't contain a leap day, Salesforce CPQ doesn't add an extra day to the full subscription term length.</p> <p>Only use 366 days if the effective subscription term actually includes a leap day.</p>	<p>term of 12 months from our start date of 05/23/19 puts the default end date at 05/22/20.</p> <p>However, because our effective subscription end date is 09/30/19, we don't include the leap day (2/29/20) in our calculations, giving us a prorate multiplier of 0.3589, or <math>(131 \div 365)</math>. Our prorated list price is <math>\\$12000 * (131 \div 365) = \\$4306.85</math>.</p>
Month	<p>Use this formula only if your subscription terms are in months.</p> <p>Divide your order product's subscription term in whole months by the product's subscription term. If your term contains a partial month, round the number of months up to the nearest whole number.</p> <p><i>(Order product's subscription term ÷ product's subscription term)</i></p>	<p><b>Term Unit = Month</b></p> <p>05/23/2019 through 09/22/2019 totals 4 whole months. Due to the remaining 8 days, Salesforce CPQ rounds the month total to 5.</p> <p>Prorate multiplier = <math>(5 \div 12) = 0.4167</math></p> <p>Prorated list price = <math>\\$12000 * (5 \div 12) = \\$5000</math></p>
Monthly + Daily	<p>Use this formula only if your subscription terms are in months.</p> <p>Calculate the subscription term's length as its number of whole months plus a decimal for any</p>	<p><b>Term Unit = Month</b></p> <p>05/23/19 through 09/22/19 = 4 months</p> <p>09/23/19 through 09/30/19 = 8 days, so we use <math>(8 \div (365 \div 12))</math> to</p>

Proration Precision Type	Calculation	Example
	<p>partial month at the end of the term, then divide this value by your subscription term. The partial month equals the number of days in the month divided by <math>(365 \div 12)</math>.</p> <p>We recommend using this formula if you sell and price products by month or year but don't regularly quote for specific periods of time.</p> <p style="background-color: #f0f8ff; padding: 5px;"><code>(Term length in whole months) + ((Remaining Days) ÷ (365/12))</code></p>	<p>calculate the partial month value in our prorate multiplier.</p> <p>Prorate multiplier = <math>(4 + (8 / (365 / 12))) / 12 = 0.3553</math></p> <p>Prorated list price = <math>\\$12,000 * ((4 + (8 / (365 / 12))) / 12) = \\$4263.01</math>.</p>
Calendar Monthly + Daily	<p>Use this formula only if your subscription terms are in months.</p> <ul style="list-style-type: none"> <li>If the first month is a partial month, divide the number of term days in the month by the total number of days in that month.</li> <li>If the last month is a partial month, divide the number of term days in the month by the total number of days in that month.</li> <li>Add those values to the number of whole months in your term.</li> <li>Divide the result by 12.</li> </ul>	<p><b>Term Unit = Month</b></p> <p>05/23/19 through 05/31/19 = <math>(9 \div 31)</math> days</p> <p>06/01/19 through 08/31/19 = 3 months</p> <p>09/01/19 through 09/30/19 = <math>(30 \div 30)</math> days</p> <p>Prorate multiplier = <math>(3 + (9 \div 31 + 30 \div 30)) \div 12 = 0.3575</math></p> <p>Prorated list price = \$4,290.32</p>

You can use this table to figure out the formula needed to provide a prorate multiplier that aligns with your proration needs.

Salesforce Billing always invoices the customer for the order product's complete balance. However, recurring order products can split that balance differently over their invoice lines based on how you calculated your order product's proration multiplier and Salesforce Billing proration package settings. To

learn how you can align your proration settings, review [Aligning Proration Between CPQ and Billing](#).

## See Also

[Proration with Invoices](#)

## Required Configurations for Types of Standalone Order Products

Standalone orders require you to fill out all the fields that would otherwise come from a CPQ quote. We Certain fields require a specific value, while some are required but (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

#### **Requirements for Standalone One-Time Order Products**

Create a standalone one-time order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

#### **Requirements for Standalone Termed Recurring Order Products**

Create a termed recurring order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

#### **Requirements for Standalone Evergreen Order Products**

Create a monthly evergreen order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

#### **Requirements for Standalone Invoice Plan Order Products**

Create a standalone invoice plan order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

#### **Requirements for Standalone Amendment Order Products**

Create a standalone amendment order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

#### **Requirements for Standalone Cancellation Order Products**

Create a standalone cancellation order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

#### **Requirements for Standalone Order Products with Consumption Schedules**

Create a monthly order product with an order product consumption schedule and consumption rates in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

#### **Requirements for Standalone Order Products with Price Schedules**

Create a standalone order product with price schedules in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

## Requirements for Standalone One-Time Order Products

Create a standalone one-time order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Winter '21 and later

## Required Fields

Product	
Field	Value
Billing Rule	[Required]
Charge Type	One-Time
Revenue Recognition Rule	[Required]
Tax Rule	[Required]

Order	
Field	Value
Account	[Required]
Effective Date	[Required]
Price Book	[Required]
Status	Draft

Order Product	
Field	Value
Billable Unit Price	Required for invoice runs or Bill Now to pick up the order product for invoicing. For One-Time products, Billable Unit Price = Unit Price x Ordered Quantity.
Billing Rule	[Get value from product]
Charge Type	One-Time
End Date	[Required]
Legal Entity	[Optional]
Order	[Required]
Ordered Quantity	[Required]
Price Book Entry	[Required]
Quantity	[Required]

Field	Value
Revenue Recognition Rule	[Get value from product]
Service Date	[Required]
Status	Draft
Tax Rule	[Get value from product]
Unit Price	[Required]

## Requirements for Standalone Termed Recurring Order Products

Create a termed recurring order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

## Required Fields

Product	
Field	Value
Billing Frequency	Monthly, Quarterly, Semiannual, or Annual
Billing Rule	[Required]
Charge Type	Recurring
Revenue Recognition Rule	[Required]
Tax Rule	[Required]

### Order

Field	Value
Account	[Required]
Effective Date	[Required]
Price Book	[Required]
Status	Draft

## Order Product

Field	Value
Billable Unit Price	Required for invoice runs or Bill Now to pick up the order product for invoicing. We recommend calculating a value based on the formula described in <a href="#">Recurring Billing and Billable Unit Price</a> .
Billing Frequency	Monthly, Quarterly, Semiannual, or Annual
Billing Rule	Get value from product
Billing Type	[Required]
Charge Type	Recurring
Default Subscription Term	[Required]
End Date	[Required]
Order	[Required]
Ordered Quantity	[Required]
Quantity	[Required]
Revenue Recognition Rule	Get value from product
Service Date	[Required]
Status	Draft
Tax Rule	Get value from product
Unit Price	[Required]

## Requirements for Standalone Evergreen Order Products

Create a monthly evergreen order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

-  **Note** Only the LIFO cancellation rule is supported. This cancellation rule is in the billing treatment that the billing rule pulls in. If no billing rule exists, the package setting is used, and LIFO must be specified in that setting. Otherwise the order can't be activated.

## Required Fields

Product	
Field	Value
Billing Frequency	Monthly, Quarterly, Semiannual, or Annual
Billing Rule	[Required]
Charge Type	Recurring
Revenue Recognition Rule	[Required]
Tax Rule	[Required]

Order	
Field	Value
Account	[Required]
Effective Date	[Required]
Price Book	[Required]
Status	Draft

Order Product	
Field	Value
Billable Unit Price	Required for invoice runs or Bill Now to pick up the order product for invoicing. We recommend calculating a value based on the formula described in <a href="#">Recurring Billing and Billable Unit Price</a> .
Billing Frequency	Monthly, Quarterly, Semiannual, or Annual
Billing Rule	Get value from product
Billing Type	[Required]
Charge Type	Recurring
Order	[Required]
Ordered Quantity	[Required]
Price Book Entry	[Required]
Product Subscription Type	Evergreen
Quantity	[Required]

Field	Value
Revenue Recognition Rule	Get value from product
Service Date	[Required]
Status	Draft
Subscription Type	Evergreen
Tax Rule	Get value from product
Unit Price	[Required]

## Requirements for Standalone Invoice Plan Order Products

Create a standalone invoice plan order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

Salesforce Billing invoices standalone invoice plan order products only when the order product has a billing treatment with a New Order Invoice Plan value. You can configure your billing rules and legal entities so that Salesforce Billing assigns your order product a billing treatment. Or, you can create your order product, and then give it a billing treatment on your own.

## Required Fields

Product	
Field	Value
Billing Frequency	Invoice Plan
Billing Rule	[Required]
Charge Type	Recurring
Revenue Recognition Rule	[Required]
Tax Rule	[Required]

### Order

Field	Value
Account	[Required]
Effective Date	[Required]

Field	Value
Price Book	[Required]
Status	Draft

**Order Product**

Field	Value
Billable Unit Price	When Enable Billing Order Validations is active, Salesforce Billing sets this value to zero for order products created with a billing frequency of Invoice Plan.
Billing Frequency	Invoice Plan
Billing Rule	[Required]
Billing Rule	Get value from product
Billing Treatment	[Must be a billing treatment with an active invoice plan]
Charge Type	Recurring
End Date	[Required]
Price Book Entry	[Required]
Quantity	[Required]
Revenue Recognition Rule	[Required]
Revenue Recognition Rule	[Get value from product]
Service Date	[Required]
Status	Draft
Tax Rule	[Required]
Tax Rule	[Get value from product]
Unit Price	[Required]

**See Also**

[Dynamic Invoice Plans](#)

**Requirements for Standalone Amendment Order Products**

Create a standalone amendment order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Winter '21 and later

Provide a value for the revised order product ID. The Revised Order Product field shows the ID of the original order product used before any amendments were made. An order product can be related to multiple amendment order products.

You can't use standalone evergreen subscription order products with amendment order products.

## Required Fields

### Product

Field	Value
Billing Frequency	Invoice Plan
Billing Rule	[Required]
Charge Type	Recurring
Revenue Recognition Rule	[Required]
Tax Rule	[Required]

### Order

Field	Value
Account	[Required]
Effective Date	[Required]
Price Book	[Required]
Status	Draft

### Order Product

Field	Value
Billable Unit Price	Required for invoice runs or Bill Now to pick up the order product for invoicing. We recommend calculating a value based on the formula described in <a href="#">Recurring Billing and Billable Unit Price</a> .
Billing Frequency	[Required]
Billing Rule	[Get value from product]
Billing Type	[Required]

Field	Value
Charge Type	[Required]
Charge Type	[Required]
Default Subscription Term	[Required]
End Date	[Required]
Order	[Required]
Ordered Quantity	[Required]
Price Book Entry	[Required]
Quantity	[Required]
Reference ID	[Required]
Revenue Recognition Rule	[Get value from product]
Revised Order Product	[Required]
Service Date	[Required]
Status	Draft
Tax Rule	[Get value from product]

## See Also

[Cancellation Order Management](#)  
[LIFO Order Product Cancellation](#)

## Requirements for Standalone Cancellation Order Products

Create a standalone cancellation order product in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

Salesforce Billing calculates canceled billings for a canceled standalone order product under the following settings.

- The cancellation order product has a Contract Action value of Cancel.
- The cancellation order product and related amendment order products all have a Revised Order Product field with the ID of the original order product used before any amendments or cancellations.
- The cancellation order products and related amendment order products all have the same terminated date.

- Contracting isn't supported for standalone order products.

 **Note** You can cancel standalone order products under both legacy cancellation and LIFO cancellation.

Product	
Field	Value
Billing Frequency	Must match the value of the Revised Order Product field
Billing Rule	[Required]
Charge Type	Recurring
Revenue Recognition Rule	[Required]
Tax Rule	[Required]

Order	
Field	Value
Account	[Required]
Effective Date	[Required]
Price Book	[Required]
Status	Draft

Order Product	
Field	Value
Billing Frequency	Must match the value of the Revised Order Product field
Billing Rule	[Get value from product]
Billing Type	[Required]
Billable Unit Price	Required for invoice runs or Bill Now to pick up the order product for invoicing. We recommend calculating a value based on the formula described in <a href="#">Recurring Billing and Billable Unit Price</a> .
Charge Type	Recurring
Contract Action	Cancel
Default Subscription Term	[Required]

Field	Value
End Date	[Required] for termed subscriptions, omit for evergreen subscriptions
Legal Entity	Must match the value of the Revised Order Product field
Order	[Required]
Ordered Quantity	Negative amount of the Revised Order Product field
Price Book Entry	Must match the value of the Revised Order Product field
Product Subscription Type	[Required]
Quantity	Negative amount of the Revised Order Product field
Revenue Recognition Rule	[Get value from product]
Revised Order Product	[ID number for the original order product used before any amendments]
Service Date	[Required]
Status	Draft
Subscription Type	[Required]
Tax Rule	[Get value from product]
Unit Price	[Required]
Terminated Date	<p>[Set to the last day the canceled order product is billed]</p> <p>Terminated Date and Contract Action fields must be populated in a separate DML after the order product is created, all other required fields are populated, and the order product is activated.</p> <p>Terminated Date and Contract Action fields must be populated after the cancellation order product is created and activated. To do so:</p> <ul style="list-style-type: none"> <li>• Populate Terminated Date on the original order product (the one that is linked in the Revised Order Product field).</li> </ul>

Field	Value
	<ul style="list-style-type: none"> <li>On the cancellation order product, populate Terminated Date and set Contract Action to Cancel.</li> <li>Save.</li> </ul>

**See Also**

- [Cancellation Order Management](#)  
[LIFO Order Product Cancellation](#)

**Requirements for Standalone Order Products with Consumption Schedules**

Create a monthly order product with an order product consumption schedule and consumption rates in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

**REQUIRED EDITIONS**


---

Available in: Salesforce Billing Winter '21 and later

---

You must create the order, order product, order product consumption schedule, and at least one order product consumption rate. Your order product must be related to a product with a consumption schedule.

Salesforce Billing creates usage summaries when you activate an order product with the following configuration.

- The order product is related to a product with a consumption schedule.
- The order product has an active order product consumption schedule with at least one consumption rate.

**Required Fields**

Product

Field	Value
Billing Frequency	Monthly, Quarterly, Semiannual, or Annual
Billing Rule	[Required]
Charge Type	Recurring
Revenue Recognition Rule	[Required]
Tax Rule	[Required]

## Order

Field	Value
Account	[Required]
Effective Date	[Required]
Price Book	[Required]
Status	Draft

## Order Product

Field	Value
Billable Unit Price	Required for invoice runs or Bill Now to pick up the order product for invoicing. We recommend calculating a value based on the formula described in <a href="#">Recurring Billing and Billable Unit Price</a> .
Billing Frequency	[Required]
Billing Rule	[Get value from product]
Billing Type	[Required]
Charge Type	[Required]
Charge Type	[Required]
Default Subscription Term	[Required]
End Date	[Required]
Order	[Required]
Ordered Quantity	[Required]
Price Book Entry	[Required]
Quantity	[Required]
Reference ID	[Required]
Revenue Recognition Rule	[Get value from product]
Revised Order Product	[Required]
Service Date	[Required]
Status	Draft
Tax Rule	[Get value from product]

### Consumption Schedule

<b>Field</b>	<b>Value</b>
Billing Rule	[Required]
Billing Term	[Required]
Billing Term Unit	[Required]
Category	[Required]
Rating Method	[Required]
Revenue Recognition Rule	[Required]
Tax Rule	[Required]
Type	[Required]

### Consumption Rates

<b>Field</b>	<b>Value</b>
Lower Bound	[Required]
Price	[Required]
Pricing Method	[Required]
Processing Order	[Required]
Upper Bound	[Optional]

### Order Product Consumption Schedule

<b>Field</b>	<b>Value</b>
Billing Rule	[Required]
Billing Term	[Required]
Billing Term Unit	[Required]
Category	[Required]
Consumption Schedule	[Optional]
Order Product	[Required]
Rating Method	[Required]
Revenue Recognition Rule	[Required]
Tax Rule	[Required]
Type	[Required]

Field	Value
Lower Bound	[Required]
Order Product Consumption Schedule	[Required]
Price	[Required]
Pricing Method	[Required]
Processing Order	[Required]
Upper Bound	[Optional]

#### Creating a Billing Order with Consumption-Based Usage (Standard Users)

Field	Value
Consumption Rates	[Required]
Consumption Schedule	[Required]
Order Item Consumption Rates	[Required]
Order Item Consumption Schedule	[Required]
Price Schedule	[Required]
SBQQ__BlockPrice__c	[Required]
SBQQ__DiscountSchedule__c	[Required]

#### Requirements for Standalone Order Products with Price Schedules

Create a standalone order product with price schedules in Salesforce Lightning or Salesforce Classic. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

**!** **Important** Salesforce CPQ doesn't automatically create price schedules for standalone order products. You must create the price schedule and its price tiers on your own. Each price schedule requires at least one price tier.

#### Required Fields

Product	
Field	Value
Billing Frequency	Monthly, Quarterly, Semiannual, or Annual

<b>Field</b>	<b>Value</b>
Billing Rule	[Required]
Charge Type	Usage
Revenue Recognition Rule	[Required]
Tax Rule	[Required]

**Order**

<b>Field</b>	<b>Value</b>
Account	[Required]
Effective Date	[Required]
Price Book	[Required]
Status	Draft

**Order Product**

<b>Field</b>	<b>Value</b>
Billing Frequency	[Required]
Charge Type	Usage
End Date	[Required]
Price Book Entry	[Required]
Price Schedule	[Required]
Quantity	[Required]
Service Date	[Required]
Unit Price	[Required]

**Price Schedule**

<b>Field</b>	<b>Value</b>
Discount Unit	[Required]
Price Type	[Required]

**Price Tier**

<b>Field</b>	<b>Value</b>
Lower Bound	[Required]
Optional	[Optional]

Field	Value
Price Model	[Required]
Price Schedule	[Required]

## Overriding Billing Dates

Salesforce Billing allows you to temporarily change important billing dates on your order products. These features help you control the timing and duration of invoice generation for your customers. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### **Override Next Billing Date**

Set a one-time custom billing date with the order product's Override Next Billing Date field. This feature is useful if you want to control when your invoice schedulers pick up certain order products for invoicing. (Salesforce Billing Managed Package)

#### **Bill Through Date Override**

Manually set billing dates on your invoices to control the timing and duration of invoice generation for your customers. This feature lets you minimize the number of invoice runs you need to manage. (Salesforce Billing Managed Package)

#### **Managing Partial Periods When Using Bill Through Date Override**

The billing rule field Period Treatment for BTDO controls how Salesforce creates invoice lines when a Bill Through Date Override value creates a partial billing period. (Salesforce Billing Managed Package)

## Override Next Billing Date

Set a one-time custom billing date with the order product's Override Next Billing Date field. This feature is useful if you want to control when your invoice schedulers pick up certain order products for invoicing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

By default, invoice runs pick up order products when the order product's Next Billing Date field falls on or before the invoice scheduler's target date. However, if the Override Next Billing Date field contains a value, Salesforce Billing uses that date as the next billing date instead. After Salesforce Billing creates the invoice line, it removes the Override Next Billing Date value and calculates the new Next Billing Date.

 **Important** Next Billing Date is a system field that users and admins shouldn't change.

 **Example** You sell a monthly music streaming subscription that bills in advance on the first of the month. Your customer wants to cancel their subscription as of 05/01/2018.

Order Product: Music Streaming Subscription

- Billing Frequency: Monthly
- Order Amount: 500
- Charge Type: Recurring
- Billing Type: Advance
- Next Billing Date: 4/1/2018
- Start Date: 4/1/2018
- End Date: 3/31/2019

Set your Override Next Billing Date to 5/1/2018. When your invoice scheduler picks up the order product on May 1, it produces an invoice line for month of March and April. That invoice line contains two months worth of billing for your subscription.

## Bill Through Date Override

Manually set billing dates on your invoices to control the timing and duration of invoice generation for your customers. This feature lets you minimize the number of invoice runs you need to manage.  
(Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and later

#### Bill Through Date Override

Provide an end date for the next invoice line created from this order product. This field's value overrides the standard End Date calculation of (last Charge To Date + Billing frequency).

Salesforce Billing clears this value after the system generates the next Invoice Charge.

Bill Through Date Override also defines the End Date of the next billing period. If the value of Bill Through Override Date is greater than the Next Charge Date and Billing frequency, Salesforce Billing creates multiple invoice lines for the order product.

For example, let's say you had an order product where Next Billing Date and Next Charge Date were set to 4/1/17 with monthly Billing Frequency. This setup leads to four Invoice Lines.

- 04/01/17 - 04/30/17
- 05/01/17 - 05/31/17
- 06/01/17 - 06/30/17
- 07/01/17 - 07/31/17

If you change the Bill Through Date on this Order Product from null to 7/15, you'd end up with the

following Invoice Lines.

- 04/01/17 - 04/30/17
- 05/01/17 - 05/31/17
- 06/01/17 - 06/30/17
- 07/01/17 - 07/15/17

### Example

If a Bill Through Date Override value extends the billing period for an order product, Salesforce Billing creates an extra invoice line and prorates it to cover the extended portion of the billing period. Consider the following order product.

#### **Service Desk Subscription**

Start Date: 01/01/17

End Date: 12/31/17

Billing Day of Month: 1

Next Billing Date: 01/01/17

Next Charge Date: 01/01/17

Bill Through Date Override: None

Billing Frequency: Quarterly

Amount: \$1200

Standard quarterly invoicing for this product would produce four invoices, each with one \$300 invoice line. Let's look at what happens when a user or action changes the order product's Bill Through Date Override between the first and second billing dates.

By default, the second billing period would have one invoice line with a start date of 04/01/17 and an end date of 06/30/17. However, since the Bill Through Date Override value added an extra month, we have an extra invoice line prorated to cover the extra time. This in turn moves the third billing period's start and end dates a month later. To cover the remaining time in the order product's year-long lifecycle, the last billing period lasts for only two months, with an invoice line that's prorated accordingly. By the end of the year, we've billed the order product for \$1200 exactly as we would have if we worked with four three-month-long billing periods for \$300 each. The only difference came in the total amount we billed for in the second and fourth billing period, and the number of invoice lines in the second period. These differences impact bookings and revenue recognition reporting.

### Example

You sell a monthly IT support subscription that bills in advance on the first of the month.

Your customer wants to cancel their subscription as of 05/15/2018.

Order Product: IT Support Subscription

- Billing Day of Month: 1
- Billing Frequency: Monthly
- Order Amount: 500
- Charge Type: Recurring
- Billing Type: Advance
- Next Billing Date: 3/1/2018
- Start Date: 3/1/2018
- End Date: 2/28/2019

Set the Bill Through Date Override to 5/15/2018. This date covers three billing periods: The month of March, the month of April, and a partial period containing May 1 through May 15. Since you're billing in advance, Salesforce Billing moves the next billing date to the beginning of the final period, 5/1/2018. The invoice line for this order product contains charges March, April, and prorated charges for the partial period of 5/2 through 5/15. The next billing date of 6/1 creates an invoice line for May's remaining partial period of 5/16 through 5/30.

If you bill in arrears, Salesforce Billing instead moves the Next Billing Date to 6/1/2018. This action creates an invoice line with charges for March, April, and May.

## Managing Partial Periods When Using Bill Through Date Override

The billing rule field Period Treatment for BTDO controls how Salesforce creates invoice lines when a Bill Through Date Override value creates a partial billing period. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '21 and later

---

The Period Treatment for BTDO field has these values.

#### Separate

Create separate invoice lines for the partial period and any remaining full periods, and include Billing Day of Month when calculating the invoice line balances.

#### Combine

Combine the partial period and any remaining full periods into one invoice line, and include Billing Day of Month when calculating the invoice line balances.

#### Exclude Billing Day of Month

Combine the partial period and any remaining full periods into one invoice line, and ignore the billing day of month when calculating invoice line balances. This calculation process works the same as earlier versions of Salesforce Billing, where the partial period treatment was ignored for orders with a Bill Through Date Override.

Before we see how they affect your invoice lines, let's look at the standard invoicing process for a sample order product. Your order product bills monthly with a Billing Day of Month of 10. It has a start date of 01/01/20, end date of 12/31/21, and unit price of \$240. The standard invoicing process creates four invoice lines for the first three months.

Invoice Line	Start Date	End Date	Balance
1	01/01/20	01/09/20	\$2.90
2	01/10/20	02/09/20	\$10.00
3	02/10/20	03/09/20	\$10.00
4	03/10/20	04/09/20	\$10.00

Let's say you added a Bill Through Override Date of 03/20/20 before invoicing. If you don't have a value for your Period Treatment for BTDO field, Salesforce Billing creates one invoice line. Salesforce Billing also ignores the billing day of month, so the periods used to calculate the invoice line's overall balance are prorated differently. These proration differences can cause unintended balances for users who expected the billing day of month to be included in invoice line balance calculation.

Invoice Line	Start Date	End Date	Balance
1	01/01/20	03/20/20	\$26.45

With a Period Treatment for BTDO value of Separate, Salesforce Billing creates separate invoice lines with the final period prorated according to the new bill through date override.

Period Treatment for BTDO: Separate

Invoice Line	Start Date	End Date	Balance
1	01/01/20	01/09/20	\$2.90
2	01/10/20	02/09/20	\$10.00
3	02/10/20	03/09/20	\$10.00
4	03/10/20	3/20/20	\$3.55

If your Period Treatment for BTDO value is Combine, Salesforce Billing uses the same four periods to calculate the invoice line balance, but wraps them all into one invoice line.

Invoice Line	Start Date	End Date	Balance
1	01/01/20	03/20/20	\$26.45

- (checkmark) **Note** For some amendments, if your Period Treatment for BTDO value is Combine, and Bill Through Date Override is set past the terminated date, Salesforce Billing creates two invoice lines: one through the terminated date and one from the terminated date through the date in Bill Through

### Date Override.

If your value is Exclude Billing Day of Month, Salesforce Billing produces one invoice line but ignores Billing Day of Month during proration calculations. As a result, the invoice line balance varies based on your bill through date override, start date, and end date. In this example, the balance is the same as if you used Combine, though other date combinations can cause a different balance.

Using Exclude Billing Day of Month produces the same result as leaving the Partial Period for BTDO field blank.

Period Treatment for BTDO: Exclude Billing Day of Month

Invoice Line	Start Date	End Date	Balance
1	01/01/20	03/20/20	\$26.45

## Order Creation and Activation

Salesforce CPQ runs several validations during the order creation and also following order activation. You have different options for adjustments based on your order's activated status. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### [Editable Order Fields](#)

Get the most out of orders by reviewing order field edit permissions. Salesforce Billing locks certain order fields from user edits based on user permission and order activation status. You also can't manually change key finance fields such as tax or amount regardless of permissions or order status. (Salesforce Billing Managed Package)

#### [Deactivating Orders in Salesforce Billing](#)

Deactivate an order product to return it to draft status and change its editable fields. (Salesforce Billing Managed Package)

## Editable Order Fields

Get the most out of orders by reviewing order field edit permissions. Salesforce Billing locks certain order fields from user edits based on user permission and order activation status. You also can't manually change key finance fields such as tax or amount regardless of permissions or order status. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

If you don't have the Edit Activated Order user permission, you can't edit any fields on an activated order. Otherwise, you can edit the following fields on an activated order.

### Edited Fields for Activated Orders

Order Name

Order Type

Order Reference Number

PO Number

Status

Bill Now

Contracted

PO Date

Contracting Method

Ship To Contact

Bill To Contact

Shipping Address

Billing Address

Salesforce CPQ and Salesforce Billing automate financial field changes to ensure that users don't make changes that have a legal impact on bookings and billings. Sales reps and admins can't manually edit these fields at any time.

### Locked Financial Fields

Order Amount

Estimated Tax

Total Amount (With Tax)

Billed Amount (Without Tax)

Billed Tax

Total Billings

Salesforce Billing locks the following relationship fields to ensure that the order record maintains data integrity.

#### Locked Relationship Fields

Original Order

Activated Date

Reduction Order

Contract Name

Contract End Date

## Deactivating Orders in Salesforce Billing

Deactivate an order product to return it to draft status and change its editable fields. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

You can invoice only active order products. Invoice schedulers don't evaluate draft order products, and clicking **Bill Now** on an order doesn't create invoice lines for draft order products. In Salesforce CPQ, a user or process activates order products through one of two methods.

- Click **Activate** on a draft order, which activates all its order products.
- Change an order product's Activated field from Draft to Activated. This lets you activate an order product even if its parent order isn't activated.

To deactivate all of an order's order products, click **Deactivate**. You can also deactivate an individual order product by deselecting its Activated field.

When you're working with deactivated order products in Salesforce Billing, remember important guidelines.

- You can't deactivate an order product related to an active invoice line.
- You can invoice an active order product even if its parent order is deactivated.

- Deactivating an order product returns it to draft status.
- Deactivating a usage order product deletes all its usage summaries and usage records. If you want to upload usage again, you'll have to reactivate the order product and then upload usage manually or by a workflow rule.

## Billing Status Fields

Review key status fields to track the position of order products and usage summaries in their billing lifecycles. They're useful if you're unsure whether an order product or usage summary has been invoiced or is available for invoicing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

## Order Products

### Invoice Run Processing Status

Shows the status of an order product as the invoice run evaluates it.

- Completed: The invoice run created and posted an invoice line for the order product. The order product won't create any more invoice lines.
- Error: The invoice run encountered an error while trying to invoice the order product. For more information, review the invoice scheduler's error log.
- In Progress: The invoice run has created an invoice line but hasn't posted it yet.
- Pending Billing: The order product can be picked up by an upcoming invoice run.
- Will Not Invoice: If a billing rule's Generate Invoice field has a value of No, all the rule's order products are created with an invoice run processing status set to Will Not Invoice. If you still want the next eligible invoice run to pick up your order product, you can change the processing status from Will Not Invoice to Pending Billing on your own. The change won't cause any previous invoice runs to retroactively pick up the invoice. For more information on order product invoicing requirements, review [Bill Now](#).

### Invoice Run Processing Status (Index)

Inherits the value of the invoice run processing status. Salesforce Billing uses this field for reference and users can't change it.

## Usage Summaries

### Status

Usage summaries use the Status field to track the summary's status in the invoicing process.

- Cancelled: The usage summary was cancelled and won't invoice or load new usage records.

- New: The usage summary doesn't contain any usage records. Invoice runs won't evaluate it for invoicing.
- Processed: An invoice run captured the usage summary for invoicing. Users can't load it with any more usage records.
- Queued for Group Calculation: A user has uploaded one or more usage records to a usage summary related to an order product with included usage. Salesforce Billing is calculating usage quantities and totals on the summary based on the included usage values. After calculations are finished, the Status field changes its value to Queued for Invoicing.
- Queued for Subtotal Calculation: A user has uploaded one or more usage records to a volume-based usage summary and Salesforce Billing is calculating the usage summary's subtotal. After calculations are finished, the Status field changes its value to Queued for Invoicing.
- Queued for Invoice: A user has uploaded one or more usage records to the usage summary. The usage summary is available for invoicing.

### Invoice Run Processing Status

Shows the status of a usage summary as the invoice run evaluates it.

- Completed: The invoice run created and posted an invoice line for the usage summary.
- Error: The invoice run encountered an error while trying to invoice the usage summary. For more information, review the error log.
- In Progress: The invoice run created an invoice line but hasn't posted it yet.
- Pending Billing: The usage summary can be picked up by an upcoming invoice run.
- Will Not Invoice: This usage summary won't be picked up for invoicing. Can be used like Hold Billing on an order product.

### Invoice Run Processing Status (Index)

Inherits the value of the invoice run processing status. Salesforce Billing uses this field for reference. We recommend not changing it.

## Proration with Invoices

Salesforce Billing uses proration to calculate balances for invoice lines that cover partial billing periods. The invoice line field Calculated Quantity performs a similar function to the quote line and order product's Prorate Multiplier fields. (Salesforce Billing Managed Package)

On the invoice line, for subscription pricing, Billable Unit Price (from the order product) \* Calculated Quantity = Subtotal.

Calculated Quantity is a proration that captures the number of billable periods on the invoice line. For example, if the invoice line covers one month and Billing Frequency is Monthly, Calculated Quantity = 1.000000.

Similarly, if the invoice line covers three months and Billing Frequency is Quarterly, Calculated Quantity = 1.000000. If the invoice line covers three months and Billing Frequency is Monthly, Calculated Quantity = 3.000000.

The calculated quantity is then used to determine the invoice line's subtotal (without tax).

$$\text{Subtotal (without tax)} = (\text{Calculated Quantity} * \text{OrderProduct.BillableUnitPrice})$$

### Partial Proration Type

The initial setting for determining an invoice line's calculated quantity. You can select Day or Month + Day.

#### Day

Salesforce Billing divides the number of days in the invoice line by the number of days in each month within the range of the last billing frequency. For example, you have an invoice line from 10/11/19 through 10/20/19, related to an order product with quarterly billing frequency. The 10-day billing period is divided by the total days in each month of the previous quarter: 31 days in July, 31 days in August, and 30 days in September, or  $(10 / 92) * \text{quantity} 1 = 0.1087$ .

#### Month + Day

If you select Month + Day, Salesforce Billing looks at the value of the Proration Type package setting.

### Proration Type

Salesforce Billing evaluates this setting only when the partial proration type has a value of Month + Day.

#### Calendar Days

Use the number of days in the first month of the billing period.

#### 30 Days

Use a flat value of 30 days per month.

#### Monthly: CPQ Formula

Use the CPQ formula of  $(365 / 12)$  days per month.



**Example** A sales rep quoted and ordered a subscription with a term from 05/23/19 through 09/30/19. Assuming our Billing Day of Month has a value of 1, let's see how each proration setting would affect the balance of the first invoice line.

Proration Type Value	Description	Invoice Line Balance Calculation
Partial Proration Type: Days	Use the number of days in the previous billing frequency.	The billing frequency is monthly, and the previous month is April, which contains 30 days. $((9 / 30) * 1) = 0.30 * \$1000 = \$300$ .

Proration Type Value	Description	Invoice Line Balance Calculation
Proration Type: Calendar Days	Use the number of days in the first month of the billing period.	<ul style="list-style-type: none"> <li>The first invoice line runs for 9 days, from May 23 through May 31. So, we use <math>((9 / 31) * 1)</math> and multiply that by the order product's \$1000 monthly price to get \$290.32.</li> </ul>
Proration Type: 30 Days	Use a flat value of 30 days per month.	$((9 / 30) * 1) = 0.30 * \$1000 = \$300$
Proration Type: Monthly (CPQ Formula)	Use the CPQ formula of $(365 / 12)$ days per month.	$((9 / (365 / 12)) * 1) = \$295.89$

 **Example** Actions that change an order product's next billing date affect the length of the invoice line and the prorated invoice line value. Actions could include changing the order's billing day of month or setting an override next billing date on the order product. Make sure to keep these downstream impacts in mind when you take such actions. Let's look at a few examples.

#### Order Product Billing Type: Advance

Action	Result	Proration Type: Calendar	Proration Type: Monthly (CPQ Formula)
Keep Billing Day of Month at 1	<b>Next Billing Date</b> 05/01/19  <b>Invoice Line Start Date</b> 05/23/19  <b>Invoice Line End Date</b> 05/31/19	$(09 / 31) * \$1000 = \$290.32$	$(9 / (365 / 12)) = 0.2959 * \$1000 = \$295.89$
Change Billing Day of Month to 11	<b>Next Billing Date</b> 05/11/19  <b>Invoice Line Start Date</b> 05/23/19	The first month of our period is May, so divide the 19 total days by the 31 days in May. $((19 / 31) * \$1000) = \$612.90$	$(19 / (365 / 12)) * \$1000 = \$624.66$

Action	Result	Proration Type: Calendar	Proration Type: Monthly (CPQ Formula)
	<b>Invoice Line End Date</b> 06/10/19		
Change Billing Day of Month to 30	<b>Next Billing Date</b> 05/30/19  <b>Invoice Line Start Date</b> 05/23/19  <b>Invoice Line End Date</b> 05/29/19	$(7 / 31) * \$1000 = \$225.80$	$(7 / (365 / 12)) * \$1000 = \$230.14$

If we bill in arrears, the invoice lines have the same value. The next billing date is the only difference – keep this in mind when configuring your invoice runs.

#### Order Product Billing Type: Arrears

Action	Result	Proration Type: Calendar	Proration Type: Monthly (CPQ Formula)
Keep Billing Day of Month at 1	<b>Next Billing Date</b> 06/01/19  <b>Invoice Line Start Date</b> 05/23/19  <b>Invoice Line End Date</b> 05/31/19	$(09 / 31) * \$1000 = \$290.32$	$(9 / (365 / 12)) = 0.2959 * \$1000 = \$295.89$
Change Billing Day of Month to 11	<b>Next Billing Date</b> 06/11/19  <b>Invoice Line Start Date</b> 05/23/19	The first month of our period is May, so divide the 19 total days by the 31 days in May. $((19 / 31) * \$1000) = \$612.90$	$(19 / (365 / 12)) * \$1000 = \$624.66$

Action	Result	Proration Type: Calendar	Proration Type: Monthly (CPQ Formula)
	<b>Invoice Line End Date</b> 06/10/19		
Change Billing Day of Month to 30	<b>Next Billing Date</b> 05/30/19  <b>Invoice Line Start Date</b> 05/23/19  <b>Invoice Line End Date</b> 05/29/19	$(7 / 31) * \$1000 = \$225.80$	$(7 / (365 / 12)) * \$1000 = \$230.14$



**Example** When billing semi-annual products, the proration calculation adjusts charges based on the Billing Day of Month. The proration multiplier determines the proportion of the billing period used, directly impacting the final invoice amount. Let's look at a few examples.

Action	Result	Calculated Quantity	Prorated Charge (on \$6,000)
Billing Day of Month at 1 Billing Period: 10/28/19 to 3/31/20	<b>Breakdown</b> 10/28/19 to 11/27/19 is 1 month 11/28/19 to 3/27/20 is 4 months 3/28/20 to 3/31/20 is partial month, 4 days / 31 days = 0.12903226	For a semi-annual calculation, we divide by 6. $((1 + 4 + 0.12903226) / 6 = 0.85483871$	$0.85483871 \times 6000 = \$5129.03$

Action	Result	Calculated Quantity	Prorated Charge (on \$6,000)
Billing Day of Month to 11 Billing Period: 10/19 to 3/9/20	<p><b>Breakdown</b></p> <p>10/10/19 to 10/31/19 is partial month, (22 days / 31 days) = 0.70967742</p> <p>11/1/19 to 2/29/20 is 4 months</p> <p>3/1/20 to 3/9/20 is partial month, (9 days / 31 days) = 0.29032258</p>	<p>For a semi-annual calculation, we divide by <math>6 \cdot (0.70967742 + 4 + 0.29032258) / 6 = 0.83333333</math></p>	$0.83333333 \times 6000 = \$5000$

## Billing and Invoice Cancellation

Salesforce Billing provides several ways to manage canceled invoices or order products that you cancel during the billing process. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### [Cancel and Rebill an Invoice](#)

Use the Cancel and Rebill option to revert the invoice to its prior state before the most recent billing cycle. Use this option to fix any errors in the invoice record. (Salesforce Billing Managed Package)

#### [Legacy Billing for Canceled Products](#)

When you cancel a subscription, Salesforce Billing updates several objects so you can track how much revenue you've canceled. (Salesforce Billing Managed Package)

#### [Cancellation Order Management](#)

Your users have to cancel recurring or usage-based services in response to customer cancellation or as part of a collections process. You can use Salesforce CPQ to perform the cancellation, and then send the transaction to Salesforce Billing on an order through a cancel order product. The cancel order product represents the number of pending billings that you plan to cancel. (Salesforce Billing

Managed Package)

#### Convert a Negative Invoice Line to a Credit Note

When you amend an order to decrease the quantity of an order product, your next invoice often has a corresponding invoice line with a negative balance. Convert these negative invoice lines to credit notes and issue those credit notes as refunds or apply them to other invoice lines. (Salesforce Billing Managed Package)

## Cancel and Rebill an Invoice

Use the Cancel and Rebill option to revert the invoice to its prior state before the most recent billing cycle. Use this option to fix any errors in the invoice record. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 7.0 and later

---

You might run into mistakes on your invoice. For example, an order product didn't produce an invoice line, or your order had an incorrect billing address. To return your invoiced order products to the state before you most recently billed them, click **Cancel and Rebill** on your invoice. You can cancel and rebill draft and posted invoices.

 **Important** For audit trail purposes and data integrity issues, we recommend that invoices are never deleted. Instead, use the Cancel and Rebill option to cancel the invoice and reset the order products to be invoiced again.

When you click Cancel and Rebill, Salesforce Billing finds the order products related to the invoice's invoice lines, then rolls each order product's next billing date back to its value on the previous billing cycle. For example, your order product bills monthly with a billing day of month of 16. You cancel and rebill an invoice on 05/13, while the order product's next billing date is 05/16. The previous next billing date was 04/16. In this case, Salesforce Billing changes the order product's next billing date to 04/16. The order product's billing metric fields, such as Pending Billing Amount (without tax) and Billed Amount (without tax) also roll back to their previous values.

When the Cancel and Rebill process rolls back an order product's next billing date, the Override Next Billing Date and Bill Through Date Override fields are set to null. If they had values during the previous billing cycle that you want to use again, you must re-enter them.

Next, Salesforce Billing follows one of two processes based on the invoice's status.

 **Note** On a credit note, the credit note date and effective tax date are aligned with the date the invoice is canceled and rebilled. This approach ensures that a credit note isn't issued with a backdated entry.

## Draft Invoice

When you change an invoice's status to Canceled, it's no longer included in invoice and payment runs, and it can't be posted. Invoices that are in draft status show these changes:

- The invoice status flag changes to white.
- The invoice payment status changes to unpaid.

## Posted Invoice

Users post an invoice when they send it to a customer. At this stage, Salesforce Billing records the invoice's financial transactions in finance books and the general ledger. Since the invoice is a legal document, you must account for all transactions against it. When you cancel and rebill an invoice, Salesforce Billing:

- Creates a credit note on the invoice that matches the total balance of the invoice.
- Creates the credit note and adds credit note lines with balances equal to each of your invoice line balances.
- Allocates the full amount of each credit note line to the corresponding invoice line.
- Changes the invoice's status to Rebilled.

 **Note** Rebilled invoices aren't evaluated by invoice runs and payment runs, and can't be posted.

- Sets the invoice's status flag to green.
- Sets the invoice's accounts receivable (AR) status to Cancel and Rebill. This value doesn't perform any other actions on its own, but you can use it to note that the invoice was canceled and rebilled when you record the invoice's data in an external AR or general ledger platform.
- Changes the invoice's payment status to Paid.

If a posted invoice has collected payments, you must unallocate them completely before Salesforce Billing allows you to cancel and rebill. You can then reallocate the payments to the correct invoice lines when you create your invoice.

If a posted invoice can't be canceled and rebilled, or failed a cancel and rebill process because of CPU timeout, use a credit note, a debit note, or an additional invoice to make adjustments.

 **Note** The maximum number of invoice lines of a posted invoice that can be canceled and rebilled depends on factors that are mentioned in the [scale testing guidelines to cancel and rebill invoices](#). Even without automation, external tax integrations, or roll-up summaries to Account records, the maximum number of invoice lines that can be successfully canceled and rebilled is between 325 and 350.

You can also change your objects before billing the rolled-back order products again, such as updating tax rates or expanding the scope of a billing rule.

### [Add the Cancel and Rebill Button to Invoice Page Layouts](#)

Help users take control of their invoices when you add the Cancel and Rebill button to invoice page

layouts. (Salesforce Billing Managed Package)

#### Scale Testing Guidelines to Cancel and Rebill Invoices

Salesforce Billing can post invoices with more lines than the number of invoice lines that can be canceled and rebilled. So, scale test the Cancel and Rebill Invoices feature in a full sandbox to avoid errors when using the feature for posting invoices in production. (Salesforce Billing Managed Package)

### Add the Cancel and Rebill Button to Invoice Page Layouts

Help users take control of their invoices when you add the Cancel and Rebill button to invoice page layouts. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later

---

1. From Setup, in the Quick Find box, enter *Objects*, and then select **Objects**.
2. Click **Invoice**. Go to Page Layouts.
3. Find the invoice layout and click **Edit**.
4. Add the Cancel and Rebill button to page layouts in Salesforce Classic.
  - a. Select Buttons from the panel.
  - b. Move the Cancel and Rebill button to the Custom Buttons section.
5. Add the Cancel and Rebill button to page layouts in Lightning.
  - a. Select Mobile and Lightning Actions from the panel.
  - b. Move the Cancel and Rebill button to the Salesforce Mobile and Lightning Experience Actions section.

### Legacy Billing for Canceled Products

When you cancel a subscription, Salesforce Billing updates several objects so you can track how much revenue you've canceled. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later

---

### Subscription Products

When you cancel a subscription order product, Salesforce CPQ sets that order product's terminated date to the day you performed the cancellation. If the order product you canceled was a revised order product, Salesforce CPQ also sets the terminated dates of the original order product and any of its amendments.

When your invoice run evaluates a canceled subscription billed in arrears or without a billing type, it does not create invoice lines for any date past the termination date. If the canceled subscription is billed in

advance, Salesforce Billing makes a credit note for the period from the termination date to the day before the next charge date. Once all billing has completed for an order product with a terminated date, its Billing Status field updates to Canceled.

## Usage Products

When you cancel a usage-based order product, Salesforce Billing performs the following actions on all the product's usage summaries.

- If the terminated date is between the usage summary's start and end dates, Salesforce Billing updates the End Date Override field to match the termination date.
- If the terminated date is before the usage summary's start date, Salesforce Billing changes the usage summary's status to Canceled.

Salesforce Billing does not issue credits for usage that you already billed. Invoice runs do not evaluate canceled usage summaries.

 **Example** You have a six-month contract for a data security subscription that began on 01/01/17. The order product for this subscription bills for \$200 on the first day of the month. Your customer cancels their subscription effective 03/31/17. Let's look at how the cancellation affects your invoice and order product billing metrics.

Invoice

Field Name	Before Initial Billing	After January Invoice	After February Invoice	After March Invoice	After 03/31/17 Cancellation
Start Date	null	01/01/17	02/01/17	03/01/17	null (no future invoices)
End Date	null	01/31/17	02/28/17	03/31/17	null (no future invoices)
Amount	null	\$200	\$200	\$200	null (no future invoices)

Order Product 1

Field Name	Before Initial Billing	After January Invoice	After February Invoice	After March Invoice	After 03/31/17 Cancellation
Billed Amount	\$0	\$200	\$400	\$600	\$600
Pending Amount	\$1200	\$1000	\$800	\$600. After cancellation, this amount	\$0

Field Name	Before Initial Billing	After January Invoice	After February Invoice	After March Invoice	After 03/31/17 Cancellation
				moves to the Canceled Amount field.	
Canceled Amount	\$0	\$0	\$0	\$0	\$600
Contract Action	New	New	New	New	New

You also have a revised order product that represents the amendment you made to bring the first order product's quantity down to zero. The revised order product has the following fields.

#### Billed Amount

\$0

#### Pending Amount

\$0

#### Canceled Amount

\$-600. This value represents the total amount of your cancellation order.

#### Contract Action

Cancel

#### Original Order Product

Order Product 1



**Example** You have a six-month contract for a data security subscription that began on 01/01/17. The order product for this subscription bills for \$200 on the first day of the month. Your customer cancels their subscription effective 03/16/17. Since they canceled halfway through the month, we have to consider the prorated costs of a half-month of service.

#### Invoice

Field Name	Before Initial Billing	After January Invoice	After February Invoice	After March Invoice	After 03/31/17 Cancellation	Post-Cancellation Invoice
Start Date	null	01/01/17	02/01/17	03/01/17	null	03/16/17
End Date	null	01/31/17	02/28/17	03/31/17	null	03/31/17

Field Name	Before Initial Billing	After January Invoice	After February Invoice	After March Invoice	After 03/31/17 Cancellation	Post-Cancellation Invoice
Amount	null	\$200	\$200	\$200	null	\$-50

## Order Product 1

Field Name	Before Initial Billing	After January Invoice	After February Invoice	After March Invoice	After 03/31/17 Cancellation	Post-Cancellation Invoice
Billed Amount	\$0	\$200	\$400	\$600	\$600	\$600
Pending Amount	\$1200	\$1000	\$800	\$600	\$0	\$0
Canceled Amount	\$0	\$0	\$0	\$0	\$600	\$600
Contract Action	New	New	New	New	New	New

You also have a revised order product that represents the amendment you made to bring the first order product's quantity down to zero. Its billing metrics change between cancellation and your final invoice. The \$-50 in pending billing after cancellation represents the prorated credit for the March period that you already invoiced.

## Revised Order Product

Field Name	After 03/31/17 Cancellation	Post-Cancellation Invoice
Billed Amount	\$0	\$-50
Pending Amount	\$-50	\$0
Canceled Amount	\$-600	\$-600
Contract Action	Cancel	Cancel

## Cancellation Order Management

Your users have to cancel recurring or usage-based services in response to customer cancellation or as part of a collections process. You can use Salesforce CPQ to perform the cancellation, and then send the transaction to Salesforce Billing on an order through a cancel order product. The cancel order product represents the number of pending billings that you plan to cancel. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Spring '18 and later

-  **Note** To cancel a subscription bundle product that contains assets, the Asset Amendment Behavior on the asset product must be set to **Allow Refund**.

## Understand Cancellation Orders

Let's have a quick review on how subscription cancellation works in Salesforce CPQ. To terminate a subscription, make an amendment quote and then reduce the quantity of the related quote line to zero. When you order that amendment quote or opportunity and then contract the order, the corresponding subscription order product has a terminated date that's one day before the amendment quote's start date. We call this type of order product a Cancel Order Product. All Cancel Order Products have these values.

- Date selected in the Terminated Date field, which is filled after product contracting is complete.
- Contract Action field equals Cancel, which is set when the amendment quote is created.
- Revised Order Product field is complete, which occurs when the order product is created.
- Cancel Order Product contains a positive or negative Total Price field.

-  **Note** Orders can be created with or without quotes, so contracting isn't always supported or required.

A Cancellation Order refers to any order that contains at least one Cancellation Order Product.

Canceling recurring products impacts your billings as well. For example, let's say you cancel a 12-month subscription halfway through its billing cycle. You must track how much you've billed from the first six months and the number of billings that you're no longer receiving. All order products have a collection of fields that show the number of billings at various stages in the billing process.

### Pending Billing Amount (without tax)

Amount remaining to be billed, not including tax. Invoice plan order products don't populate this field.

### Canceled Billing Amount (without tax)

Total that isn't billed due to subscription cancellation or swapping, not including tax.

All order products begin with a canceled balance of zero. When you activate a related Cancel Order Product, Salesforce Billing moves an order product's pending billings to its canceled billings based on the Cancel Order Product's total balance. We review the logic behind this movement later in the article.

### Terminated Billing Amount (without tax)

The amount that Salesforce Billing could cancel from an order product. Terminated billings equal the larger of two possible values.

- The amount billed from the termination date to the order product end date
- The amount billed from the next charge date to the order product end date

Terminated billings can be less than or equal to the order product's pending billings, but never greater.

Terminated billings on a cancel order product are always zero.

 **Note** To review the billing types used in the billing process, check out [Billing Frequency](#).

Often, canceling a subscription leaves you with several order products with pending balances from earlier orders. You must make sure that an invoice run doesn't include these pending balances on a future invoice. Salesforce Billing uses a Cancel Order Product to reduce the pending balances of your original and amendment order products by an amount equal to the Cancel Order Product's total. This process ensures that you don't invoice for canceled billings.

## Cancel Order Scenarios

A quote can generate any number of amendments, and thus any number of orders. Order products can also be positive or negative—for example, a discount appears as a negative order product. Businesses encounter scenarios based on the number of previous orders, the positive or negative status of both original order products, and the Cancel Order Product. Let's consider a few.

- Positive order product and negative Cancel Order Product
- Negative order product and positive Cancel Order Product
- Several positive order products and one negative Cancel Order Product
- Several negative order products and one positive Cancel Order Product
- Positive and negative order products with a negative Cancel Order Product

Regardless of the scenario, Salesforce Billing aims to reduce the pending balance to zero on all the original order products. When you activate the Cancel Order Product, Salesforce Billing checks whether it has enough pending billings to reduce the value of all prior pending billings to zero. For example, let's say your original order product has a pending balance of \$150. The amending order product has a pending balance of \$75, and your Cancel Order Product has a pending balance of -\$300. Salesforce Billing zeroes out the prior pending billings and reduces your Cancel Order Product by \$225. It invoices the remaining -\$75 the next time you run an invoice or Bill Now activation picks up your Cancel Order Product for invoicing. Each order product's Canceled Billing Amount field lists the amount moved out of pending billings.

You find cases where both the prior order products and the Cancel Order Product end up with pending balances of zero. In this case, there's nothing left for Salesforce Billing to invoice. If your prior order products don't have any pending billings, Salesforce Billing doesn't allocate anything from your Cancel Order Product. And invoices it for its full pending amount on the next invoice run or Bill Now application.

## Cancellation Rule

Salesforce CPQ allows users to create multiple amendments off an order. Amending orders often creates

scenarios where one order product is related to several order products through several different amendments. In this case, if you have more than one prior order product, Salesforce Billing must determine which of your prior order products have pending balances of zero and which have a pending balance remaining.

Sometimes, your Cancel Order product isn't enough to reduce the pending balance on all prior order products to zero.

When you activate a Cancel Order Product in this scenario, Salesforce Billing checks the value of the Cancellation Rule field on the Cancel Order Product's billing treatment.

## Legacy

Salesforce Billing performs legacy cancellation allocation. You can review legacy cancellation allocation in [Legacy Billing for Canceled Products](#).

## LIFO

LIFO stands for "Last in, first out." In this case, it means that Salesforce Billing aims to cancel pending billings on the newest order product first. The Cancellation Rule contains several LIFO options, each with a different order product date field used to prioritize cancellation, such as termination date or order product activation date.

Salesforce Billing uses this logic only when your amending order products can't fully cancel the pending balances of the original order products. Otherwise, Salesforce Billing cancels the original order products and subtracts the sum of their pending balances from the Cancel Order Product's pending balance.

### **LIFO Order Product Cancellation**

Salesforce CPQ allows users to create multiple amendments from an order. When the amendments reduce the value of an order product, users encounter scenarios where one order product is related to several order products through several different amendments. Sometimes, your amending order products (including the Cancel Order Product) aren't enough to completely cancel the pending balances of the original order products. In this case, Salesforce Billing uses the Cancel Order Product's cancellation rule to determine how to cancel these pending balances. (Salesforce Billing Managed Package)

### **Use Cases: Cancel Order Product Fully Covers Pending Billings**

Sometimes, a Cancel Order Product has a pending balance remaining after it completely cancels all prior order products. In this case, Salesforce Billing picks the Cancel Order Product's pending balance up on the next invoice run. (Salesforce Billing Managed Package)

### **Use Cases: All Pending Billings Reduced to Zero**

Sometimes, you'll have multiple order products and a Cancel Order Product with pending billings that sum to zero. After you activate the Cancel Order Product, Salesforce Billing cancels all pending billings and doesn't have anything left to invoice. (Salesforce Billing Managed Package)

### **Invoicing Canceled Evergreen Subscriptions**

When you cancel an evergreen order product, Salesforce Billing evaluates its billable unit price to create the Cancel Order Product. It then calculates pending billings for either the original product or

the Cancel Order Product based on the termination date. (Salesforce Billing Managed Package)

## LIFO Order Product Cancellation

Salesforce CPQ allows users to create multiple amendments from an order. When the amendments reduce the value of an order product, users encounter scenarios where one order product is related to several order products through several different amendments. Sometimes, your amending order products (including the Cancel Order Product) aren't enough to completely cancel the pending balances of the original order products. In this case, Salesforce Billing uses the Cancel Order Product's cancellation rule to determine how to cancel these pending balances. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

 **Note** In Salesforce Billing Summer '19, we changed the value name "LIFO based on Termination Date" to "LIFO by Order Product Creation Date." The value's function remains the same. The new value name is available by default for users whose first Salesforce Billing install was Summer '19 or newer. Users who upgraded from a version before Summer '19 can change the value's name to "LIFO by Order Product Creation Date."

When Salesforce Billing contracts a Cancel Order Product covered by a Cancellation Rule set to any LIFO value, it performs these steps.

1. Sum the value of terminated billing amounts for the original order products. We call this number the original balance.
2. Sum the value of terminated billing amounts for the amending order products and the Cancel Order Products. We call this number the canceling balance.
3. Evaluate whether the canceling balance is larger, smaller, or equal to the original balance. Remember, positive order products have negative amending order products, so Salesforce Billing ignores whether a balance is positive or negative during this evaluation.
  - If the canceling balance is larger, the Cancel Order Product reduces its pending balance by the sum of all other pending balances. All other order products reduce their pending balances to zero. Salesforce Billing picks up the Cancel Order product's remaining pending balance on the next invoice run.
  - If the canceling balance equals the original balance, Salesforce Billing reduces pending billings on all order products to zero. Since none of the order products have a pending balance, Salesforce Billing doesn't pick them up on the next invoice run.
  - If the canceling balance is smaller, Salesforce Billing determines how to allocate the canceling balance among the original and amending order products. In this case, it checks the value of the Cancel Order Product's cancellation rule. If the value is LIFO, Salesforce Billing cancels the amending order products first. It then applies the canceling balance to the original order products starting with the newest LIFO date (such as terminated date or order product creation date) and moving to the order product with the oldest LIFO date.



**Example** Let's look at an example of a cancel billing scenario where the amending order products can't completely cancel the original order product. The Cancel Order Product's Cancellation Rule has a value of LIFO based on Order Product Creation Date. So, we cancel the newest order product first in this case the amending order product. Salesforce Billing applies the US\$125 canceling balance to the original order product, reducing the original order product's pending billings to -\$25, which Salesforce Billing picks up on the next invoice run. The Canceled Billing Amount fields reflect the amount canceled from each order product.



## Use Cases: Cancel Order Product Fully Covers Pending Billings

Sometimes, a Cancel Order Product has a pending balance remaining after it completely cancels all prior order products. In this case, Salesforce Billing picks the Cancel Order Product's pending balance up on the next invoice run. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

## Cancel Order Product > Pending Billings

Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	\$600	\$450	\$150	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1	\$200	\$125	\$75	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>

Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	-\$300	\$0	-\$300	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>

In this case, the Cancel Order Product can reduce the net prior pending billings to \$0. Salesforce Billing adds the prior pending billings to the Cancel Order Product's pending billings, then cancels all pending

billings on the original order products. This leaves the Cancel Order Product with a pending balance of -\$75, which Salesforce Billing picks up with the next invoice run.

After Cancel Order Product Activation

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	\$600	\$450	\$0	\$150	<ul style="list-style-type: none"> <li>Next Billing Date: null</li> <li>Next Charge Date: null</li> <li>Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1 Result	\$200	\$125	\$0	\$75	<ul style="list-style-type: none"> <li>Next Billing Date: null</li> <li>Next Charge Date: null</li> <li>Terminated Date: 10/01/17</li> </ul>
Cancel Order Product Result	-\$300	\$0	-\$75	-\$225	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>

Positive Original Order Product, Negative Amending Order Product

Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	\$600	\$450	\$150	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1	-\$200	-\$125	-\$75	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>

Introduction of Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	-\$200	\$0	-\$200	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>

In this case, the Cancel Order Product can reduce the net prior pending billings to \$0. Salesforce Billing adds the prior pending billings to the Cancel Order Product's pending billings, then cancels all pending billings on the original order products. This leaves the Cancel Order Product with a pending balance of -\$125, which Salesforce Billing picks up on the next invoice run.

#### Cancel Order Logic Applied

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	\$600	\$450	\$0	\$150	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1 Result	-\$200	-\$125	\$0	-\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Cancel Order Product Result	-\$200	\$0	-\$125	-\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

#### No Prior Pending Billings

##### Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	-\$600	-\$600	\$0	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1	-\$200	-\$200	\$0	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>

##### Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	\$225	\$0	\$225	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> </ul>

Order Product	Total	Billed	Pending	Canceled	Dates
					<ul style="list-style-type: none"> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

When a user or process activates this Cancel Order Product, Salesforce Billing determines that none of the original order products have pending billings. Therefore, there's no need to apply any billings from the Cancel Order Product. All order products remain unchanged, and the Cancel Order Product's pending \$225 will be picked up during the next invoice run or Bill Now application. After invoicing, the Cancel Order Product's pending billings value moves to its billed amount.

#### After Cancel Order Product Activation

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	-\$600	-\$600	\$0	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1 Result	-\$200	-\$200	\$0	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Cancel Order Product Result	\$225	\$0	\$225	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

#### Negative Original Order Product > Pending Billings

##### Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	-\$600	-\$450	-\$150	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1	-\$200	-\$125	-\$75	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> </ul>

Order Product	Total	Billed	Pending	Canceled	Dates
					<ul style="list-style-type: none"> <li>Terminated Date: 10/01/17</li> </ul>

#### Introduction of Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	\$300	\$0	\$300	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>

In this case, the Cancel Order Product can reduce the net prior pending billings to \$0. Salesforce Billing adds the prior pending billings to the Cancel Order Product's pending billings, then cancels all pending billings on the original order products. This leaves the Cancel Order Product with a pending balance of \$75, which Salesforce Billing picks up on the next invoice run.

#### Cancel Order Logic Applied

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	-\$600	-\$450	\$0	-\$150	<ul style="list-style-type: none"> <li>Next Billing Date: null</li> <li>Next Charge Date: null</li> <li>Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1 Result	-\$200	-\$125	\$0	-\$75	<ul style="list-style-type: none"> <li>Next Billing Date: null</li> <li>Next Charge Date: null</li> <li>Terminated Date: 10/01/17</li> </ul>
Cancel Order Product Result	\$300	\$0	\$75	\$225	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>

## Negative Order Product and Positive Amending Order Product, Cancel Order Product > Prior Pending Billings

Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	-\$600	-\$450	-\$150	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1	\$200	\$125	\$75	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

Introduction of Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	\$200	\$0	\$200	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

In this case, the Cancel Order Product can reduce the net prior pending billings to \$0. Salesforce Billing adds the prior pending billings to the Cancel Order Product's pending billings, then cancels all pending billings on the original order products. This leaves the Cancel Order Product with a pending balance of \$125, which Salesforce Billing picks up on the next invoice run.

Cancel Order Logic Applied

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	-\$600	-\$450	\$0	-\$150	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1 Result	\$200	\$125	\$0	\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product Result	\$200	\$0	\$125	\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

## Negative Order Product, Positive and Negative Amending Order Products

Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	-\$600	-\$450	-\$150	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1	-\$200	-\$125	-\$75	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #2	\$225	\$90	\$135	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

Introduction of Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	\$150	\$0	\$150	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

In this case, the Cancel Order Product can reduce the net prior pending billings to \$0. Salesforce Billing adds the prior pending billings to the Cancel Order Product's pending billings, then cancels all pending billings on the original order products. This leaves the Cancel Order Product with a pending balance of \$60, which Salesforce Billing picks up on the next invoice run.

## Cancel Order Logic Applied

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	-\$600	-\$450	\$0	-\$150	<ul style="list-style-type: none"> <li>Next Billing Date: null</li> <li>Next Charge Date: null</li> <li>Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1 Result	-\$200	-\$125	\$0	-\$75	<ul style="list-style-type: none"> <li>Next Billing Date: null</li> <li>Next Charge Date: null</li> <li>Terminated Date: 10/01/17</li> </ul>
Amending Order Product #2 Result	\$225	\$90	\$0	\$135	<ul style="list-style-type: none"> <li>Next Billing Date: null</li> <li>Next Charge Date: null</li> <li>Terminated Date: 10/01/17</li> </ul>
Cancel Order Product Result	\$150	\$0	\$60	\$90	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 10/01/17</li> </ul>

## Cancel Order Product &gt; Pending Billings, Backdated Cancellation

## Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	\$600	\$450	\$150	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 09/01/17</li> </ul>
Amending Order Product #1	\$200	\$125	\$75	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 10/01/17</li> <li>Next Charge Date: 10/01/17</li> <li>Terminated Date: 09/01/17</li> </ul>

## Introduction of Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	-\$300	\$0	-\$300	\$0	<ul style="list-style-type: none"> <li>Next Billing Date: 09/01/17</li> </ul>

Order Product	Total	Billed	Pending	Canceled	Dates
					<ul style="list-style-type: none"> <li>• Next Charge Date: 09/01/17</li> <li>• Terminated Date: 09/01/17</li> </ul>

In this case, the Cancel Order Product can reduce the net prior pending billings to \$0. Salesforce Billing adds the prior pending billings to the Cancel Order Product's pending billings, then cancels all pending billings on the original order products. This leaves the Cancel Order Product with a pending balance of -\$75, which Salesforce Billing picks up on the next invoice run.

For remaining order products with a pending balance, Salesforce Billing sets the next billing date to the order product's terminated date. The next invoice run will pick up this balance for invoicing.

Cancel Order Logic Applied

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	\$600	\$450	\$0	\$150	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 09/01/17</li> </ul>
Amending Order Product #1 Result	\$200	\$125	\$0	\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 09/01/17</li> </ul>
Cancel Order Product Result	-\$300	\$0	-\$75	-\$225	<ul style="list-style-type: none"> <li>• Next Billing Date: 09/01/17</li> <li>• Next Charge Date: 09/01/17</li> <li>• Terminated Date: 09/01/17</li> </ul>

## Use Cases: All Pending Billings Reduced to Zero

Sometimes, you'll have multiple order products and a Cancel Order Product with pending billings that sum to zero. After you activate the Cancel Order Product, Salesforce Billing cancels all pending billings and doesn't have anything left to invoice. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

## Canceled Order Product = Pending Billings

Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	\$600	\$450	\$150	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1	\$200	\$125	\$75	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

Introduction of Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	-\$225	\$0	-\$225	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

The cancel order product's total and the pending values of the original order products have a combined value of \$0. Therefore, Salesforce Billing doesn't perform further billing. After a user or process activates the Cancel Order Product, Salesforce Billing moves the pending values of all order products to the canceled values. Then, the pending values become \$0.

Cancel Order Logic Applied

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	\$600	\$450	\$0	\$150	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1 Result	\$200	\$125	\$0	\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Cancel Order Product Result	-\$225	\$0	\$0	-\$225	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> </ul>

Order Product	Total	Billed	Pending	Canceled	Dates
					<ul style="list-style-type: none"> <li>• Terminated Date: 10/01/17</li> </ul>

Negative Original Order Product = Pending Billings

Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product, 01/01/17 - 12/31/17	-\$600	-\$450	-\$150	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>
Original Order Product #2	-\$200	-\$125	-\$75	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

Introduction of Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	\$225	\$0	\$225	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

Remember, a negative order product creates a positive Cancel Order Product. When a user or process activates this Cancel Order Product, Salesforce Billing determines that the Cancel Order Product and the pending billings of the original order products add up to zero. Therefore, no further billing is performed, all order product canceled billings receive the value of their pending billings, and all pending billings receive a value of zero.

Cancel Order Logic Applied

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	-\$600	-\$450	\$0	-\$150	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>

Order Product	Total	Billed	Pending	Canceled	Dates
Amending Order Product #1 Result	-\$200	-\$125	\$0	-\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Cancel Order Product Result	\$225	\$0	\$0	\$225	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>

Negative Order Product and Positive Amending Order Product, Cancel Order Product and Net Prior Pending Billings = 0

#### Original Order Products

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product	-\$600	-\$450	-\$150	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1	\$200	\$125	\$75	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

#### Introduction of Cancel Order Product

Order Product	Total	Billed	Pending	Canceled	Dates
Cancel Order Product	\$75	\$0	\$75	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

Since the Cancel Order Product has a greater value than prior pending billings, Salesforce Billing reduces the Cancel Order Product's pending balance by the sum of prior pending billings. It then cancels all prior pending billings.

### Cancel Order Logic Applied

Order Product	Total	Billed	Pending	Canceled	Dates
Original Order Product Result	-\$600	-\$450	\$0	-\$150	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Amending Order Product #1 Result	\$200	\$125	\$0	\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: null</li> <li>• Next Charge Date: null</li> <li>• Terminated Date: 10/01/17</li> </ul>
Cancel Order Product Result	\$75	\$0	\$0	\$75	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> <li>• Terminated Date: 10/01/17</li> </ul>

## Invoicing Canceled Evergreen Subscriptions

When you cancel an evergreen order product, Salesforce Billing evaluates its billable unit price to create the Cancel Order Product. It then calculates pending billings for either the original product or the Cancel Order Product based on the termination date. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce CPQ Spring '19 and later

---

For a general overview of cancellation orders, check out [Cancellation Order Management](#).

Salesforce Billing handles evergreen cancellations differently based on whether your order product's charge date falls before or after its terminated date.

## Next Charge Date Before Terminated Date

Salesforce Billing charges your customer on the original order product for the billing periods between the two dates.

This scenario occurs when a customer wants to cancel their evergreen subscription as of a predetermined time in the future. In this case, you'll make an amendment quote and set its start date to the date that the cancellation takes effect. After you contract your opportunity and order the amendment quote, your Cancel Order Product shows a total amount that's the opposite of your original order product's total.

For example, let's say your evergreen subscription bills monthly for \$10 at the first of the month, starting 01/01/17. At the end of May, your customer decides to cancel their subscription as of October 1. You amend your quote, set the amendment quote's start date to 10/1/2017, then contract and order it.

Order Product	Total	Billed	Pending	Canceled	Dates
Original	\$10	\$50	\$0	\$0	
Cancel	-\$10	\$0	\$0	\$0	

When you activate the Cancel Order Product, Salesforce Billing calculates the pending billings between the original order product's next charge date and its terminated date. Since there are four billing periods between 06/01/2017 and 09/31/2017, Salesforce Billing multiplies 4 by the billable unit price of \$10 to calculate the original order product's Pending Billings. This represents the amount you'll need to charge your customer between June 1 and September 31.

Order Product	Total	Billed	Pending	Canceled	Dates
Original	\$10	\$50	\$40	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 01/01/17</li> <li>• Next Charge Date: 01/01/17</li> <li>• Terminated Date: 09/31/17</li> </ul>
Cancel	-\$10	\$0	\$0	\$0	Terminated Date: 09/31/17

Since you're billing monthly, Salesforce continues to invoice your customer for \$10 per month until the terminated date.

#### 06/01/17 through 06/30/17

Invoice Line Total: \$10

#### 07/01/17 through 07/31/17

Invoice Line Total: \$10

#### 08/01/17 through 08/31/17

Invoice Line Total: \$10

**09/01/17 through 09/30/17**

Invoice Line Total: \$10

You could also change your target date to September 30 to bill for the entire pending \$40 on one invoice.

## Next Charge Date After Terminated Date

In this case, your customer has already paid for at least one billing period and you need to refund them for the periods between terminated date and the next charge date.

For example, let's say your evergreen subscription bills monthly for \$10 at the first of the month, starting 01/01/17. Your customer wanted to pay in advance, so you've billed them through the end of the year. However, they decide they want to cancel their contract as of 10/01/17.

In this case, you'll make an amendment quote and set its start date to the date that the cancellation should take effect. After you contract your opportunity and order the amendment quote, Salesforce CPQ creates a Cancel Order Product with a total that represents one canceled billing period.

Order Product	Total	Billed	Pending	Canceled	Dates
Original	\$10	\$120	\$0	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 01/01/18</li> <li>• Next Charge Date: 01/01/18</li> </ul>
Cancel	-\$10	\$0	\$0	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> </ul>

When you activate your cancel order product, Salesforce Billing calculates its total pending billings. This value equals the inverse sum of billing period charges between the Cancel Order Product's terminated date and the original order product's next charge date. Since you have three months between these dates at a rate of -\$10 per month, you have -\$30 in pending billings.

Order Product	Total	Billed	Pending	Canceled	Dates
Original	\$10	\$120	\$0	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 01/01/18</li> <li>• Next Charge Date: 01/01/18</li> </ul>
Cancel	-\$10	\$0	-\$30	\$0	<ul style="list-style-type: none"> <li>• Next Billing Date: 10/01/17</li> <li>• Next Charge Date: 10/01/17</li> </ul>

Since you're canceling the order after the customer has made several payments, Salesforce Billing considers the original order product completed and won't invoice it anymore. Salesforce Billing picks up your Cancel Order Product with the next invoice run or Bill Now process.

## Convert a Negative Invoice Line to a Credit Note

When you amend an order to decrease the quantity of an order product, your next invoice often has a corresponding invoice line with a negative balance. Convert these negative invoice lines to credit notes and issue those credit notes as refunds or apply them to other invoice lines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

You can also use REST API to evaluate several invoices at once and automatically generate the necessary credit notes. For more information, review [REST API for Converting Invoice Lines with Negative Balances](#) in the Salesforce Billing Developer Guide.

If you invoice an amended order product that has a negative price, the resulting invoice line has a negative balance. For example, let's say you amended an order to reduce the quantity of your SaaS subscriptions from 4 to 3. The reduction created an amendment order product with a price of -\$100. If your billing setup invoices the order product as one invoice line without any price changes, it will appear as an invoice line with an amount of -\$100 on your next invoice.

In this scenario, we'll assume that your invoice has an invoice line with a balance of -\$500 and an invoice line with a balance of -\$200.

1. Go to your invoice and click **Convert Negative Lines**.

Salesforce Billing displays the Create Credit Note From Negative Lines page. The page displays all your invoice's lines with a negative balance.

 **Note** After you click **Convert Negative Lines**, the Cancel and Rebill action on the invoice is disabled.

2. Select the negative invoice lines you want to convert. Click **Convert**.

3. Salesforce Billing creates a credit note containing a credit note line with a balance of \$500 and a credit note line with a balance of \$200. You can access this credit note on your invoice's Credit Notes related list.

## Aligning Proration Between CPQ and Billing

When you use CPQ and Billing together, we recommend aligning CPQ's Subscription Proration Precision with Billing's Proration Type. Otherwise, the proration methods can cause unwanted differences between an invoice line's balance and the customer's expected billings based on the product's price. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 212.5 and later

---

Let's start out by examining an order product billed \$1000 monthly in arrears for a 12-month term. CPQ uses Monthly + Daily proration settings, while Billing uses Calendar Days. A sales rep quotes it for 04/23 through 09/30/19, then orders and invoices it.



In (1), the first billing period in April is prorated to run from 04/23/19 through 04/30/19. This period represents one difference in proration between CPQ and Billing - when CPQ calculated proration, it counted forward by month starting on 04/23/19. We moved through 7 whole months and used 09/23 through 09/30 as our final period for calculating proration. However, Billing prorates based on when the order product invoices: Since we're billing monthly in advance, our first period for proration is 04/23 through 04/30.

Moving through the next four full months of invoicing at \$1000 per month, we arrive at a total of \$4,266.67 (2). The final billing period of 09/01 through 09/30 must cover the order product's remaining balance, so the invoice line has a total balance of \$996.34 (3).

However, this balance represents the discrepancy from the current proration configurations: The product had a monthly price of \$1000, yet the billing is at \$996.34 for the full month of September. Even though the invoice lines add up to \$5,263.01 like the order product's total, most customers would expect to pay the product's price of \$1000 per month in a full, non-prorated month.

## Alignment Option 1

Our best option for alignment is to continue using Monthly + Daily proration in CPQ while changing our Billing proration type to Monthly (CPQ Formula). While this value is active, Billing uses  $(365/12)$  as the length of a month. Let's see what happens Months (CPQ Method) is active in our same example.



In this case, the first period (1) has a prorated balance of \$263.01. The next four full months bring the total to \$4263.01. Now, the remaining balance in September (2) is \$1000, aligning with the expected charge based on the product's original price of \$12000 over 12 months.

## Aligning for Terms with Start and End Months of Different Lengths

Make sure to account for differences in month lengths when configuring proration between CPQ and Billing. If the first and last billing periods are in months with different lengths, the final billing period still differs from the product's monthly price even when proration methods are aligned.

Let's return to the first example of the same product, but quoted from 05/23/19 through 09/30/19. In this case, the initial billing period lasts for 9/31 days, producing a slightly larger value than 8/30. This larger billing period, alongside the shorter final month length in September, means that the final billing period has a smaller remaining balance to invoice.



When we change Salesforce Billing's prorate type to CPQ Method, our final invoice line has a slightly larger difference from the monthly rate of \$1000.



## Align CPQ and Billing Cancellation Based on Billing Periods

Salesforce CPQ and Salesforce Billing use different formulas to calculate proration periods. Depending on your order product's billing fields, Salesforce Billing sometimes splits your order product's overall term into more or fewer proration periods than were used in CPQ. When you cancel an order product when CPQ and Billing use different proration periods, a pending balance sometimes remains. If you want to avoid pending balances when you cancel an order product, align CPQ and Billing to use the same proration periods. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

Based on your proration settings, Salesforce CPQ calculates proration periods based on calendar month. Salesforce Billing calculates proration periods based on the billing period. Salesforce Billing proration

periods can vary based on order product billing fields such as billing day of month billing frequency.

These methods can cause differences in prorated period length between the two packages. Period variations cause differences in amended or canceled order amounts and amended or canceled billing amounts.

For example, let's say you quote a monthly subscription product with a term of 01/16/20 through 04/15/20 and then order and invoice it. CPQ has subscription prorate precision set to Calendar Month + Daily, while Billing has prorate precision set to Calendar Days.

Salesforce Billing calculates periods based on the order product's Next Billing Date. Next Billing Date can vary based on the order product's billing day of month, next billing date, billing frequency, and Override Next Billing Date. If your order product has a Billing Day of Month of 7, CPQ and Billing use the following periods.

Period	CPQ (Calendar Monthly + Daily)	Billing (Calendar Days)
1	January 16–January 31	January 16–February 6
2	February 1–February 29	February 7–March 6
3	March 1–March 31	March 7–April 6
4	April 1–April 15	April 7–April 15

The billing period differences sometimes cause variations between canceled order amounts and canceled billing amounts when you cancel an order product. For example, let's say our order product has a list price of \$1,200. Then, the customer cancels their order product for 03/01.

#### Original Order Product Price Calculation

With Calendar Month + Daily proration, our prorate multiplier is  $((16 \div 31) + 1 + 1 + (15 \div 30)) \div 12 = 0.251344$ .

Our order product's list price is \$1200, so the unit price is  $\$1200 * 0.2513 = \$301.61$ . Our order product's billable unit price is \$100.

#### Billing Field Calculation

Salesforce Billing calculates the Billed Amount for the first invoice based on the 22-day billing period of January 16 through February 6. Salesforce Billing uses Calendar Days proration to calculate the invoice line's subtotal, so the first month's subtotal is  $(22 \div 31) * 1 * \$100 = \$70.97$ .

Because the second invoice covers all of February, the subtotal equals the invoice line unit price of \$100.

### Original Order Product

Order Product Fields	Before First Invoice	After First Invoice	After second Invoice	After Cancellation
Billed Amount	Null	\$70.97	\$170.97	\$170.97
Total Billings	\$0	\$70.97	\$170.97	\$170.97
Pending Billing Amount	\$301.61	\$230.64	\$130.64	\$0.64
Canceled Billing Amount	\$0	\$0	\$0	\$170.97

Next, let's cancel the order product. The cancellation process covers the pending billing amount as of 03/01, which is \$130.64.

### Cancellation Order Product Price Calculation

The cancellation order product covers the period of 03/01 through 04/15. The proration multiplier is  $(1 + (15 \div 30)) \div 12 = 0.124429$ .

The cancellation order product's prorated unit price is  $\$1200 * 0.124429 = \$130.64$ .

The cancellation order product's canceled billing amount is -\$130.00.

### Cancellation Order Product

Order Product Fields	After Cancellation
Billed Amount	Null
Total Billings	\$0
Pending Billing Amount	-\$0.64
Canceled Billing Amount	-\$130.00

When Salesforce Billing invoices the cancellation order product, the customer receives an invoice for -\$0.64, which they can convert into a credit of \$0.64.

CPQ and Billing used different billing periods, so your cancellation order product had a pending balance, even though you canceled the remaining period of 03/01 through 04/15. These pending balances can be positive or negative depending on your billing amounts. To cancel an order product and get no pending balances, you can align CPQ and Billing to use the same billing periods.

### Aligning with Proration Day of Month

Salesforce CPQ Winter '21 added the Proration Day of Month (Calendar Monthly + Daily) to the Subscription Prorate Precision package setting. We also added a Proration Day of Month field to the

quote. While Proration Day of Month (Calendar Monthly + Daily) is active, Salesforce CPQ calculates proration periods based on quote's proration day of month. The setting uses the same formula as Calendar Month + Daily to calculate the prorate multipliers.

The proration day of month inherits its value from the quote's start date by default, but you can override it. Let's repeat the original example but change Subscription Prorate Precision to Proration Day of Month (Calendar Monthly + Daily) and set the quote's Proration Day of Month to 7. When you order the quote, the order inherits the proration day of month in the billing day of month field. This configuration allows CPQ and Billing to use the same boundary to calculate proration periods.

Billing Period	CPQ: Proration Day of Month (Calendar Monthly + Daily)	Billing: Calendar Days
1	January 16–February 6	January 16–February 6
2	February 7–March 6	February 7–March 6
3	March 7–April 6	March 7–April 6
4	April 7–April 15	April 7–April 15

### Original Order Product Price Calculation

With Proration Day of Month (Calendar Monthly + Daily), use the same prorate multiplier calculation formula as Calendar Month + Daily. But the different billing periods result in a slightly different prorate multiplier of  $((22 \div 31) + 1 + 1 + (9 \div 30)) \div 12 = 0.250806$ .

The order product's list price is \$1,200, so the unit price is  $\$1,200 * 0.250806 = \$300.97$ . Then, the order product's billable unit price is \$100.

### Billing Fields Calculation

Salesforce Billing calculates the Billed Amount for the first invoice based on the 22-day billing period of January 16 through February 6. Salesforce Billing uses Calendar Days proration to calculate the invoice line's subtotal, so the first month's subtotal is  $(22 \div 31) * 1 * \$100 = \$70.97$ .

Because the second invoice covers all of January, the subtotal equals the invoice line unit price of \$100.

Order Product Fields	Before First Invoice	After First Invoice	After second Invoice	After Cancellation
Billed Amount	Null	\$70.97	\$170.97	\$170.97
Total Billings	\$0.00	\$70.97	\$170.97	\$170.97
Pending Billing Amount	\$300.97	\$230.00	\$130.00	\$0.00

Order Product Fields	Before First Invoice	After First Invoice	After second Invoice	After Cancellation
Canceled Billing Amount	\$0.00	\$0.00	\$0.00	\$130.00

Next, let's cancel the order product. The cancellation process covers the pending billing amount as of 03/01, which is \$130.

#### Cancellation Order Product Price Calculation

The cancellation order product covers the period of 03/01 through 04/15. The proration multiplier is 0.1083.

The cancellation order product's prorated unit price is \$130.

The cancellation order product's canceled billing amount is -\$130.

Cancellation Order Product

Order Product Fields	After Cancellation
Billed Amount	Null
Total Billings	\$0
Pending Billing Amount	-\$0
Canceled Billing Amount	-\$130.00

Because the proration periods are aligned, the cancellation order product has no pending billings.

-  **Warning** If a customer overrides your billing period, you may still see differences in amendment and cancellation amounts between CPQ and Billing. Billing periods can be overridden by using a billing day of month that differs from CPQ's proration day of month, or by setting a Bill Through Override Date.

#### Guidelines for Using Proration Day of Month

If you use the quote's proration day of month to align CPQ and Billing proration periods, review important guidelines for the Proration Day of Month field. (Salesforce Billing Managed Package)

#### Guidelines for Using Proration Day of Month

If you use the quote's proration day of month to align CPQ and Billing proration periods, review important guidelines for the Proration Day of Month field. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Winter '21 and later

- The Proration Day of Month field must be part of the quote's Calculating Fields field set. Otherwise, Salesforce CPQ doesn't recalculate the quote's proration periods and prorated prices when a user or process changes the proration day of month.
- Salesforce CPQ populates the quote's proration day of month only while the subscription prorate precision package setting is Proration Day of Month (Calendar Monthly + Daily).
- If your subscription prorate precision is a value other than Proration Day of Month (Calendar Monthly + Daily), you can still provide the quote's proration day of month on your own. In this case, the order's billing day of month still inherits the proration day of month upon order creation.
- You can change the proration day of month on your quote at any time. When you order your quote, the order's billing day of month inherits the value of the quote's proration day of month upon order creation. However, further changes to the proration day of month don't affect the billing day of month.
- If a user or process changes the quote's proration day of month from a non-null value to null while Proration Day of Month is in the quote's Calculating Fields field set, Salesforce CPQ sets the proration day of month to the quote's start date.

## Troubleshooting Proration Issues

When you're working with proration in CPQ and Billing, review some important guidelines to ensure that your proration values align and deliver expected results. (Salesforce Billing Managed Package)

When you invoice a quote, you typically want the invoice's total to align with the quote and order's total. Misalignment between quote and invoice balances happens for three reasons.

- Rounding when an invoice line contains a fraction of a cent
- Differences between CPQ and Billing proration settings
- Changing any billing-date-related order field or order product field, such as billing day of month or billing frequency

Let's see how you can manage each.

## Rounding for Fractional Currency Values

If Salesforce Billing encounters a repeating decimal when dividing an order total into invoices over a set of billing periods, it rounds the repeating value to two decimal places and then adds any remaining balances in the final periods. During rounding, the third decimal is used to determine whether the second decimal is rounded up or down. The last invoice then has a different balance to account for the previous rounding, which ensures that the order product is still billed in its entirety. This process can't be changed or overwritten. Here are two examples.

### \$100 billed monthly for 12 months

Rounded down because 8.333 is closest to 8.33

Invoice balance for first 11 months: \$8.33

Invoice balance for 12th month: \$8.37

### \$104 billed monthly for 12 months

Rounded up because 8.666 is closest to 8.67

Invoice balance for first 11 months: \$8.67

Invoice balance for 12th month: \$8.63

## CPQ and Billing Proration Settings

CPQ and Billing package settings each contain several options for proration calculation formulas. Different combinations of the settings may produce variations in how your balances are distributed between orders and invoices. We recommend two ways as the most effective options for aligning proration between the packages.

### Option 1

- CPQ: Proration Day of Month (Calendar Monthly + Daily)
- Billing: Proration Day of Month

While Proration Day of Month (Calendar Monthly + Daily) is active in CPQ, Salesforce CPQ calculates proration periods based on the quote's proration day of month. The setting uses the same formula as Calendar Month + Daily to calculate the prorate multipliers. When you order the quote, the order inherits the proration day of month in the billing day of month field. This configuration allows CPQ and Billing to use the same boundary to calculate proration periods.

For more information on using the Proration Day of Month field, review [Align CPQ and Billing Cancellation Based on Billing Periods](#).

For a line-level review of proration differences and alignment, review [Aligning Proration Between CPQ and Billing](#).

### Option 2

- CPQ: Monthly + Daily
- Billing: Monthly (CPQ Formula)

This option is helpful for companies that align the effective date of their amendments to the beginning of the billing term.

## Changing Date-Related Order or Order Product Fields

Salesforce Billing proration formulas evaluate a set of order and order product fields to calculate proration periods, which are then used to calculate invoice balances. If you change the default values of

any of these fields, your quote and invoice totals can misalign, even if you aligned your CPQ and Billing proration settings.

Changing important billing fields on orders and order products can cause your quote and invoice balances to misalign, even if you aligned your CPQ and Billing proration settings.

 **Example** For example, consider a quote line for a monthly MDM Subscription with a list price of \$12,000 for a one-year subscription term. A sales rep negotiates a shortened term of 04/23/21 through 09/30/21 with a customer and then adds it to their quote. The prorated list price is \$5,263.01.

You've set up automation that sets the order's billing day of month to 1 by default. As Salesforce Billing invoices the order, your invoice lines for each month's invoice have the following values.

- 04/23-04/30: \$266.67
- 05/01-05/31: \$1,000
- 06/01-06/30: \$1,000
- 07/01-07/31: \$1,000
- 08/01-08/31: \$1,000
- 09/01-09/30: \$996.34

However, let's say you change the billing day of month to 15 after the first invoice and before the second invoice. While your total billed is the same, this change affects how the total amount is split up between each invoice.

- 04/23-04/30: \$266.67
- 05/01-05/14: \$460.27
- 05/15-6/14: \$1,000
- 06/15-07/14: \$1,000
- 07/15-08/14: \$1,000
- 08/15-09/14: \$1,000
- 09/15-09/30: \$536.07

Changing your order's start date also affects how your invoice total is divided among billing periods. Let's return to our MDM subscription with a term of 04/23/21 through 09/30/21, where the billing day of month is 1. If you change the order's start date to 05/23/21, Salesforce pushes the end date to 10/30/21. However, May has 31 days, as opposed to 30 in April, so the first billing period covers one extra day. Let's look at the new periods. Also remember that Salesforce CPQ adjusts the end date out so that the order covers the same amount of time.

- 05/23-05/31: \$295.89
- 06/01-06/30: \$1,000
- 07/01-07/31: \$1,000
- 08/01-08/31: \$1,000
- 09/01-09/31: \$1,000
- 10/01-10/31: \$967.12

## CPQ Billing Field Mapping

Salesforce CPQ includes several important billingfields that map between products, quotes, orders, quote lines, and order products. Review these fields to understand data flow and the implications of overriding fields. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Product to Quote Line

Field Name	API Name
Charge Type	SBQQ__ChargeType__c
Billing Frequency	SBQQ__BillingFrequency__c
Billing Type	SBQQ__BillingType__c

#### Quote to Order

Field Name	API Name
Payment Terms	<ul style="list-style-type: none"> <li>• SBQQ__PaymentTerms__c (quote)</li> <li>• SBQQ__PaymentTerm__c (order)</li> </ul>

#### Quote Line to Order Product

Field Name	API Name
Default Subscription Term	SBQQ__DefaultSubscriptionTerm__c
Subscription Term	SBQQ__SubscriptionTerm__c
Billing Type	SBQQ__BillingType__c
Billing Frequency	SBQQ__BillingFrequency__c
Charge Type	SBQQ__ChargeType__c
Prorate Multiplier	SBQQ__ProrateMultiplier__c

## Billing Order Fields

The Order object contains fields for both the CPQ and the Billing packages. API names for Billing

package fields use the **bln**g\_\_ prefix. With certain page layout and field-level security settings, some fields aren't visible or editable. (Salesforce Billing Managed Package)

Field	API Name	Data Type	Definition
Billed Amount (without tax)	<b>b</b> <u>l<u>n<u>g</u><u>__</u><u>BilledAmountwi</u> <u>thouttax</u><u>__</u><u>c</u></u></u>	Roll-Up Summary (SUM Order Product)	The total amount that has been billed for all order products in the order, excluding tax.  Equal to the sum of all Billed Amount (without tax) fields on the order products.
Billed Tax	<b>b</b> <u>l<u>n<u>g</u><u>__</u><u>BilledTax</u><u>__</u><u>c</u></u></u>	Roll-Up Summary (SUM Order Product)	The total amount of taxes that have been billed for all order products in the order.  Equals the sum of all Tax Amount fields on the order products.
Billing Account	<b>b</b> <u>l<u>n<u>g</u><u>__</u><u>BillingAccount</u><u>__</u><u>c</u></u></u>	Lookup(Account)	Not populated by default. If a user or process populates the Billing Account field before invoicing the order, the invoice's Account field inherits this value rather than the order's Account field.
Billing Day of Month	<b>b</b> <u>l<u>n<u>g</u><u>__</u><u>BillingDayOfMo</u> <u>nth</u><u>__</u><u>c</u></u></u>	Picklist	Determines the day of the month when the order's order products are billed.
Bill Now	<b>b</b> <u>l<u>n<u>g</u><u>__</u><u>BillNow</u><u>__</u><u>c</u></u></u>	Checkbox	Creates an invoice for all order products in the order.
Canceled Billing Amount (without tax)	<b>b</b> <u>l<u>n<u>g</u><u>__</u><u>CanceledBillings</u><u>__</u><u>c</u></u></u>	Roll-Up Summary (SUM Order Product)	The total amount that

Field	API Name	Data Type	Definition
			was canceled and won't be billed for all order products in the order.
			Equal to the sum of all Canceled Billing Amount (without tax) fields on the order products.
Invoice Batch	<b>bIbg__InvoiceBatch__c</b>	Picklist	Groups orders for invoicing during invoice run evaluation. Null values are assigned to the same invoice run.
Pending Billing Amount (without tax)	<b>bIbg__PendingBillings__c</b>	Roll-Up Summary (SUM Order Product)	The total amount that hasn't yet been billed for all order products in the order.  Equal to the sum of all Pending Billing Amount (without tax) fields on the order products.
Total Billings	<b>bIbg__TotalBilling__c</b>	Formula (Currency)	The total amount that has been billed for all order products on the order, including tax.  Equal to the sum of all Billed Amount (without tax) fields and all Billed Tax fields on the order products.

## Usage Rating and Processing

---

Usage products are billed based on a consumed amount of service, such as an electricity bill. Vendors will not know how much of the service was used, and therefore how much to bill the customer, until the

usage period has elapsed. Salesforce Billing helps you organize usage-based products and invoice them based on the total amount of usage. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

Usage products are represented by order products with a Charge Type field of Usage. Admins first define charge type on the product record, which passes its charge type first to the quote line and then to the order product.

Usage products are typically billed in arrears, since vendors have to gather usage data before actually billing. However, there are several other use cases associated with usage products. Usually, vendors bill usage products repeatedly over time, with each finance period containing its own tracked usage data.

Quoted usage is effectively an estimate, so it affects bookings value. The actual invoice amount may differ from the quote estimate.

 **Note** When you're using products with consumption schedules, you can't amend to increase quantity. You can only cancel.

## Key Usage Billing Definitions

### Usage Records

Salesforce Billing uses two objects to manage usage-based invoicing. The usage object represents the amount of service consumed over a period of time. For example, a company sells a cell phone data plan that charges by the megabyte. A user streams 20 megabytes of data while commuting each weekday between 5:00 and 5:30 P.M., which is logged as a usage record. Over the course of a week, their account ends up with five unique usage records.

Usage summaries act as grouping objects for usage records, summarizing individual usage quantities into a total quantity that vendors can use for invoicing. If our cell phone user's vendor wants to bill monthly, they could use a usage summary record that totals the quantities from all 20 usage records that user created over the month.

### Usage Aggregation

Usage aggregation refers to the process of evaluating usage data recorded over a predefined period of time, then grouping it by a variable. Third parties usually handle this process.

### Usage Mediation

Usage mediation refers to the process of organizing aggregated data and storing it for reference. Salesforce Billing handles usage mediation.

## Usage Rating

Once usage data has been aggregated and mediated, the usage rating process determines the price per unit for each instance of usage.

## Matching ID

A usage and a usage summary contain a lookup relationship. However, both records must have identical Matching ID fields for the usage to count toward the usage summary. Salesforce Billing also uses Matching IDs when amending usage order products. An amendment quote results in a separate order with its own order product, which in turn results in a separate usage summary and usage records. Since the new usage and summary records all have the same Matching ID, Salesforce Billing can tie them back to the original usage summary.

Users have to enter matching IDs for all of their usage summaries. You can use any value for the matching ID. We recommend using a process builder to quickly assign matching IDs to large groups of usage summaries at once.

## Currently Supported Use Cases

- Post-Paid: Collecting usage and invoicing in arrears for the actuals

## Usage Billing Overview

Usage summaries have predefined date ranges for counting usage records. For example, a usage summary for January would have a summary start date of 01/01/2018 and a summary end date of 01/31/2018. When you create an order, Salesforce Billing generates usage summaries for your usage order products based on the order product's billing day of month, billing frequency, start date, and end date. A usage order product billed monthly with a start date of 01/01/2018 and end date of 12/31/2018 would have 12 usage summaries, each the length of each month in the year.

Salesforce Billing includes usage order products in an invoice if the usage summary's next billing date ends before the order product's next invoice target date. That way, all usage summaries that end before the invoice target date are included in the invoice.



## Ordering Subscription Products with Consumption Schedules

When you order a subscription product related to a consumption schedule, Salesforce CPQ creates an order product consumption schedule with the same field values and rates as the usage product's product consumption schedule. Access your order product consumption schedule through its related list on a subscription order product. Salesforce Billing also creates usage summaries for your order product based on the order product consumption schedule's billing term. The number and length of your usage summaries varies based on whether your order product came from a standard or evergreen subscription. (Salesforce Billing Managed Package)

### Standard Usage Summaries

A usage summary is an object that groups usage records for an order product. As you load usage into your org, Salesforce Billing associates the usage records with the summary based on matching IDs. When you invoice your usage order product, Salesforce Billing prices the resulting invoice line based on the sum of usage subtotals across your entire usage summary. (Salesforce Billing Managed Package)

### Custom Usage Summaries

It's possible to create custom usage summaries and associate them to order products derived from products configured with charge type Usage (legacy usage product) or with charge type Recurring plus one or more related consumption schedules (usage based pricing, or UBP). (Salesforce Billing Managed Package)

### Usage Rating

Once usage data has been aggregated and mediated, the usage rating process determines the price per unit for each instance of usage. (Salesforce Billing Managed Package)

### Manage Matching IDs

A usage record and a usage summary contain a lookup relationship. However, both records must have identical Matching ID fields for the usage to count toward the usage summary. (Salesforce Billing Managed Package)

### Billing for Amended Usage Summaries

When you amend a usage product and order the amendment quote, Salesforce Billing updates your usage summaries to reflect cancellations or new usage periods. (Salesforce Billing Managed Package)

### Usage Summary and Usage Fields

A usage summary shows the total quantity and value of related usages, and the date ranges for including usage records. The usage record shows the number of usage charges over a predefined time period. A user can override the precalculated usage amounts with their own values. With certain page layout and field-level security settings, some fields aren't visible or editable. (Salesforce Billing Managed Package)

## Ordering Subscription Products with Consumption Schedules

When you order a subscription product related to a consumption schedule, Salesforce CPQ creates an order product consumption schedule with the same field values and rates as the usage product's product consumption schedule. Access your order product consumption schedule through its related list on a subscription order product. Salesforce Billing also creates usage summaries for your order product based on the order product consumption schedule's billing term. The number and length of your usage summaries varies based on whether your order product came from a standard or evergreen subscription. (Salesforce Billing Managed Package)

### Ordering Consumption Schedules with Standard Subscriptions

When you order a standard subscription product related to a consumption schedule, Salesforce Billing generates usage summaries based on the schedule's billing term and billing term unit. The term and term unit represent the length of one usage summary, repeated through the order product's start and end date. (Salesforce Billing Managed Package)

### **Ordering Consumption Schedules with Evergreen Subscriptions**

Evergreen subscriptions are useful when you want to invoice a customer's usage until they cancel the service. (Salesforce Billing Managed Package)

#### See Also

- [Rating Usage Summaries With a Consumption Schedule](#)
- [Invoicing Usage Summaries with Consumption Schedules](#)

## Ordering Consumption Schedules with Standard Subscriptions

When you order a standard subscription product related to a consumption schedule, Salesforce Billing generates usage summaries based on the schedule's billing term and billing term unit. The term and term unit represent the length of one usage summary, repeated through the order product's start and end date. (Salesforce Billing Managed Package)

For example, you could have a standard subscription order product that bills monthly in arrears from 12/07/2018 through 12/06/19. Its order product consumption schedule has a 1-quarter term. You get one 3-month usage summary per quarter during your order product's term, totaling 4 usage summaries.

Usage summaries have their own next billing dates and inherits its next billing date from its summary end date. During an invoice run or when you invoice an order, Salesforce Billing looks at the next billing dates on order products and usage summaries. If an invoice run with a target date of 01/06/19 evaluates our order, it picks up the order product but it won't pick up any of our usage summaries as the earliest summary has a next billing date of 03/06/19.

Usage Summary	Summary Start Date	Summary End Date	Next Billing Date	Status
1	12/07/18	03/06/19	03/06/19	New
2	03/07/19	06/06/19	06/06/19	New
3	06/07/19	09/06/19	09/06/19	New
4	09/07/19	12/06/19	12/06/19	New

If you want your order product and its usage summaries to appear on the same invoice, configure the consumption schedule so that its term matches your order product's term. In this case, we could create a consumption schedule with a billing term of `1` and a billing term unit of `Month`. When a sales rep orders the product, Salesforce Billing creates a one-month usage summary each month during your order product's term, totaling 12 usage summaries.

Usage Summary	Summary Start Date	Summary End Date	Next Billing Date	Status
1	12/07/18	01/06/19	01/06/19	New
2	01/07/19	02/06/19	02/06/19	New

Usage Summary	Summary Start Date	Summary End Date	Next Billing Date	Status
...	...	...	...	...
12	11/07/19	12/06/19	12/06/19	New

Since our order product's initial next billing date falls on 01/06/19, an invoice run with a target date of 01/07/19 will pick up both the order product and the first usage summary. Further invoice runs will continue picking up both the order product and the monthly usage summary records at the same time unless a user overrides one of the next billing dates.

When an invoice run picks up a usage summary for invoicing, Salesforce Billing updates the summary's invoice run processing status to show its position in the invoicing process. The usage summary's invoice run processing status updates independently from the parent order product's invoice run processing status.

## Ordering Consumption Schedules with Evergreen Subscriptions

Evergreen subscriptions are useful when you want to invoice a customer's usage until they cancel the service. (Salesforce Billing Managed Package)

When a sales rep orders an evergreen subscription product related to a consumption schedule, Salesforce Billing performs the following process.

- Creates 12 usage summaries.
- Each summary's length is equal to the consumption schedule's term and term unit.
- The first usage summary inherits its start date from the parent order product's start date.

When the sales rep is ready to upload usage outside the original 12 summaries, they can create a usage record related to the evergreen subscription order product. Salesforce Billing then evaluates whether the usage date falls within an active usage summary. If it doesn't, Salesforce Billing creates a usage summary. The new usage summary follows the same billing periods as the order product's previous summaries.

For example, let's say our previous 12 summaries had a one-month term, began on the first of the month, and ran from 01/01/20 through 12/31/20. If the sales rep uploads usage with a usage date of 03/20/21, Salesforce Billing creates a usage summary for 03/01/21 through 03/31/21 and assigns it the new usage.

-  **Example** Let's look at a few ways Salesforce Billing creates usage summaries for evergreen subscriptions based on the consumption schedule's term. The parent evergreen order product bills monthly in arrears, with a start date of 04/01/2020. Remember, usage summaries have their own next billing dates, which are inherited from the summary end date.

Consumption Schedule Term	Resulting Usage Summaries	First Summary Start Date	First Summary End Date	First Summary Next Billing Date
1 Month	12 One-Month Summaries	04/01/20	04/30/20	04/30/20
2 Months	12 Two-Month Summaries	04/01/20	05/31/20	05/31/20
1 Quarter	12 Three-Month Summaries	04/01/20	06/30/20	06/30/20
2 Quarters	12 Six-Month Summaries	04/01/20	09/30/20	09/30/20

During an invoice run, or when you bill an order, Salesforce Billing looks at the next billing dates on order products and usage summaries independently. Sometimes, the first billing period for your evergreen subscription order product and your first usage summary can end up on different invoices based on their next billing dates. For example, let's say we ordered the evergreen subscription with the 2-month consumption schedule. If an invoice run with a target date of 04/30/20 evaluates our order, it picks up the order product. But it doesn't pick up any of our usage summaries because the earliest summary has a next billing date of 05/31/20. To include the order product and usage summary on the same invoice, configure the consumption schedule so that its term matches the billing frequency of your order product's term. In this case, we create a consumption schedule with a billing term of 1 and a billing term unit of Month. When a sales rep orders the product, Salesforce Billing creates a one-month usage summary each month during your order product's term, totaling 12.

Usage Summary	Summary Start Date	Summary End Date	Next Billing Date	Status
1	04/01/20	04/30/20	04/30/20	New
2	05/01/20	05/30/20	05/30/20	New
...	...	...	...	...
12	12/01/20	12/31/20	12/31/20	New

Because our order product's initial next billing date falls on 04/30/20, an invoice run with a target date of 04/30/20 picks up the order product and the first usage summary. Further invoice runs will continue to pick up the order product and the monthly usage summary records at the same time, unless a user overrides one of the next billing dates. When an invoice run picks up a usage summary for invoicing, Salesforce Billing updates the summary's invoice run processing status to show its position in the invoicing process. The usage summary's invoice run processing status updates independently from the parent order product's invoice run processing status.

## See Also

[Custom Usage Summaries](#)

## Standard Usage Summaries

A usage summary is an object that groups usage records for an order product. As you load usage into your org, Salesforce Billing associates the usage records with the summary based on matching IDs. When you invoice your usage order product, Salesforce Billing prices the resulting invoice line based on the sum of usage subtotals across your entire usage summary. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

Salesforce Billing creates a usage summary when you order a product with a charge type of Usage. In this case, Salesforce Billing generates usage summaries for your usage order products based on the order product's billing day of month, billing frequency, start date, and end date. A usage order product billed monthly with a start date of 01/01/2018 and end date of 12/31/2018 would have 12 usage summaries, each the length of each month in the year.

The usage summary's Usage related list shows all your associated usage records. Remember, Salesforce Billing evaluates assigning usage to a summary only if they both records have the same Matching ID field values. We recommend using a process builder to quickly provide matching IDs to large groups of usage summaries at once.

-  **Note** You can use any value for the matching ID. However, if a usage record matches the matching IDs and start/end dates of multiple usage summaries, Salesforce Billing assigns it to one of the usage summaries at random. We recommend using the order product's record ID or number as your matching ID. That way, Salesforce Billing will always match the IDs between the usage summary and usage. If you use a more generic number such as a desk number or org ID, Salesforce Billing could assign your usage to a usage summary that's not related to the correct order product.

Usage summaries inherit their start and end dates from the order products that created them. After Salesforce Billing sees equal matching IDs between a usage record and a summary, it checks whether the usage's End Date Time field falls within the summary's start date and end date. If it does, the usage record appears in the summary's Usage related list.

You can assign usage to a usage summary between (and including) the summary's start date and end date. After the end date, the summary won't accept new usage, even if the usage and summary have equal matching IDs.

The usage summary's quantity field shows the total quantity of all related usage, and the unbilled subtotal shows the subtotal of all related usage before billing.

### See Also

[Custom Usage Summaries](#)

## Custom Usage Summaries

It's possible to create custom usage summaries and associate them to order products derived from products configured with charge type Usage (legacy usage product) or with charge type Recurring plus one or more related consumption schedules (usage based pricing, or UBP). (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Salesforce Billing recommends the consumption schedule model of usage.

In many use cases, the solution involves creating usage summaries beyond what is created with order product generation. Invoice line start and end dates align with the usage summary start and end dates so you can create custom usage summaries to support invoicing customers on different cadences that support your business requirements.

### Use Case Example 1

This use case is a nonstandard cadence for invoicing a usage product that uses Salesforce Billing to rate Usage records against a usage summary.

You can create the usage summaries at any time before Usage records are loaded for rating. The order product can be in Activated status and still create and associate a new usage summary to it.

To get the best-performing environment, during your design phase, consider what other records are being created or updated and when.

Remember that four recurring billing frequencies (monthly, quarterly, annual, and semiannual) are supported. To avoid unpredictable results, don't add a custom picklist value for the Billing Frequency global value set. Depending on the billing frequency of the product, you can end up with unneeded usage summaries that you must delete or ignore.

 **Tip** To indicate to your code that a certain usage summary is to be deleted or ignored, create an attribute, or update the Matching ID or attribute with a value.

This table lists the required fields and relationships for creating a custom usage summary for loading usage.

	Required for Consumption Schedule Model	Required for Legacy Usage Model	Required Value
Order Product	✓	✓	Related order product
Summary Start Date	✓	✓	Start date of period
Summary End Date	✓	✓	End date of period
Order Product Consumption Schedule (consumption schedule usage only)	✓		Related order product consumption schedule
Price Schedule (legacy usage only)		✓	Related price schedule
Invoice Run Processing Status	✓	✓	Pending Billing
Matching ID	✓	✓	Per design
Matching Attribute* (consumption schedule usage only)	✓		Per design
Unit of Measure* (consumption schedule usage only)	✓		Per design
Status	✓	✓	New
Unbilled Subtotal			
Unbilled Quantity			
Source	✓	✓	External

\*Not technically required but highly recommended to incorporate in your overall design

## Use Case Example 2

This use case is for generating an invoice from a custom usage summary with no Usage records loaded.

This method is typically used with an external usage mediation and rating process. It can also be used to

bill on demand or on a nonstandard cadence. These usage summaries are used to generate invoices.

Because no usage is being rated, no order product consumption schedule is needed when you're using the consumption schedule model. With legacy usage, a Price Schedule lookup is still required.

Remember that four billing frequencies (monthly, quarterly, annual, and semiannual) are supported. Depending on how the product is configured, you get unneeded usage summaries when the order product is created, which you must delete or ignore.

- Tip** To reduce the number of unneeded records in your system, create an attribute, or update the Matching ID or attribute with a value to indicate to your code that a certain usage summary is to be deleted.

This table lists the required fields and relationships for creating a custom usage summary using direct invoicing.

	Required for Consumption Schedule Model	Required for Legacy Usage Model	Required Value
Order Product	✓	✓	Related order product
Summary Start Date	✓	✓	Start date of period
Summary End Date	✓	✓	End date of period
Order Product Consumption Schedule (consumption schedule usage only)	✓		
Price Schedule (legacy usage only)		✓	Related price schedule
Invoice Run Processing Status	✓	✓	Pending Billing
Matching ID	✓	✓	
Matching Attribute (consumption schedule usage only)	✓		
Unit of Measure	✓		

	Required for Consumption Schedule Model	Required for Legacy Usage Model	Required Value
(consumption schedule usage only)			
Status	✓	✓	<i>Queued for Invoice</i>
Unbilled Subtotal	✓	✓	Subtotal to be invoiced
Unbilled Quantity	✓	✓	Quantity to appear on invoice line
Source	✓	✓	<i>External</i>

After you save the manually created usage summary, billing triggers populate these fields.

- Unique ID
- Next Billing Date (This date is derived from the criteria set at the related order product.)

## Creating Usage Summaries

You can create custom usage summaries in these ways.

- Manually (by user)
- Flow or automated process via Apex
- Data Loader

Usage summaries created via standard functionality also have these fields populated.

- Status = New
- Source = Standard
- Unit Price
- Unbilled Quantity = 0
- Invoice Run Processing Status = Pending Billing
- Invoice Run Status (Index) = Pending Billing

## Design Considerations

- If a custom usage summary is created manually and overlaps the time frame of another usage summary, make sure that you assign it a matching ID that's different from the matching ID of the other usage summary. Otherwise, if usage is loaded that matches multiple usage summaries, the usage can be associated to any one of those summaries.

- For evergreen subscription products configured by consumption schedule, at order creation, 12 usage summaries are created, with the length of each summary corresponding to the billing term and billing term unit set on the consumption schedule. Whenever usages records are loaded with Matching ID, Matching Attribute, and Unit of Measure criteria that the system has a record of, the package creates other usage summaries to support loading and rating processes.
- Salesforce Billing doesn't support configuring a product as Charge Type = Usage and Subscription Type = Evergreen and adding usage summaries via a custom process using the information in this topic.

#### See Also

[Standard Usage Summaries](#)

[Ordering Consumption Schedules with Evergreen Subscriptions](#)

## Usage Rating

Once usage data has been aggregated and mediated, the usage rating process determines the price per unit for each instance of usage. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

#### [Loading Usage](#)

To load usage from API, the Data Loader, or other loading tools and successfully assign it to a usage summary, each usage record requires values for several fields. (Salesforce Billing Managed Package)

#### [Usage Rating Methods](#)

When you add usage to a usage summary in Salesforce Billing, you have two ways to control how Salesforce rates the usage and calculates the usage summary's new quantity and price. (Salesforce Billing Managed Package)

#### [Rating Usage Summaries With a Consumption Schedule](#)

Salesforce Billing uses the order product consumption schedule to rate usage related to your usage order product. As you upload usage to a usage summary, Salesforce Billing rates each usage record based on the schedule's type and where its quantity falls within the schedule's order product consumption rates. Before invoicing, your usage summary's unbilled subtotal equals the sum of its usage subtotals. (Salesforce Billing Managed Package)

#### [Rating Usage with Price Schedules](#)

Salesforce Billing can use price schedules to price usage. When you order a usage product, Salesforce CPQ converts its discount schedule into a price schedule, which inherits the discount schedule's prices and tiers. Then, when you upload usage, Salesforce Billing prices the usage based on the rules and prices defined in the price schedule. Usage summaries created from price schedules use the same billing rules and billing terms as their parent usage product. (Salesforce Billing Managed Package)

#### [Override Dates for Usage Products](#)

Usage summaries have optional override dates for ignoring usages during a set time period. (Salesforce Billing Managed Package)

## Usage Processing

Salesforce Billing lets you customize how you track and group instances of usage. (Salesforce Billing Managed Package)

## Loading Usage

To load usage from API, the Data Loader, or other loading tools and successfully assign it to a usage summary, each usage record requires values for several fields. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

#### Standard Required Fields

Start DateTime

End DateTime (must be equal to or later than Start DateTime)

If the End DateTime falls under a different usage summary than the start date time, Salesforce Billing assigns the usage to the usage summary that covers the end date time.

Matching ID

Quantity

#### Required Fields for Prerated Usage

Prerated Amount

Prerated Quantity (required instead of Quantity)

#### Required Fields for Upload Usage to Usage Summaries with Consumption Schedules

Matching Attribute must match the consumption schedule's Matching Attribute. These values can be null.

Unit of Measure must match the consumption schedule's Unit of Measure. These values can be null.

## After Loading

If Salesforce Billing successfully assigns usage to a usage summary, it updates several usage fields.

- The Status field changes its value to Processed.
- The order, order product, account, and usage summary fields all inherit their values from the usage summary.
- The Unit Price field inherits its value from the usage summary.

- If the package setting Usage Rating Process has a value of On Trigger, the usage updates its Subtotal field.
- The usage inherits the usage summary's currency, even if the usage originally had a different currency.

If Salesforce Billing can't assign usage to a summary, it updates the following usage fields.

- The Status field changes its value to Warning - Unrated
- The Error Message field shows why Salesforce Billing couldn't assign the usage.

Unrated usage can't be changed or corrected. To add your unrated usage to usage summary, create a usage record with the necessary corrections and try loading it again.

## Usage Rating Methods

When you add usage to a usage summary in Salesforce Billing, you have two ways to control how Salesforce rates the usage and calculates the usage summary's new quantity and price. (Salesforce Billing Managed Package)

### On Trigger

When you load usage, Salesforce Billing assigns the usage to a matching usage summary, then calculates each usage record's subtotal and unit price. Then, it calculates each usage summary's quantity and unbilled subtotal based on the summary's new usage records. If your usage summary uses range pricing, Salesforce Billing also rates all the summary's other usage to determine whether the new usage moves the summary into a new pricing tier. If your usage summary uses slab pricing, Salesforce Billing evaluates the summary's total quantity to determine a pricing tier for the new usage. It then rates all the new usage that falls in that tier.

### By Process

When you load usage, Salesforce Billing doesn't automatically assign the usage to a usage summary. Instead, you can run the `usageScheduleable` Apex class. This class assigns usage to a summary, then calculates the summary's quantity and unbilled subtotal based on the Quantity field in each of the summary's new usage records. However, to reduce load times, it doesn't calculate the Subtotal and Total Price fields for each usage record.

If your summary uses range pricing, On Trigger rating can cause long load times because Salesforce Billing runs the rating process every time new usage is uploaded. We recommend customers with range-priced usage summaries and large usage quantities use By Process rating. To avoid long load times, we advise testing in your Salesforce org to determine acceptable load times and which usage rating method meets your needs.

### Schedule Batch Usage Rating

Schedule a recurring Apex class that evaluates unrated usage, assigns it to usage summaries, and recalculates usage summaries based on their new usage records. (Salesforce Billing Managed Package)

## Schedule Batch Usage Rating

Schedule a recurring Apex class that evaluates unrated usage, assigns it to usage summaries, and recalculates usage summaries based on their new usage records. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Use a scheduled Apex class for usage rating only when the Salesforce Billing package setting Usage Rating Process Based On has a value of By Process. Make sure to load your usage records to Salesforce Billing before running the `usageScheduled` Apex class.

1. From Setup, enter `Apex Classes`, and then select **Apex Classes**.
2. Click **Schedule Apex**.
3. Enter a job name, then look up or enter `UsageScheduled` for the apex class.
4. Enter apex execution schedule, then save your changes.

## Rating Usage Summaries With a Consumption Schedule

Salesforce Billing uses the order product consumption schedule to rate usage related to your usage order product. As you upload usage to a usage summary, Salesforce Billing rates each usage record based on the schedule's type and where its quantity falls within the schedule's order product consumption rates. Before invoicing, your usage summary's unbilled subtotal equals the sum of its usage subtotals. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '19 and later

---

-  **Important** Salesforce Billing recalculates your usage summary's subtotal only when you upload a new usage record. Changing the quantity of a usage record doesn't cause Salesforce Billing to recalculate the usage summary's subtotal.

Salesforce Billing associates an order product consumption schedule with a usage summary if the two records have the same Matching Attribute and Unit of Measure values, including null values. The fields are null by default, in which case Salesforce Billing uses the order product consumption schedule defined in the usage's parent usage summary.

When you upload usage, Salesforce Billing associates usage with a usage summary if the records have the same Matching ID value. We recommend using a process builder to quickly assign matching IDs, matching attributes, and units of measure to your usage and usage summary records.

-  **Important** Values added to the consumption schedule field Unit of Measure must also be added to

the Unit of Measure Global Value Set in Salesforce Billing.

As you upload usage records to your usage summary, Salesforce Billing calculates the usage's subtotal based on where the usage quantity falls within the order product consumption schedule's rates. All order product consumption rates have bounds, a price, and a pricing method. The pricing method controls whether Salesforce Billing applies the price to each individual unit within the quantity, or as a flat fee to the entire quantity of usage. Let's look at how Salesforce Billing handles different usage pricing methods.

### Per Unit

- Usage Quantity: 100
- Pricing Method: Per Unit
- Price: \$0.50
- Usage Subtotal: \$50

### Flat Fee

- Usage Quantity: 100
- Pricing Method: Flat Fee
- Price: \$20
- Usage Subtotal: \$20

For order product consumption schedules with multiple rates, the value of the schedule's Type field also affects the usage summary's unbilled subtotal. To help understand this example, let's look at the following order product consumption rates for a usage summary with a quantity of 150.

- Rate 1: Bounds 0-100, \$50
- Rate 2: Bounds 101-200, \$40
- Rate 3: Bounds 201-Null, \$30

### Range

Salesforce CPQ prices the entire quantity of usage based on the rate where that quantity falls. For per-unit rates in the above example, our usage summary would have a subtotal of  $(150 * \$40) = \$6000$ . For flat rates, our usage summary would have a subtotal of \$40.

If you're using range-based pricing and a new quantity of usage causes the overall usage quantity to move to a new rate, Salesforce CPQ recalculates your usage summary's unbilled subtotal using the new range. Let's say you added a second usage record with a quantity of 70, moving your overall usage to rate 3.

- For per-unit pricing, your usage summary has a price of  $(220 * \$30) = \$6600$ .
- For flat-rate pricing, your usage summary has a price of \$30.

### Slab

Salesforce CPQ prices usage based on where its quantity falls in each available rate. In the above example, the first 100 units fall in Rate 1, and the next 50 units fall in Rate 2. For per-unit pricing, this leads to an unbilled subtotal of  $(100 * \$50) + (50 * \$40) = \$7000$ . For flat rates,

our usage summary would have an unbilled subtotal of (\$50+\$40) = \$90.

-  **Note** For flat fee pricing method, set the lower bound to 1 or higher. If the lower bound is 0 and the quantity is 0, Salesforce Billing shows a subtotal of 0 instead of a flat fee.

When you invoice a usage summary, Salesforce Billing creates an invoice line based on your usage summary's entire unbilled subtotal. After you post the invoice, Salesforce Billing reduces the summary's unbilled subtotal and unbilled quantity to zero.

## Rating Usage with Price Schedules

Salesforce Billing can use price schedules to price usage. When you order a usage product, Salesforce CPQ converts its discount schedule into a price schedule, which inherits the discount schedule's prices and tiers. Then, when you upload usage, Salesforce Billing prices the usage based on the rules and prices defined in the price schedule. Usage summaries created from price schedules use the same billing rules and billing terms as their parent usage product. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

Salesforce CPQ and Billing let you manage usage with price schedules or consumption schedules. For more information on consumption schedules, check out [Usage-Based-Products](#).

To rate usage with a price schedule, set your product's charge type to Usage and provide it with a discount schedule. Price schedules inherit their name, account, price book, and discount unit from the discount schedule. They also have lookups to the order and order product that they target. A range-type discount schedule creates a volume-type price schedule, and a slab-type discount schedule creates a tier-type price schedule.

-  **Note** Price schedules don't support amount-based additional discounts.

A usage summary's price is based off where the summary's total quantity falls within the price schedule. So, when you use volume price schedules, your usage records and usage summary all have the same unit price. But for tiered price schedules, your usages can have different unit prices than the usage summary based on how their quantity ranges fall in each price tier.

Price tiers inherit their name, number, bounds, and discount value from the discount tier. Every price tier also has a Price field. By default, Salesforce CPQ calculates each tier's price based on the order product's unit price minus the volume discount for that tier. For example, let's say you had an order product with a value of \$100 and quantity of 1. Your price schedule contains a tier with bounds of 1 through 11 and a discount amount of 30%. Salesforce CPQ calculates a price of \$70 for that tier.

When you upload usage, each usage calculates its unit price based on where the usage would fall within the price schedule's price tiers. If you're using a tiered, per-unit price schedule, usage with a quantity of

five would fall in the first tier, leading to a unit price of \$70 and subtotal of \$350.

When you make discount schedules for a usage product, create discount tiers so that the resulting price tiers have your desired per-unit pricing for your usage records. You can still edit a price tier's price after order product creation and before order product activation.

You can also create a price schedule manually and assign it to an order. However, creating all the price schedule's tiers and prices for every order containing your usage order product would take a lot of time. We recommend always assigning a discount schedule to your usage product so that Salesforce can auto-generate all the related price schedules.

### **Important**

- Always use a lower bound of zero for your first tier, allowing users to upload fractional usage quantities between zero and one.
- Even if your usage product isn't discountable, or you don't want to discount usages with quantities less than one, you still need a discount tier with a lower bound of zero to use fractional usage.
- Always create a discount schedule with an unbounded upper tier. If there is no unbounded upper tier, then the price schedules create an extra tier to account for quantities sold outside the last discount schedule's tier's upper bound. Since this final tier does inherit a discount from a discount tier, its price is equal to your usage order product's entire unit price.
- When Salesforce Billing invoices an order product with a charge type of Usage, it invoices only the usage summary's unbilled subtotal, and doesn't include the order product's billable unit price. If you want to charge customers for both usage and a flat order product fee, create a separate order product.

### **Example** Your customer sells a data recovery response subscription as a usage product.

#### **Product**

Subscription Pricing: Fixed Price

Charge Type: Usage

Billing Frequency: Monthly

List Price: \$100

When an end user receives a data recovery response, you upload the response incident as a usage record. Your customer wants a pricing structure that provides increasing discounts for 1-10, 11-20, 21-30, and 30 or more incidents. Create a range discount schedule with a discount unit of Percent and the following discount tiers.

### Discount Tiers

Tier Name	Lower Bound	Upper Bound	Discount (%)
Response Range 1	1	11	10
Response Range 2	11	21	15
Response Range 3	21	null	20

-  **Note** Remember, range discount schedules provide the tier's discount to your entire quantity of units. If you want a schedule that provides each tier's discount to only the quantity of units within that tier – 10% for the first ten products, 15% for the next 10 products – use a slab discount schedule.

When your customer orders one subscription of this usage product, Salesforce Billing converts the discount schedule into a price schedule with a type of Volume and discount unit of Percent. It contains the following tiers.

### Price Tiers

Tier Name	Lower Bound	Upper Bound	Volume Discount	Price
Response Range 1	1	11	\$10	\$90
Response Range 2	11	21	\$15	\$85
Response Range 3	21	31	\$20	\$80

The end customer has twelve incidents in January, so your workflow rule uploads a quantity-12 usage record to the order product's usage summary. Since twelve units fall within price tier 2, the usage has a unit price of \$85 and subtotal of \$1020.

#### Usage Summary

Summary Start Date: 01/01/18

Summary End Date: 01/31/18

Total Quantity: 12

Unbilled Subtotal: \$1020

Later that day, your invoice scheduler picks up the recovery response order product for invoicing, which produces an invoice line with a subtotal of \$1020.

## Override Dates for Usage Products

Usage summaries have optional override dates for ignoring usages during a set time period. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

Usage override dates are useful for temporarily ignoring usage. For example, a vendor is providing an integration and created usages during a six-day installation and testing period. To avoid including these charges on customer invoices, you can add

- **Override Summary Start Date:** This usage summary ignores usage prior to this date.
- **Override Summary End Date:** This usage summary ignores usage after this date.

## Usage Processing

Salesforce Billing lets you customize how you track and group instances of usage. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

Usage products are represented by order products with a Charge Type field of Usage. Admins first define charge type on the product record, which passes its charge type first to the quote line and then to the order product.

Usage products are typically billed in arrears, since vendors have to gather usage data before actually billing. However, there are several other use cases associated with usage products. Usually, vendors bill usage products repeatedly over time, with each finance period containing its own tracked usage data.

Quoted usage is effectively an estimate, so it affects bookings value. The actual invoice amount may differ from the quote estimate.

The usage object represents the amount of service consumed over time. You can measure service as a single event or as a collection of expenses over time. Let's look at a few examples.

- A company sells a cell phone data plan that charges by the megabyte per day. A user streams 20 megabytes of data while commuting each weekday between 5:00 and 5:30 P.M., which is logged as a usage record. Over the course of a week, their account ends up with five unique usage records. Each record represents the 30 minutes of service on a given day. In this case, they're using one usage record to represent all thirty minutes – this is an example of a single usage record representing a non-single quantity of usage.
- A company sells a security alarm response system. Each triggering of the alarm and the subsequent response represents a unit of usage – this is an example of a single usage record representing a single quantity of usage.

Usage summaries act as grouping objects for usage records, summarizing individual usage quantities into a total quantity that vendors can use for invoicing. If our cell phone user's vendor wants to bill

monthly, they could use a usage summary record that totals the quantities from all 20 usage records that user created over the month.

### Prerated Usage

Allow your customers to define their own amounts and quantities on their usage records. A customer-defined value overrides any rated value that Salesforce Billing would calculate. Prerated usage is useful for tracking billing expenses, pass-through postage, and third-party rating charges. (Salesforce Billing Managed Package)

### Managing Usage Processing Errors

If Salesforce Billing encounters an error when trying to associate usage with a usage summary, it sets the usage's status to Error or Warning - Unrated and shows an error message explaining the issue. In this case, take the required corrective actions and then change the usage record's status to New. After the status becomes New, Salesforce Billing attempts to process the usage again. (Salesforce Billing Managed Package)

## Prerated Usage

Allow your customers to define their own amounts and quantities on their usage records. A customer-defined value overrides any rated value that Salesforce Billing would calculate. Prerated usage is useful for tracking billing expenses, pass-through postage, and third-party rating charges. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later.

---

Define prerated values with the Prerated Quantity and Prerated Amount fields on the usage record. If your usage record contains a prerated quantity and prerated amount, Salesforce Billing doesn't calculate a usage subtotal based on the standard formula of `(Unit Price) * (Quantity)`. Instead, the usage summary that consumes your usage evaluates only the prerated amount when calculating the summary's subtotal. Salesforce Billing displays an error if you save a usage record with values for a prerated field and rated field.

When your usage summary evaluates its usage records, it calculates a price for your rated records based on their quantity and unit price, then adds that to the prerated amounts of your prerated records. For example, consider a usage summary with the following usage records.

Usage Record	Unit Price	Prerated Quantity	Prerated Amount	Quantity	Subtotal
1 (Prerated)	\$0.04	20	\$2.00	none	none
2 (Not Prerated)	\$0.04	none	none	25	\$1.00

In this case, Your usage summary bases its unbilled subtotal off the calculation `$(.04 * 25) + $2 =`

\$3.00. The summary also updates its Prerated Quantity field to reflect the sum of all prerated quantities on its usage records – in this case, 25.

An order product consumption schedule requires the following configuration to inherit usage summaries with prerated quantities.

- Only one order product consumption rate.
- The consumption rate has only one price tier.
- The price tier is per unit with a lower bound of 1 and null upper bound.

When you invoice a usage summary that contains prerated and rated usage, Salesforce Billing evaluates only the summary's total amount and total quantity. An invoice line for our usage summary would have a total amount of \$3.00 and quantity of 45.

## Applying Tax to Invoice Lines Based on Usage Summaries

By default, Salesforce Billing taxes invoice lines based on their amount. This process means that a usage summary made of rated and prerated usage would create an invoice line that calculates tax based on both types of usage. If you don't want to tax the prerated usage for one of your usage products, create a separate usage product to represent only your prerated usage and associate that with a tax rule that doesn't apply tax. You can then associate that order product with its own usage summary separate from the usage summary and order product you're using for your rated usage.

If you want a separate field for prerated tax, we recommend adding a custom Prerated Tax field to your usage record. You can then create a Prerated Tax field on your invoice that displays the total of prerated tax fields across all usage records related to your invoice.

## Managing Usage Processing Errors

If Salesforce Billing encounters an error when trying to associate usage with a usage summary, it sets the usage's status to Error or Warning - Unrated and shows an error message explaining the issue. In this case, take the required corrective actions and then change the usage record's status to New. After the status becomes New, Salesforce Billing attempts to process the usage again. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

Use Case	Error Message
Usage account and usage summary account don't match	Usage and order product accounts don't match.
Usage order and usage summary order don't match	Usage and order product orders don't match.

Use Case	Error Message
Usage order product and usage summary order product doesn't match	Usage order product and usage summary order product don't match.
User uploads the usage for a billed usage summary	Can't load usage for a usage summary that's already billed. To continue, upload usage and processing date for a later usage summary.
User uploads the usage for usage summary with an invoice Run processing status of Will Not Invoice.	Can't create or load usage data into a usage summary with an invoice run processing status of Will Not Invoice. To continue, change the Invoice Run Processing Status field to Pending Billing.
User uploads the usage for usage summary with an invoice run processing status of Error.	Can't save usage data for this usage summary because there's an error in the invoice run processing status. Check the error logs, fix the errors, and try again.
Usage and usage summary currency is different	Can't create this usage. To continue, make sure the currency for usage summary and usage is the same.
The order product has more than 1 tier for price schedule	Error: Prerated Usage can be used only with "Per Unit" pricing.
Usage doesn't share Matching ID with any available usage summary.	Matching Usage Summary not found.
 <b>Note</b> In this case, the usage's status is <b>Warning-Unrated</b> .	

## Manage Matching IDs

A usage record and a usage summary contain a lookup relationship. However, both records must have identical Matching ID fields for the usage to count toward the usage summary. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: Salesforce Billing Spring '18 and later

Salesforce Billing also uses Matching IDs when amending usage order products. An amendment quote results in a separate order with its own order product, which in turn results in a separate usage summary and usage records. Since the new usage and summary records all have the same Matching ID, Salesforce Billing can tie them back to the original usage summary.

Users have to enter matching IDs for all of their usage summaries. You can use any value for the matching ID. We recommend using a process builder to quickly assign matching IDs to large groups of usage summaries at once.

-  **Note** When you're using products with consumption schedules, you can't amend to increase quantity. You can only cancel.

## Billing for Amended Usage Summaries

When you amend a usage product and order the amendment quote, Salesforce Billing updates your usage summaries to reflect cancellations or new usage periods. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

When you amend a usage product and order the amendment quote, Salesforce Billing performs the following actions.

- If the amendment occurred during a usage summary's billing period, Salesforce Billing updates the summary's Override Summary End Date field to the order product's termination date.
- Salesforce Billing sets the Status field of any unbilled usage summaries to Canceled.

When you order your amended quote, Salesforce Billing performs the following actions based on the type of amendment you performed.

- If you swapped an order product for a new one, Salesforce Billing creates usage summaries for your remaining billing periods.
- If you canceled your original order product, Salesforce Billing does not create more usage summaries for it.

-  **Example** You sell a business security monitoring and response system. One of your customers has an order for the system that runs for 01/31/18 through 12/31/18, billed quarterly. This setup creates the following usage summaries.

Usage Summary	Start Date	End Date	Status
1	01/01/18	03/31/18	Open
2	04/01/18	06/30/18	Open
3	07/01/18	09/30/18	Open
4	10/01/18	12/31/18	Open

On 07/15/18, your customer decides to upgrade to your premium monitoring and response system. They amend their order to swap their original system for the premium system. When you order the

amended quote, Salesforce Billing has to account for the partial period created by amending your order product during the third usage summary's timeframe. First, Salesforce Billing provides a value of 07/14/18 to your third summary's Override Summary End Date field. It then sets your third and fourth summary's Override Status field to Canceled. While these usage summaries remain in your order product's Usage Summary related list, Salesforce Billing does not pick them up for invoicing during invoice runs. When you order your amendment quote, Salesforce Billing creates the following usage summaries on your amended order product.

Usage Summary	Start Date	End Date	Status
1	07/15/18	09/30/18	Open
2	10/01/18	12/31/18	Open

## Usage Summary and Usage Fields

A usage summary shows the total quantity and value of related usages, and the date ranges for including usage records. The usage record shows the number of usage charges over a predefined time period. A user can override the precalculated usage amounts with their own values. With certain page layout and field-level security settings, some fields aren't visible or editable. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

## Usage Summary Fields

Usage Summary Field	Description
Applied Unit Price	This field isn't supported.
Billed Quantity	The total quantity of this summary's usage that Salesforce Billing has billed. The billed quantity can contain usage from one or more usage records.
Billed Revenue	This field isn't supported.
Included Unbilled Usage	Used only when Order Product Charge Type is defined as Usage. Not used when a usage summary has an order product consumption schedule.
Included Unbilled Exhausted	Used only when Order Product Charge Type is defined as Usage. Not used when a usage summary has an order product consumption schedule.
Invoice	The invoice that contains an invoice line for this usage summary.

Usage Summary Field	Description
Invoice Line	The invoice line that represents invoiced billings from this usage summary.
Invoice Run	The invoice run that created an invoice line for this usage summary.
Invoice Run Processing Status	<p>Shows the order product status as evaluated by the invoice run:</p> <ul style="list-style-type: none"> <li>• Pending Billing—A future invoice run can create an invoice line from the usage summary.</li> <li>• Will Not Invoice—The usage summary isn't invoiced.</li> <li>• Error—An error was encountered while trying to invoice the usage summary. For more information, review the invoice scheduler's error log.</li> <li>• In Progress—An invoice line was created but not yet posted.</li> <li>• Completed—An invoice line for the usage summary was created and posted.</li> </ul>
Invoice Run Status (Index)	Text string that mirrors the value of the Invoice Run Processing Status picklist. Users must not edit this system field.
Matching Attribute	Populated from the consumption schedule. Usages are rated against usage summaries where the Matching ID, Matching Attribute, and Unit of Measure fields all match. When Charge Type equals Usage, legacy usage pricing ignores this field.
Matching ID	Salesforce Billing uploads usage only to usage summaries with matching IDs. We recommend creating a workflow rule that populates your usages and usage summaries based on certain criteria.
Next Billing Date	The date that this usage summary is due for billing. The Target Date on an invoice run must be after this date for the usage summary to be invoiced. When you use Bill Now, the effective target date must be after this date.
Override Item Consumption Schedule	Lookup to the order item consumption schedule for pricing. Populated only for products with consumption schedules.
Override Status	<p>When this status is set to Canceled:</p> <ul style="list-style-type: none"> <li>• Prevents the usage summary from loading more usage.</li> <li>• Nulls the Unique Id field.</li> <li>• Allows the usage summary to be invoiced (doesn't prevent invoicing).</li> </ul>
Override Summary End Date	Salesforce Billing automatically calculates your summary's start and end dates based on your order product's next billing date and billing

Usage Summary Field	Description
	frequency. However, you can enter a date in this field to override the default summary end date.
Prerated Amount	The total value of prerated usage across all of this summary's usage records. When users enter a prerated amount on a usage record, Salesforce Billing uses that value for the usage's total rather than the standard calculation of usage unit price * usage quantity.
Prerated Quantity	The total value of prerated usage across all of this summary's usage records. When users enter a prerated amount on a usage record, Salesforce Billing uses that value for the usage's total rather than the standard calculation of usage unit price * usage quantity.
Source	<ul style="list-style-type: none"> <li>• Standard—The billing package created the usage summary.</li> <li>• External—A user-defined external process created the usage summary, or it was created manually.</li> </ul> <p>The source is automatically set to Standard when created by the managed package. Use External when creating custom usage summaries. See <a href="#">Custom Usage Summaries</a></p>
Status	<p>Valid values are:</p> <ul style="list-style-type: none"> <li>• New: The usage summary is available for adding and rating usage records.</li> <li>• Queued For Group Calculation: For system use only.</li> <li>• Queued For Subtotal Calculation: For system use only.</li> <li>• Queued For Invoice: The usage summary is ready to be invoiced. If Invoice Run Status = Pending Billing, additional usage can be loaded and rated. If Invoice Run Status = In Progress, no more usage can be loaded unless the invoice is canceled or rebilled.</li> <li>• Processed: The usage summary was invoiced. No additional usage can be loaded or rated.</li> <li>• Canceled: Not invoiced, and no additional usage can be loaded or rated.</li> </ul>
Summary Start Date	Salesforce Billing includes usage in a usage summary if the records have matching IDs and the usage's processing date falls within the summary's start and end dates. The first usage summary start date inherits the order product's start date. For more information on how summary start dates are calculated for additional usage summaries, see <a href="#">Changing the Billing Day of Month for Usage Products</a> .
Summary End Date	Salesforce Billing includes usage in a usage summary if the records have

Usage Summary Field	Description
	matching IDs and the usage's processing date falls within the summary's start and end dates.
Total Revenue	This field isn't supported.
Unbilled Revenue	This field isn't supported.
Unbilled Quantity	The quantity of the summary's usage that hasn't been invoiced.
Unbilled Subtotal	The amount of the summary's usage that hasn't been invoiced.
Unit Price	<p>When you invoice the usage summary, the unbilled quantity, unbilled subtotal, and unit price become 0.</p> <p>A usage summary's unit price is based on where the summary's total quantity falls within the price schedule. When you use volume price schedules, your usage records and usage summary all have the same unit price. But for tiered price schedules, your usages could have different unit prices than the usage summary based on how their quantity ranges fall in each price tier.</p> <p>When you invoice the last of your usage and the summary's unbilled quantity becomes 0, the summary's unit price changes to 0.</p>
Unit of Measure	Populated from the consumption schedule. Usages are rated against usage summaries where the Matching ID, Matching Attribute, and Unit of Measure fields all match. When Charge Type equals Usage, legacy usage pricing ignores this field.
Usage Count	The number of usage records rated against this usage summary. The number doesn't change when a rated usage record is deleted.

## Usage Fields

Usage Field	Description
Account	The account that contains this usage record.
End DateTime	The end date and time of the period during which an end-user charges for an instance of use.
Error Message	If Salesforce Billing can't calculate a value on this usage record, it shows a message with the reason for the error.
Matching Attribute	Usages are rated against usage summaries where the Matching ID, Matching Attribute, and Unit of Measure fields all match. When Charge Type equals Usage, legacy usage pricing ignores this field.

Usage Field	Description
Matching ID	Salesforce Billing rates usage only to usage summaries with matching IDs. Usages are rated against usage summaries where the Matching ID, Matching Attribute, and Unit of Measure fields all match.
Prerated Amount	When users enter a prerated amount on a usage record, Salesforce Billing uses that value for the usage's total rather than the standard calculation of usage unit price * usage quantity.
Processing Date	The value of the End DateTime field. The date is adjusted based on the company's default time zone.
Quantity	The total number of times an end customer used this service during the usage record's start and end dates.
Start DateTime	The start date and time of the period during which a user is charged for an instance of use.
Status	<p>Valid values are:</p> <ul style="list-style-type: none"> <li>• New: The Usage records are available.</li> <li>• Queued For Subtotal Calculation: The usage is ready to be rated.</li> <li>• Processed: The Usage records are rated successfully and their corresponding usage summaries are updated.</li> <li>• Warning: The usage isn't rated successfully. This status is shown when matching usage summary isn't found. The Error Message field shows the reason.</li> <li>• Error: An error occurred when trying to rate the usages. The Error Message field gives the details.</li> </ul>
Subtotal	Quantity * unit price. If the package setting for Usage Rating Process Based On is By Process, the value populates when the batch job for usage rating is run.
Unit of Measure	Usages are rated against usage summaries where the Matching ID, Matching Attribute, and Unit of Measure fields all match. When Charge Type equals Usage, legacy usage pricing ignores this field.

## Usage Fields on the Order Product

These fields are used when Charge Type equals Usage and ignored when using a consumption schedule.

Usage Field	Description
Included Usage	The Included Usage field defines a quantity that you provide for free before you begin charging for usage. If your order product has an

Usage Field	Description
	included quantity of 10 and usages sum to 12, the invoiced quantity is only 2.
Usage Floor Quantity	Usage Floor Quantity defines the minimum quantity to charge for, even for little or no usage. For example, if your order product has a usage floor quantity of 10, and usages sum to 7, the invoiced quantity is 10.

## Invoice Generation

---

Invoices display a list of purchased items and services alongside the total amount a customer must pay. The invoice record itself contains important details such as the balance, due date, and payment status. You can control several date fields in the Salesforce CPQ and Salesforce Billing packages to manage the number of invoices generated from an order, their billing dates, and the order products converted to invoice lines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

---

Salesforce Billing allows you to create an invoice manually from a single order or to automate the invoice creation process with an invoice scheduler. Invoice schedulers evaluate unbilled order products based on user-defined criteria and create an invoice with invoice lines for each matching order product.

An invoice inherits the following fields from its parent order.

- Service Start Date
- Service End Date
- Subscription Term
- Payment Terms
- Order products, which convert to invoice lines

Invoices can have the following statuses.

- Draft: Salesforce Billing has not calculated field values for this invoice or its invoice lines.
- Initiated: Salesforce Billing is calculating field values for the invoice lines on this record, and field values for the invoice itself.
- Error: Salesforce Billing was unable to finish calculating field values on this invoice. You can find information on these errors in the invoice's Error Log related list.
- Post In Progress: Salesforce Billing is working on posting this invoice. If this process finishes successfully, the Status field changes to Posted. If Salesforce Billing encounters an error, the Status field changes to Error.
- Posted: Salesforce Billing has finished calculating field values for this invoice and its invoice lines. You can now send it to a customer.

- Canceled: A user or process has canceled this invoice. Salesforce Billing has associated the invoice with a credit note equal to the remaining balance, with credit lines automatically allocated to each invoice line.
- Rebilled: A user or process has canceled this invoice and rebilled it via the **Cancel and Rebill** button.

Invoices generate with a default Status field set to Draft. This state allows admins to review the invoice for accuracy before sending it to a customer. If you're posting invoices manually, when you're ready to post your invoice, change its Status field from Draft to Posted. Otherwise, your invoice run automates changes between status fields.

### Billing Process Overview

Salesforce Billing provides several ways to automate subscription billing, usage-based product billing, milestone billing, and custom billing with recurring invoices. The invoice record represents one-time, recurring, and usage-based charges incurred over a period of time. (Salesforce Billing Managed Package)

### Proration Types for Partial Billing Periods

Salesforce Billing uses proration to calculate balances for invoice lines that cover partial billing periods. While the invoice line doesn't have a prorate multiplier field, Salesforce Billing still calculates one behind the scenes and multiplies it by the order product's billable unit price to determine the invoice line's balance. The Salesforce Billing invoice package setting Proration Type offers several options for the method used to calculate the prorate multiplier. (Salesforce Billing Managed Package)

### Grouping Order Products into Invoices

Salesforce Billing creates invoices for different groups of order products based on several order and order product fields. This process is useful for invoicing certain types of order products separately from your other order products. It's also useful if your account managers want one invoice record for all their orders. (Salesforce Billing Managed Package)

### Invoice Schedulers for Automating Invoice Creation

Set up an invoice scheduler for your billable order products so invoices can be generated automatically. (Salesforce Billing Managed Package)

### Invoice Presentation

Salesforce Billing provides several options for creating documents based on your invoices. (Salesforce Billing Managed Package)

### Managing Invoice Lines Related to Your Assets

An asset contains several fields and relationships to help you track how often and for how much you've invoiced a customer. You can create a relationship by adding an asset to the order product's Lifecycle-Managed Asset field. The asset's Invoice Line Balance field shows the sum of the Balance fields on the posted invoice lines related to the asset. (Salesforce Billing Managed Package)

### Bill Now

Users can manually invoice an order by selecting Bill Now on the order record and saving. It's a useful feature when a sales rep must quickly produce an initial or final invoice. (Salesforce Billing Managed Package)

### Invoice Fields

Before you start working with billing and invoicing, review invoice fields and attributes. (Salesforce Billing Managed Package)

## Invoice Line Fields

Before you start working with billing and invoicing, review invoice line fields and attributes. (Salesforce Billing Managed Package)

## Editable Fields on Invoices and Invoice Lines

Salesforce Billing follows Generally Accepted Accounting Principles, which require that users must always be able to track details of invoices and invoices lines. To ensure that invoice information is accurate for accounting purposes, Salesforce Billing doesn't allow you to edit or delete certain fields on posted invoices and their lines. While some other fields are editable, changing them may prevent data from passing correctly during other Salesforce Billing processes such as payments or amended orders. When you work with invoices, review the fields that you can safely edit on invoices and invoice lines. (Salesforce Billing Managed Package)

# Billing Process Overview

Salesforce Billing provides several ways to automate subscription billing, usage-based product billing, milestone billing, and custom billing with recurring invoices. The invoice record represents one-time, recurring, and usage-based charges incurred over a period of time. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Posting Invoices

Post an invoice to lock it from changes so that customers or processes can make payments against it. A posted invoice is also ready for revenue recognition reporting. (Salesforce Billing Managed Package)

### Invoicing One-Time Products

When an invoice run or manual invoice action (such as clicking Bill Now) invoices an order product with a one-time charge type, Salesforce Billing creates a single invoice line. This invoice line represents your order product line's entire quantity and total price. (Salesforce Billing Managed Package)

### Invoicing Recurring Products

Salesforce Billing invoices subscription products repeatedly over time. An order product's date fields and billing fields drive when and how frequently that order product is invoiced. (Salesforce Billing Managed Package)

### Invoicing Evergreen Subscriptions

When your order product has an Evergreen subscription type, Salesforce Billing invoices it each billing period indefinitely. This way, you can charge customers for subscriptions until they decide to cancel. Salesforce Billing determines an evergreen subscription's invoiceable balance for one billing period based off the order product's billable unit price. (Salesforce Billing Managed Package)

### Invoicing Products with Billing Schedules

Billing schedules use predetermined dates to define when and how Salesforce Billing invoices an order product. For example, a billing schedule could invoice for 30% upon order activation and the remaining 70% after 90 days. When an order product has a billing schedule, Salesforce Billing uses the schedule to drive invoice processing rather than the order product's date fields. (Salesforce Billing

Managed Package)

#### Invoicing Usage Summaries

A usage summary is invoiced independently from its parent order product. Both records have their own Next Billing Dates, allowing users to invoice them at the same time or to create different billing timeframes. You can use different billing timeframes to split your usage order product and its related usage summaries onto different invoices. Your ability to customize usage summary billing timeframes will vary based on whether you're pricing usage with price schedules or order product consumption schedules. (Salesforce Billing Managed Package)

#### Invoice an Order Using Bill Now

Create an invoice from one of your orders. (Salesforce Billing Managed Package)

#### Hold Billing

Prevent an order from billing until it meets certain criteria. (Salesforce Billing Managed Package)

## Posting Invoices

Post an invoice to lock it from changes so that customers or processes can make payments against it. A posted invoice is also ready for revenue recognition reporting. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Post an invoice by changing its Status field from Draft to Posted. If you're using invoice schedulers, you can also select the Automatically Post Invoices field. All of that scheduler's invoice runs will change a draft invoice's status to Posted once invoice creation has finished.

Posting an invoice enables the following actions.

#### Payments

Manually allocate payments to the invoice's invoice lines

Allow customers to select the invoice for payment in the payment center

Allow payment runs to automatically process the invoice for payment

#### Revenue Recognition

Create a revenue schedule for invoice lines covered by a Revenue Schedule Creation Action set to Invoice Posting.

Remember, the invoice is a legal document that represents the customer's agreement to pay you for services. Posting an invoice finalizes this agreement and prevents you from deleting the invoice or any of its invoice lines. This way, your organization always has a record for legal and bookkeeping purposes.

If you need to change the invoice – for instance, if a mistake on the account's address caused an

incorrect tax calculation – you'll use credit and debit notes to change the invoice's balance while keeping the invoice and invoice line records available for reference. For more information, check out [Issuing Credits, Creating Debits and Add-on Charges](#), and [Cancel and Rebill an Invoice](#).

 **Note** Salesforce Billing doesn't support posting multiple invoices from a list view.

## Invoicing One-Time Products

When an invoice run or manual invoice action (such as clicking Bill Now) invoices an order product with a one-time charge type, Salesforce Billing creates a single invoice line. This invoice line represents your order product line's entire quantity and total price. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

When a user, workflow, or invoice scheduler invoices an order, active order products with a next billing date on or before the invoice target date are included on the invoice. One-time order products always have next billing dates equal to their Order Product Start Date field.

For more information on how Salesforce Billing calculates the next billing date, check out [Understanding Next Billing Date](#).

## Invoicing Recurring Products

Salesforce Billing invoices subscription products repeatedly over time. An order product's date fields and billing fields drive when and how frequently that order product is invoiced. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

When a user, workflow, or invoice scheduler invoices an order, active order products with a next billing date on or before the invoice target date are included on the invoice. For more information on how Salesforce Billing calculates the next billing date, check out [Understanding Next Billing Date](#).

All subscription products have a charge type of Recurring. When you invoice a recurring order product, the resulting invoice line represents a period of service from the invoice line's start date up through its end date. By default, the start date inherits the order product's next billing date, and the end date is one term of the billing frequency afterward. So, let's say you invoice a recurring order product with monthly billing frequency and a next billing date of 10/08/2018. The resulting invoice line has a start date of 10/08/2018 and end date of 11/07/2018.

Salesforce Billing uses a formula to calculate exactly how much is billed for a recurring order product. The result of this calculation is known as the order product's billable unit price.

```
Billable Unit Price = [(Total Amount ÷ Prorate Multiplier)] ÷ Subscription Term  
* Billing Frequency Months
```

- The first part of the formula takes the quote line's net total, which is prorated, and divides it by the quote line's prorate multiplier. This provides an unprorated total amount, or what it would cost if the customer purchased the product for exactly the length of its subscription term.
- Next, the formula divides that value by the order product's subscription term (remember that the order product inherits the product's subscription term.) This gives us a temporary price per month.
- Finally, Salesforce Billing accounts for billing frequency by multiplying the price per month by the number of months covered by a given billing frequency. For example, quarterly billing frequency would give a value of 3.

We now have a value for the amount we expect to bill a recurring order product on any given invoice. When you invoice this order product, its balance by default is equal to the billable unit price. If your order had partial periods or proration, Salesforce Billing then prorates the invoice line's balance accordingly.

 **Note** The invoice line's start date equals the order product's next charge date. If the order product has a value for Bill Through Date Override, Salesforce Billing uses that value for the invoice line's end date.

## Invoicing Evergreen Subscriptions

When your order product has an Evergreen subscription type, Salesforce Billing invoices it each billing period indefinitely. This way, you can charge customers for subscriptions until they decide to cancel. Salesforce Billing determines an evergreen subscription's invoiceable balance for one billing period based off the order product's billable unit price. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '19 and later

---

When a sales rep quotes an evergreen subscription product, Salesforce CPQ sets the quote line's subscription term to 1. This allows the quote line's net total price to represent one month of billing for forecasting purposes. When they order the quote, the resulting order product inherits the subscription term of one. Users can't change an evergreen quote line or order product's subscription term. Evergreen order products always have a prorate multiplier of one as well.

Since the order product's subscription term and prorate multiplier is always one, Salesforce CPQ uses the following simplified formula to calculate the the order product's billable unit price.

```
(Order Product's Total Price * Order Product's Billing Frequency) / (Order
```

Product's Prorate Multiplier \* Default Subscription Term)

Let's look at a few examples.

Total Price	Billing Frequency	Prorate Multiplier	Default Subscription Term	Billable Unit Price
\$1200	Monthly (1)	1	12	\$100
\$4000	Monthly (1)	1	16	\$250
\$4000	Quarterly (3)	1	16	\$750

Since your evergreen subscription doesn't have an end date, Salesforce Billing invoices its order product for its billable unit price until you cancel the order product.

## Bookings Amount for Evergreen Subscriptions

The order product field Order Product Bookings represents the total amount that Salesforce Billing will invoice the order product for across all its billing periods, also known as the bookings amount. Many companies that recognize revenue from the order product use this field for forecasting purposes.

For standard subscriptions, an order product's total amount and bookings amount are always the same. However, an evergreen subscription has no predefined end date, so its total amount may not be the same as the actual amount you end up billing for. To prevent inaccurate data, Salesforce Billing always sets the bookings amount for an evergreen subscription order product to 0.

Your revenue recognition treatment's Revenue Schedule Amount field drives the billing field that Salesforce Billing uses to determine the total amount of your revenue schedule. When you recognize revenue on the order product, set your revenue schedule amount to Bookings Amount. When Salesforce Billing creates a revenue schedule for an order product related to that treatment, it sets the total revenue amount to the same value as the order product's bookings amount. This setup ensures that standard subscription products always have revenue schedules reflecting their total forecasted amount, while evergreen subscriptions always have revenue schedules with a total revenue amount of zero.

### Overriding Billable Unit Price for Evergreen Subscriptions

The order product's Override Billable Unit Price field lets you define a custom billable unit price for your evergreen subscriptions. This field is useful for changing the price that Salesforce Billing uses to invoice an evergreen subscription's order product. You can also use it to override the billable unit price on evergreen subscription records that were migrated to Salesforce Billing. (Salesforce Billing Managed Package)

## Overriding Billable Unit Price for Evergreen Subscriptions

The order product's Override Billable Unit Price field lets you define a custom billable unit price for your evergreen subscriptions. This field is useful for changing the price that Salesforce Billing uses to invoice an evergreen subscription's order product. You can also use it to override the billable unit price on

evergreen subscription records that were migrated to Salesforce Billing. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce CPQ Spring '19 and later

---

You can edit the Override Billable Unit Price field only on unactivated order products for evergreen subscriptions that haven't been invoiced.

After you start invoicing the evergreen order product, Salesforce Billing will use its Override Billable Unit Price instead of its billable unit price when creating invoice lines.

We recommend using the Override Billable Unit Price field when you need to change an evergreen subscription's billable unit price before activation. It's also useful if you're migrating an evergreen subscription order product from outside Salesforce Billing - you can quickly define an Override Billable Unit Price rather than rerun calculations to determine the billable unit price.

Percent of total subscriptions don't evaluate the Override Billable Unit price for any of their covered evergreen subscriptions. If you want to change the billable unit price on a covered evergreen subscription so that the percent of total parent covers the new price, we recommend canceling the evergreen subscription and replacing it with one that has the new price.

 **Warning** Always use Override Billable Unit Price to change your evergreen order product's billable unit price. Salesforce Billing sometimes encounters errors if you manually change the Billable Unit Price field.

## Invoicing Products with Billing Schedules

Billing schedules use predetermined dates to define when and how Salesforce Billing invoices an order product. For example, a billing schedule could invoice for 30% upon order activation and the remaining 70% after 90 days. When an order product has a billing schedule, Salesforce Billing uses the schedule to drive invoice processing rather than the order product's date fields. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer Winter '18 and later

---

For an overview of invoice plans and billing schedules, check out [Dynamic Invoice Plans](#).

 **Note** Billing schedules are supported only when the billing frequency is Invoice Plan. To use a billing schedule, set charge type to Recurring. Recurring is also used for one-time products using an invoice plan.

When you invoice a billing-schedule-covered order product, Salesforce Billing adds one billing

transaction to the schedule for each of the schedule's invoice plan lines. A transaction's billing target date determines when Salesforce creates an invoice line for your order product, and the amount to be billed determines the invoice line's balance. Each transaction has a status of Pending until it hits the billing target date, which changes the status to Billed.

 **Example** Let's say you invoice a billing-schedule-covered order product with a balance of \$1000 on 06/01/18. The billing schedule looks up to an invoice plan with the following invoice plan lines.

Name	Type	Percentage	Commencement Date Offset	Commencement Date Offset Units	Processing Order
20 Percent	Percent	20	10	Days	1
50 Percent	Percent	50	1	Months	2
Remainder	None	None	6	Months	3

When you invoice this order product, Salesforce Billing creates three billing transactions on your billing schedule.

Name	Status	Billing Target Date	Amount To Be Billed
20 Percent	Pending	06/08/18	\$200
50 Percent	Pending	06/29/18	\$500
Remainder	Pending	11/29/2018	\$300

On 06/08/18, Salesforce Billing creates one invoice line with a balance of \$200 and a start date of 06/08/18.

## Invoicing Usage Summaries

A usage summary is invoiced independently from its parent order product. Both records have their own Next Billing Dates, allowing users to invoice them at the same time or to create different billing timeframes. You can use different billing timeframes to split your usage order product and its related usage summaries onto different invoices. Your ability to customize usage summary billing timeframes will vary based on whether you're pricing usage with price schedules or order product consumption schedules. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

#### [Invoicing Usage Summaries with Consumption Schedules](#)

An order product consumption schedule inherits its billing rule, tax rule, and revenue recognition rule inherited from the original consumption schedule. When an invoice scheduler evaluates a usage

summary related to an order product consumption schedule, it uses the schedule's rules and treatments rather than the parent order product's rules and treatments. The invoicing process also evaluates a usage summary's next billing date separately from the order product's next billing date, so that usage summaries can end up on different invoices than their parent subscription order product if desired. These separate combinations let you further customize how Salesforce Billing invoices a usage order product relative to its usage summaries. (Salesforce Billing Managed Package)

### Invoicing Usage Summaries with Price Schedules

When you invoice a usage order product, Salesforce Billing evaluates its related usage summaries. If a usage summary's next billing date ends on or before the invoice's target date, Salesforce Billing includes that summary's usage data in the resulting invoice line. This way, all usage summaries that end before the invoice target date are billed on the invoice. (Salesforce Billing Managed Package)

## Invoicing Usage Summaries with Consumption Schedules

An order product consumption schedule inherits its billing rule, tax rule, and revenue recognition rule inherited from the original consumption schedule. When an invoice scheduler evaluates a usage summary related to an order product consumption schedule, it uses the schedule's rules and treatments rather than the parent order product's rules and treatments. The invoicing process also evaluates a usage summary's next billing date separately from the order product's next billing date, so that usage summaries can end up on different invoices than their parent subscription order product if desired. These separate combinations let you further customize how Salesforce Billing invoices a usage order product relative to its usage summaries. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '19 and later

---

## Next Billing Date

After Salesforce CPQ creates an order product associated with a consumption schedule, Salesforce Billing generates usage summaries based off the schedule's billing term and billing term unit. The term and unit represent the length of one usage summary, repeated through the order product's start and end date. This setup lets you create usage summaries with different billing terms from the parent order product. For example, you could have an order product that bills monthly in arrears from 12/07/2018 through 12/06/2019. Its order product consumption schedule has a 1-quarter term, so you'll have one three-month usage summary each quarter for the duration of your order product's term. This setup leads to a total of four usage summaries.



Usage summaries have their own next billing dates. A usage summary inherits its next billing date from its summary end date. When you bill an order or during an invoice run, Salesforce Billing looks at each next billing date on order products and usage summaries. If an invoice run with a target date of 01/06/2019 evaluates our order, it picks up the order product, but it won't pick up any of our usage summaries

as the earliest summary has a next billing date of 03/06/2019.

Because order product and usage summary next billing dates are independent, overriding the order product next billing date doesn't change the usage summary's next billing date. Order products and usage summaries have independent invoice run processing statuses as well. If an order product's Hold Billing field is selected, Salesforce Billing will hold billing for the order product and all its usage summaries. Bill Now and invoice runs will not pick the order product or its usage summaries up for invoicing.

-  **Important** When Bill Now or an invoice run evaluates a usage summary, it includes the usage summary on the invoice if the usage summary's next billing date is **less than** the invoice's target date (for Bill Now) or the invoice run's target date (for invoice runs). This process is different from invoicing an order product or billing transaction, which are included on the invoice if their next billing date is **less than or equal to** the invoice or invoice run's target date. If your usage summary doesn't have a next billing date, the Bill Now process or invoice run evaluates the Summary End Date field instead.

If you want your order product and its usage summaries to appear on the same invoice, configure the consumption schedule so that its term matches your order product's term. If your product bills monthly from 01/01/2019 through 12/31/2019, the consumption schedule must have a billing term of one month. When a sales rep orders the product, Salesforce Billing creates a one-month usage summary each month during your order product's term. This leads to a total of 12 usage summaries.

## Rules

You can apply different rules between your subscription order product and its usage. For example, you could apply tax to the order product while not applying tax to one of the order product's consumption schedules. You could also record order product transactions to a different GL Account than your usage summary transactions.

For more information on rules in Salesforce Billing, check out [Defining Rules and Treatments](#).

-  **Example** Your order product has a billable unit price of \$100 and bills monthly from 11/19/2018 through 11/18/2019. Its current Next Billing Date is 12/18/2018. It contains 12 usage summaries, each the duration of one month. Let's look at the billing information on your order product's first usage summary.  When your invoice scheduler evaluates these records, it creates one invoice line for each on the resulting invoice. The first invoice line reflects the order product, with a subtotal of \$100. The second invoice line reflects the usage summary, with a subtotal of \$112.50. 

-  **Tip** By default, both invoice line product names match the name of the order product that you invoiced. We recommend using automation to change the product names so you can tell which invoice line relates to the order product and which relates to the usage summary. In this example, we've changed the second invoice line's product name to reflect that it came from your usage summary for the phone plan minutes.

When you post the invoice, Salesforce Billing updates the usage summary's billed quantity and

unbilled subtotal to reflect the changed billing status.

-  **Example** Your order product bills \$100 monthly from 12/07/2018 through 12/06/2019. Its current Next Billing Date is 01/07/2019, while its Next Charge Date is 12/07/2018. Its usage summaries bill quarterly.  Since a usage summary inherits its next billing date from its summary end date, your first usage summary has a next billing date of 03/06/2019. It has an unbilled subtotal of \$50. When an invoice scheduler or Bill Now evaluates your order, it evaluates the order product's Next Charge Date and creates an invoice with an invoice line for your order product. However, since your usage summary has a future Next Billing Date, the invoice doesn't include an invoice line for the usage summary. You'll bill the usage summary later. 

## Invoicing Usage Summaries with Price Schedules

When you invoice a usage order product, Salesforce Billing evaluates its related usage summaries. If a usage summary's next billing date ends on or before the invoice's target date, Salesforce Billing includes that summary's usage data in the resulting invoice line. This way, all usage summaries that end before the invoice target date are billed on the invoice. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

For a detailed review of usage billing setup, check out [Usage Rating and Processing](#).

-  **Important** Usage summaries also have a next billing date, which Salesforce Billing uses instead of the related order product's next billing date. By default, the summary's next billing date inherits the value of the Summary End Date field. If the summary's Override Summary End Date field has a value, the next billing date uses that instead.

Remember, a usage summary multiplies its unit price by its billed quantity to determine its unbilled subtotal. As Salesforce Billing invoices the usage summary, Salesforce Billing subtracts the invoiced balance from the unbilled subtotal and adds it to the invoice line's subtotal. This procedure lets you monitor how much of your usage summary has been billed. When invoicing finishes successfully, the invoice line has the original balance of the usage summary's unbilled subtotal, while the summary's current unbilled subtotal has been reduced to zero.

Your invoice also line inherits its start date and end date from the usage summary's start date and end date.

-  **Important** You can add usage to a usage summary only when the usage summary has a status of New. During and after billing, the summary can't receive further usage.

-  **Example** Let's say you invoice a usage order product on 08/01/18. The invoice has a target date of 08/01/2018, and the order product contains the following usage summaries.

Usage Summary	Next Billing Date
US-001	07/31/18
US-002	08/31/18
US-003	09/30/18

In this case, Salesforce Billing invoices only US-001. Let's look at that summary's billing details and see how they flow into the resulting invoice line.

#### **Usage Summary US-001 (Before Invoicing)**

Summary Start Date: 07/01/18

Summary End Date: 07/31/18

Unbilled Quantity: 5000

Unit Price: \$0.50

Unbilled Subtotal: \$2500

#### **Invoice Line for US-001**

Start Date: 07/01/18

End Date: 07/31/18

Subtotal: \$2500

#### **Usage Summary US-001 (After Invoicing)**

Unbilled Quantity: 0

Billed Quantity: 5000

Unbilled Subtotal: \$0

## Invoice an Order Using Bill Now

Create an invoice from one of your orders. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. Go to the order you want to invoice. Remember, you can invoice only active orders.
2. Select the **Bill Now** field and save your order record.

Salesforce Billing creates an invoice as follows.

- The invoice sets its invoice date to the date you created the order.
- The invoice bases its due date off the invoice date and your order's payment terms. For example, if your payment terms are net 30, your due date is 30 days after the invoice date.
- The Bill Now process sets the invoice's target date to the earliest Next Billing Date found among invoiceable order products. Remember, order product Next Billing Dates are compared to the invoice target date to qualify the order product for inclusion on the invoice. Only order products with next billing dates on or before the target date are included as invoice lines. For more information on the Next Billing Date field, review [Understanding Next Billing Date](#)
- The invoice has a draft status and an unpaid payment status.
- The invoice's balance reflects the sum of your invoice line balances.

You can now perform several tasks from your invoice record.

- If you made a mistake in creating your invoice and need to change it, you can cancel and rebill it.
- If your invoice does not contain any errors and you're ready to apply payments, you can use automation or manual edits to change your invoice's status to Posted.

## Hold Billing

Prevent an order from billing until it meets certain criteria. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Set the order product field Hold Billing to Yes to hold or suspend billing. While Yes is active, Salesforce Billing keeps the order active but does not pick it up for billing until the hold is released. Release the hold by setting the Hold Billing field to No or null.

Hold Billing does not change the amount that is billed for the order product. Salesforce Billing creates an invoice as soon as an Order Product is no longer on hold.

Consider the distinction between when to activate an inactive order and when to hold billing. If your customer hasn't received a PO number, you'd want to ask the following questions:

- Can you consider the order as a booking without getting the PO number?
- Should you track revenue for this order without having the PO number?

Answering yes to the above means that you have a hold billing scenario. Otherwise, you have an order activation process.

If an order product's Hold Billing field is selected, Salesforce Billing will hold billing for the order product and all its usage summaries. Bill Now and invoice runs will not pick the order product or its usage

summaries up for invoicing.

## Proration Types for Partial Billing Periods

Salesforce Billing uses proration to calculate balances for invoice lines that cover partial billing periods. While the invoice line doesn't have a prorate multiplier field, Salesforce Billing still calculates one behind the scenes and multiplies it by the order product's billable unit price to determine the invoice line's balance. The Salesforce Billing invoice package setting Proration Type offers several options for the method used to calculate the prorate multiplier. (Salesforce Billing Managed Package)

Based on your billing period date ranges, different Proration Type values cause different invoice line balances for the same billing period. Let's look at a few examples. Your order product has the following values:

- Billable Unit Price: \$1000
- Billing Type: Arrears
- Billing Frequency: Monthly
- Start Date: 05/23/2019
- End Date: 09/30/2019
- Next Billing Date: 06/01/2019

The resulting invoice line has a 9-day period that starts on 05/23/19 and ends on 05/31/19. Its subtotal will vary based on the package's Proration Type value.

Proration Type	Description	Invoice Line Balance Calculation
Calendar Days	Divide the number of invoiced days in the month by the total number of days in the month.	The invoice line falls in May, which has 31 days. We use $(9 \div 31)$ and multiply that by the order product's billable unit price of \$1000 to get \$290.32.
30 Days	Divide the number of invoiced days in the month by 30.	$(9 \div 30) * \$1000 = \$300$
Monthly (CPQ Formula)	Divide the number of invoiced days in the month by $(365/12)$ .	$(9 \div (365 \div 12)) * \$1000 = \$295.89$

Salesforce Billing recommends choosing a proration type based on your organization's billing and invoicing needs.

## Grouping Order Products into Invoices

Salesforce Billing creates invoices for different groups of order products based on several order and order product fields. This process is useful for invoicing certain types of order products separately from your

other order products. It's also useful if your account managers want one invoice record for all their orders. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and later

---

When your invoice run evaluates an order, it checks several fields when deciding how to group order products as invoice lines on an invoice.

- By default, the invoice run groups order products by their matching Billing Account fields.
- If the order product billing account is null, the invoice run uses the order's Billing Account field.
- If the order's billing account is null, the invoice run uses the order's Account field.

If the billing account has order products with different payment terms, it then groups the order products by matching payment terms.

You can define more levels of grouping with the order product's Invoice Group field. After your invoice run groups order products by their billing account and payment terms, it then considers the order's invoice grouping.

### Contract Number

Order products are grouped by matching contract numbers. Salesforce Billing references the contract number on the order product's originating subscription record.

### Order

Salesforce Billing places all of this order's order products on the same invoice.

### Order PO Number

Order products are grouped by matching order PO numbers.

### Legal Entity

Order products are grouped by matching legal entities.

### Separate Invoice

Every order product on this order is placed into a separate invoice.

### Invoice Group ID

Users can create a custom Invoice Group ID field. Salesforce Billing groups order products by matching invoice group ID values.

Finally, if any order products in your group have different currencies, each combination of the invoice group field and a currency type receives a separate invoice.

[Create Invoice Group IDs](#)

Salesforce Billing offers a way to customize invoice groups so that customers can bill order products matching certain criteria. This process is useful for separating invoices that don't share the same billing or payment terms. It's also helpful for combining several order products into one invoice. By defining an Invoice Group ID, you get to control how and when you invoice an order product. (Salesforce Billing Managed Package)

### **Billing Rule Amendment Settings**

Use billing rules to adjust billing dates following amendments to recurring service plans. (Salesforce Billing Managed Package)

### **Consolidating Order Products to One Invoice Line**

Salesforce Billing can combine the data from an order product and a revised order product into one invoice line. This feature is useful if you invoice a recurring order product, amend it, and must account for the amended order product on future invoices. (Salesforce Billing Managed Package)

## Billing Rule Amendment Settings

Use billing rules to adjust billing dates following amendments to recurring service plans. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

Amendment orders calculate their billing day of the month the same way as standard orders: The start date is based on the Default Order Start Date option in the Salesforce CPQ package settings. Billing Day of Month inherits the order's start date if the CPQ package setting, Align billing day of month to Order start date, is enabled. Otherwise, users must manually set the billing day of the month. The billing day of the month then drives the order product's next charge date and next billing date.

Because amendment orders can have start dates that differ from the original order, there are situations where you want the original order and the amendment order to be billed on the same day. To accomplish this change, update the Billing Day of Month value in the amendment order.

You can set the billing day of the month:

- Manually.
- By using automation such as a flow.
- By using Proration Day of Month on the quote.
- By using an Order Management plug-in.
- By setting a default value on the Billing Day of Month picklist.

In some situations, account for proration impacts when aligning amended order products to the original order product's Last Charge to Date.

By default, the Align to Original Order Product field is set to Do not align amended Order Product. See [Aligning Billing Dates for Amended Order Products](#).

## See Also

- [Align CPQ and Billing Cancellation Based on Billing Periods](#)
- [Plug-in Package Settings](#)

## Consolidating Order Products to One Invoice Line

Salesforce Billing can combine the data from an order product and a revised order product into one invoice line. This feature is useful if you invoice a recurring order product, amend it, and must account for the amended order product on future invoices. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and later

---

When you amend an order product in Salesforce CPQ and order the amendment quote, you receive a new order product that represents the changes. For example, if you amended an SPM subscription to have two more units and then ordered the amended quote, the resulting order product has a quantity of two. Its Revised Order Product field also looks up to the original order product. Finally, its Contract Action field lists the type of change made from the original order product – in this case, Quantity Increase.

The original and amendment order products likely have different start dates, end dates, quantities, and prices. When you invoice the amendment order product, Salesforce Billing might consolidate the information on the original order product and the amendment order product into one invoice line. To manage this consolidation, two or more subinvoice line records are created and combined to create the invoice line on the invoice. On the invoice line, the Invoice Line Type field is set to Merged to indicate the presence of subinvoice lines. The Sub Invoice Lines related list can be added to the Invoice Line layout as needed.

The final invoice line made from subinvoice lines shows the sum or difference in important price and quantity fields. For example, let's say your original order product has a quantity of 20. You amend the order's contract and reduce the quantity of the equivalent quote line by 5, and then order the amendment quote. You end up with the original order product, which has a quantity of 20, and the new order product, which has a quantity of -5. The invoice line created from the subinvoice lines created from these order products shows a quantity of 15.

 **Note** Subinvoice lines for cancellation orders roll up to one invoice line with a quantity of zero on the resulting invoice.

Salesforce Billing checks the following fields to determine whether two order products create subinvoice lines during the invoicing process.

- Billing Account
- Billing Day of Month
- Billing Frequency
- Billing GL Treatment

- Billing Treatment
- Discount Tier
- End Date
- Finance Book
- Invoice Grouping
- Legal Entity
- Pricing Tier
- Revenue Treatment
- Revenue GL Treatment
- Revised Order Product
- Start Date
- Tax Treatment
- Any other validation fields

In Salesforce Billing Summer '18 and later, the Revised Order Product and Contract Action fields are populated when you order your amendment quote. In earlier versions, you have to contract your amended order product to populate them so that Salesforce Billing can create subinvoice lines.

As of Winter '19, Salesforce Billing creates subinvoice lines for billing transactions related to invoice plan order products.

 **Example** Your order product represents 100 six-month data security subscriptions that begin on January 1. Each subscription has a unit price of \$10, so your total order product price is \$6000. Your invoice for January contains an invoice line with the following field values.

**Invoice Line #**

0001

**Start Date**

01/01/17

**End Date**

01/31/17

**Quantity**

100

**Unit Price**

\$10

**Total**

\$1000

At the end of January, your customer decides to remove 20 subscriptions from their account. When you amend the order product, you create an amendment order product that starts on 02/01/2017. It has a

quantity of -20, unit price of \$10, and total price of \$1000.

Order Product	Revised Order Product	Start Date	End Date	Quantity	Unit Price	Total
1	OP-0001	02/01/17	02/28/17	100	\$10	\$1000
2	OP-0001	02/01/17	02/28/17	-20	\$10	\$-200

When invoicing your amendment order product in February, Salesforce Billing creates two subinvoice lines for the billing period based on your order product's pricing and quantity, and your revised order product's pricing and quantity. Salesforce Billing adds the quantity and start dates of your subinvoice lines together to create field values for your February invoice line.

**Invoice Line #**

0002

**Start Date**

02/01/17

**End Date**

02/28/17

**Quantity**

80

**Unit Price**

\$10

**Total**

\$800

## Invoice Schedulers for Automating Invoice Creation

Set up an invoice scheduler for your billable order products so invoices can be generated automatically. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later

---

## Invoice Schedulers

An invoice scheduler uses an invoice run to evaluate whether and when an order product gets invoiced. The invoice scheduler's Type field defines the time frame for your invoice run: Daily, Weekly, Monthly, or Once. After selecting the type, choose a date and time for the invoicing process to begin, such as the starting date of a subscription or service. Then set your target dates as follows:

- Target Date: Set a date for a one-time invoice run.
- Target Day of Week: Choose a number corresponding with a day of the week (1=Saturday) for weekly invoice runs.
- Target Day of Month: Choose the day of the month for monthly invoice runs.

Use your scheduler's invoice date fields to define the start date of the invoices that your invoice run creates. These dates must also match the time frame you set with the Type field.

- Invoice Date: Use this field if your invoice run is a one-time event.
- Invoice Day of Week: Choose a number for the day of the week (1=Saturday) if your invoice runs every week.
- Invoice Day of Month: Use this field if your invoice run is a monthly event.

 **Note** If the invoice day of month isn't in a certain month, the scheduler can't launch an invoice run. For example, if the invoice day of month was 31, the scheduler can't launch a run in months with 30, 29, or 28 days. You must add the Run at End of Month field to the invoice scheduler's page layout, then create an invoice schedule and select **Run at End of Month**. Select a Start Date Time that's on the last day of the current month. As long as the invoice scheduler is active, future invoice runs launch on the last day of the month.

Select one or more invoice batches using the multi-select picklist to include those batches in the run. If this field is set to null, only those orders with null in the batch are picked up.

The invoice scheduler kicks off invoice runs when it hits the start time that you specify. For example, if you set up a monthly invoice scheduler to run on the 15th of every month, it runs on the next occurrence of the 15th of the month. You can't start a monthly invoice scheduler on the same day of the month for a future month, unless the date in your current month has passed. For example, if today is August 15, you can't set up a monthly invoice schedule to start on September 20.

 **Important** The Invoice Scheduler creator must have all the permissions to create and post invoices in addition to the permission to author Apex. See [Salesforce Billing Permission Requirements](#). Daily invoice schedulers must use the record's creation date for their target date and start date time. The start date time's timestamp can be any point in the future up to 11:59:59 PM.

## Invoice Runs

When your invoice scheduler hits its target, it creates an invoice run. An invoice run is like a net that trawls for eligible order products. When an unbilled order product matches the invoice run's criteria, it's

included in the next invoice. Salesforce Billing stores Invoice Run criteria as a record in the Invoice Scheduler's Invoice Run Related List.

The invoice run inherits its Target Date from one of these values.

- Invoice scheduler's Target Date value
- A specific date based on your Target Day of Week field or Target Day of Month field values

The invoice run also inherits its Invoice Date from your scheduler's Invoice Date, or a date based on the target day of week or month. All invoices generated from an invoice run inherit their start dates from the Invoice Run's Invoice Date.

## Invoice Run Criteria

### All Order Products

- Activated checkbox is selected.
- The order's invoice batch must match the invoice scheduler's invoice batch.
- Next Billing Date or Override Next Billing Date must be before or on your invoice run's Target Date.
- In multicurrency orgs, the currency matches your invoice run's currency, or the invoice scheduler's Include All Currencies field is selected.
- Hold Billing is set to No.
- Invoice Run Processing Status has a value of Pending Billing. If a billing rule's Generate Invoice field has a value of Yes, all the rule's order products are created with an invoice run processing status set to Pending Billing. Otherwise, order products are created with an invoice run process status of Will Not Invoice. We recommend using Generate Invoice to control invoice run processing status on your order products. If an invoice run ignores an uninvoiced order product with a Will Not Invoice status, and a user changes the status to Pending Billing, Salesforce Billing won't invoice the order product until the next invoice run or Bill Now process.

### Order Products with Usage Summaries

- All of the previously listed order product criteria, and the following:
- The invoice run's parent invoice scheduler has the Bill Usage Charges field selected.
- Invoice Run Processing Status has a value of Pending Billing. When the charge type is Usage, this field's value is also controlled by the same billing rule that controls the usage summary's parent order product. Otherwise, it's controlled by the Billing Rule on the Consumption Schedule.
- The Next Billing Date of any usage summaries that you want to invoice must be before the invoice run's target date.
- Usage Summaries are picked up only for invoicing if both the invoice run's Bill Usage Charges field is True and the Next Billing Date on the Usage Summary is before the invoice run's Target Date value.

After your invoice run has evaluated its available order products, it generates an invoice based on the billing data of the invoiceable order products.

**!** **Important** Invoice schedulers update several order product fields during an invoice run. If you

added triggers, process builders, flows, or workflow rules to orders, order products, invoices, or invoice lines, we recommend testing them during an invoice run in a development org.



**Example** You want a one-time invoice scheduler that begins immediately on August 1 and picks up all your unbilled order products. Create an invoice scheduler record as follows.

- Target Date: 08/01/2018
- Start Date Time: 08/01/2018 sometime before 11:59:59 PM
- Invoice Date: 08/01/2018

Your org contains the following product and its related order product:Product

- Standard Price: \$1000
- Billing Frequency: Monthly
- Charge Type: Recurring
- Billing Type: Advance
- Active: True

Order Product

- Quantity: 20
- Start Date: 08/01/2018
- End Date: 07/31/2019

On August 1, your invoice schedule launches your invoice run, which creates the following invoice:Invoice

- Invoice Date: 08/01/2018
- Subtotal: \$200,000

Your order product also updates its billed amount to \$200,000.

### Guidelines for Invoice Runs

When you make an invoice scheduler and define parameters for its invoice runs, consider key guidelines. (Salesforce Billing Managed Package)

### Clean Up Invoice Runs

Clean-up invoice runs to correct any system errors, and reset order products and usage summaries.

After an invoice run clean-up, another invoice run picks up eligible order products and usage summaries. (Salesforce Billing Managed Package)

### Cancel Upcoming Invoice Runs

Cancel upcoming invoice runs by deleting the Scheduled Job record of the parent invoice scheduler. (Salesforce Billing Managed Package)

## Guidelines for Invoice Runs

When you make an invoice scheduler and define parameters for its invoice runs, consider key guidelines.

(Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later

---

Invoice runs are subject to Salesforce governor limits. If your invoice runs process many order products, or many order products and usage summaries, you may encounter Apex errors. Customizations on processed objects may cause Apex errors as well. We recommend testing your invoice runs for scalability when the total number of records processed per invoice run exceeds 200,000, and when your invoice runs process objects that have customizations.

You can use target dates to generate invoices ahead of the actual invoice date. For example, you can generate an end-of-month invoice before the end of the month. This process is useful if you want to create an invoice for internal review before sending it to customers. If you wanted to generate your end-of-January invoices on January 29 but keep the invoice start date as January 31, you would set the following fields on the invoice scheduler:

- Start Date/Time: 1/29/xx xx:xx:xx
- Target Date: 1/31/xx
- Invoice Date: 1/31/xx

This setup causes the invoices to generate on January 29 while billing through January 31, with a start date of January 31 as well.

By default, invoice runs generate invoices in draft status. You can then post all the run's draft invoices by selecting the invoice run's **Post All Invoices** button. The Post All Invoices process doesn't affect canceled invoices. If you select the invoice scheduler's Automatically Post Invoices field, each invoice run automatically posts its invoices at the end of the run.

 **Tip** The invoice run's **Clean Up Invoices** button cancels any of the run's invoices that have an Error, Initiated or Draft status, and rolls related order products and usage summaries from Error or In Progress back to Pending Billing. It's useful if you need to quickly reset many invoices that generate incorrectly. The Clean Up Invoices process doesn't affect posted invoices. If you must cancel posted invoices, you must use the Cancel and Rebill process on them one at a time.

If you want your invoice run to evaluate only order products from a certain account or accounts, set your scheduler and account to have matching Invoice Batch values.

Daily invoice schedulers must use the record's creation date for their target date and start date time. The start date time's timestamp can be any time in the future.

## Clean Up Invoice Runs

Clean-up invoice runs to correct any system errors, and reset order products and usage summaries. After an invoice run clean-up, another invoice run picks up eligible order products and usage summaries.

(Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Classic and Lightning Experience

---

Available in: **Professional, Enterprise, Unlimited, and Developer** Editions

---

Most invoice runs are successful but some invoice runs encounter problems that are out of your control. An invoice run can fail because of permission issues, connectivity issues, system failures, row lock issues, or exceeding a governor limit. When such errors occur, the order products and usage summaries in the run aren't processed.

To clean up invoice runs, click **Clean Up** on the Invoice Run page. In Salesforce Classic, add Clean Up as a custom button on the Invoice Run page.

Invoice run clean-up makes these changes.

- Changes the status of invoices from Error, Initiated, or Draft to Canceled.
- Changes the status of invoice lines from Error or Draft to Canceled.
- Changes the status of order products from Error or In Progress to Pending Billing.
- Changes the status of usage summaries from Error or In Progress to Pending Billing.
- Removes the invoice run and invoice run start date of order products and usage summaries.

These actions ensure that all the data from a failed invoice run is added to a new invoice run. Batch Apex Error Events generate detailed error logs and support multiple clean-up scenarios.

The Completed with errors value in the CleanUp Status field of the Invoice Run object helps identify the processed invoice run clean-up errors correctly. Add this value to the CleanUp Status field from Object Manager or from the Picklist Value Sets setup page.

You can also clean-up completed invoice runs without errors by changing the run's status from Completed to Completed with errors. This clean-up runs on all affected records, except for invoices that have already been posted.

### See Also

[Add or Edit Picklist Values](#)

[Global Picklist Value Set](#)

## Cancel Upcoming Invoice Runs

Cancel upcoming invoice runs by deleting the Scheduled Job record of the parent invoice scheduler.  
(Salesforce Billing Managed Package)

Canceling upcoming invoice runs is useful if you changed the terms or amounts of an order in a way that disqualifies it from runs created by your current invoice scheduler. After cancellation, create an invoice scheduler with invoice runs configured to pick up your revised order.

 **Warning**

- Deleting only the invoice scheduler doesn't stop its scheduled jobs. If you delete the scheduler without deleting its scheduled jobs, Salesforce Billing continues running the jobs, which can cause unnecessary or inaccurate invoices.
- You can't reinstate an invoice scheduler after you delete its scheduled job.

1. From Setup, in the Quick Find box, enter *Scheduled Jobs*, and then select **Scheduled Jobs**.
2. Find the job name of the invoice scheduler that you want to cancel. The job name matches the value of the Invoice Scheduler Name field on the invoice scheduler that created the invoice run.
3. In the Action column next to the job, click **Del**, and then click **OK**.  
We recommend making a note of the cancellation in the invoice scheduler's Notes field.

You can delete the invoice scheduler record after you delete its scheduled job. However, we recommend keeping the canceled invoice scheduler record for bookkeeping purposes.

## Invoice Presentation

Salesforce Billing provides several options for creating documents based on your invoices. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

#### [Conga Invoice Generation for Salesforce Billing](#)

Conga Invoice Generation for Salesforce Billing allows customers to customize and export invoices, statements, and receipts. (Salesforce Billing Managed Package)

## Conga Invoice Generation for Salesforce Billing

Conga Invoice Generation for Salesforce Billing allows customers to customize and export invoices, statements, and receipts. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19

---

To get started with Conga Invoice Generation for Salesforce Billing, check out Conga documentation.

- [Using Conga Invoice Generation](#)

 **Warning**

- All questions and feedback regarding Conga Invoice Generation (CIG) for Salesforce Billing and Conga Quote Generation (CQG) for Salesforce CPQ should be directed to Salesforce CPQ & Billing Support. For more information, check out [Logging a Case for Conga Quote Generation or Conga Invoice Generation](#).
- Conga Invoice Generation requires a license that's available for free with paid subscriptions to Salesforce Billing. If you're unsure whether you have this license, log a case with Salesforce CPQ & Billing Support.

## Managing Invoice Lines Related to Your Assets

An asset contains several fields and relationships to help you track how often and for how much you've invoiced a customer. You can create a relationship by adding an asset to the order product's Lifecycle-Managed Asset field. The asset's Invoice Line Balance field shows the sum of the Balance fields on the posted invoice lines related to the asset. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '20 and later

---

The invoice line's Asset field inherits the value of the parent order product's Lifecycle-Managed Asset field.

You can update the Lifecycle-Managed Asset field using a process or workflow rule. Your Lifecycle-Managed Asset value doesn't have to be the same as the order product's Asset field.

Salesforce CPQ uses the Asset field for CPQ features, such as contracting and amendments. Salesforce Billing uses the Lifecycle-Managed Asset field for Billing features, such as invoicing.

These fields aren't included in page layouts by default. To display a field to your users, add it to a page layout.

- Lifecycle-Managed Asset field on Order Product
- Invoice Line Balance field on Asset
- Asset field on Invoice Line

## Bill Now

Users can manually invoice an order by selecting Bill Now on the order record and saving. It's a useful feature when a sales rep must quickly produce an initial or final invoice. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing 8.0 and Later

Selecting **Bill Now** and saving the record triggers the invoice creation process for the order. CPQ automatically deselects Bill Now. An invoice is created if one or more products on the order meet these base requirements.

- Activated
- Next Billing Date has a value
- Hold Billing field is No or null
- Invoice Run Processing Status field is Pending Billing
- Has a tax rule that can either successfully calculate tax, or has a Taxable (Yes/No) value of No
- Has a Billing Rule
- Has a Revenue Recognition Rule
- Has a billable unit price
- Has a charge type
- Has a Billing Frequency when Charge Type isn't equal to One-Time
- Has a start date
- Usage order products require an invoiceable usage summary
- Pending Billing Amount must be greater than 0.

Bill Now works on an inactive order if one or more order products are activated and the order meets the base requirements.

Invoices are created in Draft status.

It's good practice to enable field tracking for Bill Now for troubleshooting and audit purposes.

The Bill Now process sets the invoice's target date to the earliest Next Billing Date found among invoiceable order products. Remember, order product Next Billing Dates are compared to the invoice target date to qualify the order product for inclusion on the invoice. Only order products with next billing dates on or before the target date are included as invoice lines. For more information on Next Billing Date, review [Understanding Next Billing Date](#).

### Important

- We recommend that users don't create workflow rules, custom triggers, or process builders that invoke the Bill Now process.
- We don't recommend using automation to post the invoice after it's created due to multiple updates required to create the invoice.
- Salesforce Billing can't use Bill Now to create a single invoice from an amendment order and the original order.
- Salesforce Billing doesn't support using Bill Now in a list view to select multiple invoices.
- For Bill Now, no option is available for selecting a target date. Target date is defined by comparing

the next billing dates of all order products on an order with the order's effective date. If the minimum next billing date is earlier than the effective date, we define target date as the effective date. If the minimum next billing date is later than the effective date, target date equals the minimum next billing date.

- For invoices created from a combination of usage summary and order product records using the standard tax integration, the TaxStatus on the invoice line created from the usage summary is set to Queued. TaxStatus is updated correctly when the invoice posts or a field update is made to the draft invoice line.



**Example** Your order contains three order products: Two with a Next Billing Date of February 20, and one with a Next Billing Date of February 26. If you select **Bill Now** on 01/09, your invoice has a target date of February 20. The invoice contains invoice lines for your first two order products. It doesn't contain a line for the order product with a Next Billing Date of 02/26, since that date falls after the invoice's target date.



## Invoice Fields

Before you start working with billing and invoicing, review invoice fields and attributes. (Salesforce Billing Managed Package)

Field	Definition
<b>Account</b>	Parent account for the invoice. Inherited from the order product.
<b>Action</b>	Shows whether a user has credited the invoice by either selecting the Cancel and Rebill button or the Credit button.
<b>Allocations Against Invoice</b>	Total number of payment allocations made against the invoice.
<b>Allocations Against Invoice Lines</b>	Total number of payment allocations made against the invoice's invoice lines.
<b>AR Status</b>	Shows whether the invoice has been credited or canceled and rebilled.
<b>Asset</b>	Inherited by default from the Lifecycle-Managed Asset field on the invoice's parent order product. Can be overridden with a different asset.
<b>Balance</b>	The sum of all balances on the invoice's invoice lines.
<b>Base Currency</b>	If you have currency conversion requirements, you

Field	Definition
	can define your base currency here.
<b>Base Currency Amount</b>	If you have currency conversion requirements, you can define your base currency amount here. When you export your data, you can use this value in your conversions.
<b>Base Currency FX Date</b>	If you have currency conversion requirements, you can define your base currency FX date here. When you export your data, you can use this value in your conversions.
<b>Base Currency FX Rate</b>	If you have currency conversion requirements, you can define your base currency FX rate here. When you export your data, you can use this value in your conversions.
<b>Bill To Contact</b>	The user who receives information about the invoice.
<b>Corrective Action (Payment Run)</b>	When this field has a value of Action Required, the invoice is locked from receiving payments until the issue defined in the Last Payment Run Processing Message field is resolved.
<b>Credits</b>	The sum of credit notes allocated against the invoice and its invoice lines.
<b>Days Outstanding</b>	The number of days past the invoice due date.
<b>Debit Payments</b>	The sum of payments made to debit notes allocated against the invoice and its invoice lines.
<b>Debits</b>	The sum of debit notes allocated against the invoice and its invoice lines.
<b>Declined Payment Count (Payment Run)</b>	Salesforce Billing doesn't populate this field by default. You can add your own customization to it if you want to track the number of times a payment run that targets the invoice has been unable to pay it.
<b>Default Payment Type</b>	Used in payment runs. A payment run picks up an invoice if the run and the invoice have matching Default Payment Type values.
<b>Due Date</b>	The date by which customers must pay their invoice. We've provided it for users who want to

Field	Definition
	add their own due date and collections features for AR and AP.
<b>Impact Amount</b>	Inherits the value of the invoice's total amount. We've provided this field for users who must monitor impact amounts separately as part of their billing and collections processes.
<b>Invoice Date</b>	Date when the invoice was created by either the Bill Now process or an invoice run.
<b>Invoice Line Balance</b>	Total balance of invoice lines on the invoice.
<b>Invoice Posted Date</b>	Date when a user or process posted the invoice.
<b>Invoice Run (Created By)</b>	Shows the invoice run that created the invoice.
<b>Last Payment Run</b>	Most recent payment run that evaluated the invoice for payment.
<b>Last Payment Run Processing Message</b>	If a payment run that targeted the invoice encountered an issue, this field shows information on how to correct the issue and unlock the invoice for receiving payments.
<b>Negative Invoice Line Conversion</b>	<p>The sum of all Negative Line Conversion fields on the invoice's invoice lines. It's helpful when evaluating whether to run the <code>blng__NegativeInvoiceToCreditNoteAction</code> REST service.</p> <p>When the <code>blng__NegativeInvoiceToCreditNoteAction</code> REST service evaluates an invoice and finds at least one invoice line with a negative balance, the service creates a credit note containing one credit note line for each negative invoice line. Each credit note line has a balance equal to the positive equivalent of the corresponding negative invoice line. You can then allocate the credit note lines to your negative invoice lines to zero out your invoice's negative lines.</p>
<b>Notes</b>	Users can provide additional notes and information.
<b>Number of Invoice Lines</b>	Total number of invoice lines.

Field	Definition
<b>Order</b>	Populated only when Bill Now is used to create the invoice, or during an invoice run if the order product Invoice Group field has a value of Order.
<b>Override Autopay Payment Method</b>	<p>Shows a payment method. If this field has a value when a payment run evaluates the invoice, the invoice's default payment type ignores the account's payment method. The default payment type instead inherits its Type value from the payment method defined in Override Autopay Payment Method.</p> <p>We recommend using this setup when you must temporarily change the payment method used for a specific invoice.</p>
<b>Payment Batch</b>	<p>If a payment scheduler has a payment batch value, the payment run evaluates only invoices with matching payment batch values.</p> <p>If this field is null, payment runs evaluate the invoice only if the payment run's batch is also null.</p>
<b>Payment Run</b>	The payment run that evaluated and made payment against the invoice.
<b>Payment Run Processing Message</b>	This field doesn't have any functionality by default.
<b>Payment Status</b>	A formula field that shows Paid when the invoice balance is zero (0). If payments are applied but there's still a balance, the formula returns Partially Paid. If there are no payments applied, then the field shows Unpaid. If only credit notes are allocated to an invoice and the balance is still positive, the payment status remains Unpaid. Only applying payments to the invoice can change the status to Partially Paid. If this is a requirement, create a custom field.
<b>Payments</b>	The number of payments made against this invoice.

## Invoice Line Fields

Before you start working with billing and invoicing, review invoice line fields and attributes. (Salesforce Billing Managed Package)

Field	Definition
Allocated Revenue Amount	<p>Total revenue impact of this invoice line. Can differ from the subtotal and total amount (with tax). Used when reallocating revenue among several transactions.</p> <p>Can be configured to create revenue schedules for this invoice line using this field as the amount.</p>
Allocations Against Debit Note Lines	<p>Sum of all allocations against debit note lines allocated to this invoice line. Credit note and payment allocations reduce the balance of the debit note allocations.</p>
Allocations Against Invoice Lines	<p>Sum of all allocations made against this invoice line. Credit note and payment allocations reduce the balance of the invoice line. Debit note allocations increase the balance of the invoice line.</p>
AR Status	<p>Whether the invoice line was credited or canceled and rebilled.</p>
Balance	<p>Amount of the invoice that isn't settled. Equals total amount (with tax) – allocations + unallocations.</p>
Balance without Debits	<p>Amount of the invoice that isn't settled, excluding debit note allocations. Equals total amount (with tax) – payments + unallocations against invoice lines – credits.</p>
Base Currency	<p>Base currency of this credit note line. Used for reporting purposes.</p>
Base Currency Amount	<p>Credit note line's total amount, converted into the base currency.</p>
Base Currency FX Date	<p>Date used to determine the FX rate when calculating the base currency amount.</p>
Base Currency FX Rate	<p>FX rate used to convert the credit note amount to</p>

Field	Definition
	the base currency amount.
Billing Finance Book	This invoice line's transactions report to this finance book.
Billing Finance Period	This invoice line's transactions report to this finance period within the finance book.
Billing Frequency	How often Salesforce Billing invoices this line's originating order product.
Billing GL Rule	Lookup to the GL rule that defines the GL treatment for this invoice line.
Billing GL Treatment	Lookup to the GL treatment that defines the debit and credit GL accounts that record this invoice line.
Billing Rule	Lookup to the billing rule used to generate this invoice line.
Billing Transaction	
Billing Treatment	Lookup to the billing treatment used to generate this invoice line.
Calculated Quantity	Number of billing periods charged on the invoice line.
Charge Date	
Charge Type	Whether this invoice line is recurring, one-time, or usage (legacy).
Converted Credit Note Line	For converted negative invoice lines, the credit note line that was created as part of the conversion.
Credits	Sum of all credit note lines allocated to this invoice line.
Debit Payments	Net amount of all payments made to debit note allocations on this invoice line. Equals debits (payment allocations) – debits (payment unallocations).
Debits	Sum of all debit note lines allocated to this invoice line.
Debits (Payment Allocations)	Sum of all payments made against the debit note

Field	Definition
	allocated to this invoice line.
Debits (Payment Unallocations)	Sum of all payments unallocated from debit notes allocated to this invoice line.
End Date	End date of the billed period for this invoice line. This value can impact the revenue. See Salesforce Help for details on configuring revenue recognition.
Grouped Quantity	
Impact Amount	Change to net receivables as a result of this credit note line. Positive credit note lines have a negative impact amount because they reduce the overall amount owed to the company.
Invoice	Parent invoice for this invoice line.
Invoice Line Type	<p>Whether this invoice line is marked Merged or Regular.</p> <ul style="list-style-type: none"> <li>• Merged means the invoice line contains invoice sublines.</li> <li>• Regular means the invoice line is standard and doesn't include sublines.</li> </ul>
Invoice Run	Invoice run that created this invoice line.
Legal Entity	Lookup to this invoice line's legal entity. Legal entities control which treatments are associated with this invoice line.
LegalEntityReference	
Negative Line Conversion	Amount of this invoice that converted to a credit note line. Only negative invoice lines are converted.
Net Allocations Against Invoice Lines	Net amount of all payments made against this invoice line. Equals (allocations against invoice lines) - (unallocations against invoice lines).
Notes	Reference notes that users can add. Notes fields don't have system logic associated.
Order Product	Order product that generated this invoice line.

Field	Definition
Override Initial Revenue End Date	This field lets users override the revenue period's configured end date when you're creating revenue schedules.
Override Initial Revenue Start Date	This field lets users override the revenue period's configured start date when you're creating revenue schedules.
Payments	Net amount of payments against this invoice line and any of its allocated debit note lines. Equals (amount allocated against invoice line) + (amount allocated against debit note lines) - (amount unallocated against invoice lines) - (amount unallocated against debit note lines).
Product	Product that created the order product being invoiced.
Quantity	Quantity invoiced from the order product.
Revenue Expected Amount	<p>This user-defined field is the expected revenue impact of this credit note line, if different from the credit note line amount. Can differ from the subtotal and total amount (with tax).</p> <p>Can be configured to create revenue schedules for this credit note line using this field as the amount on the revenue schedule.</p>
Revenue Liability Amount	<p>This user-defined field is the revenue liability associated with this credit note line. Can differ from the subtotal and total amount (with tax).</p> <p>Can be configured to create revenue schedules for this credit note line using this field as the amount on the revenue schedule.</p>
Revenue Most Likely Amount	This user-defined field is the most likely revenue impact of this credit note line for revenue calculations when the revenue amount is uncertain. Can differ from the subtotal and total amount (with tax).
Revenue Schedule Status	Whether revenue schedule creation for this invoice line is queued, is completed, or has encountered an error.

Field	Definition
Start Date	Start date of the billed period for this invoice line. This value can impact the revenue. See Salesforce Help for details on revenue recognition.
Status	Current state of the invoice line.
Subtotal	Amount to settle on this invoice line, excluding tax.
Tax	Total tax owed on this invoice line.
Tax City	City used during tax calculation. The invoice's tax address fields are used only if the invoice line's tax address fields are blank.
Tax Country	Country used during tax calculation. The invoice's tax address fields are used only if the invoice line's tax address fields are blank.
Tax Country (Override)	User-defined value that overrides the system-defined tax country.
Tax Error	
Tax Error Message	Descriptions of errors that occur during tax calculation.
Tax GL Rule	Lookup to the GL rule that defines the GL treatment of the tax for this invoice line.
Tax GL Treatment	Lookup to the GL treatment that defines the debit and credit GL accounts used to record the tax for this invoice line.
Tax Location Code (Override)	
Tax Percentage Applied	Total tax percentage applied to this line. Null if the line is nontaxable.
Tax Postal Code	Postal code used during tax calculation. The invoice's tax address fields are used only if the invoice line's tax address fields are blank.
Tax Rule	Lookup to the tax rule that defines the tax engine and tax data, such as address and date, to use when calculating tax for this invoice line.
Tax State	State used during tax calculation. The invoice's tax address fields are used only if the invoice line's tax address fields are blank.

Field	Definition
Tax Status	Status of the tax calculation for this invoice line.
Tax Street 1	Street address used during tax calculation. The invoice's tax address fields are used only if the invoice line's tax address fields are blank.
Tax Street 2	Street address used during tax calculation. The invoice's tax address fields are used only if the invoice line's tax address fields are blank.
Tax Treatment	Lookup to the Tax Treatment used to generate this invoice line.
Total Amount (with Tax)	Subtotal plus tax.
Unallocations Against Debit Note Lines	Sum of all unallocations made against prior allocations debit note lines on this invoice line.
Unallocations Against Invoice Lines	Sum of all unallocations made against prior allocations on this invoice line.
Unique ID	This field is system-generated and not updated or changed by users.
Unit Price	
Usage Summary	Lookup to the usage summary that was used to create the invoice line. Blank unless usage summary is the source for the invoice line.

## Editable Fields on Invoices and Invoice Lines

Salesforce Billing follows Generally Accepted Accounting Principles, which require that users must always be able to track details of invoices and invoices lines. To ensure that invoice information is accurate for accounting purposes, Salesforce Billing doesn't allow you to edit or delete certain fields on posted invoices and their lines. While some other fields are editable, changing them may prevent data from passing correctly during other Salesforce Billing processes such as payments or amended orders. When you work with invoices, review the fields that you can safely edit on invoices and invoice lines. (Salesforce Billing Managed Package)

 **Note** You can always edit custom fields on draft and posted invoices and invoice lines.

## Editable Fields for Draft and Posted Invoices

Field	API Name	Data Type	Editable on Draft Invoice	Editable on Posted Invoice
Base Currency	bIng__BaseCurrency__c	Text(20)	✓	✓
Base Currency Amount	bIng__BaseCurrencyAmount__c	Currency(16,2)	✓	✓
Base Currency FX Date	bIng__BaseCurrencyFXDate__c	Date	✓	✓
Base Currency FX Rate	bIng__BaseCurrencyFXRate__c	Number(16,2)	✓	✓
Bill To Contact	bIng__BillToContact__c	Lookup(Contact)	✓	✓
Notes	bIng__Notes__c	Long Text Area(32678)	✓	✓
Override Autopay Payment Method	bIng__PaymentMethod__c	Lookup(Payment Method)	✓	✓
Payment Batch	bIng__PaymentBatch__c	Picklist	✓	✓

## Edited Fields for Draft and Posted Invoice Lines

Field	API Name	Data Type	Editable on Draft Invoice	Editable on Posted Invoice
Base Currency	bIng__BaseCurrency__c	Text(20)	✓	✓
Base Currency Amount	bIng__BaseCurrencyAmount__c	Currency(16,2)	✓	✓
Base Currency FX Date	bIng__BaseCurrencyFXDate__c	Date	✓	✓
Base Currency FX Rate	bIng__BaseCurrencyFXRate__c	Number(16,2)	✓	✓
Notes	bIng__Notes__c	Long Text Area(32768)	✓	✓

Field	API Name	Data Type	Editable on Draft Invoice	Editable on Posted Invoice
Override Initial Revenue End Date	bIng__OverrideInitialRevenueEndDate__c	Date	✓	✓
Override Initial Revenue Start Date	bIng__OverrideInitialRevenueStartDate__c	Date	✓	✓
Revenue Allocation Amount	bIng__AllocatedRevenueAmount__c	Currency(16,2)	✓	✓
Revenue Expected Amount	bIng__RevenueExpectedAmount__c	Currency(16,2)	✓	✓
Revenue Liability Amount	bIng__RevenueLiabilityAmount__c	Currency(16,2)	✓	✓
Tax	bIng__TaxAmount__c	Currency(16, 2)	✓	
Tax Percentage Applied	bIng__TaxPercentageApplied__c	Percent(14, 4)	✓	

## Applying Taxes

---

Salesforce Billing can use internal or external tax engines to provide automatic tax calculations on your order products and invoice lines. The tax integration object acts as a bridge between your org and the external service. When you set up your org, you'll need to configure a few basic tax settings and then create your tax integration. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

---

#### Tax Calculations

Salesforce Billing provides an internal tax calculation engine and also lets you integrate with external tax engines. Both types of engines provide automatic tax calculations for your order products and invoice lines. (Salesforce Billing Managed Package)

#### Tax Integrations

Send data from Salesforce Billing to an external tax calculation service. Tax integrations are the Salesforce Billing objects that define the relationship between your Salesforce Billing org and the external service you use for tax calculation. (Salesforce Billing Managed Package)

## Tax Calculations

Salesforce Billing provides an internal tax calculation engine and also lets you integrate with external tax engines. Both types of engines provide automatic tax calculations for your order products and invoice lines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

When you use the internal tax engine, Salesforce Billing calculates tax upon order product and invoice line creation. Third-party tax engines use API tax calls for tax calculation. The tax is calculated and displayed on each order product and invoice line, which aggregate to the order and invoice tax totals respectively.

If you're also using Salesforce CPQ, you can click **Calculate Tax** on your quote to calculate an initial quote tax estimate. This functionality is available only with third-party tax engines and won't calculate using the Salesforce standard tax engine.

Quote-level and order-level tax calculation represents an estimated tax value, since these objects don't have access to complete billing information. Invoice-level tax represents the final calculated tax value that you'll use when billing customers.

The Salesforce Billing tax integration object acts as the bridge between your Salesforce org and your tax engine.

### Tax Calculation Results

Salesforce Billing updates several fields following tax calculation, depending on the active stage of the billing process. (Salesforce Billing Managed Package)

### Choosing a Tax Engine

When choosing your tax engine, consider your company's scale, complexity, and auditing plans. Third-party integrations often provide reporting services, while the standard tax engine requires you to handle reporting in a separate process. (Salesforce Billing Managed Package)

### Configure Salesforce Billing for Tax Calculation

To ensure tax calculation, Salesforce Billing requires values for several key fields. If you're hitting errors when calculating tax, make sure that these fields are populated correctly. (Salesforce Billing Managed Package)

### Line-Level Tax Calculation

Salesforce Billing calculates tax at the line level on both the order product and invoice Line objects. Each order product's tax amount sums to the Billed Tax field on the parent order, and each invoice line's tax amount sums to the Tax field on the parent invoice. (Salesforce Billing Managed Package)

## Tax Calculation Results

Salesforce Billing updates several fields following tax calculation, depending on the active stage of the billing process. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Order and invoice creation can occur at different times depending on requirements and billing process. Tax information on these records includes the actual calculation result and other relevant information, such as attributes driving the calculation and the status of the calculation process. This table provides a comprehensive list of fields that Salesforce Billing populates during and following tax calculation.

 **Note** Required fields for calculation vary based on the tax engine you're using.

 **Tip** To change the address driving tax calculation, use the Address Override fields on your order product.

Object	Field	Description
Order Product	Estimated Tax	The calculated tax estimate for the order product
Order Product	Tax Rule	Lookup to the tax rule that was applied on the product
Order Product	Tax Treatment	Lookup to the tax treatment associated to the tax rule and legal entity matching your order product
Order Product	Tax Calculation Status	Shows the status of your tax calculation. Values include: <ul style="list-style-type: none"><li>• Queued</li><li>• Processing</li><li>• Completed</li><li>• Error</li><li>• Warning</li></ul>
Order Product	Tax Error Message	Shows a description of why your tax calculation status has a value of Error
Order Product	Tax Code	Shows the tax code that

Object	Field	Description
		Salesforce Billing applied to the Order Product. The source of this field varies between the product and tax treatment records, depending on the tax engine.
Order	Estimated Tax	The total tax estimate for all taxable order products associated to the order
Invoice Line	Tax Street 1	The street address used in tax calculation
Invoice Line	Tax City	The city used in tax calculation
Invoice Line	Tax State	The state used in tax calculation
Invoice Line	Tax Postal Code	The postal code used in tax calculation
Invoice Line	Tax Country	The country used in tax calculation
Invoice Line	Tax Percentage Applied	The tax percentage that Salesforce Billing applies to your invoice line
Invoice Line	Tax Status	Shows the status of your tax calculation. Values are the same as the Tax Calculation field on the order product.
Invoice Line	Tax Error Message	Shows a description of why your tax calculation status has a value of Error
Invoice Line	Tax	The final calculated tax for the invoice line
Invoice Line	Total Tax	The sum of all taxes applied across this invoice's invoice lines
Invoice Line	Tax Status	Shows the status of your tax calculation. Values are the same as the Tax Calculation field on the order product.
Invoice Line	Tax Error Message	Shows a description of why your tax calculation status has a value

Object	Field	Description
		of Error

To change the address driving tax calculation, use the Address Override fields on your order product.

-  **Note** Taxes apply to usage products the same way as they apply to all other invoiced order products.

## Choosing a Tax Engine

When choosing your tax engine, consider your company's scale, complexity, and auditing plans. Third-party integrations often provide reporting services, while the standard tax engine requires you to handle reporting in a separate process. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

These tax engine options are available.

#### No Tax Engine

You can use Salesforce Billing without calculating tax. In this case, you don't set up your Tax Integration object. Use a tax rule with Taxable (Yes/No) set to No, and don't create a tax treatment. Salesforce Billing calculates invoice and invoice line amounts without tax.

#### Standard Tax Engine

Salesforce Billing provides an internal tax engine. Salesforce recommends using it for simple and static tax calculations, such as VAT. Enter and update all tax rates manually. See [Create a Tax Rate and Tax Code for Standard Tax Integrations](#). Salesforce doesn't provide tax rate data.

Create a tax rule with Taxable (Yes/No) set to Yes and a tax treatment that indicates the standard tax integration. See [Configure Salesforce Billing for Tax Calculation](#).

Tax is calculated only on the order products and invoice lines. The order and the invoice roll up the sum of the tax on the lines. Keep in mind that due to rounding tax values on the line, a small difference between the sum of the tax and the tax as calculated on the invoice total can occur.

When an order is created, the tax rule on each order product comes from the Product2 record. The tax treatment is assigned by matching the treatment that looks up to the rule with the same legal entity on the order product. The tax treatment is used to calculate the estimated tax on the order product. Changing the legal entity or the tax rule is allowed while the order product status is Draft. See [Defining Rules and Treatments](#).

When an invoice is created, Salesforce uses the standard integration to find a tax rate that matches the address, tax code, and legal entity. Salesforce then applies the tax percentage to the

appropriate invoice lines.

### Third-Party Tax Engine

Third-party tax engines are available in AppExchange. When using a third-party tax engine, set up the engine in the Tax Integrations object within Salesforce Billing according to the provider's instructions.

Create a tax rule with Taxable (Yes/No) set to Yes and a tax treatment that indicates the tax integration. See [Configure Salesforce Billing for Tax Calculation](#).

When an order is created, the tax rule on each order product comes from the Product2 record. The tax treatment is assigned by matching the treatment that looks up to the rule with the same legal entity on the order product. The tax treatment is used to calculate the estimated tax on the order product. Changing the legal entity or the tax rule is allowed while the order product status is Draft. See [Defining Rules and Treatments](#).

When an invoice is created, Salesforce calls the third-party engine for tax information based on the tax code from the tax treatment on the order product. Salesforce then applies the tax percentage to the appropriate invoice lines. Transactions are recorded in Salesforce and in the third-party engine.

Third-party tax engines can update their tax transaction records to align with Salesforce records. Your tax package provider can supply details. Third-party tax engines can also provide collection and remittance services that aren't available in the standard tax engine. External vendors can let you define where you collect and remit taxes. These definitions ensure that you're collecting the appropriate amount of taxes and remitting taxes to the correct jurisdiction based on city, state, or country.

-  **Note** When you use a third-party tax engines, Billing supports tax calculation for a maximum of 700 invoice lines. Custom development involving related objects such as Invoice, Invoice Line, Subinvoice Line, Order, and Order Products or any roll-up summaries that aggregate data to the related account can reduce this limit. Contact your tax integration provider for package specific details.

### Custom-Built Integrations

Salesforce Billing users can create and use custom integrations. See [Salesforce Billing Tax Integration Developer Guide](#).

## Configure Salesforce Billing for Tax Calculation

To ensure tax calculation, Salesforce Billing requires values for several key fields. If you're hitting errors when calculating tax, make sure that these fields are populated correctly. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

Object	Field	Description
Product	Tax Rule	Lookup field to a tax rule. The tax rule defines whether the product is taxable. It also usually holds one or more tax treatments that define the specific tax treatment of products based on legal entity.
Tax Treatment	Legal Entity	A legal entity creates a relationship between the tax rule and the order product and invoice line based on matching legal entities. For example, you could assign several products to look up to a tax rule that applies tax. That tax rule contains several treatments that process tax differently based on business region. Salesforce Billing applies the tax treatment to one of your products (in the group considered by the tax rule) only if the treatment and product have matching legal entities.
Product or Tax Treatment (dependent on tax engine)	Tax Code	Tax codes are unique identifiers that determine how Salesforce Billing taxes a product.
Product	Taxable	Determines whether a product is taxable for third-party tax engines. If unselected, Salesforce Billing considers the product tax-exempt.
Tax Rule	Taxable (Yes/No)	Determines whether products assigned to this tax rule are taxable.
Order Product or Invoice Line	Legal Entity	By default, a legal entity doesn't automatically populate on the

Object	Field	Description
		order product or invoice line unless your organization uses custom automation. If this field isn't populated, the order product doesn't look up to a tax treatment, so taxes don't calculate.

 **Important** You can also customize the tax calculation address. Update the address settings in the Salesforce Billing package settings.

## Line-Level Tax Calculation

Salesforce Billing calculates tax at the line level on both the order product and invoice Line objects. Each order product's tax amount sums to the Billed Tax field on the parent order, and each invoice line's tax amount sums to the Tax field on the parent invoice. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

To calculate tax on a line, Salesforce Billing uses that line's address, tax rule, the tax code on the product, tax treatment, and the tax exemption status of both the account and the product. Different tax codes have different tax statuses, depending on where the product is being consumed or delivered. Tax-exempt lines have a \$0 tax value.

The Standard Tax Engine determines tax calculations or any exemptions along with tax rules and tax treatments and rounding of tax amounts.

 **Note** Tax status on the invoice can update to Completed before the tax status on all invoice lines is completed.

## Tax Integrations

Send data from Salesforce Billing to an external tax calculation service. Tax integrations are the Salesforce Billing objects that define the relationship between your Salesforce Billing org and the external service you use for tax calculation. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and newer

---

### Standard Tax Calculation Engine

Salesforce Billing's Standard Tax Engine provides a basic tax calculation engine for testing or for simple tax use cases. (Salesforce Billing Managed Package)

## Standard Tax Calculation Engine

Salesforce Billing's Standard Tax Engine provides a basic tax calculation engine for testing or for simple tax use cases. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

 **Important** If your use case includes testing or simple tax calculation, you can use Salesforce Billing's internal tax tables, which include basic rates and requirements. These internal tables are not intended to replace full-scale tax engines. Due to the complexity of rules, addresses, and remittance requirements for taxation, we recommend using a third-party tax integration. The Standard Tax Engine is built only to handle testing or basic calculation needs.

Like any other type of tax engine, the Standard Tax Engine calculates tax by using a tax integration, tax rule, tax treatment, and tax rate. When an order product is invoiced, Salesforce Billing checks its tax rule to see whether it's taxable. Salesforce Billing then matches the order product to a tax treatment with a matching legal entity. To use the Standard Tax Engine, your treatment's Tax Integration field must look up to a Standard Tax Engine tax integration record.

The tax rate object, which looks up to a legal entity, defines the percentage value used to calculate tax on order products and invoice lines. Upon order product creation, Salesforce Billing evaluates the order product's tax treatment. If the treatment and a tax rate have matching legal entities, Salesforce Billing then evaluates the treatment and rate's tax code. If both values match, Salesforce Billing applies the tax rate to the order product. For example, if an order product had a subtotal of \$1000 and a matching tax rate had a 10% rate, the order product has an estimated tax of \$100.

During invoice line creation, Salesforce Billing makes the same evaluation using the invoice line's tax treatment and a tax rate. The result of the calculation is sent to the invoice line's Tax field and used to calculate the invoice line's Total Amount (With Tax) field.

Salesforce Billing allows a company to integrate with multiple tax engines, which can be segmented to apply the correct tax engine calculation when key requirements are met. Third-party tax reporting is more robust and can ensure compliance with tax laws and that taxes payable are in balance.

 **Example** To describe the standalone use of the Standard Tax Engine, we'll use a sample company called Local Containers. Local Containers utilizes Salesforce CPQ and Salesforce Billing. They are a smaller business, with their entire customer base operating from a single country with VAT taxation. Rates are defined at a national level; national rates are static and don't change often. If a credit and rebill occurs after the rate has changed, Local Containers can reinvoice at the original tax rate.

Reporting requirements for tax require insight into the total amount of VAT charged in their country. Reconciling tax reports out of Salesforce against information in the company's financial management system ensures that their taxes payable are in balance. Next, let's review the use of the Standard Tax Engine with a third-party integration. Local Containers has expanded their business to the United States, in addition to their country of origin. Due to company expansion, and the complex nature of tax in the U.S., Universal Containers requires more robust tax calculation services. They'll continue to use standard tax for their local VAT tax calculation, however, they also need to integrate with a third-party provider. The third party provides updated rates for appropriate jurisdictions, as tax laws change. This rate management reduces risk and complexity.

### **Setting Up a Standard Tax Integration**

To configure a standard tax integration, define a default tax address in Salesforce Billing package settings. Then create tax rates and tax codes, and associate the code with a tax treatment. (Salesforce Billing Managed Package)

### **Considerations for the Standard Tax Engine**

Before you start working with the Standard Tax Engine, review important considerations. (Salesforce Billing Managed Package)

### **Guidelines for the Standard Tax Engine**

When you're working with the Salesforce Billing standard tax engine, consider key guidelines. (Salesforce Billing Managed Package)

### **Comparing CPQ Tax and Billing Tax**

Salesforce CPQ calculates tax on the quote object, while Salesforce Billing calculates tax on the order and invoice objects. Review this and other important differences when comparing the tax calculation processes between Salesforce CPQ and Salesforce Billing. (Salesforce Billing Managed Package)

### **Configure an External Tax Integration**

Create a tax configuration to handle communication between Salesforce Billing and an external tax-calculation service. (Salesforce Billing Managed Package)

## Setting Up a Standard Tax Integration

To configure a standard tax integration, define a default tax address in Salesforce Billing package settings. Then create tax rates and tax codes, and associate the code with a tax treatment. (Salesforce Billing Managed Package)

### **1. Set Up a Standard Tax Integration**

Create a basic tax integration for Salesforce's internal tax engine. (Salesforce Billing Managed Package)

### **2. Create a Tax Rate and Tax Code for Standard Tax Integrations**

Salesforce Billing uses the tax rate object to store tax rates for standard tax integrations. To accurately perform tax calculations for a standard tax integration, Salesforce Billing requires a tax rate for every tax jurisdiction or address. (Salesforce Billing Managed Package)

## Set Up a Standard Tax Integration

Create a basic tax integration for Salesforce's internal tax engine. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. Configure package-level Standard Tax Engine settings.
  - a. From Setup, enter *Installed Packages*, and then click **Installed Packages**.
  - b. Find the Salesforce Billing package and click **Configure**.
  - c. Select the General tab.
  - d. Provide a value for the **Tax Calculation is based on?** field.
2. Create a standard tax integration record.
  - a. From the tax integration object, click **New**.
  - b. Give your tax integration a name. We recommend something simple and descriptive, like "Standard Tax Integration."
  - c. Select **Active**.
  - d. For the Tax Engine field, choose Standard.
3. Create a tax rule.
  - a. From the tax rule object, click **New**.
  - b. Select **Active**.
  - c. For the Taxable (Yes/No) field, choose Yes.
4. Create a tax treatment on your tax rule.
  - a. For the Tax Integration field, provide a lookup to the tax integration you made in Step 2.
  - b. Select **Active**.
  - c. Provide a tax GL rule and tax legal entity if needed.
  - d. Make sure the tax treatment's legal entity matches the legal entity on the order products that you want the tax rule to target.
5. Assign your tax rule to the products where you want Salesforce Billing to calculate tax using the standard tax engine.

Remember, Salesforce Billing applies the rule's tax treatment only if the product and the tax treatment have matching legal entities.

If you want to use the tax address override fields on your order product, make sure that you populate all of them. Salesforce Billing doesn't calculate standard tax for order products where only some of the tax address override fields have values.

You can have one tax treatment per legal entity on each tax rule. To implement different tax codes, create a different tax rule for each tax code. Use a flow on the order product to update the tax rule. Salesforce Billing selects the correct tax treatment to match the legal entity.

### See Also

[Flow Builder](#)

## Create a Tax Rate and Tax Code for Standard Tax Integrations

Salesforce Billing uses the tax rate object to store tax rates for standard tax integrations. To accurately perform tax calculations for a standard tax integration, Salesforce Billing requires a tax rate for every tax jurisdiction or address. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

For standard tax integrations, Salesforce Billing applies a tax rate's percentage value to an order product's subtotal during tax calculation. The results of this calculation are sent to the order product's Estimated Tax field. For example, if a 10% tax rate applies to a \$1,000 order product, the estimated tax is \$100.

1. From the tax rate page, click **New**.
2. Choose a tax identifier. This value represents the name of your tax rate record.
3. Enter your address. Provide all the information above the lowest address level that you enter. For example, if your tax integration applies tax rates at the ZIP code level, your tax rate requires a ZIP code, city, state, and country. If your integration applies tax at the state level, your tax rate requires only the state and country. The address requires at least a country.
4. Enter a priority number. When multiple rates apply to a transaction with matching priorities, Salesforce Billing combines the rates and applies them against the taxable amount. When priorities differ, Salesforce applies the lowest priority tax rate first. The sum of the taxable amount and the first tax rate is used as the taxable amount for the second tax rate calculation, and so on.

For example, you have a \$5,000 order amount with a 15% priority-0 tax rate and a 20% priority-1 tax rate. Salesforce Billing applies the following calculations.

- $\$5,000 * 0.15 = \$750$ .  $\$5,000 + 750 = \$5,750$ .
- $\$5,750 * 0.2 = \$1,150$ .  $\$1,150 + \$750 = \text{Total estimated tax amount of } \$1,900$ .

5. Enter a legal entity or leave the Legal Entity field null.

If a tax treatment and tax rate have null Legal Entity fields, Salesforce Billing still matches them. They must both be null to match.

6. Enter a tax rate value. This value is a percentage used to calculate tax for all eligible order products that fall under this tax rate.



**Note** After saving your tax rate, Salesforce Billing creates values for the rate's Address Comparison and Priority Address Comparison fields. Salesforce Billing uses these fields only for internal reference.

7. Enter a tax code. (Optional)

If a tax rate has a tax code, Salesforce Billing applies the tax rate to an order product only if the order product's tax treatment has the same tax code. You can apply a tax code to multiple tax rates. Tax codes are useful for allowing different treatments to share a legal entity but inherit different tax rates. Salesforce Billing doesn't use the order product and invoice line's tax code fields for standard tax calculation. These fields are for reference only.

8. If you gave your rate a tax code, go to the tax treatment of the tax rule that you want to associate with your tax rate. Set the treatment's tax code to the code that you made in Step 7.



**Example** Your company needs a standard tax rate for taxing workstations and apps sold in Germany. Workstations are taxed at 10%, and apps are taxed at 5%. The workstation looks up to a Workstation Tax Rule with a Workstation Tax Treatment, and the app looks up to an App Tax Rule with an App Tax Treatment. Both treatments look up to a legal entity titled Germany Legal Entity. First, you need two tax rates. Because this rate applies to all of Germany, define only the country in the tax rate's address fields.

### Tax Rate 1

Tax Identifier: Workstation Tax Rate - Germany

Country: Germany

Priority: 0

Rate: 10%

Legal Entity: Germany Legal Entity

Tax Code: WKS

### Tax Rate 2

Tax Identifier: App Tax Rate - Germany

Country: Germany

Priority: 0

Rate: 5%

Legal Entity: Germany Legal Entity

Tax Code: APP

When a customer orders a workstation, Salesforce Billing connects it to Germany Legal Entity. The tax treatment then evaluates the legal entity's tax rates for matching tax codes and finds the matching WKS code on Workstation Tax Rate - Germany. Salesforce Billing uses a 10% tax rate for the workstation's order products and invoice lines. When a customer orders the app, Salesforce Billing follows the same process to match the app's treatment with the shared APP code on App Tax Rate - Germany. Salesforce Billing uses a 5% tax rate for the app's order products and invoice lines.

If you plan to data-load tax rate records from an outside source, don't disable Salesforce Billing triggers.

To test your standard tax integration, create an order containing products that look up to your tax rule, then invoice that order. In your invoice's Price Information section, the Tax field should show the amount calculated based on your configured tax rate.

## Considerations for the Standard Tax Engine

Before you start working with the Standard Tax Engine, review important considerations. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

- Tax rates frequently change based on industry changes. Admins must manually update tax rate records in Salesforce Billing to reflect these changes.
- When setting up addresses on your tax rate, populate all the relevant fields for the specific level of address detail that you need. For example, if tax rates are applied at the zip code level, you must also populate the city, state, and country fields. If you apply tax at the state level, you need to populate only the state and country fields.
- All tax rates require a tax code, which Salesforce Billing uses to identify which tax rate is used for a given tax calculation. Salesforce Billing administrators maintain tax codes, which can be any unique identifier.
- A tax treatment and the tax rate it references must have matching tax codes.
- Salesforce Billing applies a tax treatment to an order product if both records have matching legal entities. If these records don't have matching legal entities, or if one of the legal entities is null, Salesforce Billing doesn't calculate tax for that order product.

## Guidelines for the Standard Tax Engine

When you're working with the Salesforce Billing standard tax engine, consider key guidelines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

- We strongly recommend using the standard tax engine only for simple taxation use cases. For complex tax calculation requirements, configure the standard tax engine to integrate with third-party platform.
- Salesforce Billing doesn't support customizations made to standard tax engine behavior. If you have to make customizations, consider working with a third-party vendor.
- To override the address used for order product-level taxation, use the order product address fields that end in (Override).
- When Salesforce Billing creates an invoice line, the invoice line inherits its parent order product's legal entity and tax. Further changes to the order product's legal entity and tax don't carry over to the invoice line.
- In multicurrency orgs, Salesforce Billing calculates order product and invoice line tax independently of the tax rate's currency. For example, a 10% GBP tax rate would apply 10% tax to both a USD invoice line and a GBP invoice line. Therefore, a tax rate doesn't require the same currency as its target order

- product or invoice line.
- Salesforce Billing triggers must be active while you load tax rates from an outside source.

## Comparing CPQ Tax and Billing Tax

Salesforce CPQ calculates tax on the quote object, while Salesforce Billing calculates tax on the order and invoice objects. Review this and other important differences when comparing the tax calculation processes between Salesforce CPQ and Salesforce Billing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Your tax engine provides real-time calculations to your Salesforce org.

If your use case requires tax calculation on the quote, your Salesforce org must integrate with a third-party tax engine or custom configuration. This connection lets you calculate tax on the quote and benefit from other features such as automated tax rate updates, pre-built reports, and further customizations.

## Tax Calculations for CPQ & Billing

### Estimated Tax vs. Actual Tax

Your tax engine performs tax calculations on the quote, order, and invoice records.

Since tax rates are subject to change, the quote and order tax calculations are only estimates. Salesforce Billing performs the final tax calculation on the invoice, which represents the final amount included on the customer's billing.

### Using Legal Entities for Tax in Salesforce Billing

In Salesforce Billing, tax rules may have tax treatments, which allow users to define tax calculations based on legal entity. The legal entity is a company-specific object associated with order products, rules, treatments, and other records. If an order product and a tax treatment have matching legal entities, Salesforce Billing applies the treatment to that order product. You can have multiple treatments on a rule, and each treatment can have a unique legal entity. This process lets you consider a broad group of order products under a single tax rule while specifying how smaller sets of order products in that group are treated based on their legal entities.



**Note** Tax rules and treatments aren't part of Salesforce CPQ. They calculate tax only in Salesforce Billing.

### Calculating Tax for CPQ vs. Billing

Tax engines calculate quote-level tax by evaluating the product's tax code and the related

account's shipping address. When calculating tax on orders and invoices, you can choose the tax calculation address in Salesforce Billing package settings. You can also override this field by using the Address Override fields on the order product.

Salesforce CPQ and Salesforce Billing each make API calls at different stages throughout the quote to cash process. Each tax calculation requires an independent API call.

#### Key Differences

Topic	CPQ	Billing
Engine	Use only an external engine. Only Avatax is available pre-packaged. All other tax engines are custom integrations.	Use an external engine like Avalara or Vertex, or a custom integration to an external engine. You can also use Salesforce Billing's internal engine for lightweight tax calculations.
Timing	Calculates quote line tax immediately upon quoting action	Calculates order product tax upon order creation. Calculates invoice line tax during the invoice run(s).
Limitations	Up to 1000 quote lines	Up to 2200 invoice lines in Avalara

#### API calls

Salesforce CPQ and Billing make API calls at various stages throughout the quote-to-cash process. Each tax calculation requires an independent API call.

#### Limitations

The Salesforce CPQ and Billing internal tax engine can calculate tax for up to 1000 quote lines per API call.

Avalara can calculate tax for up to 2200 invoice lines.

## Impact of Tax Amount and Tax Status

#### Estimated Tax on the Quote

When Salesforce CPQ calculates quote tax, it has only limited tax-relevant information such as the account address and the quoted product. Since tax-relevant billing information hasn't been decided yet, the quote's tax serves as an estimate for the quote recipient. We recommend calculating quote tax on quotes with high subtotals (such as \$1 million or greater) given the potential for a large tax calculation to significantly increase the quote's total price.

### Estimated Tax on the Order

When Salesforce Billing calculates order tax, it can evaluate other tax-relevant information such as tax treatments, exemptions, shipping address, or processes to skip tax calculation before manual tax preparation. Other information such as usage data is not yet available, so the order's tax remains an estimate.

### Actual Tax on the Invoice

When Salesforce Billing calculates invoice tax, it has access to all tax-relevant data for products, shipping, and usage. Salesforce Billing sends this data to either an external tax engine or its own tax engine. The Avalara and Vertex tax engines also post the data.

### Posting Tax on the Invoice

When Salesforce Billing posts to an external tax engine, most tax services use the posted data to remit the tax directly to a governing authority, or to drive tax remittance through reporting. External engines ignore unposted transactions.

## Configure an External Tax Integration

Create a tax configuration to handle communication between Salesforce Billing and an external tax-calculation service. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and later

---

Salesforce Billing doesn't own support for external tax calculation services or their Salesforce packages. This topic describes only the steps needed to create the basic Salesforce Billing objects that support an integration with an external tax processing service. Most tax service packages also install package-specific fields and custom settings in Salesforce Billing. Configure these fields and settings based on your tax service provider's documentation.

1. Create a tax integration.
  - a. Add the name of your tax calculation service provider as a value of the Tax Engine field, and then select this value.
  - b. Set the name of your tax integration to the name of your tax calculation service provider.
  - c. Check the Active field.
2. Create a tax rule, check its Active field, and then set the Taxable (Yes/No) field to Yes.
3. On your tax rule, create a tax treatment.
  - a. In the Tax Integration field, enter the tax integration that you created in Step 1.
  - b. If you need to, add a tax GL rule and tax legal entity.

Install your tax service provider's integration package, and then configure the package and Salesforce Billing further based on the tax service provider's documentation.

**!** **Important** After you create and use a tax integration, tax rule, or tax treatment, setting them to inactive doesn't prevent their use when creating an order and draft invoice. In that case, the invoice is created with a tax error and doesn't allow posting until the tax error is resolved. You can change the tax code on the tax treatment, which is effective for future orders and invoices.

## Payments and Credits

---

Collect payments against posted invoices. Salesforce Billing lets you manually collect and allocate payments or automate the payment process. You can then post the payment to keep your books up-to-date. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

**!** **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

---

Available in: All Salesforce Billing Editions

Salesforce Billing tracks payments using several objects. The first of these is the Payment object, a financial record that represents a “payment event,” such as receiving a check or credit card information. A payment record stores information about the amount of the payment, the payment type, and a lookup to the account where the payment can be applied. Users can apply payments to any posted invoice with a balance on the selected account.

Salesforce Billing allows for three ways to create payments.

- Accounts Receivable users can manually create a payment record.
- Admins can create a payment run that evaluates posted invoices on scheduled intervals. It then makes payments based on the related account's payment method.
- End customers can enter the Salesforce Billing payment center to pay invoices using their own payment methods.

Salesforce Billing creates a payment record once it receives customer funds through the payment scheduler or the payment center. The payment record's Payment Status field changes from Initiated to Completed once Salesforce Billing verifies the funds. After verification, funds can be applied to one or more invoices. Applying a full payment to an invoice moves the invoice's Status from Active or Overdue to Paid.

Payments also have a related list for payment allocation records. These records define the amount of a payment that has been allocated to a given invoice line.

Admins can also import payments via the Data Loader or manual entry. Some enterprises require manual payment entries when processing large amounts of checks through a lockbox service or when mailed directly to the enterprise. You can convert this check information to .CSV file format and then import it

via the Salesforce data-loading process. A payment has a master-detail relationship to an account.

-  **Note** Payment allocations can't change the actual amount of their parent payment.

Payments have an optional lookup to an invoice record. Users often add this lookup if the payment is intended for one invoice, though it is not required.



### Payment Methods

Payment methods store Automated Clearing House (ACH) or credit card details for a customer, and the payment gateway that accepts payments. You can use these details in scheduled or one-time payment scenarios. (Salesforce Billing Managed Package)

### Processing Payments

Salesforce Billing provides a comprehensive system for taking customer payment information and passing it to the customer's bank. You can schedule payment runs that evaluate unsettled invoices and automate payments against them, or use the Payment Center to apply payments on your own. When the customer bank responds to a payment request, several Salesforce Billing objects and fields let you track the responses and take corrective action against unsuccessful requests. (Salesforce Billing Managed Package)

### Payment Allocations

A payment allocation represents the amount of a payment that has been applied to an invoice line's balance. Users can manage allocations in Salesforce Billing's Payment Allocation page. (Salesforce Billing Managed Package)

### Issuing Credits

Credit notes allow you to allocate a negative balance change to an invoice line. This feature is useful for adjusting errors in unpaid or partially paid invoices. You can also provide credit to a user account and then allocate that credit to decrease the balance of their invoices at a later date. Finally, credit notes provide accurate and complete recordkeeping of the transactions that occur after an invoice has been posted. (Salesforce Billing Managed Package)

### Creating Debits and Add-on Charges

Debit notes allow you to allocate a positive balance change to an invoice line or positive credit note line. This feature is useful if you must change a line's balance after invoice generation, such as applying a late fee. (Salesforce Billing Managed Package)

### Applying Refunds

Refund customers accurately and efficiently with Salesforce Billing. You can provide a refund when a customer wants to cancel or change their invoiced order products. (Salesforce Billing Managed Package)

### Payment Data Model

When you're working with payment objects in Salesforce Billing, review important object relationships. (Salesforce Billing Managed Package)

## Payment Methods

Payment methods store Automated Clearing House (ACH) or credit card details for a customer, and the payment gateway that accepts payments. You can use these details in scheduled or one-time payment scenarios. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Payment methods contain important information such as payment method address, expiration date, and bank routing number. You can also store a token that the payment gateway uses to process a payment. Your customers can use payment methods to quickly submit payments by using the method on file rather than entering the payment method each time a payment occurs. Payment methods can represent recurring payments or a one-time payment that an authorized Salesforce user performs.

To process all of an account's transactions with a specific payment method, select your payment method's Autopay field and relate it to your account. You can assign multiple payment methods to the same account, but an account can have only one payment method with a selected autopay field. To change autopay methods on an account, deselect autopay on your old method before you select autopay on the new method.

Payment options include credit card, ACH, or cash. Admins must first establish a payment gateway for their enterprise before they can accept payments via credit card or ACH.

After you have a payment gateway and payment method, create a payment scheduler to process payments for posted invoices with open balances.

Your customers can also set up their payment from a virtual terminal. Whenever they receive an invoice and select the virtual terminal payment link, they see the payment method form.

-  **Note** The encryption logic isn't part of the Salesforce Billing triggers. So, even if you save payment methods when the triggers are disabled, the saved records are PCI-compliant.

## Important Payment Method Fields

### Payment Gateway

The payment gateway is a third-party software that provides connectivity to various payment processing networks, such as Visa or American Express. These third-party software products can have various roles in the payment processing lifecycle, such as a processor or acquirer, and may not describe themselves as a payment gateway. However, all these products provide the connectivity that Salesforce Billing uses to process a payment. To associate your payment method with only a certain gateway, provide a lookup to one of your payment gateway records in this field.

## Payment Gateway Token

A token is a unique value that represents the payment method with the payment gateway. When you define a token value, Salesforce Billing references that value during the payment process instead of referencing data and field values from the payment method record itself. This way, Salesforce Billing doesn't expose sensitive customer information, such as credit card numbers, during the payment process.

## Auto Pay

Salesforce Billing automatically uses this payment method to create payments toward posted invoices on this account.



**Example** This is a sample credit card payment method. Because AutoPay is enabled, Salesforce Billing uses it during payment runs that include its related account.

### Payment Type

Credit Card

### Payment Gateway

Authorize.net

### Active

Selected

### Auto Pay

Selected

### Name on Card

Joe Smith

### Card Last Four

1111



**Note** This field auto-populates when a card number is also entered.

### Card Number

\*\*\*\*\*-\*\*\*\*\*-\*\*\*\*\*-1111



**Note** To ensure PCI compliance, Salesforce Billing shows only the last four digits of the entered card numbers.

**Card Type**

Visa

**Card Expiration Month**

01

**Card Expiration Year**

2020

**Card CV**

[null]



**Note** To ensure PCI compliance, Salesforce Billing doesn't store values for the CVV numbers.

## PCI Compliance in Salesforce Billing

The Payment Card Industry Data Security Standard, also known as PCI Compliance, is an information security standard for companies that accept, process, store, or transmit credit card information from major credit card providers. Salesforce Billing became PCI Level 1 compliant in 2012 and has retained its compliance every year afterward. (Salesforce Billing Managed Package)

### Create an ACH Payment Method

Create and store a record of a customer ACH. (Salesforce Billing Managed Package)

### Create a Credit Card Payment Method

Add a credit card payment method to an account. (Salesforce Billing Managed Package)

### Lockbox Processing

Lockbox services collect customer checks through a third-party bank and send them to your business for processing. This service saves your business from spending excessive time processing high volumes of checks. Since each bank has a different lockbox file format, Salesforce doesn't have any pre-built integrations with lockbox providers. When you set up your integration between Salesforce Billing and a lockbox service, make sure that you follow key guidelines. (Salesforce Billing Managed Package)

### Managing Expiring Credit Cards

Manage expiring credit cards to avoid rejection fees. You can flag expired credit cards so they're not sent to your payment processor, and create processes that prompt customers to update near-expired cards. (Salesforce Billing Managed Package)

## PCI Compliance in Salesforce Billing

The Payment Card Industry Data Security Standard, also known as PCI Compliance, is an information security standard for companies that accept, process, store, or transmit credit card information from major credit card providers. Salesforce Billing became PCI Level 1 compliant in 2012 and has retained its compliance every year afterward. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

Salesforce Billing stays PCI compliant by never storing any credit card information before, during, or after payment method collection. Payment card information is transmitted only to payment processors through a token and never stored along the way within Salesforce. The payment method record contains the payment processor token, which links to the actual Personal Account Number stored on the payment processor. The processor token is unique to the customer, payment card, merchant, and payment processor.

Tokens allow systems like Salesforce to store a representation of the customer's payment card, and change it, without having to store their actual payment card information. In the event of a data breach, tokens aren't useful for the thief as the token works only when used with the original merchant and payment processor.

Salesforce Billing stores the following credit card information on each payment method. This information is stored either during the payment collection process or by creating a payment method related to an account. Each payment method contains the following credit card information.

- Name on card
- Last four digits of credit card number
- Card Type
- Token
- Expiration month and year

The payment center automatically creates a token during the process of successfully processing a payment. Salesforce Billing performs the following steps when using tokens during payment processing.

1. A user initiates payments through the Payment Center.
2. Salesforce passes the token to the payment gateway as an authorized request.
3. The payment gateway receives the token and routes to the credit card provided to begin processing.
4. The credit card provider accepts or declines the transaction.
5. The token and payment authorization are routed back to the payment gateway.
6. The payment gateway provides Salesforce a successful or unsuccessful response.

All Salesforce API communication is encrypted and highly secure. The encryption logic isn't part of the Salesforce Billing triggers. So, even if you save payment methods when the triggers are disabled, the saved records are PCI-compliant.

**!** **Important** If you're building a custom Salesforce component that collects user payment information, take special care to not store credit card information within Salesforce or any system that's not designed to store credit card information. If you create a payment method and provide a CVV value and full credit card number, Salesforce Billing doesn't store the CVV and stores only the last four digits of the credit card number to ensure PCI compliance. If you edit a payment method where the CVV is already populated (for example, from a data migration while Salesforce Billing

triggers were disabled), the save will fail and show a "can't change credit card information" error message. To save the edited payment method, delete the contents of the CVV field.

## Staying PCI Compliant with Salesforce Billing

If you're building a custom Salesforce component that collects user payment information, take special care to not store credit card information within Salesforce or any system that's not designed to store credit card information.

If you're migrating customers into Salesforce Billing, ensure you're also migrating payment card information in a PCI-Compliant manner. Each of your customers will likely have to work with their existing payment processors to regenerate tokens that will be valid with Salesforce.

## Create an ACH Payment Method

Create and store a record of a customer ACH. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. Go to your account's Payment Methods related list and click **New Payment Method**.
2. Set the payment type to ACH.
3. Choose the payment gateway that you want to process this payment method.
4. If you want Salesforce Billing to use this method to automatically make payments toward posted invoices, select **Auto Pay**.
5. Fill out the fields under ACH Details.
6. If you want to associate the payment method with an address, fill out the fields under Address Details.
7. Click **Save**.

## Create a Credit Card Payment Method

Add a credit card payment method to an account. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

Before you add credit cards, you must add the New Payment Method Credit Card button to the account page layout.

1. From your account, find the Payment Methods related list, and then click **New Payment Method Credit Card**.

**!** **Important** Use the New Payment Method Credit Card button to create a credit card payment method. The New Payment Method button works only for Automated Clearing House (ACH) payment methods.

2. Choose a payment gateway.
3. Enter the card holder's personal information, card details, and address.  
As of Salesforce Billing Winter '19, the Billing Email field is required on payment methods for Authorize.net, Payeezy, and all other custom payment gateways that are configured to use New Payment Method Credit Card.
4. If you want Salesforce Billing to use this credit card to make automatic payments, select **Sign up for the automatic payment option?**.
5. Save your changes.  
If you entered your information correctly, Salesforce Billing shows a success message.
6. To return to the account, click **Back**. To add another credit card to the account, click **Yes** and repeat the steps.

## Lockbox Processing

Lockbox services collect customer checks through a third-party bank and send them to your business for processing. This service saves your business from spending excessive time processing high volumes of checks. Since each bank has a different lockbox file format, Salesforce doesn't have any pre-built integrations with lockbox providers. When you set up your integration between Salesforce Billing and a lockbox service, make sure that you follow key guidelines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The bank delivers to the business a list of electronic copies of the checks, information on the customer account and payment amount, and transfers the money to the business' bank account. Businesses then use this information to record payments in their billing or financial management systems.

Ideal setups for bringing lockbox payments into Salesforce Billing vary based on the payment volume and the feasibility of your automation strategy.

### Lockbox Workflow Overview

Consider using an integration platform to collect and process data files. You can also use Salesforce for loading payments. We recommend creating a custom “staging object” called Lockbox Payment, and then loading the payments you receive from the bank and lockbox process into your staging object. This way, you can monitor payments that have been matched to a corresponding Salesforce Billing record and any discrepancies or errors in this process.

### Manually Recording Payments

If your customers remit only a small percentage of payments through lockbox processing, we recommend manually recording and applying payments in Salesforce Billing. You'll annually

create payment records along with their corresponding payment allocation records to allocate the payment to the correct invoices.

### Automatically Record Payments

Business that receive the bulk of payments through lockbox processing should use the Salesforce platform to automate cash applications. Automating payment record creation and allocation can be a time-consuming process, with the level of effort compounded by the number of lockbox services in scope, and the integrity of the remittance info that customers provide.

Your solution should involve building a system to consume the information from the lockbox service provider and then record payments in Salesforce Billing. When using this approach, consider the number of banks currently providing lockbox services to your business. Since many banking institutions use different e-file formats, the level of effort to auto-create and allocate payments grows with each additional interface.

### Considerations for Automatically Allocating Payments from Lockbox

After recording the payment in Salesforce Billing, you'll need to apply that payment to the appropriate charge. Make sure to consider quality of customer remittance information when customers pay through the lockbox – payments can't be accurately recorded and applied if remittance data and exception processes are incomplete. Here are some of the variables that you should evaluate.

- Does the receipt show which customer is paying?
- Does the receipt show which invoice or invoices are being paid?
- If the customer short-pays, how should the payment get allocated across several invoice lines?
- Does the bank take a fee per payment receipt, resulting in net cash received being lower than the invoice total?

### Advanced AR Application

Determine whether payment allocations are in-scope for the automation effort. If they are, the Advanced AR Application package setting determines whether Salesforce Billing allocates payments at the invoice or invoice line record. This decision drives whether Salesforce Billing creates payment allocation (invoice) or payment allocation (invoice line) records to satisfy open balances.



#### Example

A utility company offers their customers the ability to pay by check. Every day, hundreds of customers send checks to pay for their utility bill. Let's take a quick look at each step in this process.

1. The customers send their checks to a P.O. Box. The bank then retrieves checks from this location.
2. The bank records these payments and moves money from the customer's bank account to the utility company's bank account.
3. The bank sends a CSV file detailing the payment activity to one of the utility company's servers.
4. The utility company uses a custom automated solution to ingest the file and process it according

to the bank's CSV format.

5. For each record, the utility company creates a payment record and one or more payment allocation records. Salesforce Billing allocates these records to the invoice or invoice line based on the package's Advanced AR Application setting.
6. Records in the CSV file that don't have complete payment information are flagged for manual processing.

## Managing Expiring Credit Cards

Manage expiring credit cards to avoid rejection fees. You can flag expired credit cards so they're not sent to your payment processor, and create processes that prompt customers to update near-expired cards. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Salesforce Billing best practices recommend deactivating expired credit cards to avoid rejections from payment attempts against an expired card. You don't have to delete related payment methods, since existing payments can still map back to a payment method. Many payment gateways offer credit card updating services that reduce payments against expired credit cards. These services update credit card numbers when users receive new cards expiration date extensions.

Some payment gateway offer services that update near-expired credit cards that your customers have saved on file. Depending on your configuration, the update process occurs at the gateway (if the gateway is serving as the vault) or Salesforce may be integrated with your gateway. You can also prompt users to update their credit cards as they get closer to their expiration dates. You can also make a Salesforce workflow that sends emails to your customers, or you can prompt the customer from your customer portal. If you're prompting your customers, make sure that you only prompt users with an active subscription, and avoid prompting customers to update inactive credit cards.

You can also make a workflow that updates payment methods to inactive as they expire, preventing these cards from moving to the payment gateway as part of your payment collection process. If the expiring payment method on the account is the default payment method, the customer must choose a new payment method or enter a new card. Consider building this logic into the Salesforce workflow.

## Processing Payments

Salesforce Billing provides a comprehensive system for taking customer payment information and passing it to the customer's bank. You can schedule payment runs that evaluate unsettled invoices and automate payments against them, or use the Payment Center to apply payments on your own. When the customer bank responds to a payment request, several Salesforce Billing objects and fields let you track

the responses and take corrective action against unsuccessful requests. (Salesforce Billing Managed Package)

### Getting Started With Electronic Payments

Salesforce Billing uses payment runs and the Payment Center to collect payments against recurring charges. Payment runs are automated processes that evaluate an account's electronic payment method and use that method to make payments against invoices with outstanding balances. The Payment Center allows account handlers to make one-time payments against outstanding customer invoices. Before you begin using either method, review how Salesforce Billing handles customer payment data and sends it to external gateways and customer banks for processing. (Salesforce Billing Managed Package)

### Payment Gateways

Payment gateways are external platforms that act as a bridge for communications between Salesforce Billing and customer banks during the payment transaction process. The payment gateway record in Salesforce contains information for establishing the connection between Salesforce Billing and the external gateway. Payment gateway records are assigned to payment methods, which are then assigned to accounts. This way, all payments against an account's invoices flow through the gateway defined on the account's payment method, though you also have options for overriding payment methods when needed. (Salesforce Billing Managed Package)

### Payment Runs

Salesforce Billing uses payment runs to automate the collection of payments against recurring charges. A payment run evaluates an account's electronic payment method and uses that method to collect payments for outstanding invoices. A payment scheduler controls when Salesforce Billing launches a payment run and whether the run is a one-time event or based on a recurring schedule. (Salesforce Billing Managed Package)

### Make Payments with the Payment Center

The Salesforce Billing Payment Center allows admins, end users, and customers to pay invoices, create payments, and manage their payment methods. (Salesforce Billing Managed Package)

### Self-Service Payment Pages

Salesforce Billing supports self-service payment platforms where companies can use Salesforce Billing API or Experience Builder sites to integrate our payment processes with their external platforms. (Salesforce Billing Managed Package)

### Managing Gateway Transaction Responses

Take actions on your payment transactions based on the gateway response recorded in the transaction's Gateway Status field. If the transaction was successful, reconcile your successful transaction against the gateway's settled transactions to ensure that Salesforce Billing didn't miss recording any other successful transactions. If the transaction was indeterminate, check the payment gateway to confirm whether the transaction was settled or unsettled. If the transaction failed, take corrective action in Salesforce Billing and try the payment again. (Salesforce Billing Managed Package)

### Locked and Unlocked Invoices

Salesforce Billing locks invoices from payments when a user makes a charge request to the payment gateway from the Payment Center or a force.com payment site. It also locks invoices after a payment run picks them up for evaluation. The invoice unlocks only when Salesforce Billing confirms that the gateway provides a valid response, or when the payment run finishes processing the invoice. Users

must not risk making duplicate payments if the gateway response is indeterminate. (Salesforce Billing Managed Package)

## Getting Started With Electronic Payments

Salesforce Billing uses payment runs and the Payment Center to collect payments against recurring charges. Payment runs are automated processes that evaluate an account's electronic payment method and use that method to make payments against invoices with outstanding balances. The Payment Center allows account handlers to make one-time payments against outstanding customer invoices. Before you begin using either method, review how Salesforce Billing handles customer payment data and sends it to external gateways and customer banks for processing. (Salesforce Billing Managed Package)

Payment methods store ACH or credit card details for a customer and a link to a Salesforce Billing payment gateway record. The Salesforce Billing payment gateway object contains specific details that Salesforce Billing must communicate with the actual payment gateway, which handles the financial requests made against the customer's bank.

Payment gateway objects can be configured with different default and custom fields based on the information that the customer's payment gateway requires. For example, certain payment gateways can require specific security keys or merchant credential information. Salesforce Billing also supports third-party payment gateway packages that install additional fields on the payment gateway object. These fields are unique to the package and are used to pass information to a specific external gateway. Although Salesforce Billing supports the custom fields that the package directly installs, not all custom payment gateway fields made by users are supported.

Salesforce Billing uses payment transaction records to record the results of gateway requests and responses. During payment runs, Salesforce Billing creates a transaction in an in-progress state before the request and then updates the transaction following the gateway's response. During Payment Center applications, Salesforce Billing creates the transaction after the gateway's response.

The payment transaction has several status fields that let you know whether the communication was successful, failed, or indeterminate. If the transaction wasn't successful, you have several options for corrective action. We'll take a detailed look at these options later. After the payment run has finished creating its transactions, Salesforce Billing evaluates the successful transactions and creates payment records. It then allocates the payments to their respective invoices.

At a general level, we can break down the electronic payment process into three steps.

- Finalizing the payment information in Salesforce Billing (Blue)
- Sending requests to the payment gateway and receiving the payment gateway's responses (Green)
- Recording the results of the gateway communication in payment transactions and allocating payments for each successful transaction (Purple)



We'll take a detailed look at each of these three processes in our next topics.

## Payment Gateways

Payment gateways are external platforms that act as a bridge for communications between Salesforce Billing and customer banks during the payment transaction process. The payment gateway record in Salesforce contains information for establishing the connection between Salesforce Billing and the external gateway. Payment gateway records are assigned to payment methods, which are then assigned to accounts. This way, all payments against an account's invoices flow through the gateway defined on the account's payment method, though you also have options for overriding payment methods when needed. (Salesforce Billing Managed Package)

### **Get Started With Payment Gateways**

Payment gateways are external platforms that act as a bridge for communications between Salesforce Billing and customer banks during the payment transaction process. Your developers configured the API to help Salesforce Billing send customer payment information to the gateway. They also configured API that lets Salesforce Billing record the results of the gateway-bank communication into Salesforce Billing objects. Although you won't be working directly with the API, we recommend reviewing the general gateway communication process to understand the resulting record changes in Salesforce Billing. (Salesforce Billing Managed Package)

### **Creating Payment Gateway Records**

Create a payment gateway record and assign it to a payment method. Then assign the payment method to an account. The required fields and values on a payment gateway record vary based on the information required by the external payment gateway that you're linking to your record. If you installed and configured a payment gateway package, your payment gateway records likely have fields specific to that package. We've provided some examples for configuring payment gateway records for the payment gateway packages that Salesforce Billing supports by default. (Salesforce Billing Managed Package)

### **Processing Payments with Payment Gateways**

Salesforce Billing supports payment interfaces to process credit card and ACH transactions. Payment gateways are external service providers that process these electronic payments. Salesforce Billing uses out-of-the-box or API integrations to interface with a payment gateway. (Salesforce Billing Managed Package)

### **Payment Gateway Fields**

Before you set up payment gateways, review important fields and attributes. (Salesforce Billing Managed Package)

## Get Started With Payment Gateways

Payment gateways are external platforms that act as a bridge for communications between Salesforce Billing and customer banks during the payment transaction process. Your developers configured the API to help Salesforce Billing send customer payment information to the gateway. They also configured API that lets Salesforce Billing record the results of the gateway-bank communication into Salesforce Billing objects. Although you won't be working directly with the API, we recommend reviewing the general gateway communication process to understand the resulting record changes in Salesforce Billing. (Salesforce Billing Managed Package)

Salesforce Billing contacts a payment gateway in these scenarios.

- An electronic payment run picked up at least one eligible invoice.
- A user submitted a payment in the Payment Center.

At these stages, Salesforce Billing evaluates the customer account's payment method for the Payment Gateway field. The field contains a lookup to the Salesforce Billing payment gateway object. The payment gateway object doesn't perform any direct communication with the customer bank, and it doesn't store or handle any customer payment data. That's the external payment gateway's job. Instead, it simply tells Salesforce Billing which external gateway to contact and passes any configuration data that the external gateway needs for communicating with Salesforce Billing.

Salesforce Billing also supports payment gateway integration packages that provide additional fields and processes for establishing communications with the external payment gateway. For example, if an account uses AuthorizeDotNet, your developers likely installed an AuthorizeDotNet integration package alongside some custom automation and processes. Don't worry about these integration packages, either. We'll look at the handful of gateway-specific fields you see later.

Let's look at an overview of the payment gateway communication process. This flowchart picks up the payment process at the point where Salesforce Billing initiates communication with the external payment gateway, so it's the same regardless of whether the request came following a payment run or payment center request. In this example, the customer is using AuthorizeDotNet to handle payments. Remember that during payment runs, Salesforce Billing creates the transaction before the gateway request, but during Payment Center applications, Salesforce Billing creates the transaction following the gateway's response.



While your developers maintain the payment gateway integration, you're responsible for ensuring that funds are allocated correctly following a successful transaction and for taking corrective steps following a failed indeterminate transaction.

**!** **Important** Salesforce Billing and Salesforce don't store credit card numbers, CVV, or CVS information. Instead, the payment method record contains a payment processor token that links to the actual Personal Account Number stored on the payment processor. The processor token is unique to the merchant account. All Salesforce API communication is encrypted and highly secure.

## Creating Payment Gateway Records

Create a payment gateway record and assign it to a payment method. Then assign the payment method to an account. The required fields and values on a payment gateway record vary based on the information required by the external payment gateway that you're linking to your record. If you installed and configured a payment gateway package, your payment gateway records likely have fields specific to that package. We've provided some examples for configuring payment gateway records for the payment gateway packages that Salesforce Billing supports by default. (Salesforce Billing Managed Package)

### Configure the authorize.net Payment Gateway

The authorize.net payment gateway allows your customer's subscribers to purchase your customer's products and services online. Configure Salesforce Billing to send data to this gateway. (Salesforce Billing Managed Package)

### Configure the Payeezy Payment Gateway

The Payeezy payment gateway allows your customer's subscribers to purchase your customer's products and services online. Configure Salesforce Billing to send data to this gateway. (Salesforce Billing Managed Package)

### Configure Other Payment Gateways

Some payment gateway packages, such as PayflowPro Billing and Cybersource for Salesforce Billing, are available from AppExchange. This topic lists them and provides links to installation and support information. (Salesforce Billing Managed Package)

## Configure the authorize.net Payment Gateway

The authorize.net payment gateway allows your customer's subscribers to purchase your customer's products and services online. Configure Salesforce Billing to send data to this gateway. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 7.0 and later

---

If you're using the AuthorizeDotNet gateway, Salesforce Billing Winter '19 requires AuthorizeDotNet for Salesforce Billing 3.0 or later.

The authorize.net gateway accepts credit cards, debit cards and ACH payment types. It can also process payments through tokenization, charging, refunds, and voiding payment.

1. Add the following URLs to your Salesforce Billing org's remote site settings.
  - a. <https://api.authorize.net>
  - b. <https://apitest.authorize.net>
2. Add the following record to your custom settings.
  - a. Name: AuthorizeDotNet
  - b. Gateway Class Name: adnblng.AuthorizeDotNetAPI
3. Create a payment gateway with the following fields.
  - a. Payment Gateway Name: AuthorizeDotNet
  - b. Gateway Type: AuthorizeDotNet
  - c. APILoginId: Provided by Authorize.net
  - d. APITransactionKey: Provided by Authorize.net
  - e. Active: Selected (for active gateways)
  - f. Default: Selected (if this is the default gateway)
  - g. TestMode: Select if you're using sandbox. Leave it unselected if you're using your gateway in production to process actual payments.
4. Add the AuthorizeDotNetAdmin permission set to the user profiles of admins who process or add

credit cards.

## Configure the Payeezy Payment Gateway

The Payeezy payment gateway allows your customer's subscribers to purchase your customer's products and services online. Configure Salesforce Billing to send data to this gateway. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 7.0 and later

---

If you're using the Payeezy gateway, Salesforce Billing Winter '19 requires Payeezy for Salesforce Billing 3.0 or later.

The Payeezy gateway accepts credit cards, debit cards, and ACH payment types. It can also process payments through tokenization, charging, refunds, and voiding payment.

Obtain a security key from First Data.

1. Add the following URLs to your Salesforce Billing org's remote site settings.
  - a. <https://api.payeezy.com>
  - b. <https://api-cert.payeezy.com>
2. Add the following record to your custom settings.
  - a. Name: Payeezy
  - b. Gateway Class Name: pyzblng.PayeezyAPI
3. Create a payment gateway with the following fields.
  - a. Payment Gateway Name: Payeezy
  - b. Gateway Type: Payeezy
  - c. JSSecurityKey: Provided by Payeezy
  - d. APISecret: Provided by Payeezy
  - e. TransarmorToken: Provided by Payeezy
  - f. Token: Provided by Payeezy
  - g. Active: Selected
  - h. Default: Selected
  - i. TestMode: Selected
  - j. Enable User Identification: Checked
4. Add the PayEezyAdmin permission set to the user profiles of admins who process or add credit cards.

## Configure Other Payment Gateways

Some payment gateway packages, such as PayflowPro Billing and Cybersource for Salesforce Billing, are available from AppExchange. This topic lists them and provides links to installation and support information. (Salesforce Billing Managed Package)

## Braintree & PayPal Payflow Gateway for Billing

Paypal owns the Payflow Gateway for Salesforce Billing integration and manages its documentation.

For more information, see the [PayflowPro AppExchange listing](#).

## Cybersource for Salesforce Billing

An updated payment gateway for Cybersource is available from AppExchange.

- Customers already using the Cybersource for Salesforce Billing gateway (namespace `csb!ng`) can continue to do so and log support cases with Salesforce Revenue Cloud.
- New customers must install the version in AppExchange.

For installation and support information, see [Cybersource for Salesforce Billing](#)

## Processing Payments with Payment Gateways

Salesforce Billing supports payment interfaces to process credit card and ACH transactions. Payment gateways are external service providers that process these electronic payments. Salesforce Billing uses out-of-the-box or API integrations to interface with a payment gateway. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 7.0 and later

---

Store payment gateway information in the payment gateway object. You can make several payment gateway records so you can route payments to different gateways based on business requirements.

Create a record for each payment gateway that you want to process payments through. However, you don't have to create multiple records for a single gateway if you wish to process multiple payment types (such as credit cards and ACH) through that gateway.

To use a gateway with Salesforce Billing, install that gateway's Salesforce Billing integration package into your org. If you're using Salesforce Billing Winter '19, make sure that your gateway package is version 3.0 or later.

Salesforce Billing supports the following payment features end-to-end.

- Tokenization
- Charge
- Refund

We also support the following payment features through API.

- Tokenization
- Void Token
- Authorization
- Capture
- Charge
- Get Payment Status
- Void
- Refund
- Get Refund Status
- Void Refund
- Non-Referred Refund

**!** **Important** Neither Salesforce Billing nor the core Salesforce platform store Credit Card Numbers, Personal Account Numbers, CVV, or CVS information. Instead, the payment method record contains a payment processor token that links to the actual Personal Account Number stored on the payment processor. The processor token is unique to the merchant account. All Salesforce API communication is encrypted and highly secure.

## Required Fields

External payment gateways require several fields passed as API from the payment gateway and payment method objects. Absent or incorrect values for these fields may not cause validation errors in Salesforce Billing, but they may result in unsuccessful gateway communications depending on the specific gateway and integration.

Payment Gateway

Field	Definition
Payment Gateway Name	Name of the payment gateway record. We recommend making this the external payment gateway that Salesforce Billing communicates with through your gateway record and payment gateway integration.
Gateway Type	<p>Salesforce Billing supports payment processing integration with many payment gateways. You can find payment gateway integration applications on the <a href="#">Salesforce AppExchange</a>.</p> <p>Salesforce Billing also offers gateway integrations with CyberSource, AuthorizeDotNet, and Payeezy. These integration packages aren't available on the AppExchange. Contact your account representative if you're interested in using them.</p>

Active	Salesforce Billing doesn't use inactive gateways.
Default	When this field is active, all transactions in your Salesforce Billing org use this payment gateway. If more than one payment gateway record has a selected Default field, Salesforce Billing uses the first gateway that it finds when searching for valid gateways.
Email	Email of the customer who initiated the payment. Required as of Salesforce Billing Winter '19 for AuthorizeDotNet, Cybersource, and Payeezy gateways.

**Payment Method**

Field	Definition
Payment Method Name	Name of Payment method
Account	Payment method parent object
Active	Indicates that the payment method is active
Auto Pay	If set to true then Payment Scheduler will pick this payment method
Bank Account Name	Name of the Account holder
Bank Account Number	Account Number
Bank Account Type	The default account types are Checking, Business Checking, and Savings. Other picklist values can be added as permitted by the integration and gateway.
Bank Name	Name of the Bank
Bank Routing Code	ABA or Routing code of Bank
Billing Address	Billing Address
Card BIN	
Card CVV	CVV number
Card Expiration Month	Expiration Month of Card
Card Expiration Year	Expiration Year of Card
Card Number	Credit Card Number (16 digits) Only last four digits will be stored.
Card Type	The default card types are VISA, MasterCard,

	AmericanExpress, and Discover. Other picklist values can be added as permitted by the integration and gateway.
City	City
Company	Name of the Company
Country	Country
Email	Email of the person
Fax	Fax of the person
First Name	First Name of the person
Gateway Response	Displays success or failure message
Last Name	Last Name of the person
Name on Card	Name of Credit Card Holder
Nick Name	person nick name
Payment Gateway	Payment Gateway
Payment Gateway Token	The generated Credit Card token by tokenization process
Payment Type	Supported Payment Types • Credit Card: For Card Payment • ACH: For Bank Payment
Phone Number	Phone of the person
Postal Code	Postal Code
State	State
Street Address 1	Street 1
Street Address 2	Street 2

## Payment Gateway Fields

Before you set up payment gateways, review important fields and attributes. (Salesforce Billing Managed Package)

-  **Note** These fields show only Salesforce Billing fields. If you installed any other payment gateway packages, your payment gateway object also shows fields unique to those packages.

### Active

Not applicable. Salesforce Billing doesn't use inactive gateways.

## Default

When this field is selected, all transactions in Salesforce Billing use this payment gateway. If multiple payment gateway records have a selected Default field, Salesforce Billing uses the first default gateway that it finds when it searches for valid gateways.

## Gateway Type

Salesforce Billing supports payment processing integration with many payment gateways. You can find payment gateway integration applications on [AppExchange](#).

Salesforce Billing also offers gateway integrations with Authorize.net and Payeezy. These integration packages aren't available on AppExchange. To use these packages, contact your account representative.

## Notes

Users can provide additional information in this field.

# Payment Runs

Salesforce Billing uses payment runs to automate the collection of payments against recurring charges. A payment run evaluates an account's electronic payment method and uses that method to collect payments for outstanding invoices. A payment scheduler controls when Salesforce Billing launches a payment run and whether the run is a one-time event or based on a recurring schedule. (Salesforce Billing Managed Package)

### Payment Run Processes

A payment run begins by evaluating invoices to see whether they meet the payment run's criteria. It then evaluates eligible invoices, requests payments from the customer bank, and creates payment transactions based on the bank's response. For each successful transaction, the bank creates a payment record and allocates the payment to the invoice. Finally, it updates several invoice and payment run fields to record the results of the payment run. (Salesforce Billing Managed Package)

### Payment Schedulers

Payment schedulers define when Salesforce Billing starts a payment run. The run can be a one-time event, or it can recur daily, weekly, or monthly following the originally defined date. Your payment run and Salesforce Billing Payment package settings control how Salesforce Billing launches and manages its payment run processes. (Salesforce Billing Managed Package)

### Helpful Fields and Settings for Payment Runs

When you configure your payment runs, review helpful fields and package settings. (Salesforce Billing Managed Package)

### Payment Run Troubleshooting

In some cases, the payment run process fails before the run completes, or the payment run can't create and allocate payments following a successful transaction. In these cases, you have several options for resolving unfinished processes and taking corrective actions before attempting a new run. (Salesforce Billing Managed Package)

### Payment Run Permissions

To complete a payment run, Salesforce Billing must create payment transactions, payments, and payment allocations. If the user who submitted the payment information doesn't have access to the objects or the objects that they look up to, the payment creation process fails. (Salesforce Billing Managed Package)

## Payment Run Processes

A payment run begins by evaluating invoices to see whether they meet the payment run's criteria. It then evaluates eligible invoices, requests payments from the customer bank, and creates payment transactions based on the bank's response. For each successful transaction, the bank creates a payment record and allocates the payment to the invoice. Finally, it updates several invoice and payment run fields to record the results of the payment run. (Salesforce Billing Managed Package)

### Payment Run Criteria

A payment run evaluates an invoice against a set of conditions. If the invoice matches the criteria, the payment is included for processing. (Salesforce Billing Managed Package)

### Payment Run Steps

When a payment run finishes selecting invoices, it begins processing each invoice for payment. The payment evaluation process consists of several steps. (Salesforce Billing Managed Package)

### Payment Run Allocations

After Salesforce Billing records the payment transactions, the payment run evaluates successful transactions and creates a payment record for each. If the Advanced AR package setting is enabled, Salesforce Billing creates one payment allocation for each invoice line on the payment transaction's target invoice. Otherwise, Salesforce Billing creates one payment allocation for the entire invoice and applies it to the invoice header. (Salesforce Billing Managed Package)

### Recording Payment Run Results

When a payment run has finished processing a gateway call, Salesforce Billing updates payment run status fields on the invoice, payment, and any resulting payment transactions. Payment run status fields let you review the result of the call and whether there were any issues. You can use this information to take corrective action on unsuccessful calls and record responses in external sources. (Salesforce Billing Managed Package)

## Payment Run Criteria

A payment run evaluates an invoice against a set of conditions. If the invoice matches the criteria, the payment is included for processing. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '22 and later

---

Available in: Salesforce Classic and Lightning Experience

---

Available in: **Professional, Enterprise, Unlimited, and Developer** Editions

An invoice meeting all of these criteria is eligible for payment processing:

- The status of the invoice is posted.
- Contains a balance above zero.
- Looks up to an account with an active payment method. When using batch payments, the invoice run and the payment run must have the same payment batch.
- The Corrective Action (Payment Run) field is null. If set to Action Required, check the Last Payment Run Processing Message field for next steps. See [Finalize Payments After a Failed or Indeterminate Transaction](#).
- The Invoice Locked field is deselected. See [Locked and Unlocked Invoices](#).
- The Payment Run field has a null value. If it contains a link to a payment run, verify that the run is completed to avoid duplicate payments.
- The payment scheduler and the invoice have the same currency type, or alternatively, the Include All Currencies field is enabled in the payment scheduler.
- The payment run's target date is same as or after the invoice date or the due date based on the Payment Scheduler Pickup Date package setting.
- The invoice's payment batch value is added in the Payment Scheduler multiselect picklist. If the payment batch in the payment scheduler is null, the payment run selects all the invoices that are ready for payment, regardless of the invoice's payment batch.
- The billing account's default Payment Method field is active and set to Auto Pay. The Default Payment Type field matches the payment type of the payment method set to Auto Pay.
- On the Account record, the Default Payment Type is autopopulated with the Auto Pay payment type. For this to happen, the Payment Method must be set to Auto Pay in the payment scheduler.
- The payment gateway on the active default Payment Method field matches the gateway on the payment scheduler.

## Payment Run Steps

When a payment run finishes selecting invoices, it begins processing each invoice for payment. The payment evaluation process consists of several steps. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

For each invoice included in a payment run, Salesforce Billing performs the following steps.

1. Sets invoice's payment run ID to the ID of the current payment run, and then updates the invoice's payment run processing message based on the most recently completed payment run batch action.
2. Sends a transaction request to the account's external payment gateway, and then creates a payment transaction to record the response.
3. If the gateway authorization is successful, waits for the payment gateway to send a response, and then records it in the payment transaction.
  - a. If the response was successful, creates a payment with a balance equal to the invoice's balance. Changes the payment transaction's Authorization field to SUCCESS.

- b. If the response failed or was indeterminate, logs the response details in the payment transaction. Users can take corrective action based on the type of failed or indeterminate response.
4. For successful payment transactions, attempts to create a payment and allocate it to the target invoice.
- a. If Advanced AR Application is enabled in Salesforce Billing package settings, allocates the payment to the target invoice's invoice lines.
  - b. If Advanced AR Application is disabled, allocate the payment to the target invoice's invoice header.



After the payment run finishes evaluating all eligible invoices, it updates the following fields on the payment run detail page.

- Invoices Processed
- Successful Transactions
- Total Payments Processed
- Completed Time
- Status

## Payment Run Allocations

After Salesforce Billing records the payment transactions, the payment run evaluates successful transactions and creates a payment record for each. If the Advanced AR package setting is enabled, Salesforce Billing creates one payment allocation for each invoice line on the payment transaction's target invoice. Otherwise, Salesforce Billing creates one payment allocation for the entire invoice and applies it to the invoice header. (Salesforce Billing Managed Package)

Here's a sample flowchart showing the payment creation and allocation process on one invoice following a successful payment run.



## Recording Payment Run Results

When a payment run has finished processing a gateway call, Salesforce Billing updates payment run status fields on the invoice, payment, and any resulting payment transactions. Payment run status fields let you review the result of the call and whether there were any issues. You can use this information to take corrective action on unsuccessful calls and record responses in external sources. (Salesforce Billing Managed Package)

## Invoice

### Corrective Action (Payment Run)

This field shows Action Required if the payment gateway encountered an error or received an indeterminate response when attempting to create a payment for this invoice. While this field has a value of Action Required, payment runs ignore the invoice. If your transaction attempt was

If the payment transaction was indeterminate but succeeded in the payment gateway, you can finalize the payment by manually creating a payment and allocating it against the invoice. If the payment transaction failed, or was indeterminate and failed in the payment gateway, you can click **Attempt Refund and Unlock Invoice** to refund any settled payments in the gateway, change the corrective action to null, and enable the invoice for future payment runs.

### **Declined Payment Count (Payment Run)**

When a payment run related to this invoice creates a payment transaction with a gateway status of Decline, Salesforce Billing increases the invoice's declined payment count by 1.

### **Last Payment Run**

Shows the ID of the last payment run that evaluated the invoice.

### **Last Payment Run Processing Message**

Shows the result of the last batch action for the payment run evaluating the invoice.

### **Payment Run**

Shows the ID of the payment run that is processing the invoice.

### **Payment Run Processing Message**

Shows the result of the batch action for the payment run that is processing the invoice.

## Payment Run

### **Invoices Processed**

The number of invoices that the payment run processed after all of its batch processes finished.

### **Successful Transactions**

The number of payment transactions that the payment run created with a gateway status of Successful.

### **Failed Transactions**

The number of payment transactions that the payment run created with statuses other than Successful.

### **Total Payments Processed**

The number of payment records created across all the payment run's successful transactions.

### **Completed Time**

The amount of time that it took for all of the payment run's batch processes to finish.

## Status

Shows whether the payment run finished successfully. If the payment run failed, you have several options for clearing unfinished processes, correcting the errors, and trying the run again.

## Payment Schedulers

Payment schedulers define when Salesforce Billing starts a payment run. The run can be a one-time event, or it can recur daily, weekly, or monthly following the originally defined date. Your payment run and Salesforce Billing Payment package settings control how Salesforce Billing launches and manages its payment run processes. (Salesforce Billing Managed Package)

### Payment Scheduler Fields

Payment scheduler fields control when the scheduler sends out a payment run and whether that run happens on a recurring basis. You can also control the type of payment method that the run uses when creating payments against an account's invoices. (Salesforce Billing Managed Package)

### Create Payment Runs with the Payment Scheduler

A payment scheduler sends out a payment run to collect electronic payments for outstanding invoices. You can configure it to run one time, or you can schedule recurring payment runs. (Salesforce Billing Managed Package)

## Payment Scheduler Fields

Payment scheduler fields control when the scheduler sends out a payment run and whether that run happens on a recurring basis. You can also control the type of payment method that the run uses when creating payments against an account's invoices. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Include All Currencies

In multicurrency orgs, the payment run picks up invoices of all currencies. Salesforce Billing still evaluates other criteria such as payment batch and payment type.

### Payment Batch

If a payment scheduler has a payment batch value, the payment run evaluates only invoices with matching payment batch values. This is an optional field.

### Payment Type

Salesforce Billing includes an account in the payment run if the account contains a payment method where the Auto Pay field is selected and the Payment Type value matches the scheduler's payment type. You can also choose multiple values for the scheduler's payment type. In this case, Salesforce Billing evaluates payment methods for the first value listed in the scheduler's payment type. If that method type isn't available, it evaluates payment methods for

the other value.

### Start Date and Time

Defines when your invoice scheduler begins its first payment run. Payment schedulers require this field to start their first payment run, regardless of type. If you don't provide a value for this field, your payment scheduler doesn't send out its first payment run.

### Target Date

When a payment run evaluates invoices for payment, it includes invoices only if their invoice date or invoice due date is on or before the scheduler's target date. You can choose whether the scheduler evaluates invoice date or invoice due date in Salesforce Billing package settings.

### Target Day of Month

Sets the day of the month for monthly payment runs. Use this field if your invoice scheduler's Type field has a value of Monthly.

### Target Day of Week

Sets the day of the week for weekly invoice runs. The day is represented by a number, where 1 equals Sunday. Use this field if your invoice scheduler's Type field has a value of Weekly.

### Type

Defines whether the invoice run occurs once, daily, weekly, or monthly. This is a required field.



**Example** This payment scheduler sends out a payment run every Friday at 5:00 P.M. starting the first Friday of March. It picks up invoices with due dates on or before April 30, 2018.

## Create Payment Runs with the Payment Scheduler

A payment scheduler sends out a payment run to collect electronic payments for outstanding invoices. You can configure it to run one time, or you can schedule recurring payment runs. (Salesforce Billing Managed Package)

Configure your payment scheduler to perform a payment run one time or indefinitely on a certain day of the week or day of the month.

1. From the payment scheduler list view, click **New**.
2. Choose a payment gateway.  
Payment schedulers require a value for their Payment Gateway field as of Salesforce Billing Winter '19. If you upgraded from an earlier version, delete your currency payment run Apex jobs and re-create them as new payment schedulers with a value in the Payment Gateway field.
3. Set type to **Once** for a single run. Set Type to Daily, Weekly, or Monthly for recurring runs.
4. Choose a payment type.  
Salesforce Billing includes an account in the payment run if the account contains a payment method where the Auto Pay field is selected and the Payment Type value matches the scheduler's payment

type. You can also choose multiple values for the scheduler's payment type. In this case, Salesforce Billing evaluates payment methods for the first value listed in the scheduler's payment type. If that method type isn't available, Salesforce Billing evaluates payment methods for the other value.

5. Set the target date for your run. This date represents when the single run occurs if you set the type to **Once** or when the first run occurs if you set the type to any other value.  
If your invoice date is before your scheduler's target date, billing runs on the invoice date while charges run up until the target date.
6. If you're creating a recurring payment run, set the target day of week or target day of month as needed.  
The target day of week is represented by a number, where 1 equals Sunday, and 7 equals Saturday.
7. Optional: Set a payment batch value. Your payment run targets only accounts with a payment batch value that matches the scheduler's payment batch.

## Helpful Fields and Settings for Payment Runs

When you configure your payment runs, review helpful fields and package settings. (Salesforce Billing Managed Package)

### Fields

Salesforce Billing contains fields and package settings to help you customize how your payment runs are handled.

Object	Field Name	Notes
Account	Default Payment Type	<p>A payment run picks up an invoice if the run and the invoice have matching Default Payment Type values. Salesforce Billing provides two ways to set Default Payment Type on the invoice:</p> <p><b>Assign a payment method to the account and select the payment method's Autopay field</b></p> <p>The account's default payment type inherits the value of the payment method's Type field. All the account's invoices inherit the account's default payment type.</p>

Object	Field Name	Notes
		<p>This setup is useful for quickly setting a default payment type for all of the account's invoices.</p> <p><b>Enter a payment method in the invoice's Override Autopay Payment Method field</b></p> <p>The invoice's default payment type ignores the account's payment method and inherits the Type value from the Override Autopay Payment Method instead. Use this setup when you must temporarily change the payment method used for a specific invoice.</p>
Invoice	Override Autopay Payment Method	<p>Shows a payment method. If this field has a value when a payment run evaluates the invoice, the invoice's default payment type ignores the account's payment method. The default payment type instead inherits its Type value from the payment method defined in Override Autopay Payment Method.</p> <p>We recommend using this setup when you must temporarily change the payment method used for a specific invoice.</p>

## Package Settings

To access these settings, go to the Payment tab in Salesforce Billing package settings.

Setting	Notes
Payment Scheduler Pickup Date (on or after)	Allows users to choose whether the payment scheduler picks up the invoice for payment on the invoice date or due date.
Payment Creation Batch Size	<p>Payment runs can encounter errors when running batch Apex jobs that create large numbers of payment lines. These errors often occur around 2,000 payment lines, but the exact number varies based on your settings. You can reduce the likelihood of these errors by setting your payment creation batch size.</p> <p>For example, if you set a posted invoice batch size of 300 and your payment run must create 2,100 payment lines, Salesforce Billing runs 7 Apex batch jobs that each create 300 payments. Reducing the batch sizes slows payment run speeds but reduces the chance of errors from batches with multiple large invoices.</p> <p>If your customer's payment run encounters Apex errors, decrease the payment creation batch size. Then, return it to the maximum value when their payment runs are no longer evaluating a large volume of invoice lines.</p> <p>Salesforce Billing supports a minimum value of 1 and a maximum value of 70.</p>

## Payment Run Troubleshooting

In some cases, the payment run process fails before the run completes, or the payment run can't create and allocate payments following a successful transaction. In these cases, you have several options for resolving unfinished processes and taking corrective actions before attempting a new run. (Salesforce Billing Managed Package)

### [Manage an Aborted Payment Run](#)

When a payment run batch fails unexpectedly, payment run status fields remain as either Processing or Started, and values on invoices and payment transactions are not updated, even though the payment run is no longer active. You can resume the failed payment run and update the relevant status fields on your invoices and payment transactions. (Salesforce Billing Managed Package)

### [Guidelines for Ending a Failed Payment Run](#)

The End Payment Run button lets you restart and finalize a payment run that was unexpectedly

interrupted. When you click it, Salesforce Billing takes different actions based on how the payment run was interrupted. (Salesforce Billing Managed Package)

#### **Manage Failed Allocations for Completed Payment Runs**

When a payment run completes and creates a payment but can't perform a payment allocation, finish the process by manually allocating your payment. (Salesforce Billing Managed Package)

#### **Payment Creation Errors**

Gateways can encounter errors that prevent payment creation off the payment transaction. You can review information about the error on the invoice and transaction records, and then make corrections as needed. (Salesforce Billing Managed Package)

#### **Cancel Upcoming Payment Runs**

Cancel upcoming payment runs by deleting the Scheduled Job record of the parent payment scheduler. (Salesforce Billing Managed Package)

### Manage an Aborted Payment Run

When a payment run batch fails unexpectedly, payment run status fields remain as either Processing or Started, and values on invoices and payment transactions are not updated, even though the payment run is no longer active. You can resume the failed payment run and update the relevant status fields on your invoices and payment transactions. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

From your payment run record, click **End Payment Run**.

Salesforce Billing evaluates the aborted payment run's batch status and determines whether the run called the payment gateway.

- If the payment run never made a gateway callout, Salesforce Billing updates the payment transaction's status to Processed. It also updates invoice and payment transaction status fields to reflect the failed callout.
- If the payment run made a gateway callout but didn't receive a response, Salesforce Billing invoice and payment run fields to reflect that users should manually review the payment in the gateway. It also updates the payment transaction's Gateway Status field to Indeterminate. See [Indeterminate and Failed Transactions](#).

### Guidelines for Ending a Failed Payment Run

The End Payment Run button lets you restart and finalize a payment run that was unexpectedly interrupted. When you click it, Salesforce Billing takes different actions based on how the payment run was interrupted. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

Unexpected interruptions such as system failures and terminated apex jobs can cause a payment run's batch to fail. You can resume the stalled payment run by clicking **End Payment Run** on the payment run record. When you click End Payment Run, Salesforce Billing evaluates all invoices with the same payment run ID as your stalled payment run. It then performs a different set of actions based on the payment transaction's gateway status and when the batch failed.

### **Payment run failed during Started [Batch 1]**

Payment runs that failed during batch 1 never made a gateway callout, so we don't have to worry about making double payments. In this case, Salesforce Billing performs the following steps.

1. Change each invoice's current processing message to "Couldn't make a call to the gateway."
2. Clear out each invoice's Payment Run field value
3. Each invoice's Last Payment Run ID field inherits the value of the Payment Run ID field.
4. Change each payment transaction's status to Processed
5. Change each payment transaction's exception message to "Couldn't make a call to the gateway."

### **Payment run failed during Processing [Batch 2], transaction gateway status is None**

In this case, the payment run failed while it was processing invoices, and Salesforce Billing didn't receive any responses from the gateway.

1. For each invoice, find related payment transactions and change their gateway statuses to Indeterminate.
2. For each invoice, select the Corrective Action field.
3. Change the failed payment run's processing message to "Manual Review Needed."

### **Payment run failed during Batch 2, transaction gateway status is any value other than Indeterminate or Success**

In this case, Salesforce Billing updated the payment transaction after receiving a response from the gateway indicating some type of payment creation failure. Salesforce Billing changes each invoice's current processing message to "Received failure from gateway. No payment made."

### **Payment run failed during Batch 2, transaction gateway status is Success**

In this case, Salesforce Billing received a successful gateway response from the gateway. After clicking End Payment Run in this case, Salesforce Billing performs the following steps.

1. Each invoice's Last Payment Run ID field inherits the value of the Payment Run ID field.
2. Clear out each invoice's Payment Run ID field.
3. For each invoice, find any related payment transactions and change their statuses to Processed.
4. Sometimes when the payment run fails, Salesforce Billing creates payment transactions that don't look up to an invoice. Find these payment transactions and change their statuses to Processed.
5. Clear out each invoice's In Cleanup field.

## Manage Failed Allocations for Completed Payment Runs

When a payment run completes and creates a payment but can't perform a payment allocation, finish the process by manually allocating your payment. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

We recommend that users review their payment runs for error logs indicating failed allocations. From the error log, you can find the transaction and then the payment record that the run attempted to allocate from. You can also create workflow rules that provide updates when a payment changes its allocation status to Fail.

1. If your allocation failed due to a system error such as a user-made validation or database insert error, resolve that error first.
2. Go to the payment record that has a failed allocation status.
3. Click **Allocations**.
4. Review the invoice lines that still have balances. Usually, these balances are equal to the unallocated amount from the payment you selected in Step 1.
5. Select the invoice lines and enter amounts that reduce their balances to zero.
6. Click **Allocate**.

## Payment Creation Errors

Gateways can encounter errors that prevent payment creation off the payment transaction. You can review information about the error on the invoice and transaction records, and then make corrections as needed. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

After the failure, the payment run updates important status fields on the following objects.

#### **Invoice**

Last Payment Run Processing Message: Payment Creation Failed

#### **Payment Transaction**

Status: Processed

Since Salesforce Billing didn't create a payment, your transaction's Payment Created field remains unselected.

The payment run also adds an error log to your payment transaction. We recommend that users review

their payment runs for error logs indicating failed allocation. From the error log, you can find the transaction and then the payment where allocation failed.

-  **Tip** Admins can also create a workflow rule or process to alert users when a transaction changes its payment creation status to Payment Creation Failed. This way, users can review or correct as soon as possible.

## Cancel Upcoming Payment Runs

Cancel upcoming payment runs by deleting the Scheduled Job record of the parent payment scheduler. (Salesforce Billing Managed Package)

Canceling upcoming payment runs is useful if you've changed the configuration of an invoice in a way that would disqualify it from runs created by your current payment scheduler. After cancellation, you can create a payment scheduler with payment runs configured to pick up your revised invoice.

 **Warning**

- Deleting only the payment scheduler doesn't stop its scheduled jobs. If you delete the scheduler without deleting its scheduled jobs, Salesforce Billing continues running the jobs, which can cause unnecessary or inaccurate payments.
- You can't reinstate a payment scheduler after you delete its scheduled job.

1. From Setup, in the Quick Find box, enter *Scheduled Jobs*, and then select **Scheduled Jobs**.
2. Find the job name of the payment scheduler that you want to cancel. The job name matches the value of the Payment Scheduler Name field on the payment scheduler that created the payment run.
3. In the Action column next to the job, click **Del**, and then click **OK**.  
We recommend making a note of the cancellation in the payment scheduler's Notes field.

You can delete the payment scheduler record after you've deleted its scheduled job. However, we recommend keeping the canceled payment scheduler record for bookkeeping purposes.

## Payment Run Permissions

To complete a payment run, Salesforce Billing must create payment transactions, payments, and payment allocations. If the user who submitted the payment information doesn't have access to the objects or the objects that they look up to, the payment creation process fails. (Salesforce Billing Managed Package)

Before you deploy Salesforce Billing, we recommend testing your payment runs in a sandbox with each user who will create and own payment scheduler and payment run records. This way, you can confirm that each user's permission set contains the necessary permissions. Also make sure that your payment gateways are configured in a way that matches your gateway configuration in production.

**!** **Important** Many Salesforce Billing objects, including payments, payment transactions, and payment allocations, look up to the account. If you change the account's sharing settings to Private or Read-Only, payment run users can't complete payments because they lose access to payments, payment transactions, and other important objects. In this case, give equivalent permissions and sharing rules to run users so that they can continue using, creating, and managing payment runs.

Action	Read	Create	Edit	Delete
Create Payment Schedulers and Payment Runs	<ul style="list-style-type: none"> <li>• Account</li> <li>• Debit Note Line</li> <li>• Debit Notes</li> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Payment</li> <li>• Payment Allocation (Invoice)</li> <li>• Payment Allocation (Invoice Lines)</li> <li>• Payment Method</li> <li>• Payment Transaction</li> </ul>	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Payment</li> <li>• Payment Allocations (Invoice)</li> <li>• Payment Allocation (Invoice Line)</li> <li>• Payment Transaction</li> </ul>	None	None

## Make Payments with the Payment Center

The Salesforce Billing Payment Center allows admins, end users, and customers to pay invoices, create payments, and manage their payment methods. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

#### [Payment Center Permissions and Guidelines](#)

To process a payment in the Payment Center, Salesforce Billing must create payment transactions, payments, and payment allocations. If the user who submitted the payment information doesn't have access to any of these objects or the objects that they look up to, the payment creation process fails. (Salesforce Billing Managed Package)

#### [Pay Your Account's Balance in Full](#)

Create a payment (Salesforce Billing Managed Package)

#### [Create a Payment Record Manually](#)

To create a payment record on your account without having to allocate it immediately to an invoice, use this manual process. This process is useful for preparing a payment that a sales rep or end user can apply later or for manually recording payments made by check or cash. (Salesforce Billing Managed Package)

### Make a Payment on an Invoice

Use the payment center to create and apply a payment toward all or part of an invoice's balance. (Salesforce Billing Managed Package)

## Payment Center Permissions and Guidelines

To process a payment in the Payment Center, Salesforce Billing must create payment transactions, payments, and payment allocations. If the user who submitted the payment information doesn't have access to any of these objects or the objects that they look up to, the payment creation process fails. (Salesforce Billing Managed Package)

Before you deploy Salesforce Billing, we recommend testing the Payment Center in a sandbox with each user who submits customer payment information. This way, you can confirm that each user's permission set contains the necessary permissions. Also make sure that your payment gateways are configured in a way that matches your gateway configuration in Production.

**!** **Important** Many Salesforce Billing objects, including payments, payment transactions, and payment allocations, look up to the account. If you change the account's sharing settings to Private or Read-Only, payment run users can't complete payments as they lose access to payments, payment transactions, and other important objects. In this case, you must give equivalent permissions and sharing rules to run users so that they can continue using, creating, and managing payment runs.

Action	Read	Create	Edit	Delete
Use the Payment Center to make payments against an invoice	<ul style="list-style-type: none"> <li>• Account</li> <li>• Debit Note Line</li> <li>• Debit Notes</li> <li>• Invoice</li> <li>• Invoice Line</li> <li>• Payment</li> <li>• Payment Allocation (Invoice)</li> <li>• Payment Allocation (Invoice Lines)</li> <li>• Payment Method</li> <li>• Payment</li> </ul>	<ul style="list-style-type: none"> <li>• Invoice</li> <li>• Payment</li> <li>• Payment Allocations (Invoice)</li> <li>• Payment Allocation (Invoice Line)</li> <li>• Payment Transaction</li> </ul>	None	None

Action	Read	Create	Edit	Delete
	Transaction			

After you complete a payment in the Payment Center, reconcile the payment transaction with the equivalent transaction in your payment gateway to confirm that the payment was settled with the customer's bank. If the payment failed or was indeterminate, evaluate the equivalent transaction in the gateway and take the necessary corrective steps before trying to make the payment again.

## Pay Your Account's Balance in Full

Create a payment (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

1. Go to your account and click **Payment Center**.
2. On the Make a Payment page, click **Total Balance** and then click **Next**.
3. Choose a payment method or create a new one, then click **Pay Now**.

The payment center also displays the Payment Summary sidebar. This sidebar shows all the invoices in your account and the remaining balances on each invoice. The sidebar's Payment Total field displays the sum of all remaining invoice balances.

Salesforce Billing applies your payment across all posted invoices in your account that contain a balance. It then displays a page confirming that your payment was successful. You can return to your account or view the payment record you just made.

## Create a Payment Record Manually

To create a payment record on your account without having to allocate it immediately to an invoice, use this manual process. This process is useful for preparing a payment that a sales rep or end user can apply later or for manually recording payments made by check or cash. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

You can create a payment record either from your account or on the Payments page if the payment type is cash or check.

1. Create a payment record.

<b>To create Payment record from your account</b>	From the App Launcher, find and select <b>Account</b> .  Go to your account, and on the Payments related list, click <b>New</b> .
<b>To create Payment record on Payments page</b>	From the App Launcher, find and select <b>Payments</b> .  Click <b>New</b> .

2. Select account.

If you're creating a Payment record from an account, the Account field is automatically populated.

3. Enter amount.

4. Select the status as **Posted**.

Only posted payments can be allocated to an invoice. You can also create draft payments and post it later.

5. Select the payment type as cash or check.

6. Select the payment mode as **External**.

7. Enter payment date and currency.

8. If you select payment type as check, you can enter check number and check date if needed.

9. Save the Payment record.

#### See Also

[Payment Runs](#)

[Allocate a Payment](#)

#### Make a Payment on an Invoice

Use the payment center to create and apply a payment toward all or part of an invoice's balance.  
(Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

1. Go to your account and click **Payment Center**.

2. On the Make a Payment page, click **By Invoice** and then click **Next**.

The payment center displays a list of your account's posted invoices with balances due. You can choose to pay any number of invoices in full or to make partial payments against their balances. You can search for invoices by invoice number.

3. Check the boxes for all the invoices that you want to pay.

4. Choose **Full** or **Partial** payments for each invoice that you want to pay.

Salesforce Billing displays a Payment Summary sidebar listing the total balance paid and remaining balance for each invoice you've selected.

5. Click **Next**.
  - a. If you chose partial payments for any of your invoices, choose how much to pay against each invoice's invoice lines. Click **Next** when you're done.
6. Choose a payment method or create a new one, then click **Pay Now**.  
Salesforce Billing displays a page confirming that your payment was successful. You can return to your account or view the payment record you just made.

 **Note**

- Salesforce Billing always allocates partial payments to invoice lines, even when the Advanced A/R Application package setting is disabled.
- You cannot update your payment method's credit card number within the payment center.

## Self-Service Payment Pages

Salesforce Billing supports self-service payment platforms where companies can use Salesforce Billing API or Experience Builder sites to integrate our payment processes with their external platforms.  
(Salesforce Billing Managed Package)

### [Lightning Platform Payment Sites for Salesforce Billing](#)

Your customers can use Lightning Platform payment sites to pay their Salesforce Billing invoices without logging in to Salesforce Billing. (Salesforce Billing Managed Package)

### [Salesforce Billing Hosted Card Payments in Experience Cloud Sites and Lightning Pages](#)

The `force:cardPayment` component, also known as the Hosted Card Payments component lets customers make payments on their accounts and save credit card information. The component is available for Experience Cloud sites and Lightning pages. When customers submit their card information, Salesforce Billing sends the request to your company's payment gateway for processing. Because Salesforce Billing is PCI-compliant, you're providing a secure payment process in your Experience Cloud site. (Salesforce Billing Managed Package)

## Lightning Platform Payment Sites for Salesforce Billing

Your customers can use Lightning Platform payment sites to pay their Salesforce Billing invoices without logging in to Salesforce Billing. (Salesforce Billing Managed Package)

### [Create a Lightning Platform Payment Site for Salesforce Billing](#)

Create a Lightning Platform site so that customers can pay an invoice without logging in to Salesforce Billing. (Salesforce Billing Managed Package)

### [Allow Customers to Save Credit Card Details](#)

Allow customers to view several options for saving credit card details when they reach the end of your force.com checkout page. They can save their credit card information as a token and use it for fast payments in the Payment Center. They can also enable automatic payments so that Salesforce Billing uses the token to charge their credit card upon invoice due dates. (Salesforce Billing Managed

Package)

### Object and Field Permissions for Lightning Platform Payment Sites

External users require specific object and field permissions to work with Salesforce Billing payments in a Lightning Platform site. (Salesforce Billing Managed Package)

Create a Lightning Platform Payment Site for Salesforce Billing

Create a Lightning Platform site so that customers can pay an invoice without logging in to Salesforce Billing. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Payment sites require that your org has a domain name. If you don't have one yet, create one by following [Register a Salesforce Sites Domain](#).

1. Assign object and field permissions to users who work with the payment site. Usually, your end customer's finance teams use the payment site to make payments against outstanding invoices. You can find a list of required permissions in [Object and Field Permissions for Lightning Platform Payment Sites](#).
2. From the Sites page, click **New**, select **Active**, enter *checkout* for the active site home page, and then provide a default web address. Save your changes.  
On the Sites detail page, make a note of your payment site's URL. You can find it by going to the Custom URLs section and adding your path to the end of your domain name.
3. If your Lightning Platform site is part of a production org, provide site licenses to guest users.
  - a. On your Site details page, click **Public Access Settings**, then click **View Users**. Select the **Site Guest User** entry for your site.
  - b. In the Managed Packages section, click **Assign Licenses**.
  - c. Select **Salesforce Billing** and **Salesforce CPQ**, and then click **Add**.
4. Provide the payment site URL on invoices so that customers can pay them. The payment site URL consists of your site URL followed by `?id=` and then the invoice record's URL ID. Your record ID is the list of numbers at the end of the record's URL. For example, let's say your invoice has a URL of `https://sales-site.salesforce.com/a1W6A000000bnF9`, and your payment site has a URL of `https://sales-payment-site.salesforce.com`. In this case, your payment site URL for that invoice record would be `https://sales-payment-site.salesforce.com/?id=a1W6A000000bnF9`.

You can quickly reference the site URL by using an invoice formula field that returns the site URL with the invoice ID at the end. Here are two ways you can get it to your customers.

- Create an email template that customers receive with their invoice document. The body of the template contains a URL that sends the customer to the payment site URL defined on the invoice record.
- Use a Salesforce Billing document generation plug-in to create an invoice document template with a link to the invoice payment site URL.

When a customer clicks the URL, Salesforce Billing loads the payment site. Customers can use the

payment site only to pay the invoice you linked to with the payment site URL.



5. To enable the settings for saving credit card details for future payments and automatic payments, review [Allow Customers to Save Credit Card Details](#).

### Allow Customers to Save Credit Card Details

Allow customers to view several options for saving credit card details when they reach the end of your force.com checkout page. They can save their credit card information as a token and use it for fast payments in the Payment Center. They can also enable automatic payments so that Salesforce Billing uses the token to charge their credit card upon invoice due dates. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

After you enable credit card saving, Salesforce Billing shows two settings at the end of the Enter Credit Card Details page in your org's checkout page. Customers can select these settings as needed.

##### Save information for easy payment later?

Salesforce Billing saves as a token the credit card data that the customer entered on the force.com page. The token generates after the customer clicks **Pay Now**.

Customers can use that credit card to pay balances in the payment center without re-entering credit card details.

##### Sign up for the automatic payment option?

Customers can select this option only if they also selected **Save information for easy payment later?**

Whenever the customer's invoice reaches its due date, Salesforce Billing automatically uses their credit card to pay the invoice balance.

1. From Setup, in the Quick Find box, enter *Installed Packages*, and then select **Installed Packages**.
2. Find the Salesforce Billing package and click **Configure**.
3. Click the **Payment** tab.
4. Select **Save credit card details**.

Salesforce Billing adds credit card detail options to the Enter Credit Card Details page.



### Object and Field Permissions for Lightning Platform Payment Sites

External users require specific object and field permissions to work with Salesforce Billing payments in a Lightning Platform site. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing editions

The following objects require Read access.

- Accounts
- Cases
- Contacts
- Finance Books
- Finance Periods
- Invoice Runs
- Legal Entities
- Opportunities
- Orders
- Payment Gateways
- Payment Runs

The following objects require Read and Create access.

- Error Logs
- Invoice Lines
- Invoices
- Payment Allocations (Invoice)
- Payment Allocations (Invoice Line)
- Payment Methods
- Payment Transactions
- Payments



**Note** Not all gateway integrations are compatible with guest user access. We recommend confirming compatibility with your gateway integration provider. All gateway integrations provided by Salesforce Billing are compatible.

## Salesforce Billing Hosted Card Payments in Experience Cloud Sites and Lightning Pages

The `force:cardPayment` component, also known as the Hosted Card Payments component lets customers make payments on their accounts and save credit card information. The component is available for Experience Cloud sites and Lightning pages. When customers submit their card information, Salesforce Billing sends the request to your company's payment gateway for processing. Because Salesforce Billing is PCI-compliant, you're providing a secure payment process in your Experience Cloud site. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Winter '21 and later

## Important

- Before you can access the Hosted Card Payments component in the Experience Builder or Lightning App Builder, a developer must embed the `force:CardPayment` code in a wrapper component. For developer instructions on coding and configuring the component, read [Hosted Card Payments Lightning Component](#).

The Hosted Card Payments component requires an active Enable Billing HPP Component setting in Salesforce Billing. If your org doesn't have this setting enabled, contact Salesforce Tier 3 Support.

## Using the Hosted Card Payments Component

Developers configure the component so that you can add it to your Experience Cloud site or Lightning page. They can also customize the component to hide certain fields, make certain fields required, and adjust labels based on your org's payment and card saving needs.



For example, a developer can configure a component to save credit card and billing address information for future use (1). Or, they can configure it to receive payments with credit card information (2).

Developers can also add a properties pane that lets you customize field availability, required status, and label names. To access the properties pane, select the component when building your Experience Cloud site or Lightning page. Work with your organization to determine your component's configuration and whether you and other admins can adjust fields and labels using the pane.

The Hosted Card Payments component sends customer payment information to Salesforce Billing, which passes the information to the payment gateway. Salesforce Billing stores the information in your org after it receives a response from the gateway. Credit card information is stored as a tokenized payment method record, while payment information is stored as a payment record.

## Payment Security and PCI Compliance

Credit card tokenization and payments made with the Hosted Card Payments component are secured and PCI-compliant. Customizations made to the component within Salesforce are also secure and PCI-compliant. However, when using customizations outside Salesforce or using custom code, customers and partners are responsible for security and PCI compliance. For more information, see [PCI Compliance in Salesforce Billing](#).

Your payment gateway sends its response back to Salesforce Billing and the component, so you won't have to handle any sensitive customer payment information directly. We've also provided guardrails so that customers aren't charged more than once if they accidentally click the Pay Now button multiple times, refresh the page after sending the payment, or if there's fraudulent activity.



**Example** The Hosted Card Payments component can be used in many ways.

- A customer lands on your Experience Cloud site page for your enterprise release management app. The customer fills out a form to view a demo, which creates a Salesforce lead record. The customer finishes the demo, sees the Hosted Card Payments component, and then submits credit card information to sign up for a three-month trial. After three months, the customer can opt out or continue to use that credit card for the monthly fees.
- Your company has a phone line where customer support agents can take customer payment information to make a sale. Add the component to an account page that your customer support agents can access. The support agent can use the component to collect payment information during a customer call.

### [Set Up the Hosted Card Payments Component](#)

When you add the Hosted Card Payments Component to Experience Cloud sites and Lightning pages, you give your customers and partners a safe and convenient way to make payments and store credit card information. (Salesforce Billing Managed Package)

### [Guidelines for the Hosted Card Payments Component](#)

Review key guidelines to make sure that you get the most out of the Hosted Card Payments component. (Salesforce Billing Managed Package)

### [Permissions for the Hosted Card Payments Component](#)

The Hosted Card Payments component lets customers and partners save credit card information and make payments in an Experience Cloud site or Lightning page. They can use the component to when they have the proper object and field permissions. (Salesforce Billing Managed Package)

## Set Up the Hosted Card Payments Component

When you add the Hosted Card Payments Component to Experience Cloud sites and Lightning pages, you give your customers and partners a safe and convenient way to make payments and store credit card information. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '20 and later

---



**Important** Before you can access the Hosted Card Payments component in the Experience Builder or Lightning App Builder, a developer must embed the `force:CardPayment` code in a wrapper component. For developer instructions on coding and configuring the component, read [Hosted Card Payments Lightning Component](#).

The Hosted Card Payments component requires an active Enable Billing HPP Component setting in Salesforce Billing. If your org doesn't have this setting enabled, contact Salesforce Tier 3 Support.

### [Add the Hosted Card Payments Component to an Experience Cloud Site](#)

After a developer configures the Hosted Card Payments component, add the component to an

Experience Cloud site. Then, customers and partners can submit payments and save credit information through the site. (Salesforce Billing Managed Package)

#### Add the Hosted Card Payments Component to a Lightning Page

After a developer configures the Hosted Card Payments component, add the component to a Lightning page. Then, customers and partners can submit payments and save credit information through the page. (Salesforce Billing Managed Package)

#### Add the Hosted Card Payments Component to an Experience Cloud Site

After a developer configures the Hosted Card Payments component, add the component to an Experience Cloud site. Then, customers and partners can submit payments and save credit information through the site. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '20 and later

-  **Important** Before you can access the Hosted Card Payments component in the Experience Builder or Lightning App Builder, a developer must embed the `force:CardPayment` code in a wrapper component. For developer instructions on coding and configuring the component, read [Hosted Card Payments Lightning Component](#).

The Hosted Card Payments component requires an active Enable Billing HPP Component setting in Salesforce Billing. If your org doesn't have this setting enabled, contact Salesforce Tier 3 Support.

1. From Setup, in the Quick Find box, enter *Digital Experiences*, and then select **All Sites**. Find your Experience Cloud site and click **Builder**.
2. Click the Components icon, then go to Custom Components. Find the Hosted Card Payments component and drag it to your Experience Cloud site page.
3. If your developer has enabled the component's properties pane, you can access it by selecting your component within the builder. Use the pane to hide fields, set required fields, and update labels.

#### Add the Hosted Card Payments Component to a Lightning Page

After a developer configures the Hosted Card Payments component, add the component to a Lightning page. Then, customers and partners can submit payments and save credit information through the page. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '20 and later

-  **Important** Before you can access the Hosted Card Payments component in the Experience Builder or Lightning App Builder, a developer must embed the `force:CardPayment` code in a wrapper component. For developer instructions on coding and configuring the component, read [Hosted](#)

## [Card Payments Lightning Component.](#)

The Hosted Card Payments component requires an active Enable Billing HPP Component setting in Salesforce Billing. If your org doesn't have this setting enabled, contact Salesforce Tier 3 Support.

1. From Setup, in the Quick Find box, enter *Lightning App Builder*, and then select **Lightning App Builder**. Select an existing app or click **New**.
2. In the Lightning Components sidebar, go to Custom Components. Select the Hosted Card Payments component and drag it to the app.
3. If your developer has enabled the component's properties pane, you can access it by selecting your component within the builder. Use the pane to hide fields, set required fields, and update labels.

### Guidelines for the Hosted Card Payments Component

Review key guidelines to make sure that you get the most out of the Hosted Card Payments component. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '20 and later

---

- Customers can pay against only an account.
- In multicurrency orgs, customers can make payments only in the account's currency. In single-currency orgs, the component uses the org's currency.
- You can't change component field positions.
- The component doesn't support custom fields.
- To prevent fraudulent activity, we recommend developing guardrails around the users that can access your payments page. We also recommend that your payments page isn't the first page in your Experience Cloud site or Lightning page.

### Permissions for the Hosted Card Payments Component

The Hosted Card Payments component lets customers and partners save credit card information and make payments in an Experience Cloud site or Lightning page. They can use the component to when they have the proper object and field permissions. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

**!** **Important** Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

---

Available in: Salesforce Billing Winter '20 and later

---

You need the Enable Billing HPP Component Functionality permission enabled to use the Hosted Card Payments component. To enable the permission, contact Salesforce Tier 2 support.

The Hosted Card Payments component requires default object permissions from the Hosted Payment User Permission set. It also requires non-default object permissions, system permissions, and field permissions. We recommend creating a Hosted Payment User Additional Permissions permission set that contains all three additional permission types, then assigning the set to customers and partners who will make payments with the Hosted Card Payments component.

## Hosted Payment User Permission Set

The Hosted Payment User permission set includes the object permissions needed for users to make payments with the Hosted Card Payments component. Assign this permission set to customers and partners who use the component. If the Hosted Payment User permission set hasn't already been enabled for your org, contact Salesforce support. Enabling the permission also exposes a System Permission called **Allow Payment Using Card Hosted Payment Component**. This permission must be assigned to users who will use with the Hosted Card Payments Component.

**!** **Important** Salesforce Billing comes with a Salesforce Billing Admin User permission set that includes permissions for the package's custom objects. However, some Salesforce Billing actions require access to standard objects. As a managed package, Salesforce Billing can't give access to standard objects, so you must manually give users access to standard objects. When you assign permissions for objects with a master-detail Account relationship, such as the Invoice and Payment objects, include access to the Account object.

Hosted Payment User

Object	Read Access	Write Access	Update Access
Finance Book	Yes	No	No
Finance Period	Yes	No	No
GL Rule	Yes	No	No
GL Treatment	Yes	No	No
Legal Entity	Yes	No	No
Payment Run	Yes	No	No
Payment Transaction	Yes	Yes	Yes

## Non-Default Object Permissions

The following permissions aren't enabled by default. They're required for customers and partners to use the Hosted Card Payments component.

Object	Read Access	Write Access	Update Access
Account	Yes	Yes	Yes

Object	Read Access	Write Access	Update Access
Contact	Yes	No	No
Invoice	Yes	No	No
Payment	Yes	Yes	No
Payment Method	Yes	Yes	No

## System Permissions

Each user also requires the system permission Allow Payment Using Card Hosted Payment Component.

## Field Permissions

The following objects have fields that require Read access.

Account	Payment	Finance Period, Payment Gateway, Payment Method, Payment Transaction
<ul style="list-style-type: none"> <li>• Account Number</li> <li>• Bill To Contact</li> <li>• Customer Profile ID</li> <li>• Default Payment Type</li> <li>• Billing Street</li> <li>• Billing City</li> <li>• Billing State</li> <li>• Billing Postal Code</li> <li>• Billing Country</li> <li>• Fax</li> <li>• Name</li> <li>• Phone</li> <li>• Shipping Street</li> <li>• Shipping City</li> <li>• Shipping State</li> <li>• Shipping Postal Code</li> <li>• Shipping Country</li> </ul>	<ul style="list-style-type: none"> <li>• Transaction</li> </ul>	<ul style="list-style-type: none"> <li>• All non-required fields</li> </ul>

## Managing Gateway Transaction Responses

Take actions on your payment transactions based on the gateway response recorded in the transaction's Gateway Status field. If the transaction was successful, reconcile your successful transaction against the

gateway's settled transactions to ensure that Salesforce Billing didn't miss recording any other successful transactions. If the transaction was indeterminate, check the payment gateway to confirm whether the transaction was settled or unsettled. If the transaction failed, take corrective action in Salesforce Billing and try the payment again. (Salesforce Billing Managed Package)

### Recording Gateway Responses

Payment transactions record the results of gateway communication in their Status, Gateway Status, and Gateway Response fields. Review each field to understand what happened during gateway communication. (Salesforce Billing Managed Package)

### Reconciling Successful Transactions in the Gateway

Review settled transactions in your payment gateway to confirm that their amounts match the equivalent successful payment transaction records in Salesforce Billing. Regularly reconciling your transactions this way ensures that you don't miss instances where the payment gateway didn't record, or inaccurately recorded, a settled transaction in Salesforce Billing. (Salesforce Billing Managed Package)

### Indeterminate and Failed Transactions

An indeterminate transaction occurs when Salesforce Billing is unable to process the results of the gateway's call to the customer's bank. Depending on the type of issue that prevented communication, the transaction may or may not be successful. When you see indeterminate payment transactions in Salesforce Billing, first review your external payment gateway to determine whether the transaction was created. Then, you have several options for either recording a successful transaction on your own in Salesforce Billing, or making corrections and retrying the transaction. (Salesforce Billing Managed Package)

### Guidelines for Resolving Failed Transactions

When a payment transaction records a failed gateway status, your options for correction vary based on the type of status, payment gateway provider, and the customer bank. We've provided some generalized recommendations for corrective actions that you can take before retrying the transaction. (Salesforce Billing Managed Package)

### Finalize Payments After a Failed or Indeterminate Transaction

When Salesforce Billing receives an unsuccessful payment gateway response, it goes to the invoice related to the payment transaction and sets the invoice's Corrective Action (Payment Run) field to Action Required. As long as the invoice is in this state, payment runs don't pick up the invoice for payments. After you verify the transaction's status in the gateway, take steps to finalize the payment by manually allocating it or unlocking the invoice and attempting payment again. (Salesforce Billing Managed Package)

### Create a Refund to Represent a Voided Payment

Voided payments often occur to correct a payment that was processed incorrectly—for example, if a customer has a credit card set to autopay but decided after the payment was settled that they wanted to pay using a different card. A credit card transaction can be voided only before settlement. After settlement, you must process it as a refund. After you void a settled transaction in the payment gateway, Generally Accepted Accounting Principles require that you record the void action in Salesforce Billing. To record the void action, create a refund equal to the payment's total amount, and then allocate the refund to the payment. (Salesforce Billing Managed Package)

### Payment Transaction Fields

Before you start working with payments, review payment transaction fields and attributes. (Salesforce Billing Managed Package)

## Recording Gateway Responses

Payment transactions record the results of gateway communication in their Status, Gateway Status, and Gateway Response fields. Review each field to understand what happened during gateway communication. (Salesforce Billing Managed Package)

When Salesforce Billing creates a payment transaction during a payment run, it sets the transaction's Status field to In Progress until the gateway responds or Salesforce Billing determines that the response was indeterminate. Then, it changes the status to Processed. Because Payment Center transactions are created after the gateway's response, their payment transactions automatically have Processed statuses.

 **Note** The Status field's Completed value isn't currently used.

The Gateway Status field contains one of seven values based on the specific response sent from the gateway. Payment gateways have many different types of response codes, which often vary between gateway. However, the developers who set up your payment gateway integration configured your org to filter each of them into one of the seven gateway status values. This way, you can quickly see what happened to a transaction request without worrying about varying response types between different gateway providers.

The Gateway Response field shows one or more sentences providing a more detailed description of what happened in the payment gateway. This value appears exactly as the gateway sent it. If you must take corrective action following a failed transaction, use the gateway response for additional context.

Let's take a closer look at each of the seven gateway status values, what they mean, and what action Salesforce Billing takes after recording a given response.

Gateway Status	Definition	Salesforce Billing Action
Success	The gateway request succeeded.	Attempts payment creation and allocation.
Decline	The call to the gateway was completed successfully, but the payment gateway declined the payment. This response often happens when the customer has insufficient funds to complete the transaction.	Increases the invoice's Declined Payment Count (Payment Run) field value by 1. Sets the invoice's Corrective Action (Payment Run) field to Action Required.
Validation	Customer payment data is incorrect, such a misspelling in the credit card address or	Sets the invoice's Corrective Action (Payment Run) field to Action Required.

Gateway Status	Definition	Salesforce Billing Action
	incorrect CVV.	
Permanent Fail	<p>The gateway call failed, and future requests using the same payment method won't work. This response often happens when the external gateway or customer bank detects some kind of risk, such as attempted fraud. If you receive this response, don't make further payments using the related payment method. The customer must contact their bank to review the issue that prompted the Permanent Fail response.</p>	Sets the invoice's Corrective Action (Payment Run) field to Action Required.
Requires Review	The customer bank requires additional information before completing the payment.	Sets the invoice's Corrective Action (Payment Run) field to Action Required.
Indeterminate	Salesforce Billing didn't receive any response from the gateway. This response usually happens when Salesforce Billing times out waiting for the response. This response doesn't increase the payment run's failure count.	Sets the invoice's Corrective Action (Payment Run) field to Action Required.
System Fail	<p>Salesforce Billing ended the payment request call before receiving a response. For example, Salesforce Billing lost credentials or lost access to its server. This response increases the payment run's failure count. Salesforce Billing ends payment calls if it doesn't receive a response from the gateway in two minutes.</p>	Sets the invoice's Corrective Action (Payment Run) field to Action Required.

After you evaluate your payment transaction's gateway status, you can take your next step.

## Reconciling Successful Transactions in the Gateway

Review settled transactions in your payment gateway to confirm that their amounts match the equivalent successful payment transaction records in Salesforce Billing. Regularly reconciling your transactions this way ensures that you don't miss instances where the payment gateway didn't record, or inaccurately recorded, a settled transaction in Salesforce Billing. (Salesforce Billing Managed Package)

We recommend running a report in Salesforce to evaluate all your successful payment transactions at least one time a day. This way, you have fewer sets of transactions to review at one time compared to larger time periods such as a month. You can then compare the report to a report with the same parameters in your payment gateway. We recommend setting each report to sum the balance of every transaction in your daily or weekly period, so you can quickly see if any discrepancies exist between Salesforce Billing and the payment gateway.

When you reconcile every day, you can also find and correct any potential cash impacts immediately, which benefits your company and your customers.

 **Tip** Run payment transaction reports multiple times a day following high-impact actions such as implementations, data migrations, and package updates due to the larger potential for data errors.

If you find that Salesforce Billing didn't create a payment transaction for a settled payment, create the payment transaction in Salesforce Billing. Then, create a payment from the new payment transaction and allocate it to the invoice that should have been paid as a result of the settled transaction.

## Indeterminate and Failed Transactions

An indeterminate transaction occurs when Salesforce Billing is unable to process the results of the gateway's call to the customer's bank. Depending on the type of issue that prevented communication, the transaction may or may not be successful. When you see indeterminate payment transactions in Salesforce Billing, first review your external payment gateway to determine whether the transaction was created. Then, you have several options for either recording a successful transaction on your own in Salesforce Billing, or making corrections and retrying the transaction. (Salesforce Billing Managed Package)

When Salesforce Billing sets a payment transaction's gateway status to Indeterminate, it also changes the value of related invoice's Corrective Action (Payment Run) field to Action Required. As long as the Action Required value is active, payment runs don't evaluate the invoice for payment. We've configured invoices this way so that there's no risk of Salesforce Billing charging a customer twice with future payment runs before you figure out whether the transaction was successful. We consider an invoice with Action Required to be "locked." To unlock a locked invoice, click **Attempt Refund and Unlock Invoice** on the payment transaction record. We'll cover more information on the Attempt Refund and Unlock Invoice button later.

When you're trying to match an indeterminate transaction from Salesforce Billing with its equivalent in the gateway, we recommend searching by matching values. For example, you could search by matching

account numbers, routing numbers and transaction amounts. You can also search by instances where the payment transaction's Gateway Date field matches the date and time that the transaction was recorded in the gateway.

In some cases, such as the payment request call failing, your transaction won't exist in the gateway. In this case, unlock the invoice so that it's available for future payment attempts.

If the transaction exists in the gateway but failed with the customer's bank, correct the issues with the payment, then unlock the invoice.

If the transaction was successful and exists in the gateway, check whether the transaction is settled or unsettled. An unsettled transaction hasn't been processed by the customer's bank yet. In most cases, the bank processes an unsettled transaction at a predetermined time, such as the end of a business day. Settled transactions have been processed by the customer bank.

For successful transactions, you need to decide if you want to create a payment and manually allocate it in Salesforce Billing, or if you want to cancel the payment. If you want to cancel the transaction, your steps vary based on whether the transaction is unsettled or settled:

- Unsettled transactions can't be refunded because they haven't been applied to the customer's bank yet. Instead, follow your payment gateway provider's directions to void the transaction. Later, you'll need to record the voided transaction in Salesforce Billing.
- Settled transactions must be refunded.

If you want to record a successful transaction, create an equivalent payment in Salesforce Billing as a child of the indeterminate payment transaction, then allocate the payment to the target invoice. For example, if your indeterminate transaction had a balance of \$1000, go to the transaction's Payment related list, create a payment for \$1000, and then allocate it to your invoice.

 **Note** Although invoices with corrective action required are ineligible for payment runs, they can still be paid in the Payment Center.



## Guidelines for Resolving Failed Transactions

When a payment transaction records a failed gateway status, your options for correction vary based on the type of status, payment gateway provider, and the customer bank. We've provided some generalized recommendations for corrective actions that you can take before retrying the transaction. (Salesforce Billing Managed Package)

Gateway Status	Recommended Action
Decline	Banks can decline calls for many reasons. Customers can sometimes fix the issue, such as

Gateway Status	Recommended Action
	<p>transferring funds or calling their bank to unfreeze an account. In these cases, they can attempt a payment again. Due to the wide variety of reasons for declined calls, Salesforce only updates payment fields and leaves customers to manage a business process for declined payments.</p> <p>When a payment transaction sets its gateway response to Declined, Salesforce Billing updates Decline Payment Count (Payment Run) field on the related invoice. Users and processes can't reset or decrease this field. This field updates only in response to payment runs. Payments from the payment center aren't included. We recommend reviewing your transactions for declined responses so that you can monitor payment runs that have unexpected spikes in declined payments.</p>
Validation Error	<p>Payment runs don't create payments from transactions with a gateway status of Validation Error. Change the incorrect payment information on your payment method so that future payment runs can use it for payments. For example, if the transaction failed because the payment method listed the customer's name incorrectly, update the customer's name on the payment method.</p>
Permanent Fail	<p>The gateway doesn't accept any further requests that use the same payment method, so future payment runs using the payment method have the same result. We recommend creating a workflow rule that deselects your payment method's Auto-Pay field when a related transaction receives a permanent fail response. That way, you don't have the risk of more failed responses before you can delete or edit the payment method.</p>
Requires Review	<p>Gateways respond with a review code when the gateway call initially fails, but the payment method can still work following extra processing. For example, some banks send out this type of</p>

Gateway Status	Recommended Action
	<p>response when they have further questions about the payment request, and they provide an authorization code manually when the payment manager calls the processor.</p> <p>When a payment transaction sets its gateway status to Review, Salesforce Billing doesn't create a payment automatically. Depending on the customer's follow up with the payment processor, they can either auto-void the transaction or manually create a payment from the transaction record.</p>
System Fail	<p>Salesforce Billing provides a system fail response when it has to end its gateway payment request before receiving a response. System Fail responses can happen due to gateway server errors or invalid customer credentials or anytime the request times out before it receives a gateway response.</p> <p>System failures occur before the Salesforce Billing request reaches the gateway, so there's no risk of an unaccounted payment remaining in the gateway. If your transaction has a System Fail gateway status, you can manually create the payment or wait for another payment run to pick up your invoice.</p>
Unknown	<p>When your transaction results in an Unknown gateway status, the related invoice was manually paid or had billing data changed between the payment run's first and second batches. This situation usually results from the following actions.</p> <ul style="list-style-type: none"> <li>• Another user manually paid the invoice.</li> <li>• A customer paid the invoice in the payment center.</li> <li>• The invoice's target date was canceled.</li> <li>• The invoice was canceled and rebilled.</li> </ul> <p>If your invoice had a billing field change that's preventing payment run inclusion, fix that field so</p>

Gateway Status	Recommended Action
	that Salesforce Billing picks it up in the next payment run.

## Finalize Payments After a Failed or Indeterminate Transaction

When Salesforce Billing receives an unsuccessful payment gateway response, it goes to the invoice related to the payment transaction and sets the invoice's Corrective Action (Payment Run) field to Action Required. As long as the invoice is in this state, payment runs don't pick up the invoice for payments. After you verify the transaction's status in the gateway, take steps to finalize the payment by manually allocating it or unlocking the invoice and attempting payment again. (Salesforce Billing Managed Package)

Payment Transactions contain an Attempt Refund & Unlock Invoices button. When you select it, Salesforce Billing requests that the payment gateway attempts to refund any transactions related to the invoice in the payment gateway. If the gateway contains settled or unsettled transactions, the refund request goes through and removes the transactions from the gateway, and then Salesforce Billing changes the invoice's corrective action to null. If the gateway doesn't contain transactions, nothing else happens in the gateway and Salesforce Billing changes the invoice's corrective action to null.

-  **Note** When you evaluate an unsuccessful payment transaction in Salesforce Billing, we recommend going to the payment gateway, finding the gateway response ID for your unsuccessful transaction, and recording the ID on the Salesforce Billing transaction record. This way, you can quickly confirm that the payment transaction failed in the gateway if the customer has any questions in the future.

However, if the payment is settled in the gateway, many customers don't want to refund it and run the payment again. In this case, you can manually create a payment in Salesforce Billing and allocate it to the invoice.

1. Go to your payment transaction record in Salesforce Billing.
2. If the transaction failed, or was indeterminate in Salesforce Billing and unsettled in the gateway, click **Attempt Refund and Unlock Invoice**.  
In the payment gateway, Salesforce Billing refunds any unsettled transactions related to the invoice, and then changes the invoice's corrective action to null. The invoice is now eligible for future payment runs or Payment Center transactions.
3. If the payment was settled in the gateway, create a payment for the equivalent amount in Salesforce Billing.
  - a. Review the amount of the settled payment in the gateway.
  - b. In Salesforce Billing, create a payment with the same amount.
  - c. Allocate your payment to your invoice.

## Create a Refund to Represent a Voided Payment

Voided payments often occur to correct a payment that was processed incorrectly—for example, if a customer has a credit card set to autopay but decided after the payment was settled that they wanted to pay using a different card. A credit card transaction can be voided only before settlement. After settlement, you must process it as a refund. After you void a settled transaction in the payment gateway, Generally Accepted Accounting Principles require that you record the void action in Salesforce Billing. To record the void action, create a refund equal to the payment's total amount, and then allocate the refund to the payment. (Salesforce Billing Managed Package)

 **Important** The payment transaction's Void button is not currently supported.

1. Log in to your payment gateway and void your transaction. Take note of the transaction's amount, and confirm that it matches the value of the payment created for it in Salesforce Billing.
2. Go the Salesforce Billing payment record for your voided payment and click **Refund**.
3. Give your refund the following values.
  - Amount to Refund: the full amount of the voided payment
  - Refund Mode: External
  - Refund Payment Method: The same payment method used to create the voided payment
  - GL rule, GL treatment, and Legal entity: As needed based on your GL reporting requirementsSalesforce Billing creates a refund line (payment) for the entire balance of the payment record.

## Payment Transaction Fields

Before you start working with payments, review payment transaction fields and attributes. (Salesforce Billing Managed Package)

### Account

Account containing the invoice that the payment transaction is set to pay.

### Amount

Amount that Salesforce Billing attempts to use against the balance of an invoice.

### Bank Account Name

User-defined value. Used to record information about externally recorded ACH payments.

### Bank Account Number

User-defined value. Used to record information about externally recorded ACH payments.

### Bank Account Type

User-defined value. Used to record information about externally recorded ACH payments.

**Bank Name**

User-defined value. Used to record information about externally recorded ACH payments.

**Bank Routing Code**

User-defined value. Used to record information about externally recorded ACH payments.

**Card Expiration Month**

Information about the customer credit card related to the payment method. Sent from the payment gateway following a transaction request.

**Card Expiration Year**

Information about the customer credit card related to the payment method. Sent from the payment gateway following a transaction request.

**Card Type**

Information about the customer credit card related to the payment method. Sent from the payment gateway following a transaction request.

**Gateway Date**

The date that the transaction request occurred in the payment gateway.

**Gateway Status**

Shows one of seven values that summarizes the payment gateway's response to the payment transaction request.

**Invoice**

The invoice that Salesforce Billing is attempting to pay through the payment transaction. If the transaction is successful, Salesforce Billing creates a payment record and attempts to allocate it to the invoice.

**Notes**

Area for user-defined notes on the payment transaction.

**Payment Created**

Shows whether Salesforce Billing created a payment for a successful transaction. If this field is deselected on a successful transaction, review why Salesforce Billing didn't create the payment—for example, the user who owns the payment run didn't have permission to access payment records. After you resolve the issue, create a payment for the same amount as the payment transaction, and then allocate it to the transaction's invoice on your own.

**Payment Gateway**

The Salesforce Billing payment gateway record used to establish communication with the external payment gateway. Inherited from the payment method.

**Payment Method**

The payment method used to provide customer payment information to the payment gateway.

**Payment Run**

The payment run that created the payment transaction to record the result of external payment gateway communication.

**Request - Credit Card Number**

Shows a tokenized version of the credit card number sent to the customer bank within the transaction request. Salesforce Billing doesn't store credit card numbers or other sensitive customer payment information. Instead, it stores tokenized versions of that information and sends the tokens to the external payment gateway.

**Request - Transaction Type**

Shows the type of payment action requested of the customer bank: authorization, charge, void, or refund.

**Response**

Detailed message contained in a gateway response alongside the response code, response, response status, and response message. Although specific responses vary between gateways, the response usually provides more information about the response message.

**Response Code**

Numeric value contained in a gateway response alongside the response status, response, and response message. The code is used as an indexing value for different types of responses. You can generally ignore it and focus on the response and response message to get all the information you need about the results of your transaction request.

**Response Gateway ID**

Alphanumeric value sent in a gateway response. Used by developers for troubleshooting.

**Response Message**

Message sent from the external payment gateway to show the result of the transaction request. You can review this message for additional context regarding the value of the transaction's gateway status.

**Response Status**

Message sent from the external payment gateway to show the result of the transaction request. While this value can vary between gateways, it's often the same as the response message.

**Status**

The transaction's status is In Progress while the external payment gateway communicates with the customer bank. Salesforce Billing changes it to Completed after the gateway returns a

transaction response.

#### Type

Shows the type of action requested from the customer bank: authorization, charge, void, or refund.

## Locked and Unlocked Invoices

Salesforce Billing locks invoices from payments when a user makes a charge request to the payment gateway from the Payment Center or a force.com payment site. It also locks invoices after a payment run picks them up for evaluation. The invoice unlocks only when Salesforce Billing confirms that the gateway provides a valid response, or when the payment run finishes processing the invoice. Users must not risk making duplicate payments if the gateway response is indeterminate. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '22 and later

---

Available in: Salesforce Classic and Lightning Experience

---

Available in: **Professional, Enterprise, Unlimited, and Developer** Editions

Users can't pay invoices that are locked or that have Action Required as the Corrective Action (Payment Run) value.

If the gateway response causes the payment transaction's gateway status to be Success, Salesforce Billing unlocks the invoice.

If the gateway status is Indeterminate, External System Error, Decline, Permanent Fail, Validation Error, or Requires Review, the invoice remains locked. When the gateway status is Indeterminate, you can see the [indeterminate transaction](#) in your payment gateway's merchant portal to determine if the payment was successful. If the payment was successful, you can attempt a refund and unlock the invoice.

**!** **Important** The payment transaction's Attempt Refund and Unlock Invoice button isn't related to the Invoice Locked field. The button can change the invoice's Corrective Action field from Action Required to null following an unsuccessful or indeterminate payment run.

When a payment run doesn't process an invoice successfully, Salesforce Billing sets the invoice's Corrective Action (Payment Run) field to Action Required. See [Recording Gateway Responses](#).

The Salesforce Billing Admin permission set contains Read and Edit access to the Invoice Locked field. If your org uses custom permission sets, provide Read and Edit access to the Invoice Locked field to these users:

- Users who create or post invoices.

- Users who process payments from the Payment Center.
- Users who initiate payment runs.
- Guest users of payment sites.

## Payment Allocations

A payment allocation represents the amount of a payment that has been applied to an invoice line's balance. Users can manage allocations in Salesforce Billing's Payment Allocation page. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

When a payment run creates a payment for an invoice, it automatically creates allocations against the full balance of each invoice line. However, if a user created a payment record manually, they can also make any number of allocations as long as the total allocation amounts are within the payment's amount. In this case, an allocation can cover all or part of an invoice line's balance.

For example, let's say an account has an invoice with a \$1000 invoice line, a \$500 invoice line, and a \$300 invoice line. The same account also contains an unallocated \$1500 payment record. A user could make a \$1000 allocation against the first invoice line and a \$500 allocation against the second invoice line. They could also apply a \$300 payment allocation to each invoice line, and leave the payment record with an unallocated balance for future use. One payment's allocations can also be made to invoice lines from different invoices, as long as the invoices are on the same account.

Unallocated payments remain on an account indefinitely.

#### [Allocate a Payment](#)

Assign payments to one or more invoice lines within your account. (Salesforce Billing Managed Package)

#### [Unallocate an Invoice Line Payment](#)

If an invoice line payment was made in error, unallocate it from the invoice line. (Salesforce Billing Managed Package)

#### [Reallocate a Payment](#)

Move a payment allocation from one invoice on your account to another. (Salesforce Billing Managed Package)

#### [Pay Several Invoices at Once](#)

Allocate a payment toward several invoices on the same account. (Salesforce Billing Managed Package)

#### [Correct a Posted Payment in Lightning Experience](#)

Salesforce Billing must retain a copy of all payments for legal and accounting purposes. You can fix an error on a posted payment by refunding it. The refund reduces the erroneous credit note's balance to zero. You can then create and post a payment with the correct values. (Salesforce Billing Managed

Package)

#### Payment Allocation Errors

Sometimes, your payment run can receive a successful gateway response and create a payment, but fail to allocate the payment to your invoice. This can happen due to user-defined validation rules or database errors. To finish the allocation process, manually create an allocation record and assign it to your invoice. (Salesforce Billing Managed Package)

## Allocate a Payment

Assign payments to one or more invoice lines within your account. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. If your payment record has a status of Draft, change it to Posted. Draft payments can't be allocated.
2. From your payment record, click **Allocations**.  
The payment allocation page shows a list of posted invoice lines with a balance above zero in your account.
3. From the Invoice Lines section, select the invoice lines that you want to allocate funds toward.  
You can select any number of invoice lines. Salesforce Billing displays an Amount field for each invoice line you select.
4. For each invoice line you selected, enter the amount you wish to allocate.  
The total amount allocated must be less than or equal to your payment's balance.
5. Click **Allocate**.



**Note** If multiple browser tabs to allocate payments are open, refresh the browser before you click **Allocate** to avoid allocating the same payment twice.

If you paid off an invoice line in full, Salesforce Billing removes it from the payment allocation page and updates its status to Paid. If you made a partial payment, you'll see the invoice line on the payment allocation page with an updated balance.

## Unallocate an Invoice Line Payment

If an invoice line payment was made in error, unallocate it from the invoice line. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. From a payment record that contains at least one allocation, click **Allocations**.
2. From the Existing Invoice Line Allocations related list, find the payment allocation you want to unallocate and click **Unallocate**.

-  **Note** If multiple browser tabs to unallocate payments are open, refresh the browser before you click **Unallocate** to avoid unallocating the same payment twice.

Invoice lines that received an allocation from that payment reset their balance and payment fields to their values before the allocation was applied.

## Reallocate a Payment

Move a payment allocation from one invoice on your account to another. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

You can use this feature on only accounts that have at least two posted invoices with balances. First, you unallocate the payment from an invoice. Then you allocate the payment to invoice lines from a different invoice within your account.

1. From a payment record that contains at least one allocation, click **Allocations**.
2. From the Existing Invoice Line Allocations list, find the payment allocation you want to unallocate and click **Unallocate**.

-  **Note** If multiple browser tabs to unallocate payments are open, refresh the browser before you click **Unallocate** to avoid unallocating the same payment twice.

Invoice lines that received an allocation from that payment reset their balance and payment fields to their values before the allocation was applied.

3. From the Invoice Lines section, find the different invoice containing lines that you want to allocate the payment toward, and then select those invoice lines.
4. Choose an amount for each of the lines.
5. Click **Allocate**.

-  **Note** If multiple browser tabs to allocate payments are open, refresh the browser before you click **Allocate** to avoid allocating the same payment twice.

If you paid off an invoice line in full, Salesforce Billing removes it from the payment allocation page and updates its status to Paid. If you made a partial payment, you'll see the invoice line on the payment allocation page with an updated balance.

## Pay Several Invoices at Once

Allocate a payment toward several invoices on the same account. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

**Example**

Your account contains the posted following invoices.

- Invoice 1
  - Invoice Line 1: Balance of \$500
  - Invoice Line 2: Balance of \$300
- Invoice 2
  - Invoice Line 1: Balance of \$200
  - Invoice Line 2: Balance of \$400
  - Invoice Line 3: Balance of \$100

A customer provides a \$1500 payment to the account. Since both your invoices look up to the same account, you can allocate your payment to all the invoice lines on both invoices from the Allocation page.

1. From the payment's detail page, click **Allocations**.
2. On the Payment Allocation page, select all five invoice lines from the two invoices.
3. For each invoice line, enter the invoice line's total balance in the Amount field.
4. Click **Allocate**. If you paid off an invoice line in full, Salesforce Billing removes it from the payment allocation page and updates its status to Paid. If you made a partial payment, you'll see the invoice line on the payment allocation page with an updated balance.

## Correct a Posted Payment in Lightning Experience

Salesforce Billing must retain a copy of all payments for legal and accounting purposes. You can fix an error on a posted payment by refunding it. The refund reduces the erroneous credit note's balance to zero. You can then create and post a payment with the correct values. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Salesforce Billing follows Generally Accepted Accounting Principles. The principles require that users must always be able to track the posting of credits, debits, payments, and refunds through a series of allocations back to a business's general ledger. To ensure that these records are always available for legal and accounting purposes, Salesforce Billing doesn't allow you to delete credit notes, debit notes, payments, or refunds. If you make an error, offset the erroneous record by reducing its balance to zero, and then creating a record with the correct values.

 **Important** The payment's Refund button is available only in Lightning Experience.

1. Go to the account containing the payment that you want to offset. Navigate to the Payments related list and select your payment.

2. If your payment has any allocations, unallocate them.
3. Refund your payment.
  - a. To record the results of a refund processed outside Salesforce, follow the steps in [Record an External Payment Refund](#).
  - b. To process a refund in Salesforce Billing, follow the steps in [Issue Electronic Refunds in Lightning Experience](#).
4. After you process the refund, verify that the balance of your payment is reduced to zero.
5. Make a new payment with correct information.

## Payment Allocation Errors

Sometimes, your payment run can receive a successful gateway response and create a payment, but fail to allocate the payment to your invoice. This can happen due to user-defined validation rules or database errors. To finish the allocation process, manually create an allocation record and assign it to your invoice. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

## Failed Allocations

Failed allocations can happen in response to user-created validations or failed database inserts. After the failure, the payment run updates status fields on the following objects.

### Invoice

Last Payment Run Processing Message: Payment Allocation Failed

### Payment Transaction

Status: Processed

Payment Creation Status: Payment Allocation Failed

### Payment

Allocation Status: Fail

In some cases, the payment run updates the payment transaction with an error log showing details about why the allocation failed. However, if the payment run itself fails before making the allocation (for example, due to a lost server connection), the run is unable to make an error log on the transaction.

We recommend creating a workflow rule or process to provide an alert when a transaction changes its payment creation status to Payment Allocation Failed. If the payment run produced an error log, you can also review it to find the transaction and then the payment where the allocation failed.

Once you've identified the payment, manually create an allocation from that payment to your invoice. Manual allocation is important as future payment runs will attempt to pick up that invoice if its related payment remains unallocated.

## Duplicate Allocations

If multiple browser tabs to allocate or unallocate payments are open, refresh the browser before you click **Allocate** or **Unallocate** to avoid allocating or unallocating the same payment twice.

## Issuing Credits

Credit notes allow you to allocate a negative balance change to an invoice line. This feature is useful for adjusting errors in unpaid or partially paid invoices. You can also provide credit to a user account and then allocate that credit to decrease the balance of their invoices at a later date. Finally, credit notes provide accurate and complete recordkeeping of the transactions that occur after an invoice has been posted. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The credit note object has a lookup to the account. You can allocate credit notes only to invoice lines on that account. By default, credit notes have a draft status. Tax hasn't yet been applied and no allocations have been made. Without any additional actions or processes, the credit note remains in draft status and doesn't show up in finance periods or revenue transactions.

Credit note lines track the actual amount to be allocated across one or more invoice lines. They contain subtotal, tax, and total amount (with tax) fields, and a lookup to the product object. The product lookup provides context for why you're allocating the credit note line. For example, if your credit note allocates \$50 to a customer's invoice based on store credit, the Product field looks up to the Store Credit product record. A credit note record stores its credit note lines in a related list, and provides a summary of its total credit note values. This relationship allows you to organize your credit note lines into one credit note based on their purpose.

A positive credit note line allocation decreases the balance of an invoice line. A negative credit note line allocation increases the balance of an invoice line. Consider the following examples.

- Allocating a \$100 credit note line to a \$500 invoice line changes the invoice line's balance to \$400.
- Allocating a -\$100 credit note line to a -\$500 invoice line changes the invoice line's balance to -\$400.

Salesforce Billing provides five ways to create a credit note.

Use Case	Action	Result
Your posted invoice contains an	Click <b>Cancel and Rebill</b> on the	Salesforce Billing creates a credit

Use Case	Action	Result
<p>error, and you must rebill the customer for the billing period. Since you can't change an invoice's total amount, you must apply credit note lines to each invoice line so you can zero out the balance.</p>	<p>invoice.</p>	<p>note equal to the invoice's total balance. The credit note contains credit note lines that are allocated in full to each of your invoice lines. Salesforce Billing also evaluates each invoice line's originating order product and reverts its Next Billing Date field to its previous value.</p>
<p>Your posted invoice must be fully credited so the customer isn't rebilled for the billing period. Crediting an invoice without rebilling effectively gives customers a free product for one-time products, or a free billing period of service for recurring products.</p> <p>For example, you shipped damaged goods before their payment or charged them for an extra service period after they canceled their service and paid the final bill.</p>	<ul style="list-style-type: none"> <li>Click <b>Credit</b> on the invoice.</li> <li>In Salesforce Billing Summer '19 and later, click <b>Credit Center</b> on the account containing your invoice.</li> </ul>	<ul style="list-style-type: none"> <li>If you clicked <b>Credit</b> on the invoice, Salesforce Billing creates a credit note equal to the invoice's total balance. The credit note contains credit note lines that are allocated in full to each of the invoice lines.</li> <li>In the Credit Center, you can assign credit notes to any posted invoice with a non-zero balance on your account. You can allocate the credit notes fully or partially to your invoice lines.</li> </ul> <p>In both methods, Salesforce Billing doesn't revert credit order products to their previous next billing dates.</p>
<p>When you amend an order product and reduce its quantity or price, invoice lines from the amended order product often have a negative balance.</p>	<p>Click <b>Convert Negative Lines</b> on the invoice.</p>	<p>Salesforce Billing creates a credit note with credit lines equal to the balances of each of your negative invoice lines. You can apply these credit note lines to payments or to reduce the balance of the same invoice.</p>

Your credit note lines contain an optional lookup to a legal entity. Provide a value to this lookup if you want to associate your credit note line to a revenue schedule or GL account for bookkeeping and revenue tracking. You can also manually create a credit note and then create its credit note lines.

You can also manually create a credit note and then create its credit note lines. Manually created credit

notes use the following guidelines.

- To save a credit note line, you have to enter only the Account field and the Product field.
- If you don't enter a credit note line date, the credit note line inherits its date from the parent credit note.
- When you save your credit note line, Salesforce Billing provides its rule and treatment lookups based on the credit note line's legal entity and product.
- Credit note lines save in the draft status by default. You can post a credit note line by changing its status to Posted, posting the parent credit note record, or applying tax on the parent credit note record.

 **Tip** You can't delete draft and posted credit notes. You can cancel draft credit notes by changing the Status field to Canceled. You can't cancel a posted credit note. Instead, create a debit note with a debit note line equal to your credit note's balance and allocate it to your credit note.

#### [Cancel and Rebill an Invoice](#)

Use the Cancel and Rebill option to revert the invoice to its prior state before the most recent billing cycle. Use this option to fix any errors in the invoice record. (Salesforce Billing Managed Package)

#### [Convert a Negative Invoice Line to a Credit Note](#)

When you amend an order to decrease the quantity of an order product, your next invoice often has a corresponding invoice line with a negative balance. Convert these negative invoice lines to credit notes and issue those credit notes as refunds or apply them to other invoice lines. (Salesforce Billing Managed Package)

#### [Create a Credit Note and Apply Tax](#)

Create a credit note record and several credit note lines, then apply tax to the credit note. After applying tax, you can allocate your credit note lines to change the balance of an invoice line. (Salesforce Billing Managed Package)

#### [Create a Credit Note Without Tax](#)

Create a credit note record so you can allocate it at a future date. (Salesforce Billing Managed Package)

#### [Guidelines for Credit Note Tax](#)

Salesforce Billing calls your external tax engine to calculate tax for your credit note lines. You can also estimate tax before applying it to the credit note lines. (Salesforce Billing Managed Package)

#### [Allocate a Credit Note](#)

Allocate your credit note to decrease the change the balance of one or more invoice lines. (Salesforce Billing Managed Package)

#### [Issue Partial Credit](#)

Allocate a negative balance change to an invoice line to adjust errors in unpaid or partially paid invoices. (Salesforce Billing Managed Package)

#### [Correct Errors on Posted Credit Notes](#)

Salesforce Billing must retain a copy of all credit notes for legal and accounting purposes. To fix an error on a posted credit note, create a debit note of equal value and allocate it to each line of the original credit note. The allocation reduces the erroneous credit note's balance to zero. You can then create and post a credit note with the correct values. (Salesforce Billing Managed Package)

## Credit Center

Billing operations and customer service teams can use the Credit Center to quickly credit multiple invoice lines on an invoice. They can credit invoice lines, apply tax per credit note line, and preview the results of the credit note before posting it to the invoice. (Salesforce Billing Managed Package)

## Cancel and Rebill an Invoice

Use the Cancel and Rebill option to revert the invoice to its prior state before the most recent billing cycle. Use this option to fix any errors in the invoice record. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 7.0 and later

---

You might run into mistakes on your invoice. For example, an order product didn't produce an invoice line, or your order had an incorrect billing address. To return your invoiced order products to the state before you most recently billed them, click **Cancel and Rebill** on your invoice. You can cancel and rebill draft and posted invoices.

 **Important** For audit trail purposes and data integrity issues, we recommend that invoices are never deleted. Instead, use the Cancel and Rebill option to cancel the invoice and reset the order products to be invoiced again.

When you click Cancel and Rebill, Salesforce Billing finds the order products related to the invoice's invoice lines, then rolls each order product's next billing date back to its value on the previous billing cycle. For example, your order product bills monthly with a billing day of month of 16. You cancel and rebill an invoice on 05/13, while the order product's next billing date is 05/16. The previous next billing date was 04/16. In this case, Salesforce Billing changes the order product's next billing date to 04/16. The order product's billing metric fields, such as Pending Billing Amount (without tax) and Billed Amount (without tax) also roll back to their previous values.

When the Cancel and Rebill process rolls back an order product's next billing date, the Override Next Billing Date and Bill Through Date Override fields are set to null. If they had values during the previous billing cycle that you want to use again, you must re-enter them.

Next, Salesforce Billing follows one of two processes based on the invoice's status.

 **Note** On a credit note, the credit note date and effective tax date are aligned with the date the invoice is canceled and rebilled. This approach ensures that a credit note isn't issued with a backdated entry.

## Draft Invoice

When you change an invoice's status to Canceled, it's no longer included in invoice and payment runs,

and it can't be posted. Invoices that are in draft status show these changes:

- The invoice status flag changes to white.
- The invoice payment status changes to unpaid.

## Posted Invoice

Users post an invoice when they send it to a customer. At this stage, Salesforce Billing records the invoice's financial transactions in finance books and the general ledger. Since the invoice is a legal document, you must account for all transactions against it. When you cancel and rebill an invoice, Salesforce Billing:

- Creates a credit note on the invoice that matches the total balance of the invoice.
- Creates the credit note and adds credit note lines with balances equal to each of your invoice line balances.
- Allocates the full amount of each credit note line to the corresponding invoice line.
- Changes the invoice's status to Rebilled.

 **Note** Rebilled invoices aren't evaluated by invoice runs and payment runs, and can't be posted.

- Sets the invoice's status flag to green.
- Sets the invoice's accounts receivable (AR) status to Cancel and Rebill. This value doesn't perform any other actions on its own, but you can use it to note that the invoice was canceled and rebilled when you record the invoice's data in an external AR or general ledger platform.
- Changes the invoice's payment status to Paid.

If a posted invoice has collected payments, you must unallocate them completely before Salesforce Billing allows you to cancel and rebill. You can then reallocate the payments to the correct invoice lines when you create your invoice.

If a posted invoice can't be canceled and rebilled, or failed a cancel and rebill process because of CPU timeout, use a credit note, a debit note, or an additional invoice to make adjustments.

 **Note** The maximum number of invoice lines of a posted invoice that can be canceled and rebilled depends on factors that are mentioned in the [scale testing guidelines to cancel and rebill invoices](#). Even without automation, external tax integrations, or roll-up summaries to Account records, the maximum number of invoice lines that can be successfully canceled and rebilled is between 325 and 350.

You can also change your objects before billing the rolled-back order products again, such as updating tax rates or expanding the scope of a billing rule.

### Add the Cancel and Rebill Button to Invoice Page Layouts

Help users take control of their invoices when you add the Cancel and Rebill button to invoice page layouts. (Salesforce Billing Managed Package)

### Scale Testing Guidelines to Cancel and Rebill Invoices

Salesforce Billing can post invoices with more lines than the number of invoice lines that can be

canceled and rebilled. So, scale test the Cancel and Rebill Invoices feature in a full sandbox to avoid errors when using the feature for posting invoices in production. (Salesforce Billing Managed Package)

## Scale Testing Guidelines to Cancel and Rebill Invoices

Salesforce Billing can post invoices with more lines than the number of invoice lines that can be canceled and rebilled. So, scale test the Cancel and Rebill Invoices feature in a full sandbox to avoid errors when using the feature for posting invoices in production. (Salesforce Billing Managed Package)

 **Note** Before you test the Cancel and Rebill Invoices feature in sandbox, make sure that your sandbox is recently refreshed, and uses the same automation and data model as your production org.

We recommend scale testing starting with invoices that have over 100 lines.

## Org Limits

Scale testing helps you determine your org's limits for canceling and rebilling invoices with a large number of invoice lines.

## Automation on Objects

Rolling back invoices by using the Cancel and Rebill Invoices feature affects the Invoice, Invoice Lines, Order, Order Products, Credit Note, Credit Note Line, and the Credit Note Allocation objects. If there are rollup summary fields from the Invoice or the Credit Note objects to the Account object, using the Cancel and Rebill Invoices feature can update and trigger automation on the Account object. This automation adds to the risk of exceeding a governor limit. So, if you plan to use automation on any of these objects, we recommend testing your automation in a full sandbox org before adding the automation to production. If there are numerous automations on these objects, the org limit can be reached with fewer than 100 invoice lines.

## External Tax Integration

Using an external tax integration can reduce the maximum number of invoice lines that can be canceled and rebilled.

### See Also

[Cancel and Rebill an Invoice](#)

## Convert a Negative Invoice Line to a Credit Note

When you amend an order to decrease the quantity of an order product, your next invoice often has a corresponding invoice line with a negative balance. Convert these negative invoice lines to credit notes

and issue those credit notes as refunds or apply them to other invoice lines. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

You can also use REST API to evaluate several invoices at once and automatically generate the necessary credit notes. For more information, review [REST API for Converting Invoice Lines with Negative Balances](#) in the Salesforce Billing Developer Guide.

If you invoice an amended order product that has a negative price, the resulting invoice line has a negative balance. For example, let's say you amended an order to reduce the quantity of your SaaS subscriptions from 4 to 3. The reduction created an amendment order product with a price of -\$100. If your billing setup invoices the order product as one invoice line without any price changes, it will appear as an invoice line with an amount of -\$100 on your next invoice.

In this scenario, we'll assume that your invoice has an invoice line with a balance of -\$500 and an invoice line with a balance of -\$200.

1. Go to your invoice and click **Convert Negative Lines**.

Salesforce Billing displays the Create Credit Note From Negative Lines page. The page displays all your invoice's lines with a negative balance.

 **Note** After you click **Convert Negative Lines**, the Cancel and Rebill action on the invoice is disabled.

2. Select the negative invoice lines you want to convert. Click **Convert**.

3. Salesforce Billing creates a credit note containing a credit note line with a balance of \$500 and a credit note line with a balance of \$200. You can access this credit note on your invoice's Credit Notes related list.

## Create a Credit Note and Apply Tax

Create a credit note record and several credit note lines, then apply tax to the credit note. After applying tax, you can allocate your credit note lines to change the balance of an invoice line. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later

---

Usually, Salesforce Billing automates credit note creation when you provide a refund or rebill an invoice. However, you may sometimes need to create a credit note record on your own. In this scenario, you'll create a credit note, associate it with an account, then create several credit note lines. You can then make a call to your tax engine and apply tax to each of your credit note lines.

1. Go to the Credit Notes tab and click **New**, or click **New Credit Note** in your account's Credit Notes related list.

 **Note** Salesforce Billing doesn't support creating credit notes from the invoice's Credit Notes related list.

2. If your credit note doesn't have a value in the Account field, choose an account for your credit note. You can apply the credit note to any invoice lines on that account.

3. Set the Credit Note Source Action field to Manual.

The source action allows other users to review how the credit note was created.

4. Provide a credit note date.

Salesforce Billing records credit note transactions in the finance period that covers the credit note's date.

5. Provide an effective tax date.

Salesforce Billing sends this date to your tax engine during tax calculation. If you're manually applying tax to your credit note, this is a required field.

6. Fill out the tax address fields.

Salesforce Billing sends this address to your tax engine during tax calculation. If you're manually applying tax to your credit note, these are required fields.

7. Click **Save**.

8. From your credit note record, click **New Credit Note Line**.

9. Enter a product lookup in the Product field.

The product lookup field is for reference purposes. The relationship is important for reporting on transaction history in the general ledger. This product doesn't have to be the product related to the invoice line you want to apply your credit note toward. For example, if your credit note allocates \$50 to a customer's invoice based on store credit, the Product field should look up to the Store Credit product record.

10. If your implementation uses legal entities, enter a legal entity.

11. Give your credit note line a value for the subtotal field

This value represents the total amount available for allocation on this credit note line. It does not include tax. When you apply tax, Salesforce Billing calculates tax for each credit note line, adds it to the line's subtotal, and shows the results in the line's Total Amount (With Tax) field.

12. If you want Salesforce Billing to calculate tax for this credit note line, select **Calculate Tax?**

13. Change the value of the Tax field from 0 to null.

14. Save your credit note line.

15. Create more credit note lines if needed.

16. You can now either estimate tax or apply tax.

- a. To estimate tax, click **Estimate Tax** on your credit note.

Salesforce Billing updates the Tax field and the Total Amount (With Tax) field on each of your credit note lines. You can click **Estimate Tax** again to receive a new estimation.

- b. When you're ready to apply tax, click **Apply Tax** on your credit note.

Salesforce Billing updates the Tax field and the Total Amount (with Tax) field on each of your credit note lines. If your credit note is a draft, Salesforce Billing also posts it.

 **Note** If you're using the standard tax engine, always use Apply Tax button to post the credit note. This ensures accurate tax calculations in Tax and Total Amount (with Tax) fields. Manually changing the credit note status from draft to posted affects tax calculations.

17. After applying tax, click **Allocate** on your credit note to allocate your note to your desired invoice lines.

The creation of a credit note line represents the following journal entry.

#### Create Credit

- Debit Net Sales
- Debit Tax Payable
- Create Credit on Account (COA)

Allocating a credit note line to an invoice line represents the following journal entry.

#### Allocate Credit Note

- Debit COA
- Credit A/R

#### See Also

[Legal Entities](#)

## Create a Credit Note Without Tax

Create a credit note record so you can allocate it at a future date. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Usually, Salesforce Billing automates credit note creation when you provide a refund or rebill an invoice. However, you may sometimes need to create a credit note record on your own. In this scenario, you'll create a credit note, associate it with an account, then create several credit note lines. Your credit note is then ready for posting and allocation.

1. Go to the Credit Notes tab and click **New**, or click **New Credit Note** in your account's Credit Notes related list.  
 **Note** Salesforce Billing doesn't support creating credit notes from the invoice's Credit Notes related list.
2. If your credit note doesn't have a value in the Account field, choose an account for your credit note. You can apply the credit note to any invoice lines on that account.
3. Set the Credit Note Source Action field to Manual.  
The source action allows other users to review how the credit note was created.
4. Provide a credit note date.  
Salesforce Billing records credit note transactions in the finance period that covers the credit note's date.
5. Click **Save**.
6. From your credit note record, click **New Credit Note Line**.

7. Enter a product lookup in the Product field.

The product lookup field is for reference purposes. The relationship is important for reporting on transaction history in the general ledger. This product doesn't have to be the product related to the invoice line you want to apply your credit note toward. For example, if your credit note allocates \$50 to a customer's invoice based on store credit, the Product field should look up to the Store Credit product record.

8. If your implementation uses legal entities, enter a legal entity.

9. Give your credit note line a subtotal.

This value represents the total amount available for allocation on this credit note line. It does not include tax. Since you're not calculating tax for this credit note, Salesforce Billing passes this field's value to the line's Total Amount (With Tax) field when the note is posted.

10. Save your credit note line.

11. Create more credit note lines if needed.

12. When you're ready to allocate your credit note, change its status to Posted.

13. Click **Allocate** on your credit note to allocate your note to your desired invoice lines.

Credit notes are created in draft status and stay in draft status until posted by a user or workflow rule. You can allocate only posted credit notes.

## Guidelines for Credit Note Tax

Salesforce Billing calls your external tax engine to calculate tax for your credit note lines. You can also estimate tax before applying it to the credit note lines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Winter '18 and later

---

By default, Salesforce Billing does not calculate tax for a credit note line when you save it. This standard is useful if you plan on entering a tax amount on your own or you don't want to call a tax engine. To manage credit note line tax, you can click **Apply Tax**, **Estimate Tax**, or **Cancel Tax** on your credit note record.

#### Effective Tax Date

Provide a date for Salesforce Billing to apply tax to this credit note line. Salesforce Billing sends this date to your tax engine during tax calculation. This is a mandatory field.

#### Calculate Tax?

When you estimate or apply tax on your credit note, Salesforce Billing calculates tax for each credit note line with an effective tax date and active Calculate Tax? field. If you don't want to calculate tax for any of your credit note lines, leave their Calculate Tax? fields unselected.

#### Tax Address fields

You can enter values for the street, city, state, country, and postal code of the address you want

associated with tax calculation.

You can estimate or apply tax on a draft credit note. When you estimate or apply tax, Salesforce Billing undergoes the following steps.

1. Salesforce Billing checks each credit note line for a tax address. If the credit note line does not contain a tax address, it uses the parent credit note's address. If the parent credit note does not contain an address, Salesforce Billing does not calculate tax for the line. The tax address on your credit note or credit note line needs only one address field to apply tax.
2. Salesforce Billing evaluates each credit note line where the Calculate Tax? field is selected. It then evaluates whether each of these lines has an effective tax date. If one of these lines does not have an effective tax date, Salesforce Billing stops tax calculation and updates the credit note with a tax error message.
3. Salesforce Billing evaluates whether each credit note line has a tax rule, and a tax treatment with a tax code. If not, Salesforce Billing cancels tax calculation and updates your credit note with a tax error message.
4. For each credit note line, Salesforce Billing passes all tax fields to each of your tax engines. This process lets you calculate tax using more than one tax integration.
5. Salesforce receives tax data from your tax engines.
  - a. If you estimated tax, Salesforce Billing provides a value for the Tax field and updates the Total Amount (With Tax) field for each of your credit note lines. If you click estimate tax again, Salesforce Billing performs the same series of actions and overrides your current tax fields with the new values.
  - b. If you applied tax, Salesforce Billing provides a value for the Tax field and updates the Total Amount (With Tax) field for each of your credit note lines. If your credit note is a draft, Salesforce Billing also posts it.



**Note** Salesforce Billing doesn't currently support the Cancel Tax button. We'll let you know if that changes.

## Allocate a Credit Note

Allocate your credit note to decrease the change the balance of one or more invoice lines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later

---

You can allocate only posted credit notes.

1. Go to your credit note and click **Allocate**.  
Salesforce Billing displays the Credit Note Allocation page. You'll see all of your note's lines that have positive values. You'll also see all the invoice lines in your account where you can apply a credit note line.
2. Select the credit note line you want to allocate, and then select the invoice lines you want to allocate it

toward.

The sum of the invoice line amounts must be less than or equal to the balance of the selected credit note line. You can allocate a credit note line's entire balance or a portion of it.

### 3. Click **Allocate**.

Salesforce Billing allocates your credit note's balance to your invoice lines. The invoice lines update their Balance field and their Credit field to reflect the allocation.

The creation of a credit note line represents the following journal entry.

#### Create Credit

- Debit Net Sales
- Debit Tax Payable
- Create Credit on Account (COA)

Allocating a credit note line to an invoice line represents the following journal entry.

#### Allocate Credit Note

- Debit COA
- Credit A/R

#### Issue Partial Credit

Allocate a negative balance change to an invoice line to adjust errors in unpaid or partially paid invoices. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The invoice you want to allocate the credit to must be in the **Posted** status.

Credit notes allow you to allocate a negative balance change to an invoice line, such as to adjust errors in unpaid or partially paid invoices. You can also provide credit to a user account and then allocate that credit to decrease the balance of their invoices at a later date. Credit notes also provide accurate and complete recordkeeping of the transactions that occur after an invoice has been posted.

 **Note** You can allocate only posted credit notes.

### 1. Create the Credit Note

- From the account record, click **New** from the Credit Note related list to create a credit note.
- Populate the Credit Note Date, then click **Save**.
- From the Credit Notes related list, select the newly created credit note record.
- From the newly created credit note record, navigate to the Credit Note Line related list.
- Click **New** from the Credit Note Line related list to create a New Credit Note Line.

- f. Populate the Product field and enter an amount in the subtotal field, then click **Save**.
2. Allocate the partial credit as a credit note line.
  - a. To move forward with issuing the partial credits, change the Credit Note Status from **Draft** to **Posted**, then click **Save**. Changing the Credit Note Status also changes the Credit Note Line Status.
  - b. From the credit note record click **Allocate**.
  - c. Select the radio button next to the credit note line record that you wish to allocate.
  - d. Select the desired invoice lines. Only posted invoice lines that are associated to the account will show.
  - e. Add an amount of credit less than or equal to the credit note line balance.
  - f. Click **Allocate**.
  - g. Click **Cancel** to close the Credit Note Allocation page and return to the credit note record.

## Correct Errors on Posted Credit Notes

Salesforce Billing must retain a copy of all credit notes for legal and accounting purposes. To fix an error on a posted credit note, create a debit note of equal value and allocate it to each line of the original credit note. The allocation reduces the erroneous credit note's balance to zero. You can then create and post a credit note with the correct values. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Salesforce Billing follows Generally Accepted Accounting Principles. The principles require that users must always be able to track the posting of credits, debits, payments, and refunds through a series of allocations back to a business's general ledger. To ensure that these records are always available for legal and accounting purposes, Salesforce Billing doesn't allow you to delete credit notes, debit notes, payments, or refunds. If you make an error, offset the erroneous record by reducing its balance to zero, and then creating a record with the correct values.

 **Note** You can't change the Bill To Contact value in Posted Credit Notes. To change the Bill to Contact value, create an offsetting debit note and then create a credit note with the desired information.

1. If the credit note has any allocated lines, unallocate them first.
2. Create a debit note with one debit note line for each credit note line of the erroneous credit note.
  - a. From the affected account's Debit Notes related list, click **New Debit Note**.
  - b. Set your debit note date to the current date, and then save your changes.
  - c. Go to the debit note's Debit Note Lines related list, and click **New Debit Note Line**.
  - d. Set the debit note line's product to the same value as the credit note line's product.
  - e. Set the debit note line's subtotal to the same value as the credit note line's balance, and then save your changes.
- We recommend that you add a note explaining that you created this debit note line to offset a credit note line created in error.
- f. Repeat steps A through E for each credit note line of the credit note.

- g. Change your debit note's status to Posted.
3. Allocate each debit note line to the equivalent credit note line.
  - a. From the debit note's detail page, click **Allocate**.
  - b. Select your debit note line, and then select the credit note containing an equal balance.
  - c. On the credit note line, enter an amount equal to the debit note line's balance, and then click **Allocate**. Confirm that the credit note line now has a balance of 0.
  - d. Repeat steps B and C for each of the erroneous credit note's credit note lines.  
After you offset all the credit note lines, the credit note has a balance of 0.
4. To return to your debit note detail page, click **Cancel**.  
Clicking Cancel doesn't cancel any of the allocations that you made.



**Example** In this example, a user must offset an erroneous credit note containing two credit note lines. They created a debit note containing two debit note lines each of which matches the balance of a credit note line. They first allocate debit note line 1 to credit note line 5, and then allocate debit note line 0 to credit note line 4.

## Credit Center

Billing operations and customer service teams can use the Credit Center to quickly credit multiple invoice lines on an invoice. They can credit invoice lines, apply tax per credit note line, and preview the results of the credit note before posting it to the invoice. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '19 and later with the CPQ & Billing Plus license

---

After a billing ops user clicks **Credit Center** on their account, Salesforce Billing launches the Credit Center and shows all the posted invoices with positive or negative outstanding balances on their account. When they select an invoice, the Credit Center lets them credit any of its invoice lines. They can also calculate credit note line tax with the tax engine used for the invoice line, calculate credit note line tax manually, or choose not to calculate credit note line tax.



When they're done setting up the credit note, Salesforce Billing creates a draft credit note record with a credit note line for each invoice line that they credited. Users can review the new invoice line balances while the note is still in draft status and make further changes if needed, or post the credit note to apply the credits.

#### [Use the Credit Center to Apply Invoice Line Credits](#)

Open the Credit Center from your account and apply a partial credit to an invoice. (Salesforce Billing Managed Package)

#### [Guidelines for Using the Credit Center](#)

If you're on a billing operations or customer service team, you can use the Credit Center to credit invoice lines on a customer account. When you're working in the Credit Center, take note of a few key

guidelines. (Salesforce Billing Managed Package)

## Use the Credit Center to Apply Invoice Line Credits

Open the Credit Center from your account and apply a partial credit to an invoice. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '19 and later with the CPQ & Billing Plus license

---

Make sure that the invoice that you want to credit is posted with a non

1. Do either of the following.
  - a. If you're using Lightning Experience, from your account, click the dropdown arrow, and then click **Credit Center**.
  - b. In Salesforce Classic, from your account, click **Credit Center**.
2. The Credit Center page has Credit an Invoice selected. Click **Next**, select an invoice, and then click **Next**.
3. Choose the invoice lines you want to credit by selecting the checkbox next to each product name. By default, all invoice lines are selected.
4. Set a credit amount for an invoice line by clicking the pencil icon next to that line's credit amount and entering a number.
5. If you're taxing your credit note, you can manually add tax or calculate tax based on your tax integration.
  - a. To manually add tax, click the pencil icon next to your line's tax amount and enter a number.
  - b. To calculate tax based on your tax integration, select **Calculate Tax?** for your invoice line. Salesforce Billing calculates tax and locks the tax amount from edits. To manually add tax, deselect **Calculate Tax** and enter a value for the line's tax amount.The effective tax date for an invoice line defaults to the invoice line's start date.
6. Click **Next**.  
Salesforce Billing creates a credit note in Draft status. The credit note contains one credit note line for each invoice line that you credited. The credit note's date matches the day you created it.
7. Review your credit amounts before posting your credit note.
  - a. To return to the Edit Credit Notes page and make further changes, click **Back**.
  - b. To post your credit note to your invoice, click **Post**.
  - c. To save the credit note as a draft, click **Cancel**.

Salesforce Billing creates a draft credit note the first time that you click **Next** on Edit credit notes. This credit note's Source Action has a value of Invoice Line Credit. When you open an invoice in the Credit Center, Salesforce Billing evaluates whether the invoice has a draft credit note with a source action of Invoice Line Credit. If it does, the Credit Center loads that credit note for further edits.

## Guidelines for Using the Credit Center

If you're on a billing operations or customer service team, you can use the Credit Center to credit invoice lines on a customer account. When you're working in the Credit Center, take note of a few key guidelines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '19 and later with the CPQ & Billing Plus license

---

- The Credit Center shows up to 1,500 posted invoices at a time, starting with the oldest due date.
- The Credit Center supports invoices with up to 1500 invoice lines. If you need to credit lines on a larger invoice, we recommend doing so manually.
- Credit notes created in the Credit Center have a start date equal to their creation date. You can't edit credit note dates in the Credit Center, but you can change them on the credit note record.
- Salesforce Billing creates a draft credit note the first time that you click **Next** on Edit Credit Notes. This credit note's Source Action has a value of Invoice Line Credit. When you open an invoice in the Credit Center, Salesforce Billing evaluates whether the invoice has a draft credit note with a source action of Invoice Line Credit. If it does, the Credit Center loads that credit note for further edits. This way, you can edit an invoice's credit note any time before posting it.

## Creating Debits and Add-on Charges

Debit notes allow you to allocate a positive balance change to an invoice line or positive credit note line. This feature is useful if you must change a line's balance after invoice generation, such as applying a late fee. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '17 and later

---

Debit note records contain a related list for their debit note lines. A positive debit note line allocation increases the balance of an invoice line or positive credit note line. A negative debit note line allocation decreases the balance of an invoice line or a positive credit note line. Consider these examples.

- Allocating a \$100 debit note line to a \$500 invoice line changes the invoice line's balance to \$600.
- Allocating a -\$100 debit note line to a -\$500 invoice line changes the invoice line's balance to -\$600.

You have several options for how you can allocate your debit note lines. These options vary based on whether the debit note and the target line have positive or negative balances.

Debit Note Line	You Can Allocate it Toward These Records	Notes
Positive Balance	<ul style="list-style-type: none"> <li>Positive invoice line</li> <li>Negative invoice line</li> <li>Positive credit note line</li> </ul>	None
Negative Balance	<ul style="list-style-type: none"> <li>Positive invoice line</li> <li>Negative invoice line</li> <li>Negative credit note line</li> </ul>	You can allocate to the positive invoice line only if the allocation doesn't exceed the invoice line's balance.

You can also allocate payments to debit note lines that you allocated to a positive invoice line. When you perform this allocation, your debit note line's targeted invoice line updates its Debit Payment field to reflect the allocated payment amount. You may allocate payments to debit note lines only if the debit note line has a positive balance in its Available for Payment field.

You can create a debit note through these actions.

- Create a debit record from Salesforce Billing's Debit Notes tab. You can apply the debit note lines to any eligible lines in your org.
- Create a debit note record from your account's Debit Notes related list. You can apply the debit note lines to any eligible lines in the account.

Your debit note lines contain an optional lookup to a legal entity. Provide a value to this lookup if you want to associate your debit note line to a revenue schedule or GL account for bookkeeping and revenue tracking. You can also associate your debit note line with a product record or product name. These lookups let you quickly reference the product covered by the invoice line that you're allocating toward.

Debit notes have a default Status field set to Draft. Change this field to Posted when you're ready to allocate your debit note lines.

-  **Note** If you have single currency in Salesforce, the organization-wide currency is the debit note's currency. If you enable multi-currency, the currency of the debit note and debit note line must match the currency of the record that you want to allocate the debit note and debit note line to.

### Apply Debit Notes to Invoices

Create a debit note record and several debit note lines. Allocate your debit note lines to increase the balance of an invoice line. (Salesforce Billing Managed Package)

### Managing Tax on Debit Notes

Salesforce Billing calls your external tax engine to calculate tax for your debit note lines. First, estimate taxes on your debit notes. After you estimate the taxes, you can apply them to your debit notes. (Salesforce Billing Managed Package)

### Correct Errors on Posted Debit Notes

Salesforce Billing must retain a copy of all debit notes for legal and accounting purposes. To fix an error on a posted debit note, create a credit note of equal value and allocate it to each of the original debit note's lines. The allocation reduces the erroneous debit note's balance to zero. You can then create and post a debit note with the correct values. (Salesforce Billing Managed Package)

### Apply Partial Debits

Track and allocate additional charges with a partial debit note. (Salesforce Billing Managed Package)

## Apply Debit Notes to Invoices

Create a debit note record and several debit note lines. Allocate your debit note lines to increase the balance of an invoice line. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing 8.0 and later

---

In this example, your customer missed one of their monthly payments to your data security subscription service. Your invoice scheduler already created an invoice from the monthly charge. So, you need a debit note to apply the late fee of US\$50 to the subscription's invoice line.

1. On the account record, on the Debit Notes related list, click **New**.  
The debit note's Status field value is automatically set as Draft and the Debit Note Source Action field value is automatically set as Manual.
2. Enter a debit note date.
3. If you enabled multi-currency, enter the desired currency.
4. Save your changes.
5. On the Debit Notes related list, select the new debit note record.
6. On the Debit Note Lines related list on the new debit note, click **New**.
7. Select a product.  
This product can be the same as the product on your invoice line or any other product [configured for Salesforce Billing](#).
8. Enter *50* as the subtotal.
9. If necessary, select a legal entity.
10. Save your changes.  
You can also enter tax details. See [Managing Tax on Debit Notes](#).
11. Open the new debit note and change the status to Posted.
12. Save your changes.
13. Click **Allocations**.  
The Debit Note Allocation page shows your debit note and all its debit note lines that have positive non-zero balances. You can also see your account's invoice lines where you can apply your debit note line.
14. Select your \$50 debit note line, and then select the invoice line for this month's data security subscription. In the Amount field on the invoice line, enter *50*.
15. Click **Allocate**.

Salesforce Billing adds \$50 to your data security subscription invoice line and updates the invoice line's debit field to a value of 50.

Your invoice line has a related list with lookups to all the debit notes that you applied against it. This way, you can apply a payment to one of the invoice line's debit notes before you apply payments to the invoice line itself. If you run a payment scheduler with an auto-payment method on your account, Salesforce Billing pays the invoice line's entire remaining balance, including debit notes.

## Managing Tax on Debit Notes

Salesforce Billing calls your external tax engine to calculate tax for your debit note lines. First, estimate taxes on your debit notes. After you estimate the taxes, you can apply them to your debit notes. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '18 and later.

By default, Salesforce Billing does not calculate tax for a debit note line when you save it. This standard is useful if you plan on entering a tax amount on your own or you don't want to call a tax engine. To manage debit note line tax, you can apply, estimate, or cancel tax by clicking the appropriate button on the debit note record. Before you choose a tax action, review the following debit note line tax fields.

#### Effective Tax Date

Provide a date for when Salesforce Billing should apply tax to this debit note line. Salesforce Billing sends this date to your tax engine during tax calculation.

#### Calculate Tax?

When you estimate or apply tax on your debit note, Salesforce Billing evaluates whether Calculate Tax? is selected on each debit note line. It then sends a call to calculate tax for each debit note line with an effective tax date and

#### Tax Address fields

You can enter values for the street, city, state, country, and postal code of the address you want associated with tax calculation.

First, estimate the tax on a draft debit note using the **Estimate Tax** action. After you estimate the taxes, post the debit note with its estimated tax using the **Apply Tax** action. When you estimate or apply tax, Salesforce Billing undergoes the following steps.

-  **Note** Salesforce Billing doesn't currently support the Cancel Tax button. We'll let you know if that changes.

1. Salesforce Billing checks each debit note line for a tax address. If the debit note line does not contain

a tax address, it uses the parent debit note's address. If the parent debit note does not contain an address, Salesforce Billing does not calculate tax for the line. The tax address on your debit note or debit note line needs only one address field to apply tax.

2. Salesforce Billing evaluates each debit note line where the Calculate Tax? field is selected. It then evaluates whether each of these lines has an effective tax date. If one of these lines does not have an effective tax date, Salesforce Billing stops tax calculation and updates the debit note with a tax error message.
3. Salesforce Billing evaluates whether each debit note line has a tax code and tax treatment. If not, it cancels tax calculation and updates your debit note with a tax error message.
4. For each debit note line, Salesforce Billing passes all tax fields to each of your tax engines. This process lets you calculate tax using more than one tax integration.
5. Salesforce receives tax data from your tax engines.
  - a. If you estimated tax, Salesforce Billing provides a value for the Tax field and updates the Total Amount (With Tax) field for each of your invoice lines. If you click estimate tax again, Salesforce Billing performs the same series of actions and overrides your current tax fields with the new values.
  - b. If you applied tax, Salesforce Billing provides a value for the Tax field and updates the Total Amount (With Tax) field for each of your invoice lines. If your debit note is a draft, Salesforce Billing also posts it.

## Correct Errors on Posted Debit Notes

Salesforce Billing must retain a copy of all debit notes for legal and accounting purposes. To fix an error on a posted debit note, create a credit note of equal value and allocate it to each of the original debit note's lines. The allocation reduces the erroneous debit note's balance to zero. You can then create and post a debit note with the correct values. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Salesforce Billing follows Generally Accepted Accounting Principles. The principles require that users must always be able to track the posting of credits, debits, payments, and refunds through a series of allocations back to a business's general ledger. To ensure that these records are always available for legal and accounting purposes, Salesforce Billing doesn't allow you to delete credit notes, debit notes, payments, or refunds. If you make an error, offset the erroneous record by reducing its balance to zero, and then creating a record with the correct values.

1. If the debit note has any allocated lines, unallocate them first.
2. Create a credit note with one credit note line for each of the erroneous debit note's debit note lines.
  - a. From the affected account's Credit Notes related list, click **New Credit Note**.
  - b. Set your credit note date to the current date, then save your changes.
  - c. Go to the credit note's Credit Note Lines related list and click **New Credit Note Line**.
  - d. Set the credit note line's product to the same value as the debit note line's product.
  - e. Set the credit note line's subtotal to the same value as the debit note line's balance, then save your changes.

We also recommend adding a note explaining that you created this credit note line to offset a debit note line created in error.

- f. Repeat Steps A through E for each of the debit note's debit note lines.
  - g. Change your credit note's status to **Posted**.
3. Go to the debit note that you want to cancel and click **Allocate**.
4. Allocate each of your credit note lines to the equivalent debit note line.
- a. Select the debit note line that you want to cancel.
  - b. Select the credit note line that you created with an equivalent balance.
  - c. On the credit note line, enter an amount equal to the debit note line's balance, then click **Allocate**.
  - d. Repeat steps A through C for each of the erroneous debit note's debit note lines.
- When you finish offsetting all the debit note lines, the debit note has a balance of 0.
5. To return to your credit note detail page, click **Cancel**.
- Clicking Cancel doesn't cancel any of the allocations that you made.



**Example** In this example, a user must offset an erroneous debit note containing two debit note lines. They have created credit note CN-003, which contains two credit note lines that each match the balance of the debit note lines. They will first allocate credit note line 7 to debit note line 3, and then allocate credit note line 6 to debit note line 2.

## Apply Partial Debits

Track and allocate additional charges with a partial debit note. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. Create a debit note.
  - a. On the account record, on the Debit Notes related list, click **New**.  
The debit note's Status field value is automatically set as Draft and the Debit Note Source Action field value is automatically set as Manual.
  - b. Enter a debit note date.
  - c. If you enabled multi-currency, enter the desired currency.
  - d. Save your changes.
  - e. On the Debit Notes related list, select the new debit note record.
  - f. On the Debit Note Lines related list on the new debit note, click **New**.
  - g. Select a product.  
This product can be the same as the product on your invoice line or any other product [configured for Salesforce Billing](#)
  - h. Enter the subtotal amount.
  - i. If you enabled multi-currency, enter the debit note's currency.
  - j. Save your changes.
2. Allocate the partial debit as a debit note line.
  - a. Change the debit note status from Draft to Posted.

- b. Save your changes.
- c. From the Debit Note record, click **Allocate**.
- d. Select the Debit Note Line record that you want to allocate.  
Debit notes allocated to invoice lines increase the invoice line balance. Debit notes allocated to credit note lines lower the credit note line balance.
- e. Select the invoice lines and credit note lines.
- f. Enter an amount less than or equal to the debit note line's balance in the Amount field.
- g. Click **Allocate**.
- h. To close the Debit Note Allocation page and return to the debit note record, click **Cancel**.

## Applying Refunds

Refund customers accurately and efficiently with Salesforce Billing. You can provide a refund when a customer wants to cancel or change their invoiced order products. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Refunds fall into two categories.

- Repayment: Provide a complete refund to a customer who isn't satisfied with the product or service they received, or who changed their mind after paying.
- Amendment: Update the invoice when a customer upgrades or downgrades a subscription or service they already paid for. Switching to a less expensive service usually results in a refund of the original amount. Switching to a more expensive service creates a invoice for the entire new service and issues a refund for the original amount.

Refunds differ from credits because they reimburse an amount that was already paid, while credits are applied toward future charges.

#### [Issue Electronic Refunds in Lightning Experience](#)

Refund a payment using one of the payment gateways you've linked to Salesforce Billing. (Salesforce Billing Managed Package)

#### [Record an External Payment Refund](#)

Record a payment refund that was made in an external source. Salesforce Billing saves the information as a Refund Line (Payment) record on your payment without passing any information to a payment gateway. (Salesforce Billing Managed Package)

#### [Refund a Credit Note Line from an Account](#)

Refund a partially-allocated credit note line. (Salesforce Billing Managed Package)

#### [Void Electronic Refunds](#)

Void an electronic refund. (Salesforce Billing Managed Package)

#### [Correcting Errors on Posted Refunds](#)

Salesforce Billing must retain a copy of all refunds for legal and accounting purposes. To fix an error on

a posted refund, you can create a credit note or payment of equal value and allocate it against the invoice lines affected by the erroneous refund. You can then create and post a refund with the correct values. (Salesforce Billing Managed Package)

## Issue Electronic Refunds in Lightning Experience

Refund a payment using one of the payment gateways you've linked to Salesforce Billing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

-  **Note** To issue refunds successfully when you're using migrated payment method that has a payment gateway token, make sure that you specify the [payment method's card number, card expiration month, card expiration year, and card type](#).

1. Go to your payment and click **Refund**.
2. Click **Electronic**.
3. Enter the amount that you want to refund. By default, the Issue Refund page shows the payment's entire unallocated amount.
4. Optionally provide a legal entity, GL rule, GL treatment, and billing finance book for recording the refund transaction.
5. Click **Refund**.

Salesforce Billing calls the gateway associated with this payment's account.

If the gateway's response is successful, Salesforce Billing creates a refund and refund line for the payment. The refund line has the amount you chose in Step 3. Salesforce Billing then posts the refund.

If the gateway's response isn't successful, the Issue Refund page shows an error message. You can cancel from the page, or make changes and attempt the refund again.

-  **Important** Refunds can't be issued on transactions that aren't settled. Check your payment gateway portal for payment status. Unsettled transactions are voided in some situations. See [Void Electronic Refunds](#). Typically, refunds aren't processed on transactions older than 120 days. Check with your payment gateway for the specific time limit.

## Record an External Payment Refund

Record a payment refund that was made in an external source. Salesforce Billing saves the information as a Refund Line (Payment) record on your payment without passing any information to a payment gateway. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Lightning Experience in Salesforce Billing Winter '19 and later

1. Go to your payment record and click **Refund**.
2. Click **External**.
3. Enter the amount you want to refund. The refund must be less than or equal to the unallocated payment amount.
4. Enter the date for your external refund. By default, the Issue Refund page shows the current date, but you can change that based on when your external refund actually happened. Remember that changing the date can change the finance period where Salesforce Billing records the refund transaction.
5. The legal entity, gl rule, gl treatment, and billing finance book fields are optional based on whether and how your customer records their transactions in an external general ledger. Review your customer's billing implementation requirements and provide these fields as needed.
6. Choose your refund type. If you choose Check, enter the check number as well.
7. Click **Refund**.  
Salesforce creates a Refund Line (Payment) record on the payment.

## Refund a Credit Note Line from an Account

Refund a partially-allocated credit note line. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Summer '17 and later

1. Go to your account and click **Refund**.
2. Choose a currency.  
Salesforce Billing does not display available credit notes until you pick your currency.  
Salesforce Billing displays a list of partially-allocated credit note lines under the Refund Credit Note Line header. The Balance field for each line represents the credit that has not been allocated.
3. Select the credit note lines that you want to refund and provide a value for their Amount fields. This value can be all or part of that line's balance.
4. Provide a description and date for your refund.
5. Click **Allocate**.  
Salesforce Billing creates a refund record for each line you refunded. The refund record has an amount equal to the value you entered for the line in step 3.
6. You can view your refunded credit note lines under the Refund page's Existing Refund Credit Note Line.

## Void Electronic Refunds

Void an electronic refund. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Winter '19 and later

1. Log into the payment gateway and confirm that the refund transaction's Transaction Status is in the "Refund/Pending Settlement" status. If a transaction is settled and requires the reversal of the refund, charge the customer again using the original payment method, instead of voiding the refund.
2. Go to Salesforce and find the invoice record that you want to adjust. The invoice status must be Posted and the invoice payment status must be Unpaid.
3. From the invoice related list, go to the payment record that you unallocated when you started the refund process.
4. From the payment record dropdown menu, click **Allocations**
5. In the Payment Allocation UI under the invoice lines section, click **Corresponding Invoice Record**.
6. In the amount field, enter the amount you wish to reallocate. Make sure to include a currency sign in your amount.
7. Click **Allocate**.  
Salesforce Billing shows a message letting you know that the payment was successfully allocated. The page reloads and the invoice that you allocated remains under the Invoice Lines section.
8. Click **Cancel** to close out of the payment allocation UI page.
9. Verify that the payment status on the invoice record shows is Unpaid and the invoice status is Posted.
10. Go to the Account record dropdown menu arrow and click **Refund**.
11. In the Refund UI, under the Existing Refund Payments section, click **Unallocate** next to the corresponding refund record that you need to unallocate.  
The recently unallocated refund record is removed from the Existing Refund Payments section and is now listed under the Refund Payments section.
12. Click **Cancel** to close out of the Refund page and to go back to your account.
13. Log in to your payment gateway.
14. Void the transaction that is pending settlement. Remember, the Salesforce payment response gateway ID could be the same as the payment gateway transaction ID.  
At this stage, you've voided the refund and your values are lined up with Salesforce Billing.
15. In Salesforce Billing, go back to the payment record. The payment record's status shows as Posted.
16. From the payment's dropdown menu, click **Allocations**.
17. In the Payment Allocation UI under the Invoice Lines section, click the checkbox under the Select column to select the corresponding invoice record that you need to allocate toward.
18. Choose the amount you want to allocate and then click **Allocate**.
  - The corresponding allocated invoice record is removed from the Invoice Lines section. The Existing Invoice Line Allocations section appears and the payment associated with your corresponding allocated invoice is listed under this section.
  - To confirm you've successfully backed out the refund, check the status on the invoice record. The invoice status should now show Posted and the invoice payment status should now show Paid.

## Correcting Errors on Posted Refunds

Salesforce Billing must retain a copy of all refunds for legal and accounting purposes. To fix an error on a

posted refund, you can create a credit note or payment of equal value and allocate it against the invoice lines affected by the erroneous refund. You can then create and post a refund with the correct values. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

### **Use Credit Notes to Correct Errors on Posted Refunds**

Salesforce Billing must retain a copy of all refunds for legal and accounting purposes. If you're using a non-payment source to offset an erroneous refund, create a credit note in Salesforce Billing to offset the refund's impact on your invoice lines. (Salesforce Billing Managed Package)

### **Use Payments to Correct Errors on Posted Refunds**

Salesforce Billing must retain a copy of all refunds for legal and accounting purposes. If you're using cash or an electronic payment to offset an erroneous refund, create a payment in Salesforce Billing to offset the refund's impact on your invoice lines. (Salesforce Billing Managed Package)

## Use Credit Notes to Correct Errors on Posted Refunds

Salesforce Billing must retain a copy of all refunds for legal and accounting purposes. If you're using a non-payment source to offset an erroneous refund, create a credit note in Salesforce Billing to offset the refund's impact on your invoice lines. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Salesforce Billing follows Generally Accepted Accounting Principles. The principles require that users must always be able to track the posting of credits, debits, payments, and refunds through a series of allocations back to a business's general ledger. To ensure that these records are always available for legal and accounting purposes, Salesforce Billing doesn't allow you to delete credit notes, debit notes, payments, or refunds. If you make an error, offset the erroneous record by reducing its balance to zero, and then creating a record with the correct values.

This topic explains how to offset only a posted refund. If your refund is pending settlement, you can void it instead.

1. Create a credit note with one line for each invoice line affected by the erroneous refund.
  - a. From the affected account's Credit Notes related list, click **New Credit Note**.
  - b. Set your credit note date to the current date, then save your changes.
  - c. Go to the credit note's Credit Note Lines related list and click **New Credit Note Line**.
  - d. Set the credit note line's Product field to the same product as the invoice line affected by the erroneous refund.
  - e. Set the credit note line's subtotal to the same value as the invoice line's balance, and then save your changes.

For example, you allocated an original payment of \$500 against a Mobile Device invoice line and a Workstation invoice line on the invoice INV-006. You then created the erroneous refund for \$300 and allocated \$200 against the Mobile Device line and \$100 against the workstation line. In this case, you would create a credit note line with a balance of \$200 and set its product to your Mobile Device invoice line. You would then create another credit note line with a balance of \$100 and set its product to your Workstation invoice line.

We recommend adding a note explaining that you created this credit note line to offset a refund created in error.

- f. Repeat Steps A through E for each invoice line affected by the erroneous refund.
  2. Change your credit note's status to **Posted**.
  3. Allocate each of your credit note lines to the affected invoice lines.
    - a. Click **Allocate**.
    - b. Select a credit note line, and then select the invoice line that you want to offset.
    - c. On the credit note line, enter an amount equal to the invoice line's balance, and then click **Allocate**. Confirm that the credit note line now has a balance of 0.
    - d. Repeat steps B and C for each affected invoice line.

When you finish offsetting all the credit note lines, the credit note has a balance of 0.
  4. To return to your credit note detail page, click **Cancel**.
- Clicking Cancel doesn't cancel any of the allocations that you made.
- Your credit note has completely offset the value of your erroneous refund. The refund didn't change, but its allocations to your invoice lines have been offset by the credit note. Your invoice balance now equals the balance before the refund was applied. You can now issue a new refund with the correct values.

## Use Payments to Correct Errors on Posted Refunds

Salesforce Billing must retain a copy of all refunds for legal and accounting purposes. If you're using cash or an electronic payment to offset an erroneous refund, create a payment in Salesforce Billing to offset the refund's impact on your invoice lines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Salesforce Billing follows Generally Accepted Accounting Principles. The principles require that users must always be able to track the posting of credits, debits, payments, and refunds through a series of allocations back to a business's general ledger. To ensure that these records are always available for legal and accounting purposes, Salesforce Billing doesn't allow you to delete credit notes, debit notes, payments, or refunds. If you make an error, offset the erroneous record by reducing its balance to zero, and then creating a record with the correct values.

This topic explains how to offset only a posted refund. If your refund is pending settlement, you can void it instead.

1. Create a payment with a balance equal to the balance of the erroneous refund.

- a. From the affected account's Payments related list, click **New**.
  - b. Set your payment date to the current date, then set the amount to the amount of your refund.
  - c. Set the Invoice field to the invoice that was impacted by the erroneous refund, and then save your changes.  
For example, you allocated an original payment of \$500 against INV-006, and then applied a refund of \$300 to that payment. Your new offsetting payment would have a balance of \$300 and look up to INV-006.
  - d. On your payment, click **Allocations**.
2. Allocate the payment against the affected invoice lines.
- a. From your payment, click **Allocate**.
  - b. Allocate the payment to your affected invoice lines.  
For example, you applied \$200 of your erroneous refund to INV-006's Mobile Device invoice line, and \$100 to INV-006's Workstation invoice line. In this case, you would allocate \$200 of your offsetting payment to the Mobile Device line, and \$100 to the Workstation line.
3. To return to your payment detail page, click **Cancel**.  
Clicking Cancel doesn't cancel any of the allocations that you made.  
Your payment has completely offset the balance of your erroneous refund. The refund didn't change, but its allocations to your invoice lines have been offset by the new payment. Your invoice balance now equals the balance before the refund was applied. You can now issue a new refund with the correct values.

## Payment Data Model

When you're working with payment objects in Salesforce Billing, review important object relationships. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---



## Understanding the Revenue Recognition Process

---

Salesforce Billing uses several objects to manage the revenue recognition system. Once you define these objects and their relationships, you can automate most of the revenue recognition process. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The revenue transaction record stores information about transactions that a customer performed on one

of your order products and how much revenue those transactions created. The revenue schedule defines the period for which Salesforce Billing makes revenue transaction records for an order product. Your goal is to set up Salesforce Billing's revenue recognition objects so that your org creates revenue transactions for your desired time periods. You also define the actions that prompt Salesforce Billing to create those transactions in the first place.

Revenue recognition rules are the first object in this process. Your rule's Create Revenue Schedule? Field defines whether Salesforce Billing makes a revenue schedule for an order product. Set this field to Yes and assign your rule to any products that should have revenue transaction records once you invoice them or order. When you order those products, their order product records inherit the lookup to the revenue recognition rule.

Your revenue recognition rule has a related list for revenue recognition treatments. A treatment answers the following questions to tell Salesforce Billing how and where to recognize the product's revenue.

- Which GL rule makes a record of the revenue transaction?
- How much of a transaction's amount should be assigned to a GL rule?
- How much revenue from a transaction is assigned to a revenue schedule?
- What action causes Salesforce Billing to create a revenue schedule?

A revenue recognition rule can have several treatments. When you invoice an order, Salesforce Billing groups for evaluation order products that have the same revenue recognition rule. It then applies revenue recognition treatments based on matching legal entities between the order product and one of the rule's treatments. This process lets you apply different treatments to order products that a single rule is evaluating. It also helps you organize your rules based on the types of treatments they apply. For example, you could have one rule that makes revenue transaction records for all one-time products you sell. You could then have another rule that makes revenue transaction records for each subscription product you sell.

The revenue treatment object contains a lookup to a revenue distribution method. A revenue distribution method controls how Salesforce Billing spreads revenue over a set time period. Salesforce Billing spreads revenue into revenue transactions, grouped within a revenue schedule. Your method answers the following questions.

- When does the revenue schedule start and end?
- Should Salesforce Billing make revenue transactions automatically, or should it make only the revenue schedule and let users create transaction records manually?
- What action causes Salesforce to create a revenue transaction?
- What types of time periods should each of the schedule's transactions cover? For example, a transaction record could cover the revenue from all transactions against an object during one month. Then if your schedule lasts for five months, it contains five revenue transaction records.

The revenue transaction serves as the endpoint of your revenue recognition process. This record stores the following data.

- How much revenue you have previously recognized for this transaction
- How much revenue you have recognized on this record

- How much deferred revenue you have yet to recognize

### The Revenue Waterfall

The revenue waterfall is the distribution of revenue across one or more periods. This process represents revenue recognition as goods or services are delivered to your customers. It's a key component of revenue reporting and forecasting. (Salesforce Billing Managed Package)

### Revenue Distribution Methods

Your revenue distribution method controls how Salesforce Billing spreads revenue over a set time period. You can set up revenue to distribute over time or at a single point in time. (Salesforce Billing Managed Package)

### Organizing Your Revenue Transactions in a Revenue Schedule

A revenue schedule stores records of revenue transactions you've made against an order product. Use this object to track the total amount of revenue you've recognized within the timeframe set by your revenue recognition rule. You can also split revenue from one order product into multiple revenue schedules based on your recordkeeping needs. (Salesforce Billing Managed Package)

### Revenue Recognition Policies

Revenue recognition is an accounting principle used to determine when and how revenue is recognized or accounted for. Your business can apply different methods and policies when deciding how to recognize revenue. The type of business determines which policy to apply. (Salesforce Billing Managed Package)

### Revenue Agreements

Revenue agreements are objects with a Revenue Schedule related list. They're useful for grouping revenue schedules related to the same transaction, so that you can quickly evaluate revenue amounts for that transaction. You can use default or custom functionality to assign revenue schedules to a revenue agreement. Each revenue agreement contains several revenue amount fields that you can customize to report on revenue from the related revenue schedules based on your organization's revenue reporting standards. (Salesforce Billing Managed Package)

### Revenue Adjustments

When your actual revenue differs from your initial revenue schedule, you may need to increase or decrease your revenue schedule's available balance, or transfer it to the available balance on a different schedule. You can use revenue adjustments to change your revenue schedule's available balance. (Salesforce Billing Managed Package)

## The Revenue Waterfall

The revenue waterfall is the distribution of revenue across one or more periods. This process represents revenue recognition as goods or services are delivered to your customers. It's a key component of revenue reporting and forecasting. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Salesforce Billing stores revenue waterfall information in revenue schedules and revenue transactions:

- Revenue schedule: Salesforce Billing creates a revenue schedule record based on the fields defined in your revenue distribution method. The revenue schedule represents the total amount of revenue that comes from the delivery of a good or service. A revenue schedule record contains the following key fields.
  - The total amount of revenue to be recognized
  - The current period balance of deferred revenue
  - The total recognized revenue through the end of the current billing period
- Revenue transaction: Each revenue transaction record represents revenue distribution for a single billing period. This object contains information about revenue to be recognized in the current period, the deferred revenue balance at the end of the period, and previously recognized revenue. This is the primary object used for posting recognized revenue to your general ledger.



**Example** A customer purchases access to a video streaming service for the period of January 1 through March 31. The customer pays \$120 up-front for the three-month period, which allows unlimited access to the service within those three months. In this case, you should already have a revenue distribution method with a Distribution Method field set to Monthly. When you invoice this purchase, Salesforce Billing creates a revenue schedule for \$120 and 3 related revenue transaction records.

#### Revenue Transaction 1

- Period: January 1 through January 31
- Recognized Amount: \$40
- Deferred Balance: \$80
- Previously Recognized Revenue: \$0

#### Revenue Transaction 2

- Period: February 1 through February 28
- Recognized Amount: \$80
- Deferred Balance: \$40
- Previously Recognized Revenue: \$40

#### Revenue Transaction 3

- Period: March 1 through March 31
- Recognized Amount: \$120
- Deferred Balance: \$0
- Previously Recognized Revenue: \$80

## Revenue Distribution Methods

Your revenue distribution method controls how Salesforce Billing spreads revenue over a set time period. You can set up revenue to distribute over time or at a single point in time. (Salesforce Billing Managed

Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing editions

---

Your revenue distribution method defines how Salesforce Billing creates revenue schedules and the term for each schedule. A revenue recognition treatment contains a lookup to a distribution method. This structure allows you to associate several treatments, and thus several distribution methods to one order product. For example, you could have one treatment associate your revenue schedule with a standard finance book and another treatment associate your method with a deferred finance book. The schedule and transactions you create has lookups to different finance books. You could also create a distribution method for recognizing revenue based on order product term and another distribution method for tracking revenue based on invoice line term.

We'll review some key distribution method fields here and then take a detailed look at the layout and use cases for several types of revenue schedules.

### Monthly Recognition Proration

If your revenue transaction came from a partial period (for example, 15 days in a monthly distribution method), Salesforce Billing prorates the transaction's amount based on this value. Currently, you can prorate based only on the number of days in the partial period.

### Type

Choose the action that causes Salesforce Billing to create a revenue transaction. If you select Order, Salesforce Billing creates the revenue transaction when an order product activates. If you select Invoice, Salesforce Billing creates the revenue transaction when an invoice posts.

### Revenue Transaction Creation Process

Choose whether Salesforce Billing automatically creates a revenue schedule and transaction in response to the value you defined in the Type field. If you want to create a revenue schedule on your own, set this field's value to Manual.

### Distribution Method

Salesforce Billing prorates Revenue Amount pricing on revenue transactions based on the Distribution Method field.

- Monthly: Prorate the revenue amount based on the number of days in the month. For example, if your order product bills at \$80 for an initial period of 01/10/20 through 01/31/20, your revenue amount will be  $(22/31) * 80 = \$56.77$ .
- Full Recognition: Salesforce Billing fully recognizes all revenue for the order product or invoice line within the revenue transaction.
- Manual: When you create a revenue schedule, Salesforce Billing doesn't run the batch process to create revenue transactions and assign them to the schedule. Instead, you must create the revenue transactions and assign them to the correct revenue schedule. You can use workflow rules and process builders to automate the manual revenue transaction creation process.

### **Revenue Schedule Term Start Date and Revenue Schedule Term End Date**

These fields define when the revenue schedule starts and ends. You can pull start and end dates from the order product, invoice line, credit note, or debit note. If you choose Manual, Salesforce Billing references user-defined code to determine revenue distribution.

### **Full Recognition Date**

This field defines when Salesforce Billing recognizes revenue in full. Give this field a value only if you set your distribution method to Full Recognition.

## Organizing Your Revenue Transactions in a Revenue Schedule

A revenue schedule stores records of revenue transactions you've made against an order product. Use this object to track the total amount of revenue you've recognized within the timeframe set by your revenue recognition rule. You can also split revenue from one order product into multiple revenue schedules based on your recordkeeping needs. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Salesforce billing automatically creates revenue schedules for a transaction when the revenue recognition rules evaluating that transaction have a Create Revenue Schedule? field set to Yes. The treatment or treatments determine the revenue distribution method for your transaction. Your method also defines whether Salesforce Billing creates revenue transaction records automatically or manually. We recommend automating the transaction creation process unless you use a revenue recognition model that Salesforce Billing doesn't support.

Your revenue schedule defines its term based on your method's Revenue Schedule Term Start Date field and Revenue Schedule Term End Date field. The revenue schedule contains a number of revenue transaction records based on your schedule's revenue start date, revenue end date, and your distribution method. Here are few examples of revenue transaction creation. The first two examples display revenue recognized ratably over a period of time. The third example displays revenue recognized all at once.

Revenue Start Date	Revenue End Date	Distribution Method	Number of Revenue Transactions
01/01/2018	12/31/2018	Monthly	12
01/01/2018	12/31/2018	Daily	12
01/01/2018	12/31/2018	Full Recognition	1

Let's review a few ways to structure your revenue schedules based on your revenue recognition needs. In the first example, Salesforce Billing creates one revenue schedule for an entire service period. However, you can also create revenue schedules that run in parallel by associating several revenue recognition

treatments with one order product. This is useful if you want to record revenue based on different terms or record it into multiple finance books.

In these examples, you're recognizing revenue for a one-time order product with the following values. Your revenue distribution method defines the revenue terms by the order product start date and end date.

- Subscription Term: 12
- Start Date: 01/09/2018
- End Date: 01/09/2019
- Charge Type: One-Time

Revenue Schedule Example	Revenue Distribution Method Setup	Resulting Revenue Schedule
I want a revenue schedule that tracks revenue distributed ratably by month for my order product's entire service period.	<ul style="list-style-type: none"> <li>• Revenue Transaction Creation Process: Automatic</li> <li>• Type: Order</li> <li>• Distribution Method: Monthly</li> <li>• Revenue Schedule Term Start Date: Order Product Start Date</li> <li>• Revenue Schedule Term End Date: Order Product End Date</li> </ul>	Your revenue schedule has a start date of 01/09/2018 and an end date of 01/08/2019. It contains 13 revenue transaction records – one for each month of 2018, and one for January 1 through January 8 of 2018. The January 2018 and January 2019 transactions are not for the full month, so Salesforce Billing prorates them accordingly.
<p>I want a revenue schedule that tracks revenue distributed ratably by month for my order product's entire service period. Due to the potential for amendments and changes to the billing date, I want two revenue schedules.</p> <ul style="list-style-type: none"> <li>• The first revenue schedule bases its term off my order product's term</li> <li>• The second revenue schedule bases its term off invoice line's term</li> </ul> <p>The two schedules should record to different finance books.</p>	<p>Since you need 2 revenue schedules, your revenue recognition rule should contain two treatments. Each treatment defines a different finance book and points to one of the following distribution methods.</p> <ul style="list-style-type: none"> <li>• Revenue Transaction Creation Process: Automatic</li> <li>• Type: Order</li> <li>• Distribution Method: Monthly</li> <li>• Revenue Schedule Term Start Date: Order Product Start Date</li> <li>• Revenue Schedule Term End Date: Order Product End Date</li> <li>• Revenue Transaction Creation</li> </ul>	<p>Your first revenue schedule has a start date of 01/09/2018 and an end date of 01/08/2019. It contains 13 revenue transaction records – one for each month of 2018, and one for January 1 through January 8 of 2018. The January 2018 and January 2019 transactions aren't for the full month, so Salesforce Billing prorates them accordingly.</p> <p>Your second revenue schedule has a start date of 01/09/2018 and an end date of 01/08/2019. However, you may have a different number of revenue transactions if your amendments created an invoice line with</p>

Revenue Schedule Example	Revenue Distribution Method Setup	Resulting Revenue Schedule
	Process: Automatic <ul style="list-style-type: none"> <li>• Type: Invoice</li> <li>• Distribution Method: Monthly</li> <li>• Revenue Schedule Term Start Date: Invoice Line Start Date</li> <li>• Revenue Schedule Term End Date: Invoice Line End Date</li> </ul>	different a start date than the associated order product.

### Revenue Transaction Balance Calculations for Invoice Lines and Order Products

When Salesforce Billing creates a revenue transaction for a revenue schedule, it uses a financial formula to determine the starting value for the Current Balance field. The formula results for the schedule's first revenue transaction can vary slightly based on whether the schedule recognizes revenue from an order product or invoice line. (Salesforce Billing Managed Package)

### Order-Based Revenue Recognition Reporting

Generating revenue from order products allows high visibility into a business's future revenue streams. When you create revenue schedules based on order products, you can forecast your future deferred revenue liability for the life of the specific performance obligation. Businesses using revenue forecasts for decision-making will have an up-front view of revenue forecasts for one-time and recurring products. (Salesforce Billing Managed Package)

### Invoice-Based Revenue Recognition Reporting

Invoice line-level revenue recognition reporting is ideal for businesses who use Salesforce Billing for most of their revenue recognition reporting needs. Recognizing revenue on the invoice line provides a more streamlined view of the revenue stream in many edge cases, such as amendment or usage-pricing scenarios (Salesforce Billing Managed Package)

### Configure Revenue Schedules to Use VSOE

Vender-specific objective evidence refers to the process of determining how much revenue you assign to specific items within a multi-item sale. Use VSOE during revenue recognition to accurately assign revenue to your bundle components. (Salesforce Billing Managed Package)

### Revenue Schedule Fields

Revenue schedules contain several fields that let you review how much revenue has been distributed, and how much, if any, remains to be distributed. (Salesforce Billing Managed Package)

### Create a Revenue Schedule Manually

Create a revenue schedule and enter its revenue transactions on your own. This is useful if you use a revenue recognition model that Salesforce Billing does not support. (Salesforce Billing Managed Package)

### Overriding Revenue Schedule Dates

Salesforce Billing recommends automating the revenue schedule and creation process. However, you can override revenue schedule and transaction fields if you use a revenue recognition model that Salesforce Billing doesn't support. (Salesforce Billing Managed Package)

### Validating Revenue Recognition Treatments

The Validate Revenue Recognition Treatment button lets you review whether your revenue recognition process contains all the revenue schedule creation actions that your org requires. (Salesforce Billing Managed Package)

## Revenue Transaction Balance Calculations for Invoice Lines and Order Products

When Salesforce Billing creates a revenue transaction for a revenue schedule, it uses a financial formula to determine the starting value for the Current Balance field. The formula results for the schedule's first revenue transaction can vary slightly based on whether the schedule recognizes revenue from an order product or invoice line. (Salesforce Billing Managed Package)

Salesforce Billing uses these variables when calculating the initial value of the Current Amount field on a revenue transaction.

- A—Total source transaction amount. The transaction source is an order product, invoice line, credit note, or debit note, based on the revenue schedule creation action of the revenue recognition treatment that created the revenue schedule. The field used varies based on the source object.
  - Order product: Total Price
  - Invoice line: Subtotal
- U—Number of full billing periods in the transaction source.
- P1—Number of days in a partial period. If no partial periods exist, use 0.
- P2—Number of days in the partial period's month. If a partial period covers multiple months, use the number of days in the starting month.
- F1—Number of days that the revenue transaction exists within the current finance period.
- F2—Total number of days in the current finance period.

Salesforce Billing then uses the following formula.

$$A \div (U + (P1 \div P2)) * (F1 \div F2)$$

Remember that an order product represents the total amount and all billing periods for a given transaction, while an invoice line represents only the invoiced amount and billing periods for the same transaction. These variations in amounts and billing periods mean that the revenue transaction for a given business transaction will vary slightly based on whether revenue is recognized from the order product or the invoice line. Further, when calculating the final revenue transaction balance for an invoice line, Salesforce Billing subtracts the sum of the previous revenue transaction balances from the revenue schedule's total amount.

- !** **Important** The revenue schedules for a fully-billed invoice still recognize the same amount of revenue over time as the revenue schedule for the source order product. The difference is that all the order product's revenue is accounted for in one revenue schedule, while the invoice's revenue is split up over one or more revenue schedules depending on its number of invoice lines.

For example, you have a monthly order product for a \$1,000 subscription from 05/12/21 through 12/31/21, with a Billing Day of Month of 1. The order product has a total amount (with tax) of \$765.75.

Order Product	Start Date	End Date	Amount
OP-01	05/12/21	12/31/21	\$765.75

The order product's billing rule combines partial periods, so invoicing the first line has these values.

Invoice Line	Start Date	End Date	Amount
IL-01	05/12/21	06/30/21	\$161.29

We can already see the first difference in the amounts that Salesforce Billing will use during revenue transaction calculation. Let's look at the calculations for each object to see how billing period variations between order products and invoice lines affect the resulting revenue transaction balances. For both examples, assume that you use a monthly revenue distribution method.

## Revenue Transactions for Order Products

If your order product's revenue recognition treatment creates revenue schedules upon order activation, you'll have a revenue schedule for 5/12/21 through 12/31/21 and a total amount of \$765.75. It has eight revenue transactions.

Revenue Transaction	Start Date	End Date	Current Amount
RT-01	05/12/21	05/31/21	\$64.62
RT-02	06/01/21	06/30/21	\$100.16
RT-03 through RT-08	...	...	\$100.16 each

Salesforce Billing uses these values for the first revenue transaction.

- A = \$765.75, from the order product's Total Price field.
- F = 7, covers the seven full periods starting from 05/12/21 through 06/11/21 and ending with 11/12/21 through 12/11/21.
- P1 = 20, for the days in the partial period of 12/12/21 through 12/31/21.
- P2 = 31, for the 31 days in the month of December.
- F1 = 20, for the 20 revenue transaction days in the finance period of 05/12/12 through 05/31/12.
- F2 = 31, for the 31 days in the finance period for the month of May.

So your first transaction's initial current amount is  $\$765.75 \div (7 + (20 \div 31)) * (20 \div 31) = \$64.62$ .

Because the second revenue transaction covers the full finance period of June, both F1 and F2 become 30 to represent the full month of June. These values give an initial current amount of  $\$765.75 \div (7 + (30 \div 31)) * (30 \div 31) = \$76.57$ .

$(20 \div 31) * (30 \div 33) = \$100.61$ . This formula continues for each of the remaining full finance periods, using F1 and F2 values of 31 for months with 31 days.

## Revenue Transactions for Invoice Lines

Here's how Salesforce Billing calculates revenue transactions for the same transaction when the revenue schedule is created on invoice posting. Because the invoice line ends on 06/30, Salesforce Billing creates a revenue schedule for 05/12/21 through 06/30/21. It has two revenue transactions.

Revenue Transaction	Start Date	End Date	Current Amount
RT-01	05/12/21	05/31/21	\$63.71
RT-02	06/01/21	06/30/21	\$97.58

Salesforce Billing uses these values for the first revenue transaction.

- A = \$161.29, from the order product's Total Price field.
- F = 1, covers the one full period from 05/12/21 through 06/11/21.
- P1 = 19, for the days in the partial period of 06/12/21 through 06/30/21.
- P2 = 30, for the 30 days in the month of June.
- F1 = 20, for the 20 revenue transaction days in the finance period of 05/12/12 through 05/31/12.
- F2 = 31, for the 31 full days in the finance period for the month of May.

The first transaction's initial current amount is  $\$161.29 \div (1 + (19 \div 30)) * (20 \div 31) = \$63.71$ .

Because the second revenue transaction covers the remaining invoice period, Salesforce Billing calculates its current balance by subtracting the first period's balance from the revenue schedule's total.

$$\$161.29 - \$63.71 = \$97.58$$

## Comparing Periods

To verify how Salesforce Billing calculated the revenue transaction values for the first revenue transactions for each source, let's compare the differences between the source amount (the A value) and billing periods (U, P1, and P2 values).

Revenue Transaction	Source Object	Source Amount	Billing Period Calculation Used	Notes
RT-01	Order product	\$765.75	$(7 + (20 \div 31))$	Because Salesforce Billing recognizes revenue for all the order product's

Revenue Transaction	Source Object	Source Amount	Billing Period Calculation Used	Notes
				billing periods, the F value represents seven full billing periods. P1 and P2 represent the partial period of 20 days in the month of December.
RT-01	Invoice line	\$161.29	(1 + (19 ÷ 30))	Because Salesforce Billing recognizes revenue for only the billing periods covered by the invoice line, the F value represents one full billing period. P1 and P2 represent the partial period of 19 days in the month of June.

## Order-Based Revenue Recognition Reporting

Generating revenue from order products allows high visibility into a business's future revenue streams. When you create revenue schedules based on order products, you can forecast your future deferred revenue liability for the life of the specific performance obligation. Businesses using revenue forecasts for decision-making will have an up-front view of revenue forecasts for one-time and recurring products. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### [When to Create Revenue Schedules from Order Products](#)

When deciding whether to recognize revenue from order products, consider key business needs regarding how your company uses revenue information in Salesforce Billing. For example, it's important to know whether you'll use Salesforce Billing for revenue forecasting or as the system of record to meet revenue recognition guidelines. (Salesforce Billing Managed Package)

#### [Guidelines for Setting Up Order Product-Based Revenue Recognition Rules](#)

When you're setting up order product-based revenue recognition rules, review important guidelines. (Salesforce Billing Managed Package)

### Considerations for Order-Based Revenue Recognition Reporting

When you're setting up invoice-based revenue recognition reporting, review important consideration (Salesforce Billing Managed Package)

### Creating Revenue Schedules with the Revenue Recognition Service

Developers can call the Revenue Recognition Service API to run revenue recognition for any Salesforce object in response to triggers, process builders, and REST API calls. The service is useful for recognizing or forecasting revenue on objects other than invoices, credit notes, and debit notes. For example, you could forecast quote line revenue following quote line approval, or recognize revenue after a contract is signed and activated. (Salesforce Billing Managed Package)

## When to Create Revenue Schedules from Order Products

When deciding whether to recognize revenue from order products, consider key business needs regarding how your company uses revenue information in Salesforce Billing. For example, it's important to know whether you'll use Salesforce Billing for revenue forecasting or as the system of record to meet revenue recognition guidelines. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Business Need	Should I Recognize Revenue from the Order Product?
I need to monitor my revenue cycle to predict future revenue trends and act on them.	Yes
The pricing for this product is one-time or subscription based, and I need to see the revenue it generates ahead of time.	Yes
The pricing for this product relies on usage, and I need to see the revenue it generates actively, rather than for predictive purposes.	No
I need to consolidate revenue for amendments together with revenue from the original deal.	No

Let's review some scenarios for recognizing revenue from an order product.

### Full Recognition from Order Products

To recognize revenue on the order product in full, apply a rule with an order-based, full-recognition revenue distribution method to the Product. Salesforce Billing generates a revenue schedule upon order activation. The revenue schedule contains one revenue transaction for the

order product's full amount. The revenue schedule also sets its Full Recognition Date field based on the revenue distribution method's settings – either the invoice line start date or end date.

For example, a company wants to recognize revenue up front for a one-time, \$120 hardware purchase. They could use the following revenue recognition rule configuration, which results in a revenue schedule with one revenue transaction for the full \$120.

- Revenue Recognition Rule
  - Create Revenue Schedule?: Yes
- Revenue Recognition Treatment
  - Revenue Schedule Creation Action: Order Activation
- Revenue Distribution Method
  - Type: Order
  - Distribution Method: Full Recognition
  - Full Recognition Date: Order Product Start Date or Order Product End Date
  - Revenue Transaction Creation Process: Automatic

### Pro Rata or Ratable Recognition for Order Products

Forecast revenue pro rata by applying an order-based, ratable revenue recognition rule to the product. Salesforce Billing still creates the revenue schedule upon activation, but it has several revenue transactions instead of one. The number of transactions varies based on the order product's start and end dates, and the Distribution Method field on the revenue distribution method.

For example, a company wants to recognize revenue up front for a subscription-based software service, purchased for a term of 1 year at \$10 per month. You could use a full-recognition rule to create one revenue schedule and a revenue transaction for the full \$120. However, they may also want to report revenue as it becomes recognizable over the course of the deal, otherwise known as pro rata, or ratable recognition. The following rule results in a revenue schedule with 12 transactions each for \$10. Salesforce Billing creates each transaction upon order activation.

- Revenue Recognition Rule
  - Create Revenue Schedule?: Yes
- Revenue Recognition Treatment
  - Revenue Schedule Creation Action: Order Activation
- Revenue Distribution Method
  - Type: Order
  - Distribution Method: Monthly
  - Revenue Schedule Term Start Date: Order Product Start Date
  - Revenue Schedule Term End Date: Order Product End Date
  - Revenue Transaction Creation Process: Automatic

### Recognize Revenue from Order Products with Multiple Revenue Obligations

Sometimes, a product has both flat fee and ratable revenue obligations. You can handle this by using multiple treatments on your revenue recognition rule. For example, you could make a rule which recognizes 20% of an order product line ratably and 80% up front.

- Revenue Recognition Rule
  - Create Revenue Schedule?: Yes
- Revenue Recognition Treatment
  - Type: Percentage
  - Percentage: 80
  - Revenue Schedule Creation Action: Order Activation
- Revenue Distribution Method
  - Type: Order
  - Distribution Method: Full Recognition
  - Full Recognition Date: Order Product Start Date or Order Product End Date
  - Revenue Transaction Creation Process: Automatic
- Revenue Recognition Treatment
  - Type: Percentage
  - Percentage: 20
  - Revenue Schedule Creation Action: Order Activation
- Revenue Distribution Method
  - Type: Order
  - Distribution Method: Monthly
  - Revenue Schedule Term Start Date: Order Product Start Date
  - Revenue Schedule Term End Date: Order Product End Date
  - Revenue Transaction Creation Process: Automatic

### Recognize Revenue from Order Products on Amendment

On amendments, the same revenue recognition rule applies to both the original order product and the amendment order product. Salesforce Billing creates a revenue schedule for the amendment order product based off its effective quantity.

If the amendment is a reduction or cancellation, the resulting revenue schedule has a negative value to reconcile the revenue difference.

### Guidelines for Setting Up Order Product-Based Revenue Recognition Rules

When you're setting up order product-based revenue recognition rules, review important guidelines. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Revenue Distribution Method

Configure your revenue distribution method first. Remember, your order product-based revenue recognition rule can reference one or several revenue distribution methods. You can configure the following fields based on the type of recognition your business needs.

- Distribution Method

- Full Recognition Date (if applicable)
- Revenue Schedule Term Start Date
- Revenue Schedule Term End Date

However, the following fields have required values for order-based revenue recognition.

- Type: Order
- Revenue Transaction Creation Process: Automatic

### Revenue Recognition Rule

Configure your rule after your revenue distribution method. Think of the rule as a container for revenue recognition treatments. After Salesforce Billing groups products under a revenue recognition rule, it applies that rule's treatments based on matching legal entities between the treatment and a product. Make sure your rule's Create Revenue Schedule? field has a value of Yes.

### Revenue Recognition Treatment

Your revenue recognition rule contains one or more revenue recognition treatments. When you're using an order-based revenue recognition rule, make sure your rule's treatments have the following setup.

- Associated with a revenue distribution method.
- Revenue Schedule Creation Action: Order Activation
- Type: Fill out as needed.
- Choose a percentage or a flat amount.
- If you're using several treatments on your rule, make sure that their total amount adds up to the full amount of the product's price.
- For complex configurations, we recommend creating a final treatment with a type of Remainder. This treatment ensures that your rule always recognizes remaining revenue that falls outside your regular billing periods.

## Considerations for Order-Based Revenue Recognition Reporting

When you're setting up invoice-based revenue recognition reporting, review important consideration (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Recognizing Revenue From Credit Note Lines

When you use the Cancel & Rebill button to cancel an invoice, Salesforce Billing creates credit notes to adjust the canceled invoice's balance to zero. Since cancellations happen after order activation, you can't handle this type of adjustment on a revenue schedule generated from an order product. If you need to support recognizing revenue with this scenario, you'll need credit note-based revenue recognition treatments as well.

For example, you can add credit note-based revenue recognition treatments to your revenue recognition rules to cover the possibility of a cancel and rebill scenario. Here's a credit note-based treatment added to a pro-rata revenue recognition rule.

### Order Product Deactivation

Salesforce Billing doesn't change revenue schedules and transactions when a related order is deactivated. Make sure to consider this scenario when using order-based recognition.

### Integration

Large enterprises often integrate to an ERP for final revenue recognition reporting. Consider where you want to recognize revenue from when you set up these integrations. When you're using invoice-based revenue recognition rules, the push to the back-end system can happen any time after order posting, but make sure that you consider whether to factor amendment or cancellation support into the integration architecture.

Also consider whether you want to integrate through the revenue schedule or the revenue transaction. Revenue transactions provide more detailed revenue information. However, you may only need header-level totals if your ERP breaks down revenue through more complex mechanisms than what revenue transactions support.

## Creating Revenue Schedules with the Revenue Recognition Service

Developers can call the Revenue Recognition Service API to run revenue recognition for any Salesforce object in response to triggers, process builders, and REST API calls. The service is useful for recognizing or forecasting revenue on objects other than invoices, credit notes, and debit notes. For example, you could forecast quote line revenue following quote line approval, or recognize revenue after a contract is signed and activated. (Salesforce Billing Managed Package)

For more information on the Revenue Recognition Service, review the [Revenue Recognition Service Developer Guide](#) in Salesforce Billing Developer Documentation.

## Setup

If a developer didn't already configure objects and fields for use with the revenue recognition service, follow the steps below.

To successfully pass a revenue recognition rule to the revenue recognition service, the rule's revenue distribution methods and revenue recognition treatments require several picklist fields to have a value of *Other*. The value indicates that Salesforce Billing should override the default revenue recognition process and use parameters passed to the APEX service instead. When you're configuring a revenue recognition rule for use with the service, make sure that the following fields have a value of *Other*.

### Revenue Distribution Method

Full Recognition Date

Revenue Term End Date (Needed only for daily or monthly recognition)

Revenue Term Start Date

Type

#### **Revenue Recognition Treatment**

Revenue Schedule Amount

Revenue Schedule Creation Action

If you're upgrading to Salesforce Billing Spring '20, add *Other* as a picklist value to each of these fields before using the service.

## Invoice-Based Revenue Recognition Reporting

Invoice line-level revenue recognition reporting is ideal for businesses who use Salesforce Billing for most of their revenue recognition reporting needs. Recognizing revenue on the invoice line provides a more streamlined view of the revenue stream in many edge cases, such as amendment or usage-pricing scenarios (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### **When to Use Invoice-Based Revenue Recognition**

When you're working with invoice line-level recognition, Salesforce Billing doesn't recognize revenue until it's ready for customer billing. This setup is necessary for usage-based products and useful in many other situations. For example, when businesses frequently use amendment orders, invoice-line level recognition may give a simpler view of the revenue stream. (Salesforce Billing Managed Package)

#### **Guidelines for Setting Up Invoice-Based Revenue Recognition Rules**

When you're setting up invoice-based revenue recognition rules, review important guidelines. (Salesforce Billing Managed Package)

#### **Considerations for Invoice-Based Revenue Recognition Reporting**

When you're setting up invoice-based revenue recognition reporting, review important consideration (Salesforce Billing Managed Package)

## When to Use Invoice-Based Revenue Recognition

When you're working with invoice line-level recognition, Salesforce Billing doesn't recognize revenue until it's ready for customer billing. This setup is necessary for usage-based products and useful in many other situations. For example, when businesses frequently use amendment orders, invoice-line level recognition may give a simpler view of the revenue stream. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

Let's review some common business needs and whether they're ideal for invoice line-based revenue recognition

Business Need	Should I Recognize Revenue from the Invoice Line
I need to report on Salesforce Billing revenue data to meet revenue recognition guidelines.	Yes
I need to see revenue information when revenue becomes recognizable.	Yes
The pricing for this product relies on usage, and I need to accurately track the revenue it generates.	Yes
I need to use Salesforce Billing data to forecast my revenue pipeline in advance.	No

### Full Revenue Recognition from Invoice Lines

To recognize revenue on the invoice line in full, apply a rule with an invoice-based, full distribution method to the product. Salesforce Billing creates a revenue schedule upon invoice posting. The revenue schedule contains one revenue transaction for the invoice line's full amount. The revenue schedule sets its Full Recognition Date field based on the revenue distribution method's settings – either the invoice line start date or end date.

For example, a company wants to recognize revenue upon invoicing for a one-time, \$120 hardware purchase. They could use the following setup, which results in a revenue schedule with one revenue transaction for the full \$120.

- Revenue Recognition Rule
  - Create Revenue Schedule?: Yes
- Revenue Recognition Treatment
  - Revenue Schedule Creation Action: Invoice Posting
- Revenue Distribution Method
  - Type: Invoice
  - Distribution Method: Full Recognition
  - Full Recognition Date: Invoice Line Start Date or Invoice Line End Date
  - Revenue Transaction Creation Process: Automatic

### Pro Rata or Ratable Revenue Recognition from Invoice Lines

When recognizing revenue pro rata, the revenue distribution method has a Distribution Method field set to Daily or Monthly rather than Full Recognition. On the first invoice posting, Salesforce Billing creates a revenue schedule with revenue transactions corresponding to the amount of the

invoice line, over the period of the invoice start and end dates. The revenue schedule has values for its Revenue Start Date and Revenue End Date, rather than a full recognition date. When Salesforce Billing creates more invoice lines for that order product, it adds revenue transactions to the same revenue schedule when the invoice lines post.

For example, a company wants to recognize revenue for a subscription-based software service, purchased for a term of 1 year at \$10 per month. You could use a full-recognition rule to create one revenue schedule and a revenue transaction for the full \$120. However, they may also want to report revenue as it becomes recognizable over the course of the deal, otherwise known as pro rata, or ratable recognition. The following setup results in a revenue schedule with 12 transactions each for \$10. Salesforce Billing creates each transaction when its corresponding invoice gets posted.

- Revenue Recognition Rule
  - Create Revenue Schedule?: Yes
- Revenue Recognition Treatment
  - Revenue Schedule Creation Action: Invoice Posting
- Revenue Distribution Method
  - Type: Invoice
  - Distribution Method: Monthly
  - Revenue Schedule Term Start Date: Invoice Line Start Date
  - Revenue Schedule Term End Date: Invoice Line End Date
  - Revenue Transaction Creation Process: Automatic

### Recognize Revenue from Invoice Lines on an Amendment

Recognizing revenue from an amendment quote's invoice lines will produce the same results as a net new deal. Salesforce CPQ and Billing handle all the adjustments necessary for the amendment process are handled at the order level. Usually, the amendment order product is grouped with the original order product into the same invoice line, resulting in one revenue schedule.

On a software product using the same pro rata rule described above, a customer decides to add a license. The total price for the purchase rises to \$20 per month. While you'll have separate orders and order products for the two licenses, you can still invoice them on the same invoice line, depending on the product's billing rule amendment settings. In this case, Salesforce Billing doesn't make another revenue schedule. Instead, the amount of the revenue transaction generated each month increases to \$20.

### Forecasting Revenue from the Order Product and Recognizing Revenue from the Invoice Line

You can also make a rule for recognizing revenue at the order product and invoice line levels at the same time. Use this type of rule to separate revenue pipeline forecasting from the actual revenue reporting process. This type of rule has two revenue recognition treatments, each associated to its own revenue distribution method. As a result, Salesforce Billing makes two revenue schedules for each product purchased: one upon order product activation, for forecasting, and another upon invoice line posting, for reporting.

For example, a company wants to recognize revenue for a software purchase upon invoicing. They also want to use the order product to forecast the revenue that will be generated. Let's say the subscription software was sold for 12 months at \$10 per month. You can use the following setup to create two revenue schedules, both with 12 revenue transactions for \$10 each. One schedule is associated to the order product, and the other is associated to the invoice line. This distinction lets the company track both revenue forecasting and reporting.

- Revenue Recognition Rule
  - Create Revenue Schedule?: Yes
- Revenue Recognition Treatment
  - Type: Percentage
  - Percentage: 100
  - Revenue Schedule Creation Action: Order Activation
- Revenue Distribution Method
  - Type: Order
  - Distribution Method: Monthly
  - Revenue Schedule Term Start Date: Order Product Start Date
  - Revenue Schedule Term End Date: Order Product End Date
  - Revenue Transaction Creation Process: Automatic
- Revenue Recognition Treatment
  - Type: Percentage
  - Percentage: 100
  - Revenue Schedule Creation Action: Invoice Posting
- Revenue Distribution Method
  - Type: Invoice
  - Distribution Method: Monthly
  - Revenue Schedule Term Start Date: Invoice Line Start Date
  - Revenue Schedule Term End Date: I End Date
  - Revenue Transaction Creation Process: Automatic

## Guidelines for Setting Up Invoice-Based Revenue Recognition Rules

When you're setting up invoice-based revenue recognition rules, review important guidelines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Revenue Distribution Method

Configure your revenue distribution method first. Remember, your invoice-based revenue recognition rule can reference one or several revenue distribution methods. You can configure the following fields based on the type of recognition your business needs.

- Distribution Method
- Full Recognition Date (if applicable)
- Revenue Schedule Term Start Date

- Revenue Schedule Term End Date

However, the following fields have required values for invoice-based revenue recognition.

- Type: Invoice
- Revenue Transaction Creation Process: Automatic

### Revenue Recognition Rule

Configure your rule after your revenue distribution method. Think of the rule as a container for revenue recognition treatments. After Salesforce Billing groups products under a revenue recognition rule, it applies that rule's treatments based on matching legal entities between the treatment and a product. Make sure your rule's Create Revenue Schedule? field has a value of Yes.

### Revenue Recognition Treatment

Your revenue recognition rule contains one or more revenue recognition treatments. When you're using an invoice-based revenue recognition rule, make sure your rule's treatments have the following setup.

- Associated with a revenue distribution method.
- Revenue Schedule Creation Action: Invoice Posting
- Type: Fill out as needed.
- Choose a percentage or a flat amount.
- If you're using several treatments on your rule, make sure that their total amount adds up to the full amount of the product's price.
- For complex configurations, we recommend creating a final treatment with a type of Remainder. This treatment ensures that your rule always recognizes remaining revenue that falls outside your regular billing periods.

## Considerations for Invoice-Based Revenue Recognition Reporting

When you're setting up invoice-based revenue recognition reporting, review important consideration (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Recognize Revenue from Invoice Lines for a Usage Product

Use invoice-based revenue recognition rules for usage products. If you create an order-based revenue schedule for a usage product, the schedule's net amount will always be zero.

Usage invoice line revenue schedules look similar to recurring invoice line revenue schedules. With a distribution method of Monthly or Daily, Salesforce Billing creates revenue transactions across the entire invoice period, for the full amount of the related usage summary. With a distribution method of Full Recognition, the revenue schedule contains one revenue transaction

for the full amount of usage consumed at the end of the deal's term.

### Recognize Revenue from Credit Note Lines

When you use the Cancel & Rebill button to cancel an invoice, Salesforce Billing creates credit notes to adjust the canceled invoice's balance to zero. You'll need credit-note based revenue recognition treatments in this scenario so that you can accurately update revenue information. Otherwise, Salesforce Billing recognizes the revenue again for the same invoice line on the revised invoice, without adjusting the original balance down to zero.

For example, you can add credit note-based revenue recognition treatments to your revenue recognition rules to cover the possibility of a cancel and rebill scenario. Here's a credit note-based treatment added to a pro-rata revenue recognition rule.

### Integrations

Large enterprises often integrate to an ERP for final revenue recognition reporting. Consider where you want to recognize revenue from when you set up these integrations. When you're using invoice-based revenue recognition rules, the push to the back-end system can happen any time after invoice posting. Also consider whether you want real-time or scheduled integration. If the integration is real-time, you'll need to cover other scenarios such as cancel and rebill.

## Configure Revenue Schedules to Use VSOE

Vendor-specific objective evidence refers to the process of determining how much revenue you assign to specific items within a multi-item sale. Use VSOE during revenue recognition to accurately assign revenue to your bundle components. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

When customers buy bundles, purchasing an entire bundle is often cheaper than purchasing each component individually. When this happens, you split the bundle during revenue recognition into its component services. You can customize your revenue recognition rule and treatment to apply a VSOE process that calculates the revenue Salesforce Billing assigns to each component.

1. Within your bundle, identify all the component goods and services where you want to recognize revenue.
2. Use VSOE to determine the standalone selling price of each good and service.
3. Use the standalone selling price for each good or service to determine the portion of the bundle's overall revenue attributable to that good or service.

For example, let's say you have a bundle with 3 component services. The sum of the standalone selling price of all component services is \$10,000. Your first service has a selling price of \$5,000. Therefore, it represents 50 percent of your bundle's revenue.

4. Create a revenue recognition rule that contains a revenue treatment for each component that you

want to track.

Each revenue treatment should contain at least the following fields and values.

- Name
- Active: Selected
- Type: Percentage
- Percentage: Enter the percentage you calculated in step 3

5. Make the revenue recognition rule active and assign it to your bundle product.

When you invoice this bundle product, Salesforce Billing creates one revenue schedule for each bundle component. Remember, Salesforce Billing uses the following formula to calculate revenue for each revenue schedule.

```
Total revenue amount = (Percentage from revenue treatment) * (Total order product amount for the bundle)
```

## Revenue Schedule Fields

Revenue schedules contain several fields that let you review how much revenue has been distributed, and how much, if any, remains to be distributed. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

#### Account

Account containing the revenue schedule. Not populated by default.

#### Adjustments

The total of changes created through revenue adjustments to this revenue schedule.

#### Available

The total amount on this revenue schedule, including adjustments, that hasn't been distributed to any revenue transactions.

Calculated as Total Amount + Adjustments - (Recognized + Unrecognized)

Revenue teams can use this field to identify revenue schedules that require further review and distribution of revenue transactions.

#### Base Currency

Has no default value or functionality. We've provided this field so that you can define base currencies on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates this field based on currency rates that match the date of the transaction.

**Base Currency Amount**

Has no default value or functionality. We've provided this field so that you can define base currency amounts on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates this field based on currency rates that match the date of the transaction.

**Base Currency FX Date**

Has no default value or functionality. We've provided this field so that you can define base currency dates on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates this field based on currency rates that match the date of the transaction.

**Base Currency FX Rate**

Has no default value or functionality. We've provided this field so that you can define base currency rates on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates this field based on currency rates that match the date of the transaction.

**Contract**

Lookup field for a contract record. This field has no default value or functionality. You can populate it manually or configure automation to do so. Contract references may be useful for reporting purposes.

**Credit Note Line**

Credit note line used to calculate revenue transactions for the revenue schedule. Populated only when Salesforce Billing creates a revenue schedule under a revenue recognition treatment with a Revenue Schedule Creation Action field set to Credit Note Posting.

**Debit Note Line**

Debit note line used to calculate revenue transactions for the revenue schedule. Populated only when Salesforce Billing creates a revenue schedule under a revenue recognition treatment with a Revenue Schedule Creation Action field set to Debit Note Posting.

**Deferred**

The total amount of revenue that hasn't been recognized yet, plus any adjustments.

Deferred revenue is an important value as it represents future revenue that an organization has already booked.

Calculated as Unrecognized + Available

**Estimated Revenue Transaction Count**

When Salesforce Billing creates a revenue schedule, it estimates the number of revenue

transactions that the schedule will contain. It bases this estimation on the revenue start date, revenue end date, and revenue distribution method. For example, a start date of 01/01/2021, end date of 12/31/2021, and distribution method of Monthly would produce an estimated transaction value of 12. This number may differ from the final number of revenue transactions if the transaction source is canceled or deleted, or if you use custom automation or API to change the default revenue transaction creation process.

### **Full Recognition Date**

The date by which the revenue schedule will recognize all revenue for its transactions. Inherited from the revenue distribution method's Full Recognition Date field.

### **Invoice Line**

Invoice line used to calculate revenue transactions for the revenue schedule. Populated only when Salesforce Billing creates a revenue schedule under a revenue recognition treatment with a Revenue Schedule Creation Action field set to Invoice Posting.

### **Notes**

User-provided notes for the revenue schedule.

### **Order Product**

Order product used to calculate revenue transactions for the revenue schedule. Populated only when Salesforce Billing creates a revenue schedule under a revenue recognition treatment with a Revenue Schedule Creation Action field set to Order Activation.

### **Recognized**

The sum of this revenue schedule's revenue transactions that have been recognized and distributed to closed finance periods.

### **Revenue Agreement**

Lookup field for a revenue agreement. This field has no default value or functionality. You can populate it manually or configure automation to do so. Revenue agreement references may be useful for reporting purposes.

### **Revenue Agreement Allocation Status**

Text field with no default value or functionality. We've provided it in case you want to use custom automation to indicate whether any revenue has been allocated to or from the revenue agreement related to a revenue schedule.

### **Revenue Allocation Amount**

Currency field with no default value or functionality. We've provided it in case you want to use custom automation to indicate how much revenue has been allocated to or from the revenue schedule.

**Revenue Distribution Method**

Inherited from the revenue recognition treatment that created the revenue schedule.

**Revenue End Date**

The earliest date for revenue recognized under the revenue schedule. Salesforce Billing calculates this date based off the revenue distribution method's Revenue Schedule Term End Date field. For example, let's say the term end date has a value of Order Product End Date. The revenue end date would be the end date of the order product that the revenue schedule is recording.

**Revenue Expected Amount**

The amount of revenue that is expected to be received. Different organizations have different ways to calculate this value based on their revenue and accounting standards, so we've left the field with no default calculations. You can configure it as needed.

**Revenue Finance Book**

The finance book that records transactions related to the revenue schedule. Inherited from the revenue recognition treatment's Revenue Finance Book field.

**Revenue Legal Entity**

Legal entity used to associate the revenue schedule with other transactional records in Salesforce Billing. Inherited from the revenue recognition treatment's Revenue Legal Entity field.

**Revenue Liability Amount**

The amount of revenue collected for which a product or service hasn't been rendered. Different organizations have different ways to calculate this value based on their revenue and accounting standards, so we've left the field with no default calculations. You can configure it as needed.

**Revenue Most Likely Amount**

The amount of revenue that is most likely to be received. Different organizations have different ways to calculate this value based on their revenue and accounting standards, so we've left the field with no default calculations. You can configure it as needed.

**Revenue Recognition GL Rule**

GL rule used to organize the revenue schedule along with other entities related to the same GL rule. These relationships are helpful if want to export Salesforce Billing data to an external general ledger system. Inherited from the revenue recognition treatment's Revenue GL Rule field.

**Revenue Recognition GL Treatment**

GL treatment used to organize the revenue schedule along with other entities related to the same GL treatment. These relationships are helpful if want to export Salesforce Billing data to an external general ledger system. Assigned based on the revenue recognition gl rule's treatments.

**Revenue Recognition Rule**

Revenue recognition rule that was used to configure field values and relationships for the revenue schedule.

**Revenue Recognition Treatment**

Revenue recognition treatment that was used to configure field values and relationships for the revenue schedule.

**Revenue Start Date**

The earliest date for revenue recognized under the revenue schedule. Salesforce Billing calculates this date based off the revenue distribution method's Revenue Schedule Term Start Date field. For example, let's say the term end date has a value of Order Product Start Date. The revenue start date would be the start date of the order product that the revenue schedule is recording.

**Revenue Transaction Status**

Shows the status of the process that Salesforce Billing runs to create revenue transactions for the revenue schedule.

- Complete: The revenue transactions have been successfully created.
- Queued: The Apex job to create the revenue transactions has been triggered and is running or will run soon.
- Error: Salesforce Billing encountered an error creating the revenue transactions and recorded information on it in a revenue schedule error log. After you've fixed the errors, set the revenue transaction status to Queued to trigger the revenue transaction creation job.

**Total Revenue Amount**

The total amount available when the revenue schedule is created. If you set up Salesforce Billing to automatically create revenue transactions, the schedule automatically distributes its available amount to revenue transactions. This value doesn't change based on adjustments.

**Unrecognized**

The sum of this revenue schedule's revenue transactions that haven't been recognized and distributed to open finance periods.

## Create a Revenue Schedule Manually

Create a revenue schedule and enter its revenue transactions on your own. This is useful if you use a revenue recognition model that Salesforce Billing does not support. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

1. Create a revenue recognition rule and set its Create Revenue Schedule field to No or None.
2. Create a revenue distribution method and set its Revenue Transaction Creation Process field to Manual. We recommend giving it a name like “Manual Transaction Creation.”
3. Create a revenue recognition treatment on your rule. Give your treatment’s Revenue Distribution Method field a lookup to the manual method you created in Step 2.
4. Add the Revenue Schedule related list to your revenue distribution method and create a revenue schedule. Give your revenue schedule a lookup to the rule you created in step 1 and the treatment you created in step 2.
5. Fill out the rest of your revenue schedule’s fields as needed and save.
6. Create your revenue transaction records on your revenue schedule. Since you’re creating the transactions manually, you have to assign each transaction a lookup to a revenue finance period. Fill out the rest of your revenue transaction fields as needed and save.

## Overriding Revenue Schedule Dates

Salesforce Billing recommends automating the revenue schedule and creation process. However, you can override revenue schedule and transaction fields if you use a revenue recognition model that Salesforce Billing doesn’t support. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: All Salesforce Billing Editions

### Revenue Schedules

Overridable Field	Default Value	Override Guidelines
Revenue Start Date	<p>This field calculates its value based off your revenue distribution method’s Revenue Schedule Term Start Date field, which can be any of the following values.</p> <ul style="list-style-type: none"> <li>• Order Product Start Date</li> <li>• Invoice Line Start Date</li> <li>• Credit Note Start Date</li> <li>• Debit Note Start Date</li> <li>• Manual</li> </ul>	If you want to override your revenue start date, we recommend leaving your distribution method’s revenue schedule term start date to Manual. That way, Salesforce Billing creates your revenue schedule without a revenue start date. You can then set the revenue start date on your own.

Overridable Field	Default Value	Override Guidelines
Revenue End Date	<p>This field calculates its value based off your revenue distribution method's Revenue Schedule Term End Date field, which can be any of the following values.</p> <ul style="list-style-type: none"> <li>• Order Product End Date</li> <li>• Invoice Line End Date</li> <li>• Credit Note End Date</li> <li>• Debit Note End Date</li> <li>• Manual</li> </ul>	If you want to override your revenue end date, we recommend leaving your distribution method's revenue schedule term end date to Manual. That way, Salesforce Billing creates your revenue schedule without a revenue end date. You can then set the revenue end date on your own.

### Note

- If the revenue distribution method is configured to manual transaction generation, overriding revenue start and end dates or not overriding them makes no difference. The dates aren't populated to the revenue schedule.
- If the revenue distribution method is configured to automatic transaction generation, transactions are generated according to the override start and end dates provided by the customer.

## Validating Revenue Recognition Treatments

The Validate Revenue Recognition Treatment button lets you review whether your revenue recognition process contains all the revenue schedule creation actions that your org requires. (Salesforce Billing Managed Package)

A revenue recognition rule requires at least one revenue recognition treatment to create a revenue schedule. A treatment requires a value for the Revenue Schedule Creation Action field, which defines whether the treatment creates a revenue schedule upon order activation, invoice posting, credit note posting, or debit note posting.

Some use cases need revenue schedule creation for two or more actions, which requires multiple treatments on your rule, each with different revenue schedule creation actions. For example, your rule may need one treatment for revenue schedule creation upon invoice posting, and a second treatment for revenue schedule creation upon debit note posting. When you click **Validate Revenue Recognition Treatments** on your revenue recognition rule, Salesforce Billing lists the revenue schedule creation actions not included on any of your rule's treatments. You're not required to use all the values that the button returns. We provide the values only for reference so you can double-check whether your particular implementation is missing any of the values listed.

## Revenue Recognition Policies

Revenue recognition is an accounting principle used to determine when and how revenue is recognized or accounted for. Your business can apply different methods and policies when deciding how to recognize revenue. The type of business determines which policy to apply. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Revenue is recognized based on rules applied through accrual accounting and the matching principle. Accrual accounting states that revenue is recognized when it's realized and earned, independent of when cash is received. Realized means the good or service has been received, and earned means the good has been provided or a service has been delivered. Finally, the matching principle states that revenue and associated costs, such as costs of goods or commission, should be accounted for in the same period.

The method for recognizing revenue depends on the type of business transaction and the stipulations laid out in the contract. For example, revenue can be recognized daily, monthly, or all at once. Revenue could also be recognized beginning on service start or end dates, or based on the date of invoice.

The International Accounting Standards Board (IASB) and Financial Accounting Standards Board (FASB) introduced the Accounting Standards Codification (ASC) 606 in 2014. ASC 606 states that revenue is recognized when the delivery of promised goods or services matches the amount of consideration expected in exchange for the goods and service. To comply with that standard, businesses align their processes to five steps.

1. Identify the contract with the customer
2. Identify the contract's performance obligations.
3. Determine the transaction price
4. Allocate the transaction price to the contract's performance obligations
5. Recognize revenue when (or as) the organization meets a performance obligation

ASC 606 aims to create a transparent revenue recognition framework for use across all business sectors. While the core concepts for when and how to recognize revenue remain the same, the ASC606 principle standardizes the practice to provide a more concise recognition process.

Let's review a few basic recognition examples.

### Full Recognition Based on Invoice Date

A painting company has a contract to repaint a customer's home. The contract states that the price is \$3000. Painting starts June 1 and ends July 31.

The customer received an invoice with an invoice date of Jun 1 and net 15 payment terms. They have until June 15 to pay the invoice. Once payment has been processed, the painting company recognizes the full amount of revenue with a June 1 date.

### Monthly Recognition Based on Service Activation Date

A company sells 12-month MDM subscriptions at \$10 per month. Customers can pay for their subscription monthly or upfront.

If a customer wants to be billed monthly, they would be billed \$10 a month for 12 months on the service activation date of their subscription. In this case, revenue is recognized immediately.

If a customer wants to be billed upfront, they would be billed \$120 on the service activation date of their subscription. However, in this case, revenue would be deferred at 1/12 per month for the 12-month life of the subscription.

### Monthly Recognition Based on Service Activation Date

An online security company sells annual subscriptions of security software. They sell a \$12,000 subscription in October for the following calendar year.

The company bills the customer in October. Even though the customer paid the full amount in October, revenue recognition doesn't start until January 1, when the services are activated.

Starting in January, revenue is recognized on a monthly basis at \$1000 per month for each of the 12 months of service.

#### [Set Up a Revenue Recognition Rule for Daily Proration](#)

Create a revenue recognition rule that proportionately distributes the amount paid for a product or service on a daily recurring basis over the course of the billing term. (Salesforce Billing Managed Package)

#### [Set Up a Revenue Recognition Rule for Daily Proration](#)

Create a revenue recognition rule that proportionately distributes the amount paid for a product or service on a monthly recurring basis over the course of the billing term. (Salesforce Billing Managed Package)

#### [Set Up a Revenue Recognition Rule for Full Recognition on a Specific Day](#)

Create a revenue recognition rule that recognizes on a specific day the full amount paid for a product or service. (Salesforce Billing Managed Package)

#### [Set Up a Manual Revenue Recognition Rule](#)

Create a revenue recognition rule you can use to manually create revenue schedules and revenue transactions for each invoice line. (Salesforce Billing Managed Package)

## Set Up a Revenue Recognition Rule for Daily Proration

Create a revenue recognition rule that proportionately distributes the amount paid for a product or service on a daily recurring basis over the course of the billing term. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

We're using invoice-based revenue recognition for this revenue recognition rule.

1. Create your revenue recognition rule.
  - a. From the revenue recognition rule object, click **New**.
  - b. Give your revenue recognition rule a name. We recommend something simple and descriptive, such as *Ratable Daily Proration*.
  - c. Select **Active**.
2. On your rule, create a revenue recognition treatment.
  - a. Click **New Revenue Recognition Treatment**.
  - b. Set the revenue schedule creation action to Invoice Posting.
  - c. Set the revenue schedule amount to Transaction Amount.
  - d. Set your revenue legal entity, revenue gl rule, and revenue finance book based on your business needs.
3. Create your revenue distribution method.
  - a. From the revenue distribution method object, click **New**.
  - b. Set the monthly recognition proration to Number of Days.
  - c. Set the type to Invoice.
  - d. Set the distribution method to Daily.
  - e. Select **Active**.
  - f. Set the revenue schedule term start date to Invoice Line Start Date.
  - g. Set the revenue schedule term end date to Invoice Line End Date.
4. Return to the revenue recognition treatment you made in step 2 and set its revenue distribution method to the record you created in Step 3.

## Set Up a Revenue Recognition Rule for Daily Proration

Create a revenue recognition rule that proportionately distributes the amount paid for a product or service on a monthly recurring basis over the course of the billing term. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

We're using invoice-based revenue recognition for this revenue recognition rule.

1. Create your revenue recognition rule.
  - a. From the revenue recognition rule object, click **New**.
  - b. Give your revenue recognition rule a name. We recommend something simple and descriptive, such as *Ratable Monthly Proration*.
  - c. Select **Active**.
2. On your rule, create a revenue recognition treatment.
  - a. Click **New Revenue Recognition Treatment**.
  - b. Give your revenue recognition rule a name. We recommend something simple and descriptive, such as *Ratable Monthly Proration Treatment*.

- c. Set the revenue schedule creation action to Invoice Posting.
  - d. Set the revenue schedule amount to Transaction Amount.
  - e. Set your revenue legal entity, revenue gl rule, and revenue finance book based on your business needs.
3. Create your revenue distribution method.
    - a. From the revenue distribution method object, click **New**.
    - b. Set the monthly recognition proration to Number of Days.
    - c. Set the type to Invoice.
    - d. Set the distribution method to Monthly.
    - e. Select **Active**.
    - f. Set the revenue schedule term start date to Invoice Line Start Date.
    - g. Set the revenue schedule term end date to Invoice Line End Date.
  4. Return to the revenue recognition treatment you made in step 2 and set its revenue distribution method to the record you created in Step 3.

## Set Up a Revenue Recognition Rule for Full Recognition on a Specific Day

Create a revenue recognition rule that recognizes on a specific day the full amount paid for a product or service. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

We're using invoice-based revenue recognition for this revenue recognition rule.

1. Create your revenue recognition rule.
  - a. From the revenue recognition rule object, click **New**.
  - b. Give your revenue recognition rule a name. We recommend something simple and descriptive, such as *One-Day Full Recognition*.
  - c. Select **Active**.
2. On your rule, create a revenue recognition treatment.
  - a. Click **New Revenue Recognition Treatment**.
  - b. Give your revenue recognition rule a name. We recommend something simple and descriptive, such as *One-Day Full Recognition Treatment*.
  - c. Set the revenue schedule creation action to Invoice Posting.
  - d. Set the revenue schedule amount to Transaction Amount.
  - e. Set your revenue legal entity, revenue gl rule, and revenue finance book based on your business needs.
3. Create your revenue distribution method.
  - a. From the revenue distribution method object, click **New**.
  - b. Set the monthly recognition proration to Number of Days.
  - c. Set the type to Invoice.
  - d. Set the distribution method to Full Recognition.
  - e. Select **Active**.

- f. Set the full recognition date to Invoice Line Start Date or End Date, depending on your business needs.
4. Return to the revenue recognition treatment you made in step 2 and set its revenue distribution method to the record you created in Step 3.

## Set Up a Manual Revenue Recognition Rule

Create a revenue recognition rule you can use to manually create revenue schedules and revenue transactions for each invoice line. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

1. Create your revenue recognition rule.
  - a. From the revenue recognition rule object, click **New**.
  - b. Give your revenue recognition rule a name, ideally something simple and descriptive such as *Ratable Daily Proration*.
  - c. Select **Active**.
2. On your rule, create a revenue recognition treatment.
  - a. Select *New Revenue Recognition Treatment*.
  - b. Give your treatment a name, such as *Manual Recognition Treatment*
  - c. Select **Order Activation** as the revenue schedule creation action.
  - d. Set the processing order to 1.
  - e. Select **Percentage** as the type, and set the percentage to 100.
  - f. Select **Transaction Amount** as the revenue schedule amount.
  - g. Select **Manual Distribution Method** as the revenue distribution method.
  - h. Set your revenue legal entity, revenue GL rule, and revenue finance book based on your business needs.
3. Create your revenue distribution method.
  - a. From the revenue distribution method object, click **New**.
  - b. Give your method a name,  
, such as *Manual.Distribution.Method*
  - c. Select **Number of Days** as the monthly recognition proration.
  - d. Select **Order** as the type.
  - e. Select **Daily**, **Monthly**, or **Full Recognition** as the distribution method.
  - f. Select **Active**.
  - g. If the distribution method is Daily or Monthly, set the revenue schedule term start and end dates to Manual, and leave the full recognition date set to None. If the distribution method is Full Recognition, set the full recognition date to Manual and set the revenue schedule term start and end dates as None.

After you activate the order product, a revenue schedule is generated and linked to the order product. You can then create revenue transactions manually on the revenue schedule.

When you activate an order, all order products are activated. You can also activate the order products individually. If, in step 2b above, you set the revenue schedule creation action to Invoice Posting, the revenue schedule is generated and linked to the invoice line when the invoice is posted. Posting an invoice posts all invoice lines, but invoice lines can't be posted individually.

## Revenue Agreements

Revenue agreements are objects with a Revenue Schedule related list. They're useful for grouping revenue schedules related to the same transaction, so that you can quickly evaluate revenue amounts for that transaction. You can use default or custom functionality to assign revenue schedules to a revenue agreement. Each revenue agreement contains several revenue amount fields that you can customize to report on revenue from the related revenue schedules based on your organization's revenue reporting standards. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '18 and later

---

The revenue recognition treatment has a Revenue Agreement Association field, which controls when Salesforce Billing makes revenue agreements and when it assigns revenue schedules to them.

#### Not Applicable

Salesforce Billing doesn't create revenue agreements.

#### Manual

Salesforce Billing doesn't create revenue agreements. Users must create and assign them on their own. While this value functions the same as Not Applicable, we recommend using it to indicate situations where users should create revenue agreements and custom automation on their own.

#### New

Salesforce Billing creates one revenue agreement for an order upon revenue schedule creation for that order. If the order has any amendment orders, Salesforce Billing creates and assigns them to a new revenue agreement upon revenue schedule creation.



#### Revised Order Product

Salesforce Billing creates one revenue agreement for an order upon revenue schedule creation for that order. The order's related amendment orders are related to the same revenue agreement.



## Revenue Agreement Fields

Revenue agreements contain 8 revenue amount fields. Different organizations have different ways to calculate revenue, so we've left the fields without any default formulas or calculations. You can customize them to calculate revenue based on your organization's standards. Since most revenue agreement will be related to one or more revenue schedules, you can quickly reference their fields in your revenue amount calculations.

Revenue Bookings Fields	Revenue Billings Fields
Total Transaction Amt(Bookings)	Total Transaction Amt(Billings)
Total Revenue Allocation Amt(Bookings)	Total Revenue Allocation Amt(Billings)
Total Revenue Liability Amt(Bookings)	Total Revenue Liability Amt(Billings)
Total Revenue Expected Amt(Bookings)	Total Revenue Expected Amt(Billings)

For example, let's say you have a Revised Order Product revenue agreement association. You could configure the Total Revenue Liability Amount (Bookings) field with a formula to use the Unrecognized field value from each revenue schedule related to your revenue agreement.



You can also use custom automation to populate the revenue agreement's Order, Contract, and Order Product fields, which may be helpful for your revenue reporting.



**Example** Let's say you want customized revenue agreement reporting that assigns the revenue schedules for an order product's related invoice lines, credit note lines, and debit note lines to a revenue agreement. Since invoice lines, credit note lines, and debit note lines are typically recognized as billings, this configuration would help you quickly evaluate the bookings and billings for a given transaction.

1. Create a revenue recognition rule and treatment where the treatment's Revenue Agreement Association as a value of Manual.
2. Write custom automation for the Manual value.
  - a. Salesforce Billing creates a revenue agreement upon revenue schedule creation for a given order product.
  - b. Salesforce Billing creates revenue schedules for each invoice line, credit note line, and debit note line for the order product, then assigns those revenue schedules to the revenue agreement.

## Revenue Adjustments

When your actual revenue differs from your initial revenue schedule, you may need to increase or decrease your revenue schedule's available balance, or transfer it to the available balance on a different schedule. You can use revenue adjustments to change your revenue schedule's available balance. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

Your actual revenue can differ from the deferred revenue on your revenue schedule. For example, let's say you invoiced a three-month data plan contract (January through March) assuming your customer would use \$100 of data each month. This creates a revenue schedule with an unrecognized balance of \$300 and three revenue transactions of \$100 each. However, your customer actually uses \$80 of data in January.

To ensure that all changes are tracked for auditing and bookkeeping, Salesforce Billing doesn't let you directly edit revenue schedule balances – other objects have to apply updates so that you always have records of revenue balance changes. While you can edit revenue transaction balances, these changes don't roll up into the revenue schedule's amount.

You can change your revenue schedule's balance by using a revenue adjustment. Revenue adjustments contain revenue adjustment lines, whose balances roll up to the revenue adjustment's total amount. Each revenue adjustment line has a lookup to a revenue schedule. This way, you can associate different lines with different schedules if needed.

Revenue adjustments allow for two types of revenue changes.

- **Adjustment:** Change a revenue schedule's balance. You can change one revenue schedule or have multiple lines that each target a different revenue schedule.
- **Transfer:** Transfer available balances between revenue schedules. In Salesforce Billing, this means that your revenue adjustment has a balance of zero. For example, you could have a revenue adjustment with a \$10 line targeting Revenue Schedule A and a -\$10 line targeting revenue schedule B. You could also have a revenue adjustment with a \$30 line targeting Revenue Schedule A, a -\$15 line targeting Revenue Schedule B, and a -\$15 line targeting revenue schedule C.

All adjustments and transfers are made to and from a schedule's available balance.

**Important** A revenue adjustment line can't decrease a positive revenue schedule's available balance below zero, or increase a negative revenue schedule's balance above zero.

When you post your revenue adjustment, Salesforce Billing posts all its lines and adds their balances to their respective schedules. For our data plan example, you could make an adjustment-type revenue adjustment with one line that has a balance of -\$20. When you post your adjustment, the line adjusts your schedule's schedule amount plus adjustments amounts to \$280.

Since revenue adjustments can have a financial impact, you'll need to make sure that you track adjustment or transfers. Each revenue adjustment and revenue adjustment line requires a lookup to a legal entity, GL rule, GL treatment, finance book, and finance period. The legal entity, GL rule, and GL treatment default to the values from the revenue schedule where you created your revenue adjustment record. However, you can change them as needed.

### Create a Revenue Adjustment

Create a revenue adjustment and revenue adjustment lines to change or transfer a revenue schedule's available balance. Post the adjustment once you're ready to apply it. (Salesforce Billing Managed Package)

### Revenue Schedule Fields

Revenue schedules contain several fields that let you review how much revenue has been distributed, and how much, if any, remains to be distributed. (Salesforce Billing Managed Package)

### Revenue Adjustment Use Cases

Revenue adjustments let you adjust or transfer balances between any number of revenue schedules. To learn how to set up these adjustments, review common use cases. (Salesforce Billing Managed Package)

## Create a Revenue Adjustment

Create a revenue adjustment and revenue adjustment lines to change or transfer a revenue schedule's available balance. Post the adjustment once you're ready to apply it. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

1. From the Revenue Adjustment list view, click **New**.
2. Give your revenue adjustment the following field values, then click **Save**.
  - Type: Adjustment or Transfer
  - Status: Draft
3. From your revenue adjustment, click **New Adjustment Line**.
4. Choose the revenue schedule you want to adjust, then enter an amount.
5. Click **Save**.
6. Repeat steps 3 through 5 if you need to make additional lines.
7. When you're ready to apply your revenue adjustment, change its status to **Posted**. If your revenue adjustment type is Transfer, make sure your revenue adjustment's balance is zero. The adjustment's balance is equal to the sum of its lines.
8. Click **Save**.

## Revenue Schedule Fields

Revenue schedules contain several fields that let you review how much revenue has been distributed,

and how much, if any, remains to be distributed. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

### Account

Account containing the revenue schedule. Not populated by default.

### Adjustments

The total of changes created through revenue adjustments to this revenue schedule.

### Available

The total amount on this revenue schedule, including adjustments, that hasn't been distributed to any revenue transactions.

Calculated as Total Amount + Adjustments - (Recognized + Unrecognized)

Revenue teams can use this field to identify revenue schedules that require further review and distribution of revenue transactions.

### Base Currency

Has no default value or functionality. We've provided this field so that you can define base currencies on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates this field based on currency rates that match the date of the transaction.

### Base Currency Amount

Has no default value or functionality. We've provided this field so that you can define base currency amounts on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates this field based on currency rates that match the date of the transaction.

### Base Currency FX Date

Has no default value or functionality. We've provided this field so that you can define base currency dates on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates this field based on currency rates that match the date of the transaction.

### Base Currency FX Rate

Has no default value or functionality. We've provided this field so that you can define base currency rates on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates this field based on currency rates that match the date of the transaction.

**Contract**

Lookup field for a contract record. This field has no default value or functionality. You can populate it manually or configure automation to do so. Contract references may be useful for reporting purposes.

**Credit Note Line**

Credit note line used to calculate revenue transactions for the revenue schedule. Populated only when Salesforce Billing creates a revenue schedule under a revenue recognition treatment with a Revenue Schedule Creation Action field set to Credit Note Posting.

**Debit Note Line**

Debit note line used to calculate revenue transactions for the revenue schedule. Populated only when Salesforce Billing creates a revenue schedule under a revenue recognition treatment with a Revenue Schedule Creation Action field set to Debit Note Posting.

**Deferred**

The total amount of revenue that hasn't been recognized yet, plus any adjustments.

Deferred revenue is an important value as it represents future revenue that an organization has already booked.

Calculated as Unrecognized + Available

**Estimated Revenue Transaction Count**

When Salesforce Billing creates a revenue schedule, it estimates the number of revenue transactions that the schedule will contain. It bases this estimation on the revenue start date, revenue end date, and revenue distribution method. For example, a start date of 01/01/2021, end date of 12/31/2021, and distribution method of Monthly would produce an estimated transaction value of 12. This number may differ from the final number of revenue transactions if the transaction source is canceled or deleted, or if you use custom automation or API to change the default revenue transaction creation process.

**Full Recognition Date**

The date by which the revenue schedule will recognize all revenue for its transactions. Inherited from the revenue distribution method's Full Recognition Date field.

**Invoice Line**

Invoice line used to calculate revenue transactions for the revenue schedule. Populated only when Salesforce Billing creates a revenue schedule under a revenue recognition treatment with a Revenue Schedule Creation Action field set to Invoice Posting.

**Notes**

User-provided notes for the revenue schedule.

**Order Product**

Order product used to calculate revenue transactions for the revenue schedule. Populated only when Salesforce Billing creates a revenue schedule under a revenue recognition treatment with a Revenue Schedule Creation Action field set to Order Activation.

**Recognized**

The sum of this revenue schedule's revenue transactions that have been recognized and distributed to closed finance periods.

**Revenue Agreement**

Lookup field for a revenue agreement. This field has no default value or functionality. You can populate it manually or configure automation to do so. Revenue agreement references may be useful for reporting purposes.

**Revenue Agreement Allocation Status**

Text field with no default value or functionality. We've provided it in case you want to use custom automation to indicate whether any revenue has been allocated to or from the revenue agreement related to a revenue schedule.

**Revenue Allocation Amount**

Currency field with no default value or functionality. We've provided it in case you want to use custom automation to indicate how much revenue has been allocated to or from the revenue schedule.

**Revenue Distribution Method**

Inherited from the revenue recognition treatment that created the revenue schedule.

**Revenue End Date**

The earliest date for revenue recognized under the revenue schedule. Salesforce Billing calculates this date based off the revenue distribution method's Revenue Schedule Term End Date field. For example, let's say the term end date has a value of Order Product End Date. The revenue end date would be the end date of the order product that the revenue schedule is recording.

**Revenue Expected Amount**

The amount of revenue that is expected to be received. Different organizations have different ways to calculate this value based on their revenue and accounting standards, so we've left the field with no default calculations. You can configure it as needed.

**Revenue Finance Book**

The finance book that records transactions related to the revenue schedule. Inherited from the revenue recognition treatment's Revenue Finance Book field.

**Revenue Legal Entity**

Legal entity used to associate the revenue schedule with other transactional records in Salesforce Billing. Inherited from the revenue recognition treatment's Revenue Legal Entity field.

**Revenue Liability Amount**

The amount of revenue collected for which a product or service hasn't been rendered. Different organizations have different ways to calculate this value based on their revenue and accounting standards, so we've left the field with no default calculations. You can configure it as needed.

**Revenue Most Likely Amount**

The amount of revenue that is most likely to be received. Different organizations have different ways to calculate this value based on their revenue and accounting standards, so we've left the field with no default calculations. You can configure it as needed.

**Revenue Recognition GL Rule**

GL rule used to organize the revenue schedule along with other entities related to the same GL rule. These relationships are helpful if want to export Salesforce Billing data to an external general ledger system. Inherited from the revenue recognition treatment's Revenue GL Rule field.

**Revenue Recognition GL Treatment**

GL treatment used to organize the revenue schedule along with other entities related to the same GL treatment. These relationships are helpful if want to export Salesforce Billing data to an external general ledger system. Assigned based on the revenue recognition gl rule's treatments.

**Revenue Recognition Rule**

Revenue recognition rule that was used to configure field values and relationships for the revenue schedule.

**Revenue Recognition Treatment**

Revenue recognition treatment that was used to configure field values and relationships for the revenue schedule.

**Revenue Start Date**

The earliest date for revenue recognized under the revenue schedule. Salesforce Billing calculates this date based off the revenue distribution method's Revenue Schedule Term Start Date field. For example, let's say the term end date has a value of Order Product Start Date. The revenue start date would be the start date of the order product that the revenue schedule is recording.

**Revenue Transaction Status**

Shows the status of the process that Salesforce Billing runs to create revenue transactions for the revenue schedule.

- Complete: The revenue transactions have been successfully created.
- Queued: The Apex job to create the revenue transactions has been triggered and is running or will run soon.
- Error: Salesforce Billing encountered an error creating the revenue transactions and recorded information on it in a revenue schedule error log. After you've fixed the errors, set the revenue transaction status to Queued to trigger the revenue transaction creation job.

### Total Revenue Amount

The total amount available when the revenue schedule is created. If you set up Salesforce Billing to automatically create revenue transactions, the schedule automatically distributes its available amount to revenue transactions. This value doesn't change based on adjustments.

### Unrecognized

The sum of this revenue schedule's revenue transactions that haven't been recognized and distributed to open finance periods.

## Revenue Adjustment Use Cases

Revenue adjustments let you adjust or transfer balances between any number of revenue schedules. To learn how to set up these adjustments, review common use cases. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

#### [Adjust a Revenue Schedule](#)

Adjust the balance of available revenue on one revenue schedule. (Salesforce Billing Managed Package)

#### [Adjust Multiple Revenue Schedules](#)

Use a revenue adjustment to adjust the revenue of multiple revenue schedules. (Salesforce Billing Managed Package)

#### [Transfer Revenue Between Two Revenue Schedules](#)

Transfer available revenue from one revenue schedule to another. (Salesforce Billing Managed Package)

#### [Transfer Revenue Between Several Revenue Schedules](#)

Transfer available revenue between four revenue schedules. (Salesforce Billing Managed Package)

### Adjust a Revenue Schedule

Adjust the balance of available revenue on one revenue schedule. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Winter '19 and later

In this example, we want to reduce the balance of the following revenue schedule by \$10.

- Name: RS-001
- Recognized: \$30
- Unrecognized: \$50
- Adjustments: \$0
- Available: \$10
- Deferred: \$60

1. From the revenue adjustment related list, click **New**.
2. Give your revenue adjustment the following field values.
  - Type: Adjustment
  - Status: Draft
3. Click **Save**.
4. From your revenue adjustment, go to the Revenue Adjustment Lines related list and click **New**.
5. Give your revenue adjustment line the following values.
  - Name: -\$10 Adjustment
  - Amount: -10
  - Target Revenue Schedule: RS-001
6. Click **Save**.
7. Go to your revenue adjustment and change its status to Posted.

After posting your revenue adjustment, you'll see the following changed values on your revenue schedule.

- Available: \$0
- Deferred: \$50

## Adjust Multiple Revenue Schedules

Use a revenue adjustment to adjust the revenue of multiple revenue schedules. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

Available in: Salesforce Billing Winter '19 and later

In this example, we have two revenue schedules. We want to reduce the first schedule's balance by \$10 and increase the second schedule's balance by \$100.

Name	Total Revenue	Recognized Revenue	Unrecognized Revenue	Adjustments	Available Revenue	Deferred Revenue	Revenue Transactions
RS-001	\$90	\$30	\$50	\$0	\$10	\$60	<ul style="list-style-type: none"> <li>• \$10 Recognized</li> <li>• \$20 Recognized</li> <li>• \$25 Deferred</li> <li>• \$25 Deferred</li> </ul>
RS-002	\$60	\$30	\$0	\$0	\$30	\$30	<ul style="list-style-type: none"> <li>• \$10 Recognized</li> <li>• \$10 Recognized</li> <li>• \$5 Recognized</li> <li>• \$5 Recognized</li> </ul>

Here, you'll use one revenue adjustment with two revenue adjustment lines.

1. From RS-001's Revenue Adjustment related list, click **New**.
  2. Give your revenue adjustment the following field values.
    - a. Type: Adjustment
    - b. Status: Draft
  3. Click **Save**.
  4. From your revenue adjustment, go to the Revenue Adjustment Lines related list and click **New**
  5. Give your revenue adjustment line the following values.
    - a. Name: -\$10 Adjustment
    - b. Amount: -10
    - c. Target Revenue Schedule: RS-001
  6. Click **Save**.
  7. From the same revenue adjustment, create another revenue adjustment line with the following values.
    - a. Name: \$100 Adjustment
    - b. Amount: 100
    - c. Target Revenue Schedule: RS-002
  8. Go to your revenue adjustment and change its status to Posted.
- After posting your revenue adjustment, you'll see the following changed values on your revenue schedules.

Name	Adjustments	Available Revenue	Deferred Revenue
RS-001	-\$10	\$0	\$50
RS-002	\$100	\$130	\$130

## Transfer Revenue Between Two Revenue Schedules

Transfer available revenue from one revenue schedule to another. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '19 and later

---

In this example, we want to transfer \$10 from our first revenue schedule to our second revenue schedule.

Name	Total Revenue	Recognized Revenue	Unrecognized Revenue	Adjustments	Available Revenue	Deferred Revenue	Revenue Transactions
RS-001	\$90	\$30	\$50	\$0	\$10	\$60	<ul style="list-style-type: none"> <li>• \$10 Recognized</li> <li>• \$20 Recognized</li> <li>• \$25 Deferred</li> <li>• \$25 Deferred</li> </ul>
RS-002	\$60	\$20	\$25	\$0	\$15	\$40	<ul style="list-style-type: none"> <li>• \$10 Recognized</li> <li>• \$10 Recognized</li> <li>• \$15 Deferred</li> <li>• \$10 Deferred</li> </ul>

Here you'll use one revenue adjustment with two revenue adjustment lines. Remember, the sum of the line's balances must equal zero.

1. From RS-001's Revenue Adjustment related list, click **New**.
2. Give your revenue adjustment the following field values.
  - a. Type: Transfer
  - b. Status: Draft
3. Click **Save**.

4. Give your revenue adjustment line the following values.
  - a. Name: -\$10 Transfer
  - b. Amount: -10
  - c. Target Revenue Schedule: RS-001
5. Click **Save**.
6. From the same revenue adjustment, create another revenue adjustment line with the following values.
  - a. Name: \$10 Transfer
  - b. Amount: 10
  - c. Target Revenue Schedule: RS-002
7. Click **Save**
8. Go to your revenue adjustment and change its status to Posted.

After posting your revenue adjustment, you'll see the following changed values on your revenue schedules.

Name	Adjustments	Available Revenue	Deferred Revenue
RS-001	-\$10	\$0	\$50
RS-002	\$10	\$25	\$50

## Transfer Revenue Between Several Revenue Schedules

Transfer available revenue between four revenue schedules. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in: Salesforce Billing Winter '19 and later

In this example, we have four revenue schedules. We want to transfer \$10 from the first schedule and \$10 from the second schedule to the third and fourth schedules.

Name	Total Revenue	Recognized Revenue	Unrecognized Revenue	Adjustments	Available Revenue	Deferred Revenue	Revenue Transactions
RS-001	\$90	\$30	\$50	\$0	\$10	\$60	<ul style="list-style-type: none"> <li>• \$10 Recognized</li> <li>• \$20 Recognized</li> <li>• \$25 Deferred</li> <li>• \$25 Deferred</li> </ul>
RS-002	\$60	\$20	\$25	\$0	\$15	\$40	<ul style="list-style-type: none"> <li>• \$10</li> </ul>

Name	Total Revenue	Recognized Revenue	Unrecognized Revenue	Adjustments	Available Revenue	Deferred Revenue	Revenue Transactions
							Recognized <ul style="list-style-type: none"> <li>• \$10 Recognized</li> <li>• \$15 Deferred</li> <li>• \$10 Deferred</li> </ul>
RS-003	\$75	\$30	\$15	\$0	\$30	\$45	<ul style="list-style-type: none"> <li>• \$15 Recognized</li> <li>• \$15 Recognized</li> <li>• \$10 Deferred</li> <li>• \$5 Deferred</li> </ul>
RS-004	\$50	\$30	\$20	\$0	\$0	\$20	<ul style="list-style-type: none"> <li>• \$20 Recognized</li> <li>• \$10 Recognized</li> <li>• \$10 Deferred</li> <li>• \$10 Deferred</li> </ul>

Here you'll use one revenue adjustment with four revenue adjustment lines. Remember, the sum of the balances among all lines must equal zero.

1. From RS-001's Revenue Adjustment related list, click **New**.
2. Give your revenue adjustment the following field values.
  - a. Type: Adjustment
  - b. Status: Draft
3. Click **Save**.
4. Give your revenue adjustment line the following values.
  - a. Name: -\$10 Transfer
  - b. Amount: -10
  - c. Target Revenue Schedule: RS-001
5. From the same revenue adjustment, create another revenue adjustment line with the following values.
  - a. Name: -\$10 Transfer
  - b. Amount: -10
  - c. Target Revenue Schedule: RS-002
6. Create another revenue adjustment line.
  - a. Name: \$5 Transfer

- b. Amount: 5
  - c. Target Revenue Schedule: RS-003
7. Create another revenue adjustment line.
- a. Name: \$15 Transfer
  - b. Amount: 15
  - c. Target Revenue Schedule: RS-004
8. Go to your revenue adjustment and change its status to Posted.
- After posting your revenue adjustment, you'll see the following changed values on your revenue schedules.

Name	Adjustments	Available Revenue	Deferred Revenue
RS-001	-\$10	\$0	\$50
RS-002	-\$10	\$5	\$30
RS-003	\$5	\$35	\$50
RS-004	\$15	\$15	\$35

## Closing and Reopening Finance Periods

---

Finance departments close accounting and revenue finance periods at the end of each accounting period. After a period closes, the finance department can prepare statements and users can review their financial status. Finance periods need to close at least once a year, though most businesses close them monthly for easier bank statement reconciliation, sales tax report submission, and paying and sending out invoices. Salesforce Billing allows for convenient creation, management, and closing of your business's finance periods. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

#### Close Finance Periods

Closing the finance periods makes it easier to reconcile bank statements, submit sales tax reports, pay invoices, and send out invoices. (Salesforce Billing Managed Package)

#### Reopen a Finance Period

When an order requires an amendment in a closed finance period, reopen the period to make your changes. (Salesforce Billing Managed Package)

#### Managing Finance Period Errors

If Salesforce Billing encounters validation errors while closing or reopening a finance period, we show error logs with detailed information on why each error happened. You can access the logs on the finance period and on each of its revenue transactions where an occurred. Review the logs to determine what corrective action to take to close or reopen your finance period. (Salesforce Billing Managed Package)

## Close Finance Periods

Closing the finance periods makes it easier to reconcile bank statements, submit sales tax reports, pay invoices, and send out invoices. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

The finance department has completed all the updates required in Salesforce, and has reviewed all the numbers for this month's finance period. They have requested to close the finance period.

 **Warning** It is recommended to close the finance books in sequential order, starting from the oldest. Failure to do so can result in revenue for a transaction being recognized in older open months, skipping closed months, and then continuing with newer open months.

1. Navigate to the Finance Period record that is to be closed.
2. Change Period Status to **Closed**, and then click **Save**.

Any revenue transactions from an order that falls in the timeframe of a finance period that has already been closed will begin in the next available finance period that is still open.

The Next Open Period field becomes unchecked on this finance period record. The checkmark will be automatically added to the next sequential finance period record that is still open. The revenue transactions on the finance period related list will have their status changed from **Deferred** to **Recognized**.

## Reopen a Finance Period

When an order requires an amendment in a closed finance period, reopen the period to make your changes. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

To reopen finance periods, users need Read, Create, Edit, Delete, Modify All Records, and View All Records permissions for all billing objects.

To ensure Salesforce Billing records transactions correctly following an amendment, you'll have to open any periods closed after the period that contained the amended transaction. For example, let's say you've closed all periods before October. During October, you find that an order closed in August requires amending. To correctly amend this transaction, you have to reopen August and September finance periods to ensure that Salesforce Billing calculates revenue for both months as well.

When you reopen a period, we recommend informing financial departments to hold off on making transactions until you've closed that period. This way, your reopened period won't include accidental records like backdated transactions, canceled transactions, or subscriptions with amended dates that align to the reopened period. Each of these records could incorrectly add or remove transactions from the reopened period, which would have a financial impact on your business.

1. Find the finance period that you want to reopen.
2. Change its Period Status field to Open and save.

Salesforce Billing selects the period's Next Open Period field. The finance period's revenue transactions change their statuses from Recognized to Deferred.

## Managing Finance Period Errors

If Salesforce Billing encounters validation errors while closing or reopening a finance period, we show error logs with detailed information on why each error happened. You can access the logs on the finance period and on each of its revenue transactions where an occurred. Review the logs to determine what corrective action to take to close or reopen your finance period. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '20 and later

---

When a user or process closes or reopens a finance period, Salesforce Billing runs validations to ensure the finance period and its revenue transactions are properly configured. During this process, the finance period's Period Status field changes to Pending Closed or Pending Open. If the finance period or any of its revenue transactions encounter validation errors, Salesforce Billing sets the finance period's status to Error. It then creates one error log for each error. You can find each log in the Error Logs related list on the finance period and revenue transaction objects.

 **Tip** Add the Status field to the fields that show on the Revenue Transactions related list on your finance period page layout. That way, you can quickly find which transaction wasn't able to open or close, and then view its Error Log.

Let's review some important error log fields.

#### Error Origin

The Salesforce Billing service that encountered the error

#### Full Error Log

A detailed description of the error

#### ReferenceId

The ID of the record where the error occurred

## Summary

A summarized description of the error, and steps for correction

When you've finished reviewing your error logs and made the appropriate corrections, change your finance period's status from Error to your original attempted value.



## Example



# Reporting Essentials

---

Salesforce Billing provides a collection of customizable options for reporting on stages within the billing process. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

### Managing Invoice Debt with Reports

Account balance snapshots let you view the total amount of invoice debt for an account, organized by periods of time past invoice due date. You can automate Salesforce Billing to create these snapshots on a recurring basis. This feature is useful for reporting on bad debt and ensuring your customers collect all their owed revenue. (Salesforce Billing Managed Package)

### Currency Reporting in Salesforce Billing

To accurately report on currencies in Salesforce Billing, add several Salesforce currency fields to key Salesforce Billing objects. (Salesforce Billing Managed Package)

## Managing Invoice Debt with Reports

Account balance snapshots let you view the total amount of invoice debt for an account, organized by periods of time past invoice due date. You can automate Salesforce Billing to create these snapshots on a recurring basis. This feature is useful for reporting on bad debt and ensuring your customers collect all their owed revenue. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

The Account Balance Snapshot record displays the total amount of invoice debt in your account at a given date. It contains up to 10 Aging Buckets, each of which contain an invoice debt balance from the total number of invoices in a timeframe. For example, one aging bucket could show the total invoice debt for invoices that are 0–30 days late. Another bucket on the same snapshot could show debt for invoices that are 31–60 days late.

To gather information on past-due invoices, Salesforce Billing admins create a balance snapshot scheduler. On a one-time, daily, weekly, or monthly basis, the scheduler performs a run that evaluates all your account's invoices for past-due balances. The scheduler then creates an Account Balance Snapshot record and assigns balance debt information to the aging buckets in your snapshot.

-  **Note** Salesforce Billing uses the balance snapshot scheduler to automate creation and field population of Account Balance Snapshot records. Admins cannot manually create an account balance snapshot and then assign it a balance snapshot scheduler.

The balance snapshot run also evaluates the total of the following records across your account.

- Posted and unapplied payments
- Posted and unapplied credit notes
- Posted and unapplied debit notes

You can view this information on your account balance snapshot, as well the names of each record. Finally, the snapshot's Account Balance field displays your overall invoice debt against the sum of such payments, credit notes, and debit notes.

### Schedule a Recurring Balance Snapshot Run

Use the Balance Snapshot Scheduler to create a Balance Snapshot Run that analyzes your account's invoices on a scheduled basis for past-due balances. Recurring snapshot runs ensure that you always have an accurate view of past-due invoices for an account. (Salesforce Billing Managed Package)

### Schedule a One-Time Balance Snapshot Run

Use the Balance Snapshot Scheduler to create a one-time Balance Snapshot Run that analyzes your account's invoices for past-due balances. A one-time snapshot is useful if you do not need recurring updates for your snapshots. (Salesforce Billing Managed Package)

### Guidelines for Account Balance Snapshots

To get the most from Account Balance Snapshots, review a few key guidelines. (Salesforce Billing Managed Package)

### Account Balance Snapshot Fields

The account balance snapshot shows a list of invoice debt in an account, organized into buckets based on the number of days past the invoice's due date. It also displays the amount of unapplied credit notes and amount of unapplied payments in the same account. (Salesforce Billing Managed Package)

## Schedule a Recurring Balance Snapshot Run

Use the Balance Snapshot Scheduler to create a Balance Snapshot Run that analyzes your account's invoices on a scheduled basis for past-due balances. Recurring snapshot runs ensure that you always have an accurate view of past-due invoices for an account. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

1. From the Balance Snapshot Schedulers object, click **New**.
2. Provide a name for your balance snapshot scheduler.
3. Choose Daily, Weekly, or Monthly for your scheduler's Type field.
4. Provide a start date and time for your scheduler to launch its first run, then save your changes.
5. Click **Save**.

## Schedule a One-Time Balance Snapshot Run

Use the Balance Snapshot Scheduler to create a one-time Balance Snapshot Run that analyzes your account's invoices for past-due balances. A one-time snapshot is useful if you do not need recurring updates for your snapshots. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

1. From the Balance Snapshot Schedulers object, click **New**.
2. Provide a name for your balance snapshot scheduler.
3. Choose One-Time for your scheduler's Type field.
4. You can optionally provide a start date and time for your scheduler to launch its single run. If you leave this field blank, the scheduler launches the run when you save the scheduler record.
5. Click **Save**.

## Guidelines for Account Balance Snapshots

To get the most from Account Balance Snapshots, review a few key guidelines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

- Account Balance Snapshots record transaction records differently based on when the account balance run occurs relative to when the transaction record was created. When a balance snapshot run evaluates a transaction record, it includes only records that were created before the run launched. For example, let's say your account balance run launches on February 05, 2017 at 1:00 P.M. After the run finishes, you create an unapplied payment record for \$500 and then launch another run. In this case, your first account balance snapshot doesn't contain data from your \$500 payment, while your second balance snapshot does.
- If you're using multicurrency, Account Balance Snapshots pick up transaction records only if they match the currency on the balance snapshot scheduler that created the balance snapshot run. To pick up all records, create balance snapshot schedulers for each of your active currencies.
- The balance snapshot run picks up invoices only if their due dates aren't null. If you're using custom payment terms, see [Invoice Dates](#) for information about setting up custom due dates.

- The balance snapshot run picks up transaction records only if their status is Posted.

## Account Balance Snapshot Fields

The account balance snapshot shows a list of invoice debt in an account, organized into buckets based on the number of days past the invoice's due date. It also displays the amount of unapplied credit notes and amount of unapplied payments in the same account. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Spring '18 and later

---

#### Account

Salesforce Billing evaluates this account's invoices. This field is automatically populated when you click **New** in your account's Account Balance Snapshot related list.

#### Account Balance

This amount of unpaid balances across all the account's invoices. Salesforce Billing calculates this value using the following formula: `Sum of Invoice Balances - Amount of Account's Unapplied Payments - Amount of Account's Unapplied Credit Notes - Amount of Account's Unapplied Debit Notes`

#### Account Balance Snapshot Name

Admins can provide a name to help easily identify this account balance snapshot.

#### Balance Snapshot Run (Created By)

The balance snapshot run that created this account balance snapshot.

#### Invoices

A list of the invoices that the balance snapshot run evaluated to create this account balance snapshot, shown by record name.

#### Invoice Aging Bucket (1 - 10)

An aging bucket represents a range of dates beyond an invoice's due date. For example, one bucket could represent 1 through 30 days past due, while another bucket represents 31 through 60 days past due. Users can define up to 10 aging buckets per org. When your org performs a balance snapshot run, it groups invoice debt into aging buckets based on the debt's past-due date. If your account had three invoices that were 45 days past due, their debt would be added up and shown in your "31 through 60" aging bucket.

By default, Salesforce Billing contains the following aging buckets. Users can edit the buckets, delete them, and add up to four more.

- Current
- 1–30 Days Past Due

- 31–60 Days Past Due
- 61–90 Days Past Due
- 91–120 Days Past Due
- 121+ Days Past Due

To customize aging bucket values, update records on the Aging Buckets object. Edit the Name, Starting Days from Past Due, and Ending Days from Past Due fields to show the values you want.

 **Note** Use caution as you're editing the Starting Days from Past Due and Ending Days from Past Due fields. No validations exist to enforce gaps or overlaps on these fields.

### Snapshot Date

The day that the balance snapshot run created this account balance snapshot.

### Total Invoice Amount

The sum of all invoice balances on this account, including debit notes.

This field is different from the Total Invoice Amount field that's described in your contractual terms for Salesforce Billing.

### Unapplied Credit Note Amount

The sum of all unapplied credit note amounts on this account.

### Unapplied Credit Notes

A list of the unapplied credit notes that the balance snapshot run evaluated, shown by record name.

### Unapplied Payment Amount

The sum of all unapplied payments in this account.

### Unapplied Payments

A list of the unapplied payments that the balance snapshot run evaluated, shown by record name.

## Currency Reporting in Salesforce Billing

To accurately report on currencies in Salesforce Billing, add several Salesforce currency fields to key Salesforce Billing objects. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: All Salesforce Billing Editions

---

Salesforce Billing contains the following fields frequently used in currency reports. By default, these fields

have no default value and no functionality. We've provided them so that you can define base currencies on transactional records manually or through an integration. We recommend using a scheduled batch process that evaluates an internal or external currency table and populates these fields based on currency rates that match the date of the transaction.

 **Important** To optimize performance and scale, we advise that users avoid populating these fields using triggers or process builders.

Field	Type	Setup
Base Currency	Text	<ul style="list-style-type: none"> <li>Default = Null</li> <li>User-populated</li> <li>Not required</li> </ul>
Base Currency FX Date	Date	<ul style="list-style-type: none"> <li>Default = Null</li> <li>User-populated</li> <li>Not required</li> </ul>
Base Currency FX Rate	Decimal	<ul style="list-style-type: none"> <li>Default = Null</li> <li>User-populated</li> <li>Not required</li> </ul>
Base Currency Amount	Currency	<ul style="list-style-type: none"> <li>Default = Null</li> <li>User-populated</li> <li>Not required</li> </ul>

To enable detailed currency reporting in Salesforce Billing, add these fields to the page layouts of the following objects.

- Invoice
- Invoice Line
- Usage
- Usage Summary
- Payment
- Refund
- Refund Line
- Revenue Schedule
- Revenue Transaction
- Credit Note
- Credit Note Line
- Debit Note Line

# Finance Logging

---

Finance transactions show details about a financial action performed against one of your financial records. Finance balance snapshots show details on the state of a header-level financial record's financially significant values following an action. Use Salesforce and Tableau reports to track important financial information and address aging or bad debt circumstances across all the records affected by a financial action. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

-  **Note** You can't delete finance transactions and finance balance snapshot records, but they don't count towards your storage limits.

### [Get Started with Finance Logging](#)

Finance transactions show details about any type of financial action performed in your Salesforce org. Finance balance snapshots show the state of a header-level financial object's financially significant values following an action. Use Salesforce and Tableau reports to track important financial information and address aging or bad debt circumstances across all the records affected by a financial action. (Salesforce Billing Managed Package)

### [Guidelines for Finance Logging](#)

When you're getting ready to work with finance transactions, consider important guidelines. (Salesforce Billing Managed Package)

### [Permissions for Finance Logging](#)

Remember to enable Salesforce profile permissions for users working with finance logging. (Salesforce Billing Managed Package)

### [Create an As Of Date Balance Report with Finance Logging and Tableau](#)

Create an As Of Date balance report in Tableau to allow finance users to review balances of top-level financial records by a date set by the user. The Tableau report uses finance balance snapshot data exported as a spreadsheet from Salesforce Billing. (Salesforce Billing Managed Package)

### [Finance Transaction Fields](#)

Finance transactions allow customers to view the state of balances, totals, and other important financial data on an entity in response to a financial action. These records can't be deleted. (Salesforce Billing Managed Package)

### [Finance Balance Snapshot Fields](#)

Finance balance snapshots inherit the financial details of finance transactions related to header-level billing objects. You can then use Salesforce reports to track important financial information and address aging or bad debt circumstances across all the records affected by a financial action. These records can't be deleted. This object is available in API version 49.0 and later. (Salesforce Billing Managed Package)

## Get Started with Finance Logging

Finance transactions show details about any type of financial action performed in your Salesforce org. Finance balance snapshots show the state of a header-level financial object's financially significant values following an action. Use Salesforce and Tableau reports to track important financial information and address aging or bad debt circumstances across all the records affected by a financial action. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

A finance transaction contains a collection of fields to describe the state of a record following a financial action. Financial actions include actions such as posting an invoice or allocating a payment to an invoice line. The transaction's fields track important financial information such as amounts and the resulting balance following the action. We recommend creating one finance transaction for each record created or changed as a result of the action. For example, let's say you posted an invoice with two invoice lines. You can create one finance transaction for the invoice, one for the first invoice line, and one for the second invoice line.

The record that a financial transaction tracks is called the reference entity. When you create a finance transaction, you update the transaction's subtotal, tax amount, and total amount with tax fields to show values from the reference entity's same fields. The transaction's Charge Amount, Impact Amount, and Resulting Balance fields let you show the results of the financial action.



In this sample, a vendor created a finance transaction to represent the posting of an invoice. The invoice has a total balance of \$500 with 6% tax applied. The invoiced product also contained a 10% pretax service charge, which the user has noted in the Charge Amount field.

Later, a customer makes a \$300 payment against the invoice header. The vendor creates another finance transaction to represent the invoice's state following the payment.



When you save a finance transaction related to an invoice, payment, refund, credit memo, or debit memo, Salesforce Billing creates a finance balance snapshot. The snapshot represents the balances of a header-level financial record following a financial action. Most of its fields inherit their values from the same fields on the parent finance transaction. The snapshot's Balance field inherits its value from the finance transaction's Resulting Balance field.

## Reporting on Finance Balance Snapshots

Finance balance snapshots provide a convenient way to track changes to header-level financial objects over time. We recommend running reports to view groups of finance balance snapshots filtered by values

such as account, reference entity, and created date. This process allows you to track the changes made to important financial records over their lifecycle. For example, let's say the vendor provided a \$50 credit to the customer's invoice. You want to track all the changes made to your invoice in your billing period of 08/10/20 through 09/09 20, so you run a report. The report lets you view history of your invoice following each financial action in the billing period.



**Tip** Finance transactions provide an easy source for exporting transaction data into an external ERP. We recommend using Salesforce's [Change Data Capture](#) services to create events and synchronize updates between your finance transactions and your ERP.



**Example** On 03/01/20, you post an invoice that contains a \$77 invoice line and a \$33 invoice line. In this case, you make one transaction to represent the posted invoice, and one transaction for each of the posted invoice lines.

Finance Transactions: Post Invoice

Transaction Name	Reference Entity	Reference Entity Type	Subtotal	Total Amount with Tax	Resulting Balance	Event Type	Event Action
FT-03	IL-02	Invoice Line	\$30.00	\$33.00	\$33.00	Posted	Post an Invoice
FT-02	IL-01	Invoice Line	\$70.00	\$77.00	\$77.00	Posted	Post an Invoice
FT-01	INV-01	Invoice	\$100.00	\$110.00	\$110.00	Posted	Post an Invoice

When you save FT-01, Salesforce Billing creates a finance balance snapshot for your invoice.

Balance Snapshot: Post Invoice

Snapshot Name	Transaction Name	Balance	Reference Entity	Reference Entity Type	Subtotal	Total Amount with Tax	Event Type
FBS-01	FT-01	\$110	INV-01	Invoice	\$100	\$110	Posted

On 03/10/20, a customer makes a payment of \$77.00 against the first invoice line. In this case, we add four transactions to represent changes following the action of paying the invoice:

- FT-04 represents the reduction of the payment's balance.
- FT-05 represents allocating the payment to the invoice line.
- FT-06 represents the reduction of the invoice line's balance.
- FT-07 represents the reduction of the invoice's balance.

## Finance Transactions: Pay Invoice Line

Transaction Name	Reference Entity Type	Source Entity	Destination Entity	Subtotal	Total Amount with Tax	Impact Amount	Resulting Balance	Event Type	Event Action
FT-07	Invoice	null	null	\$100	\$110	\$110	\$33.00	Allocated	Pay Invoice Line
FT-06	Invoice Line	null	null	\$70.00	\$77	\$77	\$0	Allocated	Pay Invoice Line
FT-05	Payment Invoice Line Application	P-01	IL-01	null	\$77	-\$77	null	Allocated	Pay Invoice Line
FT-04	Payment	null	null	\$100	\$110	\$110	\$110	Allocated	Pay Invoice Line

When you save FT-04 and FT-07, Salesforce Billing creates two more snapshots. Notice that we can now compare the state of the invoice after posting (FBS-01) and the state of the invoice after payment application (FBS-03).

## Balance Snapshots: Pay Invoice Line

Snapshot Name	Transaction Name	Balance	Reference Entity	Reference Entity Type	Subtotal	Total Amount with Tax	Impact Amount	Event Type
FBS-03	FT-07	\$33	INV-01	Invoice	\$100	\$10	\$110	Allocated
FBS-02	FT-04	\$0	P-01	Payment	null	\$77	-\$77	Allocated
FBS-01	FT-01	\$110	INV-01	Invoice	\$100	\$10	\$110	Posted

Next, your customer issues a credit of \$22 against your second invoice line, reducing its balance to \$11. For this action, you need five new finance transactions to represent the action of crediting the invoice line: Three entities represent the newly created credit memo, credit memo line, and credit memo invoice line application respectively. The other two represent our original invoice and \$33 invoice line following the credit application.

- FT-08 represents the creation of the credit memo.
- FT-09 represents the creation of the credit memo line.

- FT-10 represents the application of the credit memo to the invoice line.
- FT-11 represents the reduction of the invoice line's balance.
- FT-12 represents the reduction of the invoice's balance.

Finance Transactions: Credit Invoice Line

Transaction Name	Reference Entity	Reference Entity Type	Source Entity	Destination Entity	Subtotal	Total Amount with Tax	Impact Amount	Resulting Balance	Event Type	Event Action
FT-12	INV-01	Invoice	null	null	\$100	\$110	\$110	\$11	Allocated	Credit Invoice Line
FT-11	IL-02	Invoice Line	null	null	\$30	\$33	\$33	\$11	Allocated	Credit Invoice Line
FT-10	CMLA-01	Credit Memo Invoice Line Application	CM-01	IL-02	null	\$22	-\$22	\$0	Allocated	Credit Invoice Line
FT-9	CML-01	Credit Memo Line	null	null	\$20	\$22	-\$22	\$0	Allocated	Credit Invoice Line
FT-8	CM-01	Credit Memo	null	null	\$20	\$22	-\$22	\$0	Allocated	Credit Invoice Line

When you save FT-08 and FT-12, Salesforce Billing creates a snapshot to represent the posted credit memo (FBS-04). It also creates a snapshot to represent the invoice following the credit memo application. Viewing all five balance snapshots together lets us understand the financial changes made to our invoice across its lifecycle, and the objects that contributed to those changes.

Balance Snapshots: Credit Invoice Line

Snapshot Name	Transaction Name	Balance	Reference Entity	Reference Entity Type	Subtotal	Total Amount with Tax	Impact Amount	Event Type
FBS-05	FT-12	\$11	INV-01	Invoice	\$100	\$110	\$110	Allocated
FBS-04	FT-11	\$0	CM-01	Credit	\$20	\$22	-\$22	Allocated

Snapshot Name	Transaction Name	Balance	Reference Entity	Reference Entity Type	Subtotal	Total Amount with Tax	Impact Amount	Event Type
				Memo				
FBS-03	FT-07	\$33	INV-01	Invoice	\$100	\$110	\$110	Allocated
FBS-02	FT-04	\$0	P-01	Payment	null	\$77	-\$77	Allocated
FBS-01	FT-01	\$110	INV-01	Invoice	\$100	\$110	\$110	Posted

 **Note** You can't delete finance transactions and finance balance snapshot records, but they don't count towards your storage limits.

## Guidelines for Finance Logging

When you're getting ready to work with finance transactions, consider important guidelines. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

- Finance transactions and finance balance snapshots support custom fields with updatable values. You can change the values of custom fields at any time, but you can't change a finance transaction or finance balance snapshot's standard fields after you've created the record.
- Salesforce Billing supports sharing for finance transactions and finance balance snapshots.
- Salesforce Billing doesn't support changing the picklist values of standard finance transaction fields.
- Finance Transaction and Finance Balance Snapshot records don't count towards your storage usage limit.

## Permissions for Finance Logging

Remember to enable Salesforce profile permissions for users working with finance logging. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

 **Important** You can't update standard fields on a finance transaction or finance balance snapshot after you create the record.

Task	Finance Transaction	Finance Balance Snapshot
Create finance transactions	Create	None
View finance transactions	Read	Read
View finance balance snapshots	Read	Read
Update custom finance transaction fields	Edit	None
Update custom finance balance snapshot fields	None	Edit

## Create an As Of Date Balance Report with Finance Logging and Tableau

Create an As Of Date balance report in Tableau to allow finance users to review balances of top-level financial records by a date set by the user. The Tableau report uses finance balance snapshot data exported as a spreadsheet from Salesforce Billing. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Winter '21 and later

---

In this example, you have already exported your finance balance snapshot data from Salesforce into an external spreadsheet file. You want to build a report that tracks Account, Reference Entity, Reference Entity Type, Currency ISO Code, Balance, Total Amount, and Transaction Date.

1. In Tableau, open a new workbook.
2. Bring in your finance balance snapshot data from an exported file.
  - a. Go to Data, and then select **New Data Source**.
  - b. Select your spreadsheet type and then open your file.
3. Add the dimensions and measures for your report.
  - a. Set Balance and Total Amount with Tax as discrete dimensions.
  - b. Set Transaction Date as an exact date and a discrete dimension.
4. Create an As of Date parameter that allows you to input a date value and return the latest record as of the entered date. Enter the following parameter values.
  - Name: Select Date
  - Data Type: Date & Time
  - Allowable Values: All
5. On your parameter, select **Show Parameter Control**.  
This option lets you input values for the parameter.
6. Create a calculated dimension field with logic that evaluates whether the record's transaction date falls before or after the As Of Date parameter. This setting allows you to filter for only records up to the As Of Date. The calculated dimension field has the following values.

- Name: Is On or Before As Of Date
  - [Transaction Date] <= [As Of Date]
7. Add your calculated dimension as a filter and set its value to True.
8. Create a calculated dimension field with logic that evaluates whether a record on your reference entity is the latest record relative to the As Of Date. The calculated dimension has the following values.
- Name: Is Latest Record
  - Logic:

```
if([Transaction Date]) = ({fixed[Reference Entity], [Is On or Before As Of Date]} : MAX([Transaction Date])) )
then
    true
else
    false
END
```

9. Add the Is Latest Record field as a filter and set its value to True.

You can now pass your As of Date input parameter and see the latest balance for a record of your reference entity as of that date. You can also filter out reference entities where the balance is already settled by filtering against balances of zero.

## Finance Transaction Fields

Finance transactions allow customers to view the state of balances, totals, and other important financial data on an entity in response to a financial action. These records can't be deleted. (Salesforce Billing Managed Package)

### Fields

Field	Definition
<b>AccountId</b>	Parent account of the reference entity.
<b>AdjustmentAmount</b>	The adjustment contributing to the transaction's subtotal.
<b>BaseCurrencyAmount</b>	Define this value based on any currency conversion needed for the total amount with tax.
<b>BaseCurrencyBalance</b>	Define this value based on any currency conversion needed for the resulting balance.
<b>BaseCurrencyFxDate</b>	Date used to determine the foreign exchange rate. Define this value to use later when you calculate currency conversions.
<b>BaseCurrencyFxRate</b>	Rate of exchange according to the Base Currency FX Date. Define this value to use later when you calculate currency conversions.

Field	Definition
<b>BaseCurrencyIsoCode</b>	Currency of the country defined for this account. Define this value to use later when you calculate currency conversions.
<b>ChargeAmount</b>	The charge contributing to the transaction's subtotal.
<b>CreationMode</b>	The process used to create the financial transactions. This field doesn't control any business logic.
<b>CreditGLAccount</b>	Reference to the Credit GL account associated with the finance transaction.
<b>DebitGLAccount</b>	Reference to the Debit GL account associated with the finance transaction.
<b>DestinationEntity</b>	Target entity for reference entities that have made allocations.
<b>DueDate</b>	Date when the invoice is due.
<b>EffectiveDate</b>	Date when the financial transaction takes effect.
<b>EventAction</b>	The financial action that caused the financial transaction.
<b>EventType</b>	The general type of action that caused a change to the transaction's reference entity.
<b>FinanceBook</b>	Reference to the finance book that the finance transaction is associated with.
<b>FinancePeriod</b>	Reference to the finance period that the finance transaction is associated with.
<b>FinanceSystemIntegration Mode</b>	The type of the integration to the external financial system. Possible examples include Invoice to ERP, Cash to ERP, or GL to ERP). This field is a Salesforce field separate from the Salesforce Billing package.
<b>FinanceSystemIntegration Status</b>	The status of the integration with the external financial system used to recognize the financial transaction. This field is a Salesforce field separate from the Salesforce Billing package.
<b>FinanceSystemName</b>	The name of the external financial system used to recognize the financial transaction. This field is a Salesforce field separate from the Salesforce Billing package.
<b>FinanceSystemTransaction Number</b>	The number of the transaction that's related to the external financial system. This field is a Salesforce field separate from the Salesforce Billing package.
<b>FinanceTransactionNumber</b>	Reference number assigned to the finance transaction.
<b>GLRule</b>	Reference to the GL rule used to determine the GL accounts for the

Field	Definition
	finance transaction.
<b>GLTreatment</b>	Reference to the GL treatment used to determine the GL accounts for the finance transaction.
<b>ImpactAmount</b>	The total change in a reference entity's amount as a result of the event action. This value is positive for positive changes and negative for negative changes.
<b>LegalEntityId</b>	Legal entity associated with the finance transaction. This field is an optional lookup that you can add to associate the finance transaction to GL rules for external ledger processing.
<b>Original Credit GL Account Name</b>	The finance transaction's credit GL account name when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original Credit GL Account Number</b>	The finance transaction's credit GL account number when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original Debit GL Account Name</b>	The finance transaction's debit GL account name when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original Debit GL Account Number</b>	The finance transaction's debit GL account number when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original Finance Period Name</b>	The finance transaction's finance period name when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original Finance Period Start Date</b>	The finance transaction's finance period start date when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original Finance Period End Date</b>	The finance transaction's finance period end date when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original Finance Period Status</b>	The finance transaction's finance period status when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original GL Rule Name</b>	The finance transaction's GL rule name when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.

Field	Definition
<b>Original GL Treatment Name</b>	The finance transaction's GL treatment name when the finance transaction was created. This field is a Salesforce field separate from the Salesforce Billing package.
<b>Original Finance Book Name</b>	The finance transaction's finance book name when the finance transaction was created.
<b>Original Reference Entity Type</b>	The finance transaction's reference entity type when the finance transaction was created.
<b>Original Type Action</b>	The finance transaction's event type when the finance transaction was created.
<b>Original Event Action</b>	The finance transaction's event action when the finance transaction was created.
<b>ParentReferenceEntity</b>	Parent entity of the reference entity, if any.
<b>ReferenceEntityType</b>	An object that was created or updated as a result of the event action. One event action can cause the creation of multiple objects. For example, if you allocate a payment to an invoice line, you can create two finance transactions: One represents the changes to the payment, and one to represent the changes to the invoice line.
<b>ReferenceEntity</b>	The financial object that was created or updated as a result of a customer action.
<b>ResultingBalance</b>	The balance of your reference entity as a result of the event action.
<b>SourceEntity</b>	Source entity for reference entities that have made allocations.
<b>Subtotal</b>	Total of the charge amount and adjustment. This field doesn't include the tax amount.
<b>TaxAmount</b>	The total amount of tax for the change made to your reference entity as a result of the event action.
<b>TotalAmountWithTax</b>	The subtotal plus the tax amount.
<b>TransactionDate</b>	The date when the finance transaction occurred.

## Finance Balance Snapshot Fields

Finance balance snapshots inherit the financial details of finance transactions related to header-level billing objects. You can then use Salesforce reports to track important financial information and address aging or bad debt circumstances across all the records affected by a financial action. These records can't be deleted. This object is available in API version 49.0 and later. (Salesforce Billing Managed Package)

## Fields

Field	Definition
<b>AccountId</b>	Parent account of the reference entity.
<b>AdjustmentAmount</b>	The adjustment contributing to the transaction's subtotal.
<b>Balance</b>	Final balance of the snapshot's reference entity. Inherited from the finance transaction's resulting balance if the finance transaction's reference entity is a credit memo, debit memo, invoice, payment, or refund.
<b>BaseCurrencyAmount</b>	Define this value based on any currency conversion needed for the total amount with tax.
<b>BaseCurrencyBalance</b>	Define this value based on any currency conversion needed for the resulting balance.
<b>BaseCurrencyFxDate</b>	Date used to determine the foreign exchange rate. Define this value to use later when you calculate currency conversions.
<b>BaseCurrencyFxRate</b>	Rate of exchange according to the Base Currency FX Date. Define this value to use later when you calculate currency conversions.
<b>BaseCurrencyIsoCode</b>	Currency of the country defined for this account. Define this value to use later when you calculate currency conversions.
<b>ChargeAmount</b>	The charge contributing to the transaction's subtotal. Inherited from the parent finance transaction's <b>ChargeAmount</b> field.
<b>CreditGLAccount</b>	Reference to the Credit GL account associated with the finance transaction.
<b>DebitGLAccount</b>	Reference to the Debit GL account associated with the finance transaction.
<b>DueDate</b>	Date when the invoice is due. Inherited from the parent finance transaction's <b>DueDate</b> field.
<b>EffectiveDate</b>	Date when the financial transaction takes effect. Inherited from the parent finance transaction's <b>EffectiveDate</b> field.
<b>EventType</b>	<p>The general type of action that caused a change to the transaction's reference entity. Inherited from the parent finance transaction's <b>EventType</b> field. Possible values are:</p> <ul style="list-style-type: none"> <li>• <b>Allocated</b></li> <li>• <b>Canceled</b></li> </ul>

Field	Definition
	<ul style="list-style-type: none"> <li>• <code>Posted</code></li> <li>• <code>Unallocated</code></li> <li>• <code>Void</code></li> </ul>
<code>FinanceBook</code>	Reference to the finance book that the finance transaction is associated with.
<code>FinancePeriod</code>	Reference to the finance period that the finance transaction is associated with.
<code>FinanceSystemIntegrationMode</code>	The type of the integration to the external financial system. Possible examples include Invoice to ERP, Cash to ERP, or GL to ERP). This field is a Salesforce field separate from the Salesforce Billing package.
<code>FinanceSystemIntegrationStatus</code>	The status of the integration with the external financial system used to recognize the financial transaction. This field is a Salesforce field separate from the Salesforce Billing package.
<code>FinanceSystemName</code>	The name of the external financial system used to recognize the financial transaction. This field is a Salesforce field separate from the Salesforce Billing package.
<code>FinanceSystemTransactionNumber</code>	The number of the transaction related to the external financial system. This field is a Salesforce field separate from the Salesforce Billing package.
<code>FinanceTransactionId</code>	Parent finance transaction for the snapshot. The snapshot's fields inherit the transaction's values only if the finance transaction's reference entity type is an invoice, payment, credit memo, debit memo, or refund.
<code>GLRule</code>	Reference to the GL rule used to determine the GL accounts for the finance transaction.
<code>GLTreatment</code>	Reference to the GL treatment used to determine the GL accounts for the finance transaction.
<code>ImpactAmount</code>	The finance transaction's impact on the customer's finances. Inherited from the parent finance transaction's <code>ImpactAmount</code> field.
<code>LegalEntityId</code>	Legal entity associated with the finance transaction. This field is an optional lookup that you can add to associate the finance transaction to GL rules for external ledger processing. Inherited from the parent finance transaction's <code>LegalEntityId</code> field.
<code>Name</code>	Name of the finance balance snapshot.
<code>ReferenceEntity</code>	The financial object that was created or updated as a result of a customer action.

Field	Definition
<b>ReferenceEntityType</b>	An object that was created or updated as a result of the event action. One event action can cause the creation of multiple objects. For example, if you allocate a payment to an invoice line, you can create two finance transactions: One represents the changes to the payment, and one to represent the changes to the invoice line.
<b>Subtotal</b>	Total of the charge amount and adjustment. Doesn't include the tax amount. Inherited from the parent finance transaction's <b>Subtotal</b> field.
<b>TaxAmount</b>	The total amount of tax for the change made to your reference entity as a result of the event action. Inherited from the parent finance transaction's <b>Tax Amount</b> field.
<b>TotalAmountWithTax</b>	The subtotal plus the tax amount. Inherited from the parent finance transaction's <b>TotalAmountWithTax</b> field.
<b>TransactionDate</b>	The date when the finance transaction occurred. Inherited from the parent finance transaction's <b>TransactionDate</b> field.

## Salesforce Billing Tableau Dashboards

---

Salesforce Billing provides a collection of Tableau dashboards that help you get business insights from your Salesforce Billing data. Each dashboard comes with a default layout that you can customize and extend based on your data tracking needs. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '20 and later

---

#### [Configure the Pending Billings Dashboard](#)

Connect the Tableau Pending Billings Dashboards to your Salesforce org. The dashboard helps you estimate your future billing amounts based on the pending billings amount for each of your activated order products. You can use this data to evaluate your expected future cash flow. The dashboards are configured for monthly billing by default, but you can also use them for quarterly or annual billing analysis. (Salesforce Billing Managed Package)

#### [Configure the Daily Pending Billings Dashboard](#)

Connect the Tableau Daily Pending Billings Dashboards to your Salesforce org. The dashboard helps you estimate your future daily billing amounts based on the pending billings amount for each of your activated order products. You can use this data to evaluate your expected future cash flow. (Salesforce Billing Managed Package)

#### [Configure the All Billings Dashboard](#)

Connect the Tableau All Billings Dashboard to your Salesforce org. The dashboard helps you review your overall billings and evaluate how much has been invoiced, debited, and credited, as well as your

pending billings. (Salesforce Billing Managed Package)

#### [Configure the Payments Dashboard](#)

Connect the Tableau Payments Dashboard to your Salesforce org. The dashboard helps you review processed payments and refunds issued to the payments. You can also review declined payments or payments with errors and take the necessary corrective actions. The dashboards organize your payment data by time period, account, payment method, and payment gateway. You can set a tolerance level for refunds, declines, and errors. This configuration lets you focus on areas where payment transactions are above acceptable levels. (Salesforce Billing Managed Package)

#### [Configure the Revenue Dashboards](#)

Connect the Tableau Revenue Dashboards to your Salesforce org. The Revenue Dashboards help you track your overall revenue recognition, including deferred, recognized, and aggregated previously earned revenue. You can also use it to evaluate how your outstanding deferred revenue will be recognized over time. You can view the data altogether or segmented by time, product, legal entity, finance book, and distribution method. (Salesforce Billing Managed Package)

## Configure the Pending Billings Dashboard

Connect the Tableau Pending Billings Dashboards to your Salesforce org. The dashboard helps you estimate your future billing amounts based on the pending billings amount for each of your activated order products. You can use this data to evaluate your expected future cash flow. The dashboards are configured for monthly billing by default, but you can also use them for quarterly or annual billing analysis. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '20 and later

To use the Pending Billing dashboard, download the [Month\\_ID.csv](#) file and have your Salesforce credentials ready so you can replace the demo data with your own order data. The Month\_ID.csv spreadsheet contains one column labeled Month ID, with rows for the numbers 0 through 120. When you configure the Pending Billing dashboard, Tableau performs a full outer join between Month\_ID.csv and your order data. This join creates a table that includes all pending billings within a 10-year period for each of your order products. If you need a smaller future outlook, edit Month\_ID.csv to contain fewer rows, such as 60 rows for a 5-year billing period.

1. From the [Tableau Downloads page](#), download the Salesforce Dashboard Starters package.
2. Open the Pending\_Billings.twbx file in Tableau Desktop.  
The table is populated with demo data. Replace the demo data with the data from your org in the next steps.
3. From your Tableau home page, go to the **Data** menu and select **Future Billings Standard**, and then select **Edit Data Source....**
4. Enter the credentials of the Salesforce org that you want your Pending Billings dashboard to evaluate. If you're asked to allow access, click **Allow**.
5. After Tableau has finished authenticating your org, it will display a window for selecting your data

source. Select Month\_ID.csv from the directory where you saved it.

Tableau performs an outer join between the Month\_ID.csv file and your order products with non-invoice plan billing frequencies.

6. From the Tableau dashboard, select the Billing Revenue by Month Overview tab.
7. From the Tableau toolbar, select **Data** and select **Future Billings Invoice Plan**, and then select **Edit Data Source....**

Configure the Future Billings Invoice Plan data source even if your org doesn't bill with invoice plans. Otherwise, the Tableau dashboard will still contain demo data.

8. Repeat steps 3, 4, and 5 to complete setting up your dashboard for order products with invoice plans. Tableau performs a full outer join between the Month\_ID.csv file and your order products with invoice plan billing frequencies. The Connections sidebar in Tableau shows connections for your Salesforce org and your Month\_ID.csv file.

The Pending Billings Dashboard provides a Billing Revenue by Month Overview tab and a Details tab. By default, the Billing Revenue by Month tab shows bar graphs for your overall Billings by Months. It also shows your billings by month organized by product, account, billing frequency, and legal entity. The Details page shows the same information as a spreadsheet. You can customize the filter fields and values in each tab as needed.

To reconnect to your Salesforce org data source or refresh your data, go to **Data** and click **Refresh all Extracts**.

## Configure the Daily Pending Billings Dashboard

Connect the Tableau Daily Pending Billings Dashboards to your Salesforce org. The dashboard helps you estimate your future daily billing amounts based on the pending billings amount for each of your activated order products. You can use this data to evaluate your expected future cash flow. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '20 and later

---

To use the Pending Billing dashboard, download the [Day\\_ID.csv](#) file and have your Salesforce credentials ready so you can replace the demo data with your own order data. The Day\_ID.csv spreadsheet contains one column labeled Day ID, with rows for the numbers 0 through 120. When you configure the Pending Billing dashboard, Tableau performs a full outer join between Day\_ID.csv and your order data. This join creates a table that includes all pending billings within a 120-day period for each of your order products. If you need a smaller or larger future outlook, edit the number of rows in the Day\_ID.csv file. Larger numbers of rows can cause slower performance after configuring your dashboard in Tableau.

1. From the [Tableau Downloads page](#), download the Salesforce Dashboard Starters package.
2. Open the Pending\_Billings\_Daily.twbx file in Tableau Desktop.  
The table is populated with demo data. Replace the demo data with the data from your org in the next steps.

3. From your Tableau home page, go to the **Data** menu and select **Future Billings Standard**, and then select **Edit Data Source....**
4. Enter the credentials of the Salesforce org that you want your Daily Pending Billings dashboard to evaluate. If you're asked to allow access, click **Allow**.
5. After Tableau has finished authenticating your org, it will display a window for selecting your data source. Select Day\_ID.csv from the directory where you saved it.  
Tableau performs an outer join between the Day\_ID.csv file and your order products with non-invoice plan billing frequencies.
6. Rename the Billing Revenue by Month Overview tab to *Billing Revenue by Day*.
7. From the Tableau dashboard, select the Billing Revenue by Day tab.
8. From the Tableau toolbar, select **Data** and select **Future Billings Invoice Plan**, and then select **Edit Data Source....**  
Configure the Future Billings Invoice Plan data source even if your org doesn't bill with invoice plans. Otherwise, the Tableau dashboard will still contain demo data.
9. Repeat steps 3, 4, and 5 to complete setting up your dashboard for order products with invoice plans.  
Tableau performs a full outer join between the Day\_ID.csv file and your order products with invoice plan billing frequencies. The Connections sidebar in Tableau shows connections for your Salesforce org and your Day\_ID.csv file.

The Pending Billings Dashboard provides a Billing Revenue by Day Overview tab and a Details tab. By default, the Billing Revenue by Day tab shows bar graphs for your overall Billings by Day. It also shows your billings by day organized by product, account, billing frequency, and legal entity. The Details page shows the same information as a spreadsheet. You can customize the filter fields and values in each tab as needed.

To reconnect to your Salesforce org data source or refresh your data, go to **Data** and click **Refresh all Extracts**.

## Configure the All Billings Dashboard

Connect the Tableau All Billings Dashboard to your Salesforce org. The dashboard helps you review your overall billings and evaluate how much has been invoiced, debited, and credited, as well as your pending billings. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '20 and later

---

To use the All Billings dashboard, download the [4\\_Rows.csv](#) file and have your Salesforce credentials ready so you can replace the demo data with your own order data. The 4\_Rows.csv spreadsheet contains one column labeled Copy, with rows for the numbers 1 through 4. When you configure the Pending Billing dashboard, Tableau performs a full outer join between 4\_Rows.csv and your order data. This join creates a table that includes your order product details by invoice amount, debit amount, credit amount, and pending amount.

1. From the [Tableau Downloads page](#), download the Salesforce Dashboard Starters package.
2. From the Salesforce Dashboard Starters package, open the All\_Billings.twbx file in Tableau Desktop.  
The table is populated with demo data. Replace the demo data with the data from your org in the next steps.
3. From your Tableau home page, go to the **Data** menu and select **Invoice, Credits, Debits, and Pending (Join X 4)**, and then select **Edit Data Source....**
4. Enter the credentials of the Salesforce org that you want your All Billings dashboard to evaluate. If you're asked to allow access, click **Allow**.
5. After Tableau has finished authenticating your org, it will display a window for selecting your data source. Select 4\_Rows.csv from the directory where you saved it.  
Tableau performs a full outer join between your 4\_Rows.csv file and the order products in your Salesforce org. The Data Source tab in Tableau shows connections for your Salesforce instance and your 4\_Rows.csv file.

The All Billings Dashboard provides an All Billings Overview tab and a Details tab. By default, the Billing Revenue by Month tab shows bar graphs for your total invoice, debit, credit, and pending amounts. It also includes your total amounts ordered by legal entity, product, and account. The Details page shows the same information as a spreadsheet. You can customize the filter fields and values in each tab as needed.

To reconnect to your Salesforce org data source or refresh your data, go to **Data** and click **Refresh all Extracts**.

## Configure the Payments Dashboard

Connect the Tableau Payments Dashboard to your Salesforce org. The dashboard helps you review processed payments and refunds issued to the payments. You can also review declined payments or payments with errors and take the necessary corrective actions. The dashboards organize your payment data by time period, account, payment method, and payment gateway. You can set a tolerance level for refunds, declines, and errors. This configuration lets you focus on areas where payment transactions are above acceptable levels. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '20 and later

---

To use the Payments dashboard, download the [4\\_Rows.csv](#) file and have your Salesforce credentials ready so you can replace the demo data with your own order data. The 4\_Rows.csv spreadsheet contains one column labeled Copy, with rows for the numbers 1 through 4. When you configure the Payments dashboard, Tableau performs a full outer join between 4\_Rows.csv and your payments data.

1. From the [Tableau Downloads page](#), download the Salesforce Dashboard Starters package.
2. Open the Payments.twbx file in Tableau Desktop.  
The table is populated with demo data. You'll replace the demo data with the data from your org in the next steps.

3. From your Tableau home page, go to the **Data** menu and select **Payments**, and then select **Edit Data Source....**
  4. Enter the credentials of the Salesforce org that you want your Payments dashboard to evaluate. If you're asked to allow access, click **Allow**.
  5. After Tableau has finished authenticating your org, it will display a window for selecting your data source. Select **4 Rows.csv** from the directory where you saved it.
  6. Change the connection name if needed.
- Tableau performs a full outer join between your **4\_Rows.csv** file and the order products in your Salesforce org. The Data Source tab in Tableau shows connections for your Salesforce instance and your **4\_Rows.csv** file.

The Payments Dashboard provides tabs for Payment Overview, Payment Details, Payment Transaction Overview, and Transaction Details.

To reconnect to your Salesforce org data source or refresh your data, go to **Data** and click **Refresh all Extracts**.

## Configure the Revenue Dashboards

Connect the Tableau Revenue Dashboards to your Salesforce org. The Revenue Dashboards help you track your overall revenue recognition, including deferred, recognized, and aggregated previously earned revenue. You can also use it to evaluate how your outstanding deferred revenue will be recognized over time. You can view the data altogether or segmented by time, product, legal entity, finance book, and distribution method. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in: Salesforce Billing Summer '20 and later

---

To use the Revenue Billings dashboards, download the [4\\_Rows.csv](#) file and have your Salesforce credentials ready so you can replace the demo data with your own order data. The **4\_Rows.csv** spreadsheet contains one column labeled **Copy**, with rows for the numbers 1 through 4. When you configure the Revenue dashboards, Tableau performs a full outer join between **4\_Rows.csv** and revenue data.

1. From the [Tableau Downloads page](#), download the Salesforce Dashboard Starters package.
2. Open the **Revenue.twbx** file in Tableau Desktop.  
The table is populated with demo data. Replace the demo data with the data from your org in the next steps.
3. From your Tableau home page, go to the **Data** menu and select **Revenue**, and then select **Edit Data Source....**
4. Enter the credentials of the Salesforce org that you want your Revenue dashboards to evaluate. If you're asked to allow access, click **Allow**.
5. After Tableau has finished authenticating your org, it will display a window for selecting your data source. Select **4 Rows.csv** from the directory where you saved it.

Tableau performs a full outer join between your 4\_Rows.csv file and the revenue schedules in your Salesforce org. The Data Source tab in Tableau shows connections for your Salesforce instance and your 4\_Rows.csv file.

The Revenue Dashboard provides a Revenue Recognition Overview tab, a Deferred Revenue Overview tab, and a Details tab. By default, the Revenue Recognition Overview table shows your recognized, deferred, and previously earned revenue organized by month. It also shows your total revenue amount organized by product, legal entity, finance book, and distribution method. The Deferred Revenue Overview tab shows your deferred revenue organized by month, product, legal entity, finance book, and distribution method. The Details page shows the same information as a spreadsheet. You can customize the filter fields and values in each tab as needed.

To reconnect to your Salesforce org data source or refresh your data, go to **Data** and click **Refresh all Extracts**.

## Understand What Your Customers Have Bought by Using Customer Asset Lifecycle Management

---

Customer Asset Lifecycle Management gives you visibility into products your customers have bought, from initial sale through the end date of a subscription or service. As an account, sales, or service rep, you see an asset's quantity, amount, and monthly recurring revenue at any point during an asset's lifecycle. You can also see related invoice lines, the source of a change, and other information. Your business consolidates purchases and changes in one system, making subscriptions and other complex products easier to manage, and showing trends in a dashboard and reports. Developers or integrators automate creation of, changes to, and cancellation of lifecycle-managed assets using objects and fields that enhance the Asset object. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

#### [View Changes to an Asset During Its Lifecycle](#)

View the current quantity, amount, and monthly recurring revenue (MRR) for an asset, and see trends throughout the asset's lifecycle. (Salesforce Billing Managed Package)

#### [How Lifecycle Management Works](#)

An asset without lifecycle management represents a product that a customer has bought, but the information stops at the time of sale. By contrast, a lifecycle-managed asset shows information about a product such as a subscription or a warranty after it's sold. Customer Asset Lifecycle Management is driven by a custom automated process. For each lifecycle-managed asset, you see information in a dashboard and on related pages in Salesforce. (Salesforce Billing Managed Package)

#### [Invoicing Considerations When Using Customer Asset Lifecycle Management](#)

The Asset field is empty by default. When Salesforce Billing invoices an order through an invoice run or

Bill Now, invoice lines inherit the value of the field. Review these considerations if you're entering a value for the Asset field. (Salesforce Billing Managed Package)

### [Lifecycle Management Objects and Fields in Salesforce Billing](#)

Some field values on asset lifecycle management objects are system populated, and some are API populated. You can't customize asset lifecycle management fields or add custom fields to asset lifecycle management objects. (Salesforce Billing Managed Package)

### [Administer Customer Asset Lifecycle Management](#)

Your developer or integrator uses the Connect REST API to automate creation of, changes to, and cancellation of lifecycle-managed assets. The system uses data from Salesforce or from third-party ecommerce, quoting, order management, and other applications. You grant permissions, set up page layouts, and customize the views of lifecycle-managed assets to support your business processes for account, sales, and service reps. (Salesforce Billing Managed Package)

#### See Also

[Connect REST API Developer Guide: Customer Asset Lifecycle Management](#)

## View Changes to an Asset During Its Lifecycle

View the current quantity, amount, and monthly recurring revenue (MRR) for an asset, and see trends throughout the asset's lifecycle. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

#### USER PERMISSIONS NEEDED

To view the dashboard and details for lifecycle-managed assets:      Read on the Asset and Asset Action objects

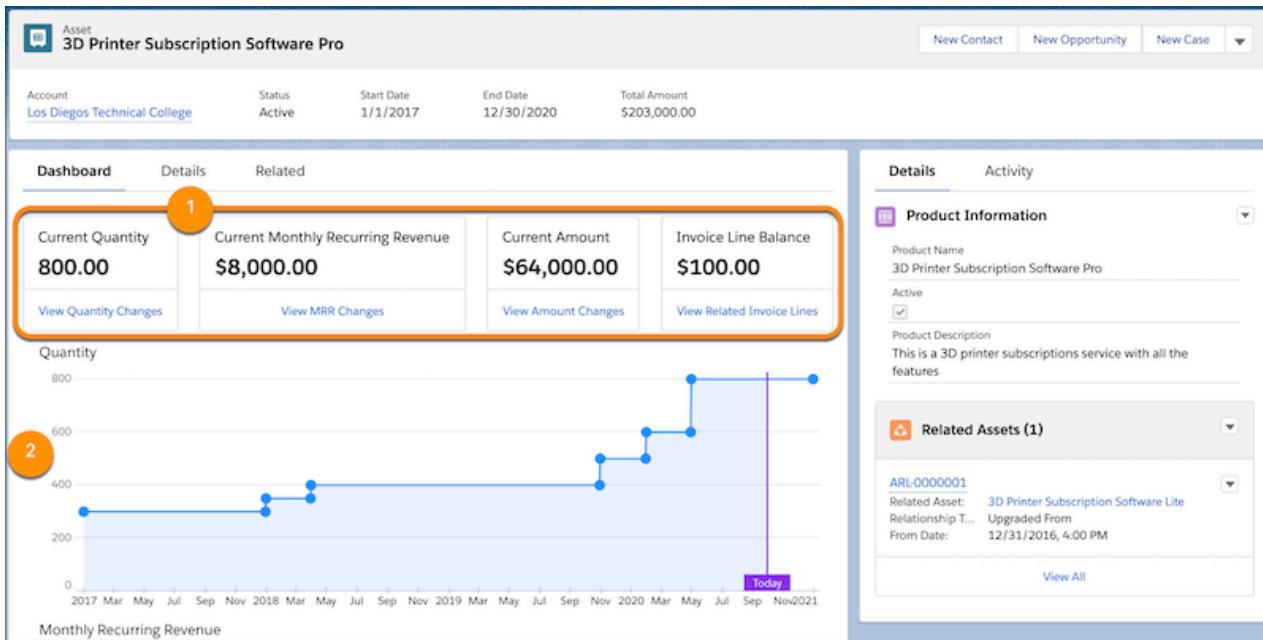
---

1. Navigate to a lifecycle-managed asset to see its dashboard.

If you don't see the dashboard, ask your Salesforce admin about Customer Asset Lifecycle Management.

The dashboard for the asset shows the current quantity, MRR, amount, and invoice line balance as of today's date (1). It's updated at the end of the current period in the asset's lifecycle.

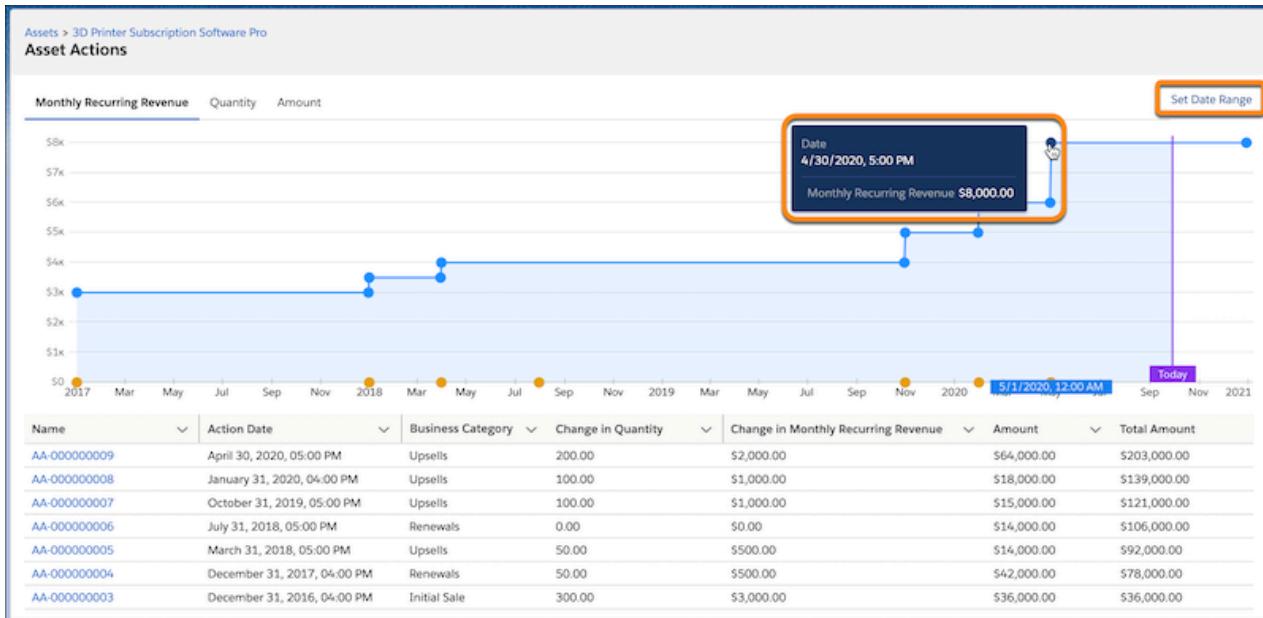
The dashboard also includes charts showing changes in quantity and MRR throughout the asset's lifecycle (2).

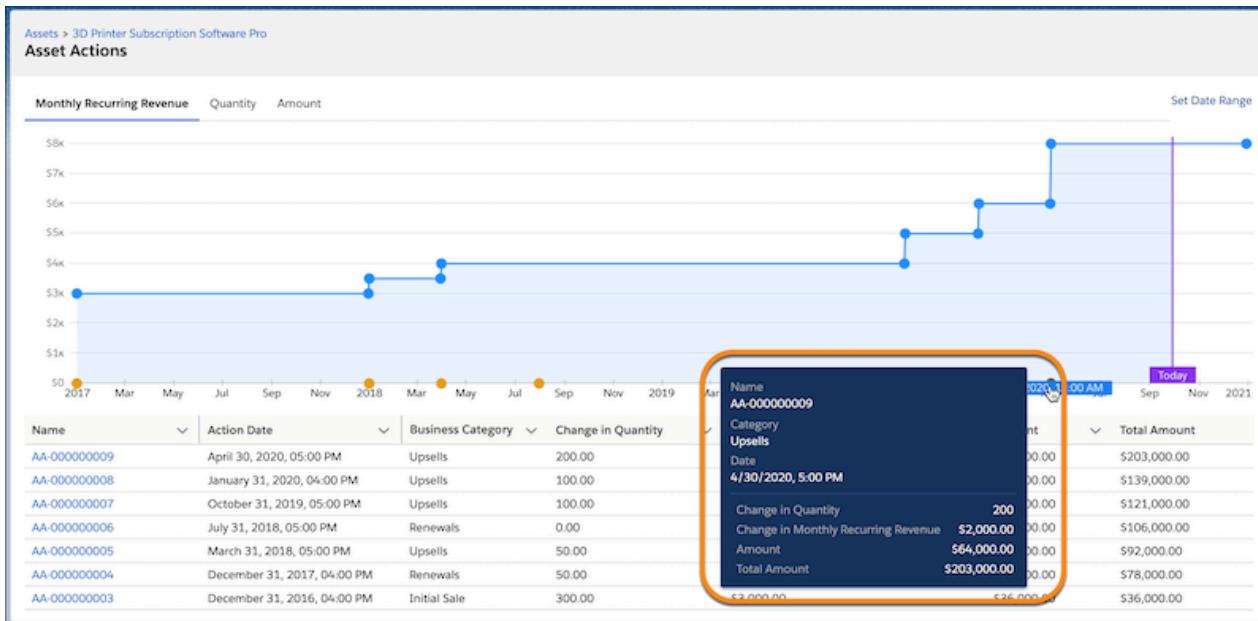


2. Use the **View** links in the dashboard to see details on the quantity, MRR, and amount.

Each asset action tab shows the effect of each change or cancellation on the quantity, MRR, and amount.

On the asset action page, narrow your view of a chart by clicking **Set Date Range**. To see details about a change, hover over a dot. A blue dot in a chart indicates an asset state period change, and an orange dot indicates an asset action change.





Some lifecycle-managed assets show asset action sources, such as a Salesforce order product or work order line item, or a reference to another system. Click the item under Name, and then click the Related tab.

Some lifecycle-managed assets show the business category of each change, such as initial sale or renewals. Include these categories in reports to understand where revenue is rising or falling.

### 3. View Related Invoice Lines opens the Invoice Lines related list.

Product Name	Invoice	Due Date	Total Amount (With Tax)	Payments	Balance	Days Past Due
1 3D Printer Subscription Software Pro	INV-0000	8/6/2020	\$100.00	\$0.00	\$100.00	54

### See Also

[The Lifecycle of a Subscription Asset](#)

## How Lifecycle Management Works

An asset without lifecycle management represents a product that a customer has bought, but the information stops at the time of sale. By contrast, a lifecycle-managed asset shows information about a product such as a subscription or a warranty after it's sold. Customer Asset Lifecycle Management is driven by a custom automated process. For each lifecycle-managed asset, you see information in a dashboard and on related pages in Salesforce. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

### Lifecycle-Managed Assets

Lifecycle-managed assets and related objects contain information about the type and timing of changes to the asset. After your business enables Customer Asset Lifecycle Management, a developer or integrator builds a custom system to automatically generate and update lifecycle-managed assets and related objects. The system uses data from order products, work order line items, or an external system to generate the lifecycle-managed assets that you see in Salesforce. (Salesforce Billing Managed Package)

### The Lifecycle of a Subscription Asset

When customers buy a subscription product, they can add to, upgrade, or cancel the subscription over time. Customer Asset Lifecycle Management represents these changes through asset actions, and one or more asset state periods. An asset action represents a change to quantity, amount, and monthly recurring revenue (MRR). An asset state period represents the time span when a change applies to an asset, forming a timeline of the asset's lifecycle. Depending on where today's date falls within this timeline, the dashboard shows a period's quantity, amount, and MRR as current. A nightly background process compares the current date to the start dates of the asset state periods. If the system finds a match, it updates the dashboard and related pages, labeling the next period's data as current.

(Salesforce Billing Managed Package)

## Lifecycle-Managed Assets

Lifecycle-managed assets and related objects contain information about the type and timing of changes to the asset. After your business enables Customer Asset Lifecycle Management, a developer or integrator builds a custom system to automatically generate and update lifecycle-managed assets and related objects. The system uses data from order products, work order line items, or an external system to generate the lifecycle-managed assets that you see in Salesforce. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

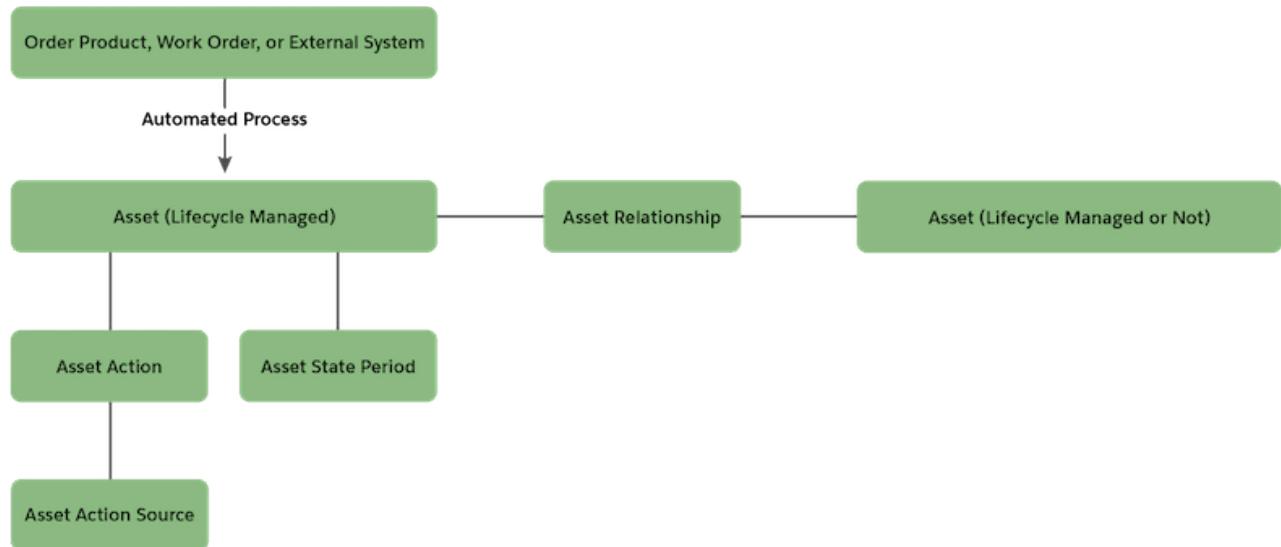
---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

When a lifecycle-managed asset is generated, the Lifecycle-managed asset field is selected. You can't switch an asset to a lifecycle-managed asset or the reverse.

With a lifecycle-managed asset, an *asset action* represents changes made to the asset, and an *asset state period* denotes the time span when the changes apply. An *asset action source* provides an optional way to track the sources of changes. Together, these three child objects provide visibility into changes made to a lifecycle-managed asset throughout its lifecycle.



The system or your Salesforce admin can create relationships between lifecycle-managed assets and other assets, lifecycle-managed or not. Here are some examples:

- In an asset relationship with a type of Upgrade, the primary asset is the upgraded asset, and the related asset is the original asset.
- In an asset relationship with a type of Coverage, the primary asset is covered by the related asset.

Related assets have separate lifecycles.

#### See Also

[Manage and Optimize Assets in Field Service](#)

## The Lifecycle of a Subscription Asset

When customers buy a subscription product, they can add to, upgrade, or cancel the subscription over time. Customer Asset Lifecycle Management represents these changes through asset actions, and one or more asset state periods. An asset action represents a change to quantity, amount, and monthly recurring revenue (MRR). An asset state period represents the time span when a change applies to an asset, forming a timeline of the asset's lifecycle. Depending on where today's date falls within this timeline, the dashboard shows a period's quantity, amount, and MRR as current. A nightly background process compares the current date to the start dates of the asset state periods. If the system finds a match, it updates the dashboard and related pages, labeling the next period's data as current. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

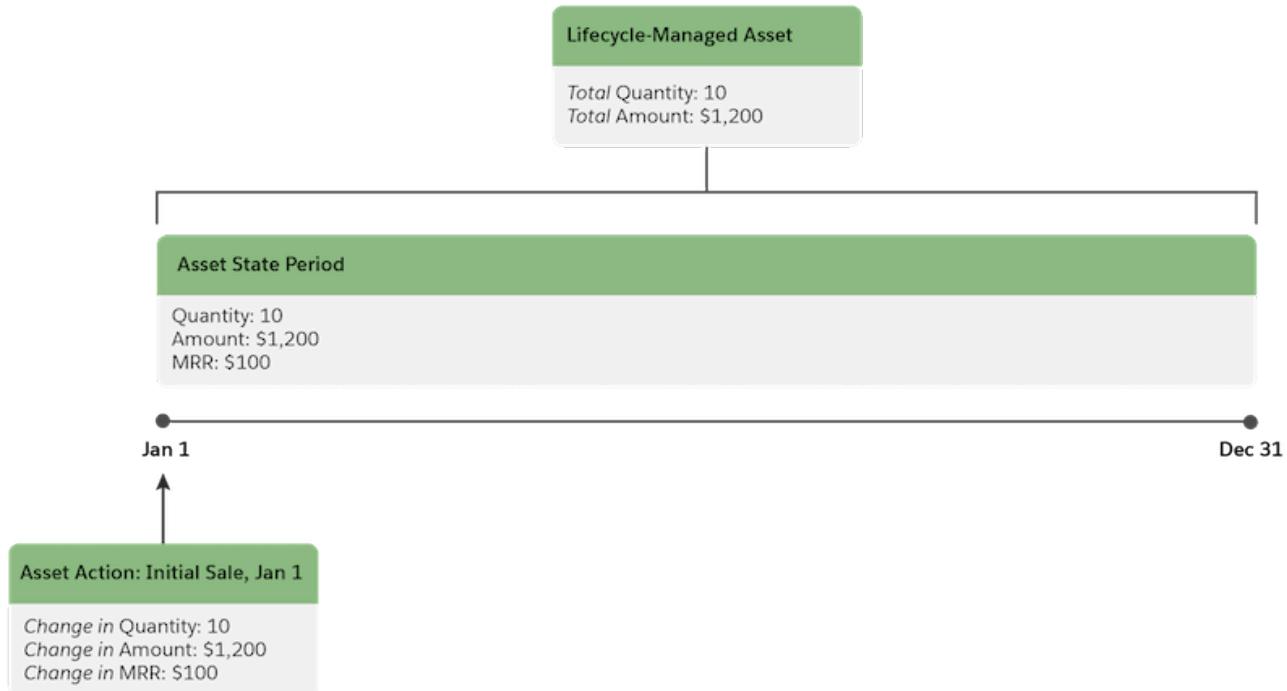
Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

Let's look at an example: A customer purchases seat licenses for a subscription product and then, after a

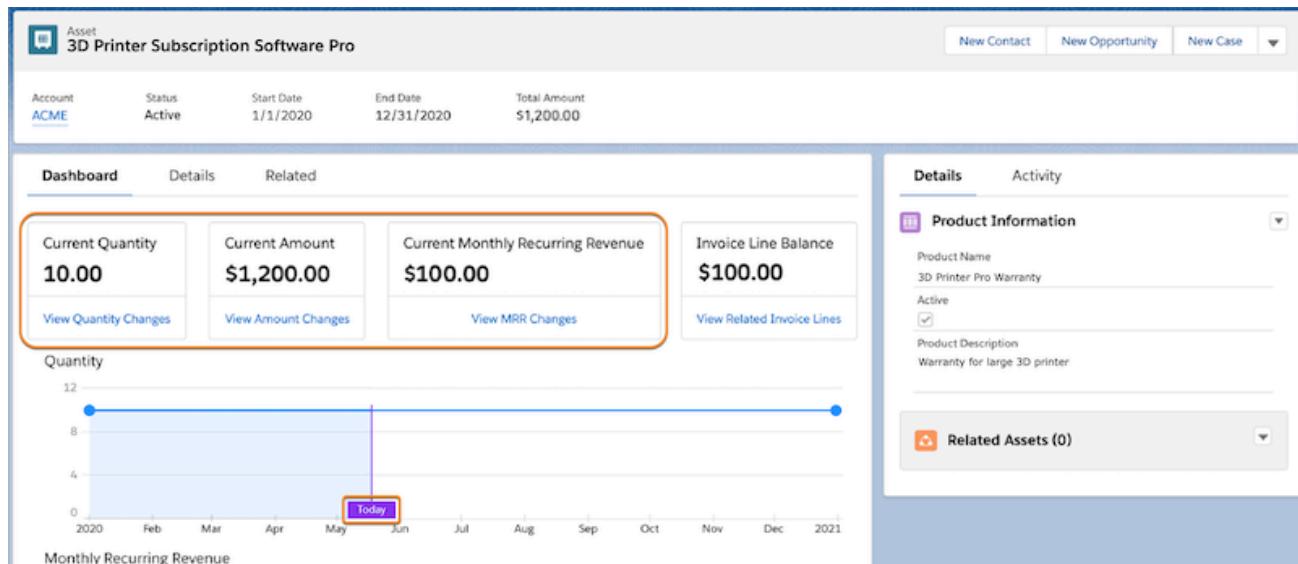
few months, adds more seats.

## Initial Purchase: Customer Buys 10 1-Year Licenses @ \$10 per Month

The customer's subscription (represented by an asset record) starts January 1 and costs \$1,200 for one year. The system generates one asset state period. An asset action represents the initial sale.

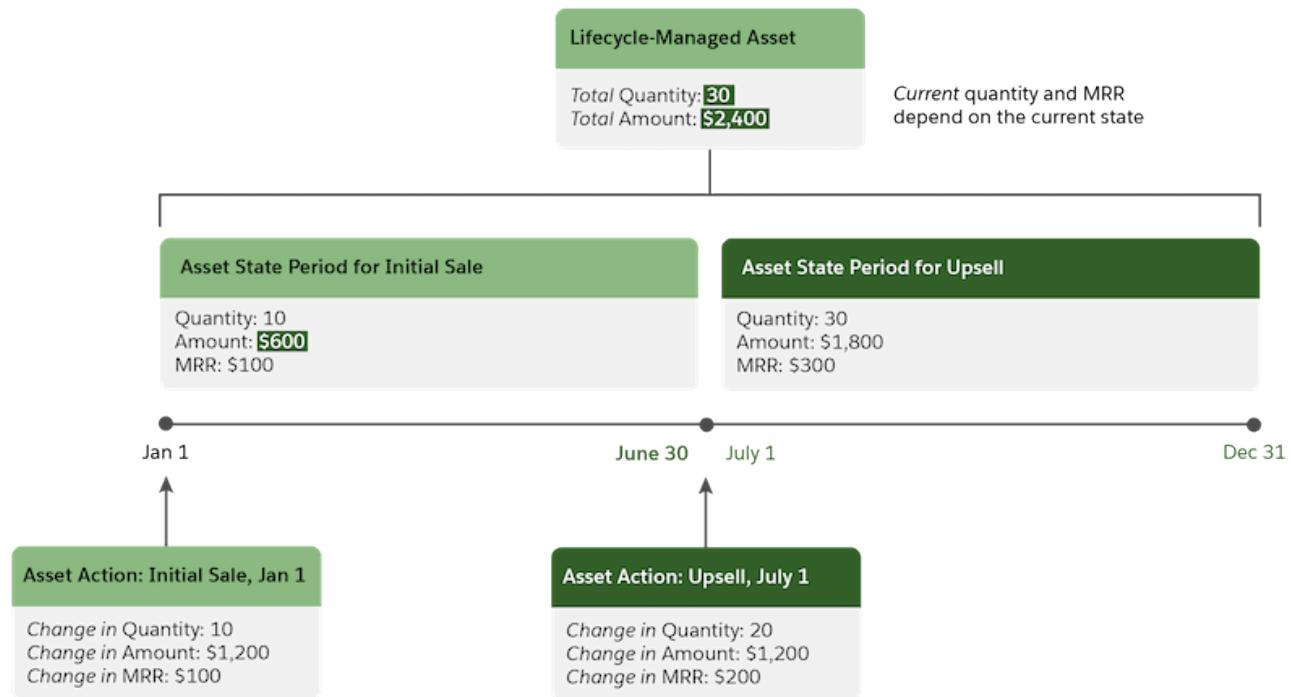


Here's the dashboard representing the asset's lifecycle after the purchase.

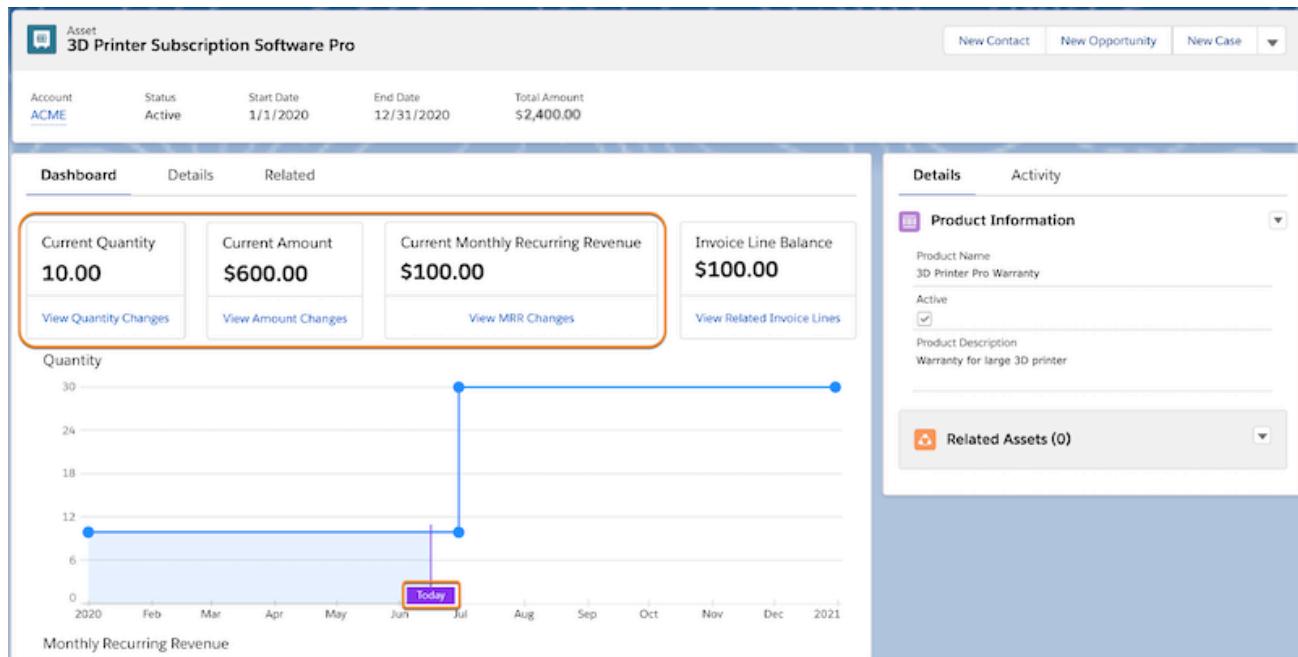


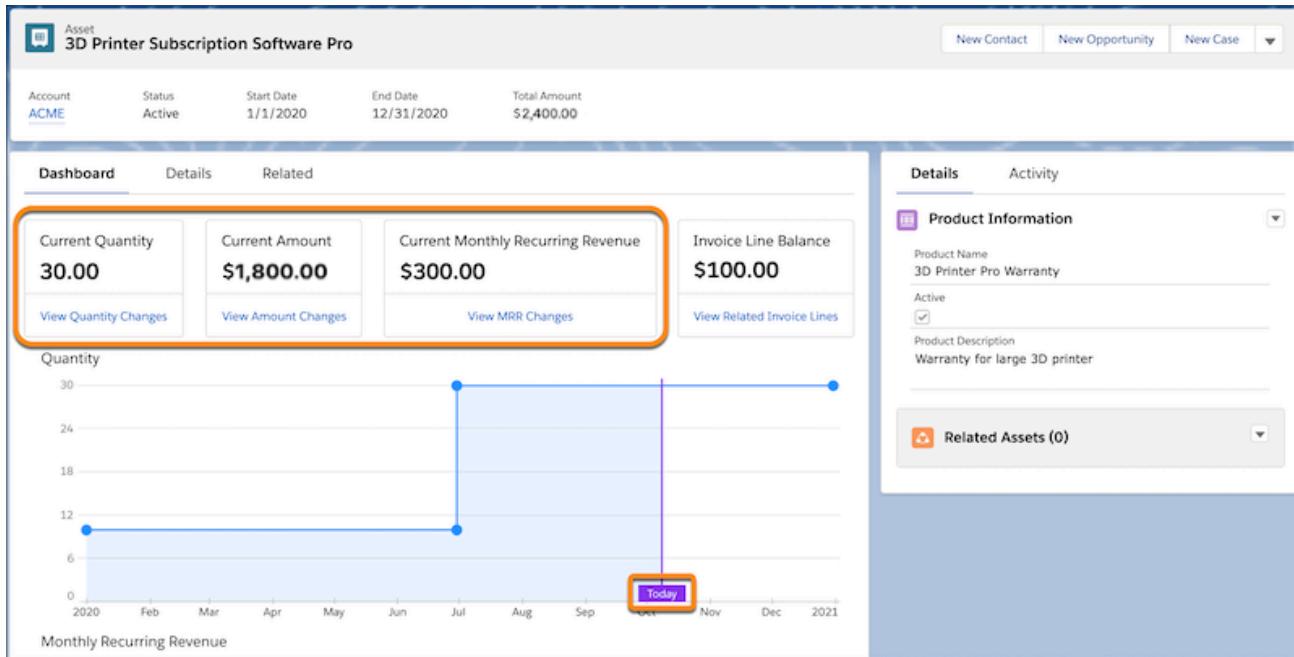
## Upsell: Customer Buys 20 More Licenses

Starting July 1, the customer adds 20 licenses to the initial 10. The system adjusts the original asset state period and adds one: the first period shows the time span when the license quantity is 10, and a second period shows the time span when the quantity is 30. Asset state periods are back to back within a lifecycle, without gaps or overlaps.



Here's the dashboard representing the asset's lifecycle on a day before the upsell takes effect.





Sometimes the current date falls outside the asset's lifecycle—that is, the start and end dates of the first and last asset state periods, respectively. In that case, the dashboard shows the current quantity and other data as 0.

The nightly background process that updates the dashboard and related pages runs at 1:00 AM in the time zone of your org's server. For more information, contact your Salesforce admin.

#### See Also

- [Lifecycle Management Objects and Fields in Salesforce Billing](#)
- [Manage and Optimize Assets in Field Service](#)

## Invoicing Considerations When Using Customer Asset Lifecycle Management

The Asset field is empty by default. When Salesforce Billing invoices an order through an invoice run or Bill Now, invoice lines inherit the value of the field. Review these considerations if you're entering a value for the Asset field. (Salesforce Billing Managed Package)

#### REQUIRED EDITIONS

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce Billing

#### Updates to the Asset Field on Invoice Lines

You can edit an invoice line's Asset field at any time, including overriding the automated field value.

## Updates to the Asset Field on Order Products

You can also edit the order product's Asset lookup field at any time. However, if you change it after the order product has been invoiced, you must update the invoice line's Asset field. It's not updated automatically.

## Requirements for Order Products Linked via Revised Order Field

Suppose that Salesforce Billing invoices an order product linked to one or more order products via the Revised Order field. The invoice line populates its Asset field only if all order products in the chain have the same value for the Asset field.

On an order product, the Asset lookup field can point to assets with or without lifecycle management. However, on an invoice, the Asset field doesn't indicate which type of asset it points to.

### See Also

[Managing Invoice Lines Related to Your Assets](#)

## Lifecycle Management Objects and Fields in Salesforce Billing

Some field values on asset lifecycle management objects are system populated, and some are API populated. You can't customize asset lifecycle management fields or add custom fields to asset lifecycle management objects. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

 **Note** You can't delete Asset Actions, Asset Action Sources, and Asset State Period records, but they don't count toward your storage limits.

### [Lifecycle Management Fields on Assets in Salesforce Billing](#)

Lifecycle-managed assets have fields for recording asset lifecycle-related information and financial values. You can't delete lifecycle-managed asset records. (Salesforce Billing Managed Package)

### [Asset Action Object and Fields in Salesforce Billing](#)

An asset action represents a change made to a lifecycle-managed asset. The fields can't be edited and the records can't be deleted. (Salesforce Billing Managed Package)

### [Asset Action Source Object and Fields in Salesforce Billing](#)

The asset action source object is an optional way to record what transactions caused changes to lifecycle-managed assets. Use it to trace financial and other information about asset actions. This object supports Salesforce order products and work order line items, and transaction IDs from other systems. The fields can't be edited and the records can't be deleted. (Salesforce Billing Managed Package)

### [Asset State Period Object and Fields in Salesforce Billing](#)

An asset state period represents a time span when an asset has the same quantity, amount, and

monthly recurring revenue (MRR). An asset has as many asset state periods as there are changes to it (asset actions) during its lifecycle. The dashboard and related pages show the current asset state period. The fields can't be edited and the records can't be deleted. (Salesforce Billing Managed Package)

## Lifecycle Management Fields on Assets in Salesforce Billing

Lifecycle-managed assets have fields for recording asset lifecycle-related information and financial values. You can't delete lifecycle-managed asset records. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

#### **Account**

The account related to the asset. This field can't have a lookup to a contact.

#### **Current Amount**

Reserved for future use.

#### **Current Lifecycle End Date**

Represents the end of the period shown as current in the dashboard and related pages. System-populated field inherited from the end date of the current asset state period. If that field is empty, as with an evergreen subscription, the Current Lifecycle End Date field is also empty.

#### **Current Monthly Recurring Revenue**

The asset's monthly recurring revenue during the current asset state period. System-populated field inherited from the monthly recurring revenue on the current asset state period. If no asset state period is current, the value is 0.

#### **Current Quantity**

The asset's quantity during the current asset state period. System-populated field inherited from the quantity on the current asset state period. If no asset state period is current, the value is 0.

#### **Lifecycle-Managed Asset**

Marked True when the system generates a lifecycle-managed asset. You can't switch an asset to a lifecycle-managed asset or the reverse. This field is system populated.

#### **Invoice Line Balance**

The sum of the Balance fields on the posted invoice lines related to the asset.

## ISO Currency

If multiple currencies are enabled, represents the currency of the asset, asset action, asset action source, and asset state period. Inherited from the currency defined when the asset is created. The currency used for transactional records such as order products and work orders can differ from the currency on a lifecycle-managed asset. This field can't be edited.

## Lifecycle End Date

Represents the end of the asset's lifecycle. System-populated field inherited from the end date of the final asset state period. If that field is empty, as with an evergreen subscription, the lifecycle has no end date.

## Lifecycle Start Date

Represents the beginning of the asset's lifecycle. System-populated field inherited from the start date of the earliest asset state period. This field can't be edited. When a new asset action affects the start date of an asset state period, the period is deleted and a new one is generated.

## Product

The product related to the asset.

## Total Lifecycle Amount

The total of a lifecycle-managed asset across its initial sale, amendments, renewals, and other changes. As of Spring '21, the field is populated on all lifecycle-managed assets. On other assets, it's null.

## See Also

- [Manage and Optimize Assets in Field Service](#)
- [Manage Multiple Currencies](#)

## Asset Action Object and Fields in Salesforce Billing

An asset action represents a change made to a lifecycle-managed asset. The fields can't be edited and the records can't be deleted. (Salesforce Billing Managed Package)

## REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

## Action Date

The date when an asset action change is recorded. This date can differ from the start date of the related asset state period. For example, suppose that a customer cancels a subscription in June (action date of the asset action). The cancellation's effective date (start date of the asset state period) can be in October.

### Amount

The delta in the total asset amount resulting from an asset action.

### Asset

The related lifecycle-managed asset.

### Business Category

The business category of the asset action, for use in reporting. Asset action totals are broken out by the picklist values on this required field, and those totals are in turn reflected on assets. The following categories are available. They aren't customizable.

- Initial Sale
- Upsells
- Downsells
- Cancellations
- Terms And Conditions Changes
- Renewals
- Other

### Category (Deprecated)

A category to apply to the asset action. In your layouts and reports, replace this optional picklist with the required Business Category picklist. The following categories are available.



**Warning** To avoid issues with asset lifecycle management, don't add, edit, replace, delete, or deactivate this picklist's values. They must remain identical to the Business Category picklist values.

- Initial Sale
- Upsells
- Downsells
- Cross-Sells
- Cancellations
- Transfers
- Terms And Conditions Changes
- Renewals
- Other

### Change in Actual Tax

Rollup of actual tax from all asset action sources. This field is populated by the system.

### Change in Adjustment Amount

Rollup of adjustment amount from all asset action sources. This field is populated by the system.

### Change in Estimated Tax

Rollup of estimated tax from all asset action sources. This field is populated by the system.

### **Change in Monthly Recurring Revenue**

The delta in the asset's MRR resulting from an asset action. For example, suppose that the MRR during an asset state period is \$200 and the next asset action adds \$100. This field's value is \$100.

### **Change in Product Amount**

Rollup of product amount from all asset action sources. This field is populated by the system.

### **Change in Quantity**

The delta in the asset quantity resulting from an asset action. For example, suppose that the asset quantity during an asset state period is 20 and the next asset action adds 10. This field's value is 10.

### **Change in Subtotal**

Rollup of subtotal from all asset action sources. This field is populated by the system.

### **Total Amount**

The sum of the current and previous asset action amount. This field is populated by the system.

### **Total Cancellations Amount**

The sum of current and previous asset action amounts categorized as Cancellations. This field is populated by the system.

### **Total Cross-Sells Amount**

The sum of current and previous asset action amounts categorized as Cross-Sells. This field is populated by the system.

### **Total Downsells Amount**

The sum of current and previous asset action amounts categorized as Downsells. This field is populated by the system.

### **Total Initial Sale Amount**

The sum of current and previous asset action amounts categorized as Initial Sale. This field is populated by the system.

### **Total Monthly Recurring Revenue**

The sum of the MRR for the current and previous asset action. This field is populated by the system.

### **Total Other Amount**

The sum of current and previous asset action amounts categorized as Other Amount. This field is populated by the system.

**Total Quantity**

The sum of the changes in quantity for the current and previous asset action. This field is populated by the system.

**Total Renewals Amount**

The sum of current and previous asset action amounts categorized as Renewals. This field is populated by the system.

**Total Terms and Conditions Changes Amount**

The sum of current and previous asset action amounts categorized as Terms and Conditions Changes. This field is populated by the system.

**Total Transfers Amount**

The sum of current and previous asset action amounts categorized as Transfers. This field is populated by the system.

**Total Upsells Amount**

The sum of current and previous asset action amounts categorized as Upsells. This field is populated by the system.

**Type**

The REST API used to generate the asset action. This field is system populated. Possible values are:

- Generate
- Change
- Cancel

## Asset Action Source Object and Fields in Salesforce Billing

The asset action source object is an optional way to record what transactions caused changes to lifecycle-managed assets. Use it to trace financial and other information about asset actions. This object supports Salesforce order products and work order line items, and transaction IDs from other systems. The fields can't be edited and the records can't be deleted. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

 **Note** An action can have more than one source. For example, suppose that an initial sale is a ramped deal with two order products. Depending on how Customer Asset Lifecycle Management is configured for your company, you see two asset actions with one source each, or one asset action

with two sources.

## Fields

### **Actual Tax**

The region-specific tax amount determined at time of the order. This field is not used for price and tax calculations.

### **Adjustment Amount**

An adjustment to the product amount, such as a discount.

### **Asset Action**

The related asset action, that is, the change caused by a source transaction.

### **End Date**

The end date of the service or change.

### **External Reference**

The ID of a source transaction originating in a system outside Salesforce.

### **External Reference Data Source**

A system outside Salesforce that contains source transactions.

### **Estimated Tax**

Estimate of the region-specific tax amount made at time of the transaction.

### **Name**

The ID of the asset action source.

### **Product Amount**

The product amount after the source transaction.

### **Quantity**

The product quantity or the change in product quantity after the source transaction.

### **Reference Entity Item**

The ID of a source transaction originating in Salesforce. The transaction can be an order product or a work order line item.

### **Start Date**

The start date of the service or change.

### **Subtotal**

Sum of the product amount and the adjustment amount.

### **Transaction Date**

The date of a source transaction, such as an order date.

## Asset State Period Object and Fields in Salesforce Billing

An asset state period represents a time span when an asset has the same quantity, amount, and monthly recurring revenue (MRR). An asset has as many asset state periods as there are changes to it (asset actions) during its lifecycle. The dashboard and related pages show the current asset state period. The fields can't be edited and the records can't be deleted. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

## Considerations for Asset State Periods

### **Dates on Asset State Periods**

All dates on an asset state period are inclusive. Asset state periods are back to back, without gaps between or overlaps. Each asset state period starts 1 second after the previous period's end date and spans 24-hour increments. A lifecycle-managed asset must have at least one asset state period at least 24 hours long.

### **Effect of Changes and Cancellations**

Only the newest asset state periods are maintained. Older asset state period records aren't retained after an asset is changed or canceled.

## Fields

### **Amount**

An asset's total amount during an asset state period.

### **Asset**

The asset related to an asset state period.

### **End Date**

The end date and time of an asset state period. On an asset that is an evergreen subscription, the last or only asset state period has no end date.

## Monthly Recurring Revenue

An asset's MRR during an asset state period.

### Name

The ID of the asset state period.

### Quantity

The total quantity of an asset during an asset state period.

### Start Date

The start date and time of an asset state period.

### See Also

[The Lifecycle of a Subscription Asset](#)

## Administer Customer Asset Lifecycle Management

Your developer or integrator uses the Connect REST API to automate creation of, changes to, and cancellation of lifecycle-managed assets. The system uses data from Salesforce or from third-party ecommerce, quoting, order management, and other applications. You grant permissions, set up page layouts, and customize the views of lifecycle-managed assets to support your business processes for account, sales, and service reps. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

---

#### [Add Lifecycle Management Components to the Asset Layout](#)

Configure Customer Asset Lifecycle Management components so that reps see the dashboard. The dashboard consists of asset summary cards and charts. (Salesforce Billing Managed Package)

#### [Customize the Asset Action and Invoice Line Pages](#)

Customize the columns and hover details on the asset action and invoice line pages as necessary to support your business process. (Salesforce Billing Managed Package)

## Add Lifecycle Management Components to the Asset Layout

Configure Customer Asset Lifecycle Management components so that reps see the dashboard. The dashboard consists of asset summary cards and charts. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

---

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with

---

## Salesforce CPQ Plus and Salesforce Billing

Add the components Asset Summary (1) and Asset State Period Chart for quantity and monthly recurring revenue (MRR) (2 and 3) to the asset page layout.



1. In Setup, in the Object Manager, find **Asset**. Click **Lightning Record Pages**, and create or open a layout.
2. Display the summary cards by dragging the Asset Summary component into the layout. You can hide individual cards using field-level security. The Asset Summary component can't be otherwise customized.
3. Display the two charts by dragging the Asset State Period Chart component into the layout twice.
  - a. Select the first chart you dragged into the layout. In the Page sidebar, under Field to Display, select **Quantity**.
  - b. Select the second chart you dragged into the layout. In the Page sidebar, under Field to Display, select **Monthly Recurring Revenue**.
4. On assets that aren't lifecycle-managed, Customer Asset Lifecycle Management components still appear, but without data. To hide each component on those assets, under Set Component Visibility, add a filter with the following values.
  - Field: Lifecycle-managed asset
  - Operator: *Equal*
  - Value: *True*

### See Also

[Lightning App Builder](#)

## Customize the Asset Action and Invoice Line Pages

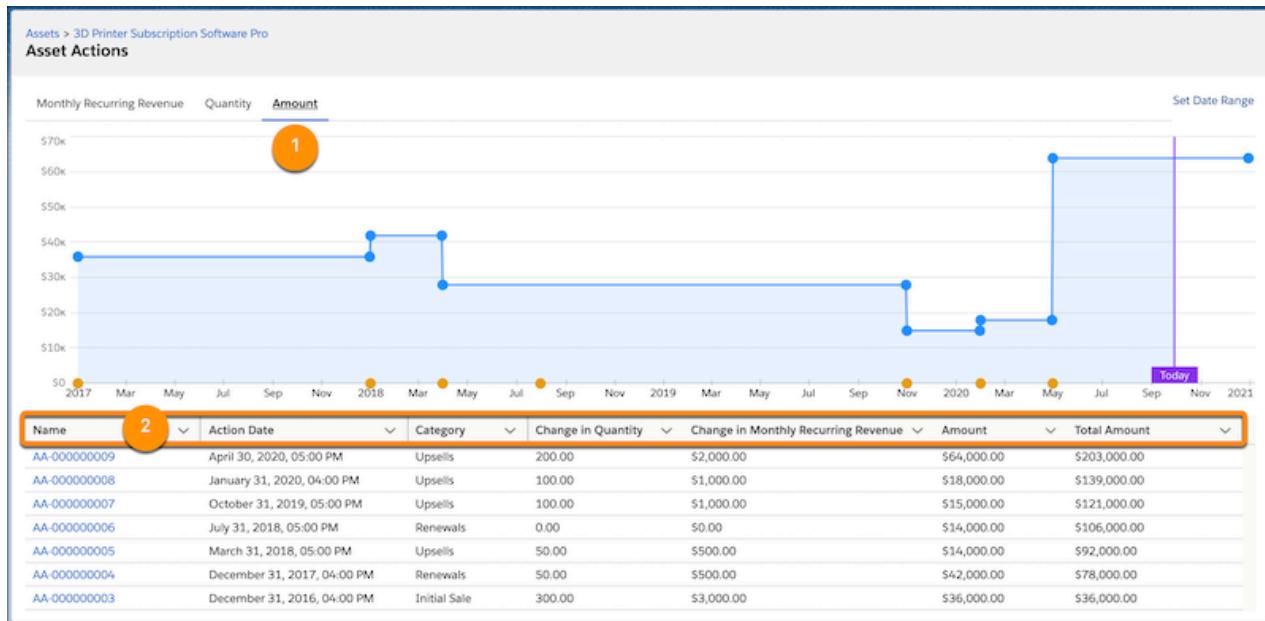
Customize the columns and hover details on the asset action and invoice line pages as necessary to support your business process. (Salesforce Billing Managed Package)

### REQUIRED EDITIONS

Available in Lightning Experience in Professional, Enterprise, Unlimited, and Developer Editions with Salesforce CPQ Plus and Salesforce Billing

In Setup, in the Object Manager, find **Asset**. Click **Page Layouts**, and create or open a layout.

1. To change the columns on the asset actions page, add the Asset Actions related list to the layout and edit the fields.  
Each tab on the asset actions page includes a chart (1) and the Asset Actions related list with the columns you select (2).



- To change the columns on the invoice lines page, add the Invoice Lines related list to the layout and edit the fields.

The invoice lines page includes the Invoice Line related list with the columns you select.

The screenshot shows the 'Invoice Lines' section for the '3D Printer Subscription Software Pro' asset. It displays a table with columns: Product Name, Invoice, Due Date, Total Amount (With Tax), Payments, Balance, and Days Past Due. A single row is shown for the asset, with a callout '1' pointing to it.

Product Name	Invoice	Due Date	Total Amount (With Tax)	Payments	Balance	Days Past Due
1 3D Printer Subscription Software Pro	INV-0000	8/6/2020	\$100.00	\$0.00	\$100.00	54

## See Also

[Asset Action Object and Fields in Salesforce Billing](#)

[Field-Level Security](#)

## Printable Tip Sheets & User Guides – For Salesforce Billing

Printable implementation guides for Salesforce Billing admins. (Salesforce Billing Managed Package)

### Salesforce Billing Printed Guides

These documents are for admins and partners.

- [Salesforce CPQ and Billing Proration Implementation Guide](#)