APTTUS®

Billing Management on Salesforce Winter 2018 Administrator Guide

12/05/2018

Table of Contents

Getting Started

Apttus Billing Management is a key component of the Apttus Revenue Pillar, helping organizations automate accurate invoice generation.This guide aims to help administrators understand, configure and manage Apttus Billing System.

Key Benefits

- **Boost Revenue and Maximize Margins-** Create more intelligent quotes.
- **Complete CPQ Cycle-** Shorten cycles and reduce sales friction.
- **Streamline-** Improve customer relationships.
- **Scalability-** Rapidly grow your business with the most advanced cloud platform.

Key Capabilities

- **Manage Assets-** Change, Amend or Cancel assets.
- **Scalability-** Rapidly grow your business with the most advanced cloud platform.
- **Dynamic Reports-** Get reports on your billing data.

The document is divided into four key sections:

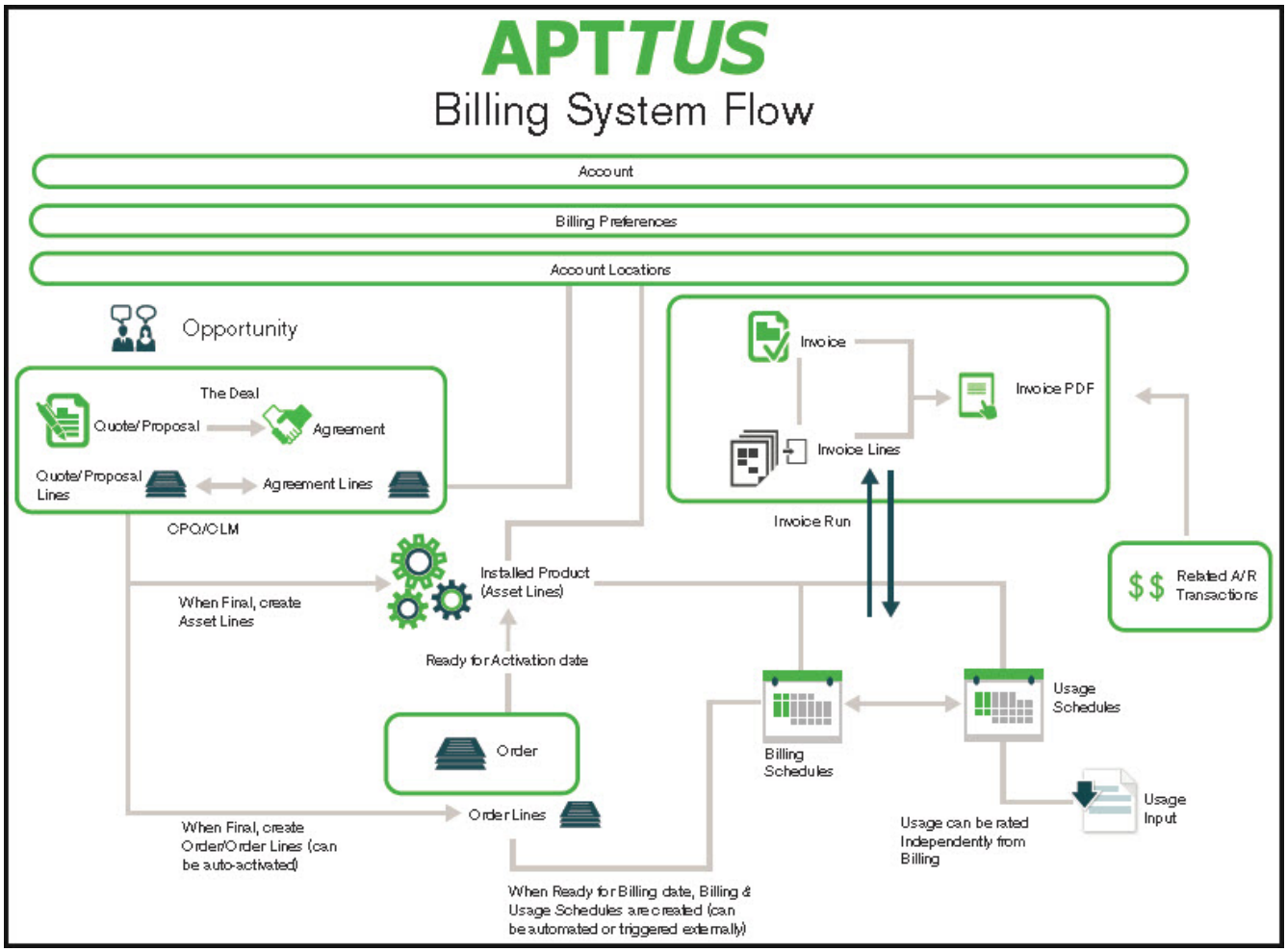
- Billing Concepts
- Installation
- Basic Settings
- Advanced Settings

What's New

The following table lists changes in the documentation to support each release.

Document	Topic	Description
Winter 2018	Billing Schedules	Modified Topic. Updated the Bill Cycle Start Date field under Billing Preferences section, to capture the relevance of Period Start Date picklist.
	Billing Preferences	Modified Topic. Updated the following: • Added a new field, Ready for Invoice Date Offset • Updated the Bill Cycle Start Date field under Billing Preferences section, to capture the relevance of Period Start Date picklist.
	Legal Entity based Custom Numbering	New Topic. New feature for this release.
	Legal Entity based Template Configuration	New Topic. New feature for this release.
	Setup Billing Preference	Modified Topic. Updated the Bill Cycle Start Date field to capture the relevance of Period Start Date picklist.
Summer 2018	Understanding Proration	Modified Topic. Updated the topic to include the feature, Combine with First Period.
	Dunning Policy	Modified Topic. Updated the topic with a note.
	Email Invoice Attachment Callback Class	Modified Topic. Updated the code sample to include the new parameter, default email body.
	Billing System Properties	Modified Topic. Updated the topic to include information about the following settings: • Delete Forecast • No Proration on Renewal

System Overview



Apttus Billing Management functions use several Objects and fields that you must define before you can generate an invoice for your customers.

1. Account is a Salesforce object that all Apttus applications inherit defined values from. Apttus Billing Management uses values from the **Account Relationship** and the **Invoice**, **Invoice Statement**, **Billing** and **Shipping Address** fields defined in the Account object.
2. You can define each customer's Billing Preference to suit their requirements and convenience.
3. Account Locations are attached to the account so you can associate more than one location to each account. You can use different account locations as shipping address, billing address, and tax invoice address.
4. On the **Proposals** page, you can select products, adjust pricing, adjust billing settings, change bill preference, initiate approvals processes and you can automate the process to activate an order, create a Bill, and create revenue.
5. On the **Order** detail page, if it is not done automatically, you can define when an order is ready for activation and billing.

6. You can use the **Billing Schedules** function to track all billing information related to an asset. With **Billing Schedules**, you can spread the net amount due, over a period of time that you can define.
7. A **Usage Schedule** is associated to **Billing Schedules** that are Usage-based. Usage or metering data is loaded to an order and then usage rating associates to a **Usage Schedule**. You can define usage price tiers that use flat pricing, or tiered pricing.
8. With **Usage Input**, you can rate usage manually or this object is used while creating Salesforce batch jobs to rate usage.
9. You can define payment terms and by default, a separate invoice is automatically created for products with different payments. A payment term is a condition or guideline under which you can make or receive a payment.
10. After you complete all the procedures listed above, you can perform an **Invoice run** which is a scheduled invoice generation batch job.
11. You can issue Credit Memos - manually or automatically as part of the Invoice runs.
12. Integrate with 3rd party tax engines to display correct taxes on the invoice automatically.
13. You can use the **Related A/R transactions** to track, measure, and record incoming payments, credits, debits, and refunds.
14. Finally, you can use reports to track and measure your billing assignments and forecast requirements.

As an administrator, it is essential to understand the key concepts driving the system. Let us go to the Billing Concepts to comprehend the system functions.

Installing Apttus Billing

You must install Apttus Billing package and other associated packages to implement Apttus Billing Management. Salesforce provides a simple wizard-led installation process for these packages.

> **Caution**
> Apttus recommends downloading and upgrading Apttus packages into a sandbox before provisioning into a production environment. Please contact **Apttus Support** before installing if you need assistance.

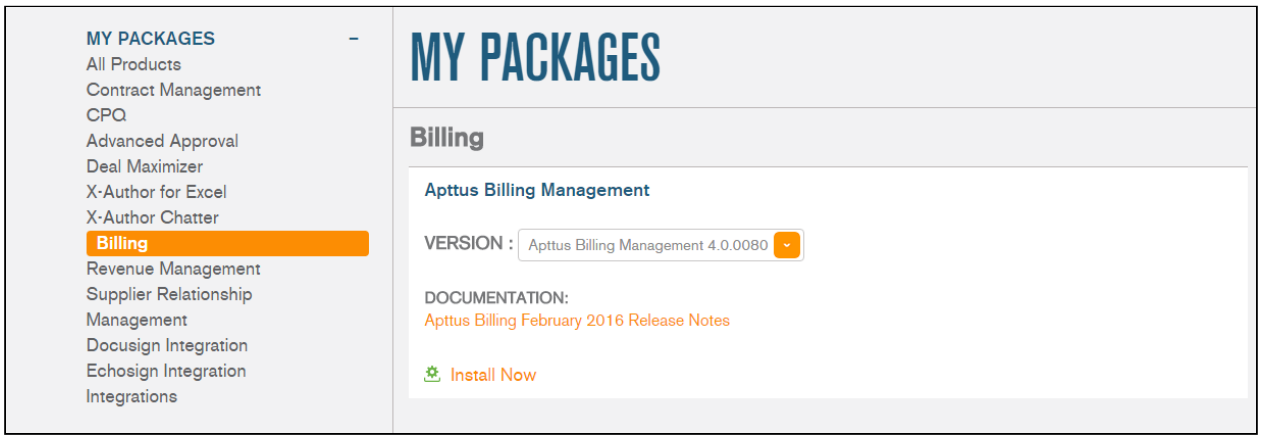
You can check out the Release Notes to get the information on latest package versions. Install the packages in the following order:

Order	Package	Install Center tab to Access the package
1	Apttus Contract Management	Contract Management
2	Apttus Proposal Management	CPQ
3	Apttus Configuration & Pricing	CPQ
4	Apttus Quote/Proposal-Configuration Integration	Integrations
5	Apttus CPQ API	CPQ
6	Apttus Quote/Proposal-Asset Integration	CPQ
7	Apttus Billing Management	Billing

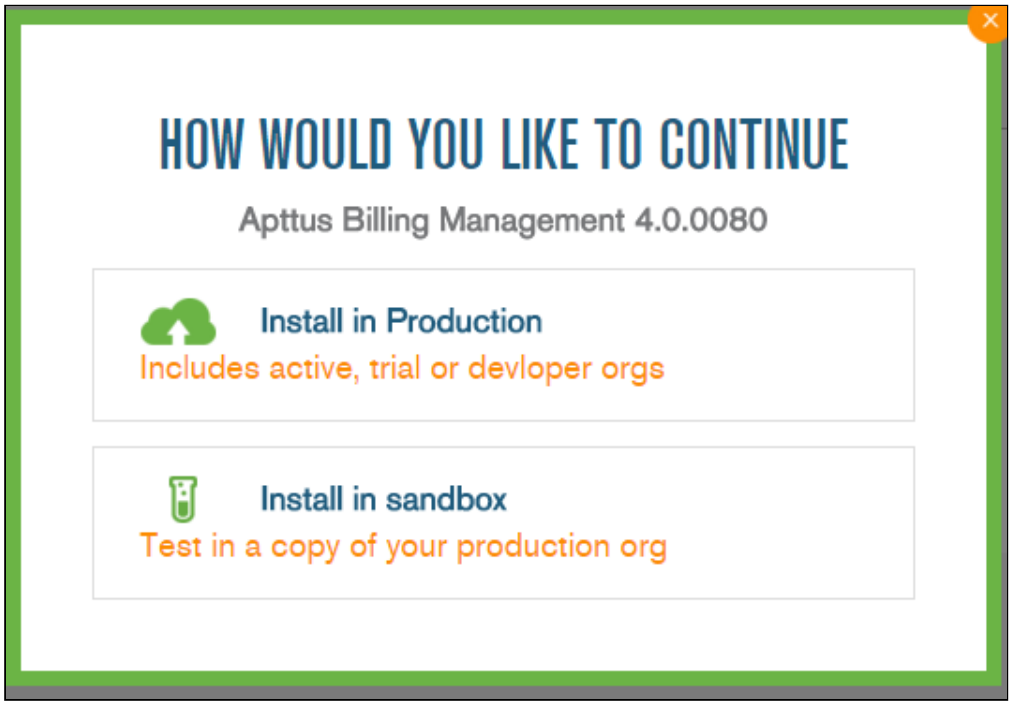
> **Note**
> You must have Apttus-provided login credentials to access the Apttus Community Portal and download packages.

To install the Apttus Billing Management package

1. Go to the **Install Center** within the *Apttus Customer Portal.*
2. Under My packages, select **Billing** and click **Install Now**.



3. Select an environment where you want to install the package: **Production** OR **Sandbox**.



4. You will be re-directed to the Salesforce AppExchange, where you will use your Salesforce credentials to access the Apttus-managed package.
5. Agree to the to the terms and conditions and click **Confirm** and **Install**!. You may have to log in to Salesforce again with your credentials.
6. On the Upgrade page, enter the Apttus provided **password**.
7. Select the security level. If you know the required settings, select the third option; otherwise, it is recommended that you select the second option and refine the security settings as required later on:
    - **Grant access to admins only**
    - **Grant access to all users**

- **Select security settings**

8. Click **Next** and then click **Install/Upgrade**.

After you complete installing this package, you must customize your settings before you can start creating invoices.

Post-Installation Configurations

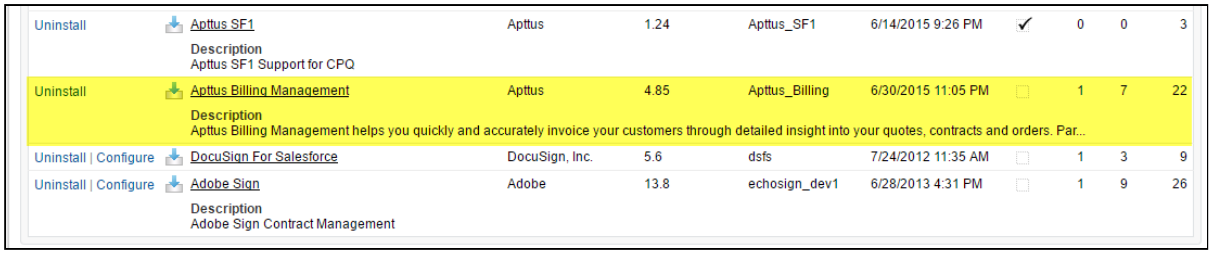
After installing Apttus Billing Management, you need to configure it to meet the business objectives of your implementation. You should set up the user profiles, assign them appropriate access and update the Billing object layout.

The system administrator for Tier1 Solutions Sam, needs to:

- Set up Object Layouts
- Assign User Permissions
- Billing Setup on PLI and Quote Proposal

Verify Billing Management Package Objects

Billing Management is installed with several key objects that drive the application behavior. Go to **Setup** > **Deploy** > **Installed Packages** to verify the Billing package.



To use the Apttus Config System Properties custom setting, you must create a dataset named *System Properties*.

If your organization has already installed and/or have been using the Apttus CPQ /CLM solutions, the *System Properties* dataset might be configured. You must customize the system properties to use the Billing Management features.

- Add Picklist Values
- AutoActivate Order

Add Picklist Values

1. Go to **Billing Preferences** custom object. Under **Custom Fields & Relationships** search for **Billing Cycle Start** field. Click **Billing Cycle Start** and scroll down to see the **Picklist Values**. Click **New** to add a new picklist value. Add following values for the Billing Cycle Start: *Billing Day of Month*, *Ready*

for Billing Date, *Period Start Date*, *Order Date,* and *Account Billing Day of Month*.

2. Go to **Invoice Run** custom object. Under **Custom Fields & Relationships** search for the **Category** field. Click **Category** and scroll down to see the **Picklist Values**. Click **New** to add a new picklist value.

Create an Order on Proposal Acceptance

To create an Order on the acceptance of a Quote/Proposal, go to **Custom Settings** > **Proposal System Properties** and check **Auto Create Order**.

On selecting this field, Order and Assets will be created as soon as you click **Accept** on the Quote/ Proposal.

Create Custom Labels

The *Display Action Settings* page provides a simple interface for selecting actions, applying styles, and reordering them. You can even create and use your own custom action.

In order to create custom actions, you must first create a custom label.

To create a custom label

1. Go to **Setup** > **Create** > **Custom Labels** and click **New Custom Label**.
2. Type a mandatory **Short Description** and type a mandatory **Name.**
3. In **Categories**, type comma-separated values that can be used in filter criteria when creating custom label list views. For example: Pages, Label, Components, and more.
4. In **Value**, type mandatory text in the form of a brief description or copy and paste the field name from Step 1.
5. Click **Save.**

A custom label is created and saved. In the **Display Action Settings,** enter the label name that you created in the field **Action Label Name**.

Pre-requisites for Invoice PDF Generation

For invoice generation in the PDF format, you must perform the following configurations:

1) Go to **Billing Console** and click **Billing Settings**. Enter the **Api user name** and **Password**. The Password is your salesforce login password followed by the security token.

2) Go to **Setup** > **Custom Settings** > **Comply System Properties** and set the **Merge Webserivce End point** to *https://mwsdev.apttus.net/cgi-bin/Janus/MergeServer/Bin/MMCGI.exe* under **Comply System Settings**.

3) Go to **Setup** > **Remote Site Settings** and click **New remote site** to add a remote site. Specify **Remote Site Name** and set **Remote Site URL** to the base URL of production merge server, for example, *https://mwsdev.apttus.net*.

Update Existing Billing Plans and Billing Plan Templates

Admins, Advanced Uses, Implementation Experts, and partners can update the existing billing plans and billing plan templates to change the existing percent field to customized percent field.

To update the existing billing plans,

1. Go to Developer Console of your Salesforce Org.
2. Execute this Apex code:

```
Database.executeBatch(new UpdateBillingPlanItemBatchJob());
```

To update the existing billing plans templates,

1. Go to Developer Console of your Salesforce Org.
2. Execute this Apex code:

```
Database.executeBatch(new UpdateBillingPlanTemplateItemBatchJob());
```

Manage Object Layouts

Each object has a standard view when you install the packages. Because of this you might miss out on some important fields that are available but not visible on the interface. By Editing a page layout you can show or hide fields and custom buttons on the detail pages.

In this section, you'll see the recommended layout for following objects:

- Account
- Invoice Run
- Credit Memo

Accounts

Page layouts define the organization of various fields in an object. It is recommended to add following formula fields to the layout for Account object:

- **Set Invoice Email Template**
  Go to **Account** Object > Click **View Fields** > Add a *new* **formula field** from **Custom Fields & Relationships**
  Enter the Formula value as

```
 HYPERLINK("/apex/Apttus_billing__SetAccountInvoiceEmailTemplate?id="&Id ,IMAGE("/
resource/Apttus_billing__Button_SetInvoiceEmailTemplate", "Set Inv Temp"),"_self")
```

- **Set Billing Contacts**
  Go to **Account** Object > Click **View Fields** > Add a *new* **formula field** from **Custom Fields & Relationships**
  Enter the Formula value as

```
HYPERLINK("/apex/Apttus_Billing__SetAccountBillingContacts?id="&Id, IMAGE("/
resource/Apttus_Billing__Button_SetBillingContacts", "Set Billing
Contacts"),"_self")
```

- **Set CM Default Template**
  Go to **Account** Object > Click **View Fields** > Add a *new* **formula field** from **Custom Fields & Relationships**
  Enter the Formula value as

```
HYPERLINK("/apex/Apttus_billing__SetAccountDefaultCMTemplate?id="&Id ,IMAGE("/
resource/Apttus_billing__Button_SetDefaultCreditMemoTemplate", "Set CM
Temp"),"_self")
```

- **Set CM Default Email Template**
  Go to **Account** Object > Click **View Fields** > Add a *new* **formula field** from **Custom Fields & Relationships**
  Enter the Formula value as

```
HYPERLINK("/apex/Apttus_billing__SetAccountCreditMemoEmailTemplate?
id="&Id ,IMAGE("/resource/Apttus_billing__Button_SetCreditMemoEmailTemplate", "Set
CM Temp"),"_self")
```

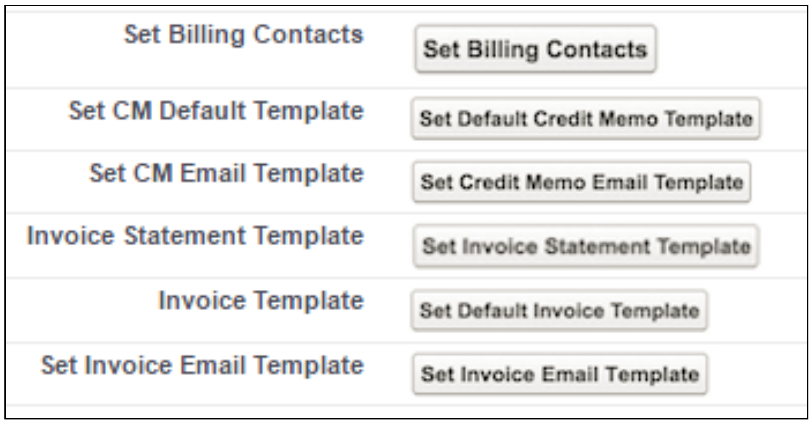
- **Set Invoice Statement Template**
  Go to **Account** Object > Click **View Fields** > Add a *new* **formula field** from **Custom Fields & Relationships**
  Enter the Formula value as

```
HYPERLINK("/apex/Apttus_Billing__SetAccountInvoiceStatementTemplate?id="&Id,
IMAGE("/resource/Button_SetInvoiceStatementTemplate", "Set Default Invoice
Statement Template"),"_self")
```

- **Invoice Template**
  Go to **Account** Object > Click **View Fields** > Add a *new* **formula field** from **Custom Fields & Relationships**
  Enter the Formula value as

```
HYPERLINK("/apex/Apttus_Billing__SetAccountDefaultInvoiceTemplate?id="&Id,
IMAGE("/resource/Apttus_Billing__Button_SetDefaultInvoiceTemplate", "Set Default
Invoice Template"),"_self")
```

You should see following buttons on the Account page.

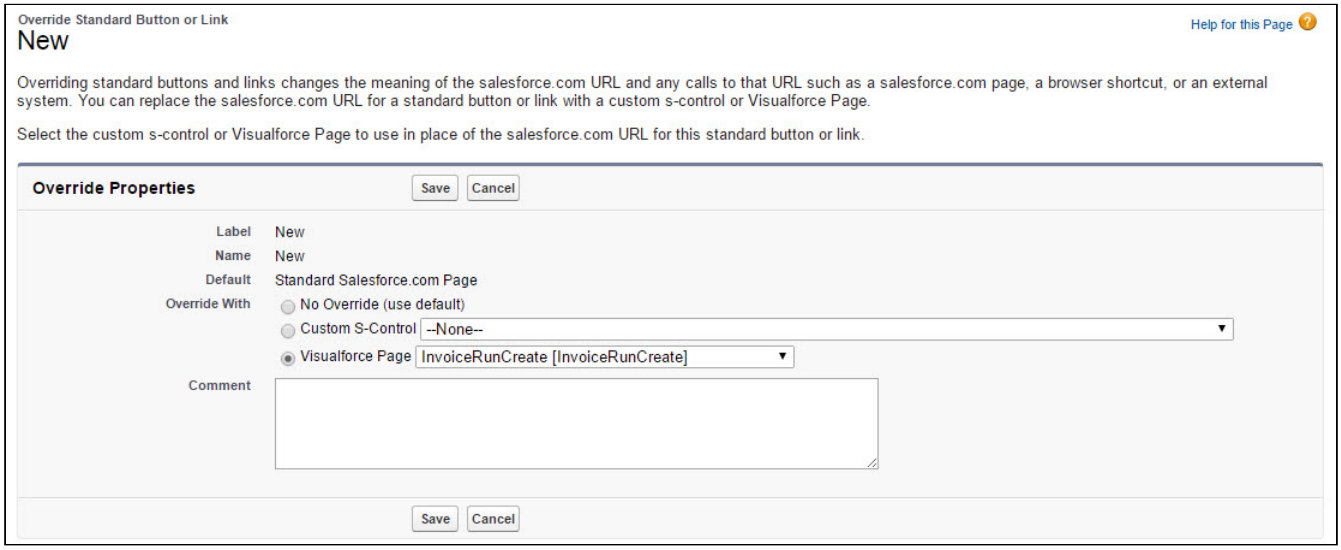


Follow the same procedure to add these buttons on the **Account Location** page. Go to **Account Location** Object > Click **View Fields** > Add a new formula field from **Custom Fields & Relationships** for each of the templates.

Invoice Runs

You must override the standard Invoice Run page with a Visualforce page to drive the Invoice Run behavior as required.

Go to **Invoice Runs** object. Under **Buttons, Links, and Actions** section, edit **New** label to select a VisualForce page, *InvoiceRunCreate[InvoiceRunCreate]* as shown in the screenshot.



Creating Custom Filters for Invoice Runs

Apart from the four filters, Account, Account Location, Billing Schedule, and Product, you can create Custom Filters to sort and process Invoices.

For example, you are a Billing executive in a retail chain and you want to generate only those invoices for a custom object that is local to your organization and is not a standard object in Apttus CPQ or Apttus Billing Management. You can define **Custom Settings** to call the custom Object in your implementation of Apttus Billing Management.

> **Note**
> The **Invoice Filter Callback Class** value in the above illustration is a class name. Consult Apttus Professional Services or your Apttus Implementation Engineer for the value you must use in your instance of Apttus Billing Management.

To define the Custom Settings,

1. Click **Setup** > **Custom Settings** > **Manage** (to the left of **Billing Custom Classes)** > **System Properties** > **Edit**.
2. Enter a value for **Invoice Filter Callback Class** and click **Save.**

When you define a value for the **Invoice Filter Callback Class**, the Invoice Run User interface is updated with the **Custom Options** section that has 2 fields, **Category** and **Sub Category**.

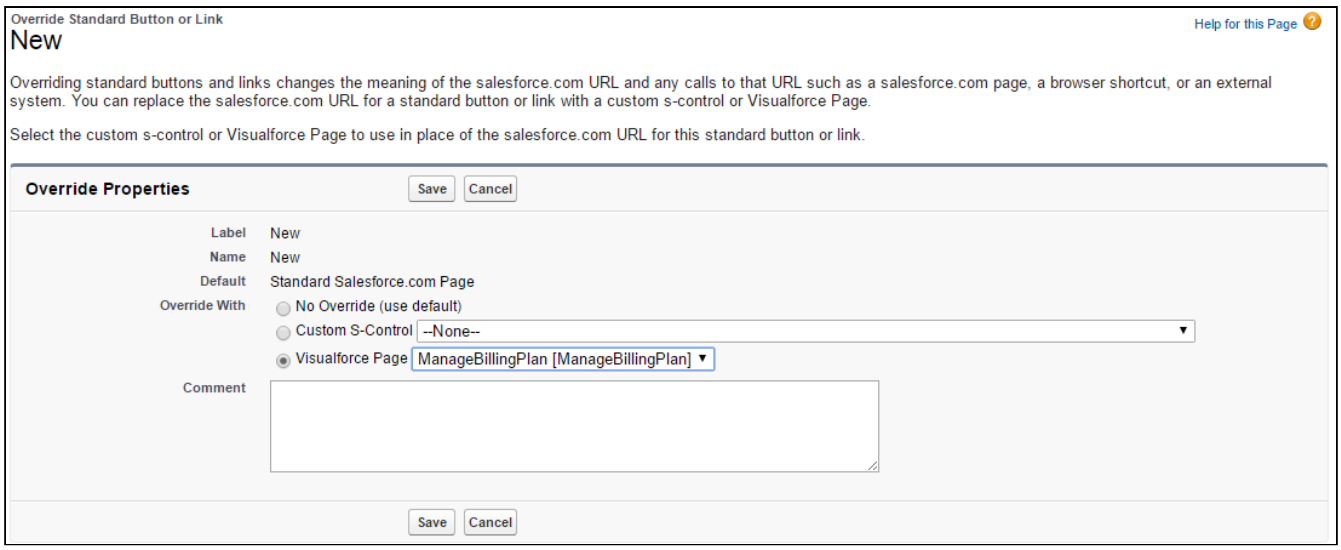
Select the values with the from each field with the picklist you have defined and the resulting Invoice Run is processed based on the fields you define here. For example, if a customer wants to only apply the Invoice Run to one or more Business Partner countries (a custom field added to the Asset Line Item) then the **Category** pick list should be updated to contain all countries applicable to their application: Germany, France, Italy, and others. Based on the country selected in Category, selected Billing Schedules will be

picked by Invoice Run. If further granular filtering is needed then you can leverage the **Sub Category** pick list.

Billing Plan

Go to **View Object** from the Billing Plan screen. From **Buttons, Links, and Actions** edit **New** label to select a *ManageBillingPlan* **VisualForce page** as shown in the screenshot.

> Select ManageBillingPlan2 Visualforce page to use Full Term Billing Plan Functionality.



Manage Users

The following permissions need to be granted for user profiles associated with users who will be working with Apttus Billing Management. If you do not grant these permissions, user may face issues while using and configuring Apttus Billing Management.

Please note that user roles and their access might differ during the actual implementation. The following table intends to give you a direction for granting user permissions.

Custom Object Layouts	Finance Administrator	Finance User
Account Location	Default	Default
Order	Default	Default

Account Contacts	Default	Default
Custom Tab Settings	**Finance Administrator**	**Finance User**
Billing Preference	Default On	Default On
Accounts	Default On	Default On
Contacts	Default On	Tab Hidden
Payment Terms	Default On	Tab Hidden
Tax Codes	Default Off	Tab Hidden
Tax Certificates	Default On	Tab Hidden
Reports	Default Off	Tab Hidden
Custom Object Permissions	**Finance Administrator**	**Finance User**
Asset Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Asset Transaction History	Read, Create, Edit, Delete, View All, Modify All	Read
Asset Usage Price Tiers	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Asset Usage Price Tier History	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Billing Console	Read, Create, Edit, Delete, View All, Modify All	Read
Billing Plan	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Billing Plan Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Billing Preferences	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Billing Schedules	Read,Delete, View All, Modify All	Read
Billing Settings	Read, Create, Edit, Delete, View All, Modify All	
Invoices	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Invoice Runs	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Invoice Run Results	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Invoice Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Invoice Line Item Tax Breakups	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Opportunities	Read, Create, Edit, Delete, View All, Modify All	Read, Create

Orders	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Order Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Proposal Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Related A/R Transactions	Read, Create, Edit, Delete, View All, Modify All	Read
Tax Codes	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Tax Certificates	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Tax Breakups	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Templates	Read, Create, Edit, Delete, View All, Modify All	Read, Create
Credit Memos	Read, Create, Edit, Delete, View All, Modify All	Read, Create

Billing Concepts

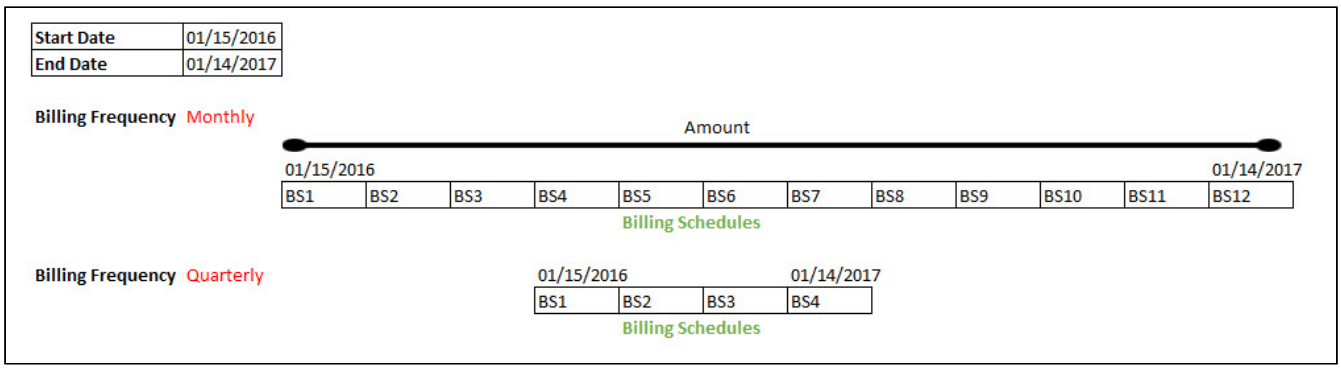
The process of Billing embodies various sub-processes that drive the entire system.

As an administrator, you should know the key concepts that run behind the scenes and define the system behavior. They are:

- Billing Schedules
- Proration
- Usage Billing
- Related A/R Transactions
- Billing Preferences
- Credit Memo
- Payment Terms

Billing Schedules

Billing Schedules for a product (whether one-time or recurring) are a distribution of the net amount of a contract, based on the chosen Billing Frequency. Billing Schedules determine the amount that should be billed to the customer once or over a period of time.



Input Required

From CPQ

Field Name	Description
Start Date	The date when a customer starts getting billed for the purchase. This field translates to **Period Start Date** on Billing Schedules. This value is stored as **Expected Start Date** on the Proposal.
End Date	The date till when a customer is billed for the purchase.This field translates to **Period End Date** on Billing Schedules. This value is stored as **Expected End Date** on the Proposal.

Field Name	Description
Price	The total amount of purchase.

From Product PLI (Price List Item)

Field Name	Description
Billing Rule	For every Billing Schedule, the date when invoice should be generated is based on the Billing Rule. The field **Ready for Invoice Date** on Billing Schedules is processed from the Billing Rule. Following are the options for the Billing Rule: • **Bill in Advance** - To invoice the receivables at the beginning of the schedule. • **Bill in Arrears** - To invoice the receivables at the end of the schedule. • **Bill on Ready for Billing Date** - To bill your customer with a consolidated invoice, on a day of their choice.
Billing Frequency	Billing Frequency is how the product price over a period is distributed and billed to an account. The Billing Frequency for a product can be set to: • **Monthly** - To generate a bill once every month • **Quarterly** - To generate a bill once every three months • **Half-yearly** - To generate a bill once in the middle of a calendar or financial year. • **Yearly** - To generate a bill once every year • **Usage** - To generate a bill based on usage or consumption • **One-Time** - To generate a one-time bill. For example, a product 'Health App' priced $3000 has the **Billing Frequency** set to **Monthly**. Monthly Billing Schedules will be created and invoiced at a price of $250/month. If Billing Frequency for the same product is changed to **Quarterly**, quarterly Billing Schedules will be created and invoiced at a price of $750 each quarter.

From Billing Preferences

Field Name	Description
Bill Cycle Start Date	Billing Cycle Start can be any day on which you want to bill customers. Billing Schedules are aligned on the basis of Billing Cycle Start. You can select one of the following options: • **Billing Day of the Month** - value specified in the **Billing Day of Month** field. • **Ready for Billing Date** - value from **Ready for Billing Date** field on the accepted Quote/Proposal. • **Period Start Date** - value from **Expected start date** field on Asset Line Item • **Order Date** - value from **Order Date** field in the activated order. • **Account Billing Day of the Month** - value from **Billing Day of Month** field on Accounts. Select this option, if you want to set a common billing day for all orders in an account.

Field Name	Description
Billing Day of the Month	Day of the month when you want to bill your customer. For example, to set '10' as the Billing Day, select **10th of the month** from the picklist. If you select **End of the month**, the Billing Day is set to the **last day** of each month as and when the schedule advances. For example, Billing Day of Month will be '31' for March and '30' for April. **Note:** This field holds significance only if you set 'Billing Cycle Start' to 'Billing Day of Month'.
Proration Period Treatment	To define how you want to distribute the billing amount proportionally between the first and last Billing Schedules, select from one of the following picklist options: • **Separate Period** - To create a separate Billing Schedule for the outstanding amount. • **Ignore** - To levy no charge for the Initial partial period. • **Charge Full** - To levy no charge for the last partial period. • **Combine with First Period** - To charge each partial period along with the next full period. To see the impact Proration has on Billing Schedules, please refer Proration.

Billing Schedule Alignment

Billing Schedules are aligned on the basis of **Billing Cycle Start** field on **Billing Preferences**.

Billing Schedule for an asset takes the **Expected Start Date** mentioned on the Proposal Line and Order Line as the starting point for generating the schedules. To align rest of the duration, it looks up to the value provided in the **Billing Cycle Start**. You have the flexibility to choose Billing Day from other objects such as Account, Quote/Proposal, and Order.

The **Ready for Invoice Date** is calculated based on the **Billing Rule** as follows,

- For **Bill in Arrears: Ready for Invoice Date= Period End Date + 1**
- For **Bill in Advance: Ready for Invoice Date= Period Start Date**

Use Case

Standalone Recurring product

Tier1 system sells 'SecureDevice' ($100/unit) which is billed monthly for a contract period of one year. The contract is in effect from 04/20/2016. The customer wants an invoice on 15th of every month.

Steps:

1. Add a product 'SecureDevice' with **List Price** as *$100* and **Price Type** as *Recurring*.
2. Under the Tax and Billing tab on PLI, set the **Billing Rule** as *Bill in Advance* and **Billing Frequency** as *Monthly*.
3. Create a new Billing Preference and set **Billing Cycle Start** to *Billing Day of month.*
4. Set *Billing Day of month* to 15th of the month.
5. Create a Quote/Proposal. Set **Start Date** as 04/20/2016 and **End Date** as 04/19/2017.
6. Add the product to your cart and **Finalize**.
7. **Present** the proposal and **Accept** it after the reviews.
8. Order and Asset are created the moment a Proposal is accepted. Activate the order.

Resulting Billing Schedules

Period Start Date	Period End Date	Ready for Invoice Date	Amount
4/20/2016	5/14/2016	4/20/2016	$83.33
5/15/2016	6/14/2016	5/15/2016	$100
6/15/2016	7/14/2016	6/15/2016	$100

Billing Schedule Amendment

When an ABO amendment is applied to the underlying Billing Schedules of an Asset Line item, a new set of Billing Schedules is generated for the Start and End Dates of the amended Asset. The existing Billing Schedules that correspond to the new Schedules are evaluated and the collective set of Billing Schedules are updated accordingly. How an existing Billing Schedule is evaluated and which schedule updates are performed is a function of whether or not the existing Schedule pertains to a partial period or a full period. Following examples will help you understand how amendment impacts the Billing Schedules.

Mid-cycle Amendment with Pending Billing Schedules

Before Amendment

- Billing Day of Month = 1st
- Billing Frequency = Monthly
- Net Price = 400.00 i.e. 100.00 per month

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Credit Schedule
BS1	3/1/2015	3/31/2015	Pending Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Pending Invoiced	100.00		
BS3	5/1/2015	5/31/2015	Pending Billing	100.00		
BS4	6/1/2015	6/30/2015	Pending Billing	100.00		

After Amendment

- Date of Amendment = 4/16
- Net Price = 500.00 for 2.5 months i.e. 200.00 per month

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Credit Schedule
BS1	3/1/2015	3/31/2015	Pending Invoiced	100.00		
BS2	4/1/2015	4/30/2015	**Superseded**	100.00	**Yes**	
BS5	**4/1/2015**	**4/15/2015**	**Pending Billing**	**50.00**		
BS6	**4/16/2015**	**4/30/2015**	**Pending Billing**	**100.00**		
BS3	5/1/2015	5/31/2015	**Superseded**	100.00	**Yes**	
BS7	**5/1/2015**	**5/31/2015**	**Pending Billing**	**200.00**		
BS4	6/1/2015	6/30/2015	**Superseded**	100.00	**Yes**	
BS8	**6/1/2015**	**6/30/2015**	**Pending Billing**	**200.00**		

Mid-cycle Amendment with Invoiced Billing Schedules

Before Amendment

- Billing Day of Month = 1st
- Billing Frequency = Monthly
- Net Price = 400.00 i.e. 100.00 per month

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Credit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00		
BS3	5/1/2015	5/31/2015	Invoiced	100.00		
BS4	6/1/2015	6/30/2015	Pending Invoiced	100.00		

After Amendment

- Date of Amendment = 4/16
- Net Price = 500.00 for 2.5 months i.e. 200.00 per month

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Credit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00	**Yes**	
BS5	**4/16/2015**	**4/30/2015**	**Pending Billing**	**-50.00**		BS2
BS6	**4/16/2015**	**4/30/2015**	**Pending Billing**	**100.00**		
BS3	5/1/2015	5/31/2015	Invoiced	100.00	**Yes**	
BS7	**5/1/2015**	**5/31/2015**	**Pending Billing**	**100.00**		

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Credit Schedule
BS4	6/1/2015	6/30/2015	Superseded	100.00	Yes	
BS8	6/1/2015	6/30/2015	Pending Billing	200.00		

Since the amendment is mid-cycle and the April Billing Schedule (BS2) has been invoiced it will be marked as superseded and will be augmented with two new Billing Schedules.

- The 1st Billing Schedule (BS5) reflects the amount (at the old rate) that was invoiced for the partial period at/after the amendment date and will appear as a credit when an Invoice is generated. Please note that the credit has a reference back to the source of the credit (BS2).
- The 2nd Billing Schedule (BS6) reflects the increased amount for the partial period that was amended and will appear as a charge when an Invoice is generated.

Change Billing Frequency

Before changing Billing Frequency

Billing Day of Month = 1st

Billing Frequency = Monthly

Calendar Cycle Start = June

Net Price = $600.00 ($100.00 per month)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Credit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00		
BS3	5/1/2015	5/31/2015	Invoiced	100.00		
BS4	6/1/2015	6/30/2015	Invoiced	100.00		
BS5	7/1/2015	7/31/2015	Invoiced	100.00		
BS6	8/1/2015	8/31/2015	Pending Invoiced	100.00		

After changing Billing Frequency

Amendment Date = 4/16

Billing Frequency = Quarterly

Calendar Cycle Start = June

Net Price = $450.00
Since the new Term is now 4/16 – 8/31 (4 ½ months) and the new Net Price is $450, the Fee amount will be $300.00 per quarter.

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Credit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00	**Yes**	
BS7	**4/16/2015**	**4/30/2015**	**Pending Billing**	**-50.00**		**BS2**
BS8	**4/16/2015**	**5/31/2015**	**Pending Billing**	**150.00**		
BS3	5/1/2015	5/31/2015	Invoiced	100.00	**Yes**	
BS9	**5/1/2015**	**5/31/2015**	**Pending Billing**	**-100.00**		**BS3**
BS4	6/1/2015	6/30/2015	Invoiced	100.00	**Yes**	
BS10	**6/1/2015**	**6/30/2015**	**Pending Billing**	**-100.00**		**BS4**
BS11	**6/1/2015**	**8/31/2015**	**Pending Billing**	**300.00**		
BS5	7/1/2015	7/31/2015	Invoiced	100.00	**Yes**	
BS12	7/1/2015	**7/31/2015**	**Pending Billing**	**-100.00**		**BS5**
BS6	8/1/2015	8/31/2015	**Superseded**	100.00	**Yes**	

> **Note**
> Because the amendment took place on 4/16 and the Calendar Cycle Start is June, the 1st period will be from 4/16 - 5/31. After that, the quarters will align with 6/1 so the next period will be from 6/1 to 8/31. If the Calendar Cycle Start was None then the 1st period would be from 4/16 - 4/30.

Credits for Billing Frequency Change with Invoiced Billing Schedules

Initial Billing Schedules (Billing Day of Month = 1st, Billing Frequency = Monthly, 100.00 per month)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00		
BS3	5/1/2015	5/31/2015	Invoiced	100.00		
BS4	6/1/2015	6/30/2015	Invoiced	100.00		

Amendment #2 from prior Use Case (Amend on 4/16: **Decrease** rate to 25.00 per month, then Invoice)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00	Yes	
BS5	4/16/2015	4/30/2015	Invoiced	-50.00	Yes	BS2
BS6	4/16/2015	4/30/2015	Invoiced	100.00	Yes	
BS9	4/16/2015	4/30/2015	Invoiced	-100.00		BS6
BS10	4/16/2015	4/30/2015	Invoiced	12.50		
BS3	5/1/2015	5/31/2015	Invoiced	100.00	Yes	
BS7	5/1/2015	5/31/2015	Invoiced	100.00	Yes	

BS11	5/1/2015	5/31/2015	Invoiced	-100.00		BS3
BS12	5/1/2015	5/31/2015	Invoiced	-75.00		BS7
BS4	6/1/2015	6/30/2015	Superseded	100.00	Yes	
BS8	6/1/2015	6/30/2015	Invoiced	200.00	Yes	
BS13	6/1/2015	6/30/2015	Invoiced	-175.00		BS8

Amendment #3 (Amend on 4/16: **Change** to Quarterly 90.00 per quarter or 20.00 per month)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00	Yes	
BS5	4/16/2015	4/30/2015	Invoiced	-50.00	Yes	BS2
BS6	4/16/2015	4/30/2015	Invoiced	100.00	Yes	
BS9	4/16/2015	4/30/2015	Invoiced	-100.00	**Yes**	BS6
BS10	4/16/2015	4/30/2015	Invoiced	12.50	**Yes**	
BS14	**4/16/2015**	**4/30/2015**	**Pending Billing**	**-12.50**		**BS10**
BS15	**4/16/2015**	**6/30/2015**	**Pending Billing**	**75.00**		
BS3	5/1/2015	5/31/2015	Invoiced	100.00	Yes	
BS7	5/1/2015	5/31/2015	Invoiced	100.00	Yes	
BS11	5/1/2015	5/31/2015	Invoiced	-100.00	**Yes**	BS3
BS12	5/1/2015	5/31/2015	Invoiced	-75.00	**Yes**	BS7
BS16	**5/1/2015**	**5/31/2015**	**Pending Billing**	**-25.00**		**BS6**
BS4	6/1/2015	6/30/2015	Superseded	100.00	Yes	

BS8	6/1/2015	6/30/2015	Invoiced	200.00	Yes	
BS13	6/1/2015	6/30/2015	Invoiced	-175.00	**Yes**	BS8
BS17	**6/1/2015**	**6/30/2015**	**Invoiced**	**-25.00**		**BS8**

Credits for Invoiced Billing Schedules

Initial Billing Schedules (Billing Day of Month = 1st, Billing Frequency = Monthly, 100.00 per month)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00		
BS3	5/1/2015	5/31/2015	Invoiced	100.00		
BS4	6/1/2015	6/30/2015	Pending Billing	100.00		

After Amendment #1 (Amend on 4/16: **Increase** rate to 200.00 per month)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	3/1/2015	3/31/2015	Invoiced	100.00		
BS2	4/1/2015	4/30/2015	Invoiced	100.00	**Yes**	
BS5	**4/16/2015**	**4/30/2015**	**Pending Billing**	**-50.00**		**BS2**
BS6	**4/16/2015**	**4/30/2015**	**Pending Billing**	**100.00**		
BS3	5/1/2015	5/31/2015	Invoiced	100.00	**Yes**	
BS7	**5/1/2015**	**5/31/2015**	**Pending Billing**	**100.00**		

BS4	6/1/2015	6/30/2015	Superseded	100.00	**Yes**	
BS8	**6/1/2015**	**6/30/2015**	**Pending Billing**	**200.00**		

Since the amendment is *mid-cycle* and the April Billing Schedule (BS2) has been invoiced it will be marked as *superseded* and will be *augmented* with two new Billing Schedules.

- The 1st Billing Schedule (BS5) reflects the amount (at the old rate) that was invoiced for the partial period at/after the amendment date and will appear as a *credit* when an Invoice is generated.

Notice how the *credit* has a reference back to the source of the *credit* (BS2).

- The 2nd Billing Schedule (BS6) reflects the increased amount for the partial period that was amended and will appear as a *charge* when an Invoice is generated.

  Since the Billing Schedule for May (BS3) has been invoiced it will be marked as *superseded* and will be augmented with a Billing Schedule (B76) that reflects the additional amount to charge for. The period for June is unbilled its Billing Schedule (BS4) will be superseded and replaced by a Billing Schedule (BS8) that reflects the new amount to bill for.

Billing Schedule Cancellation

Before Amendment

- Billing Day of Month = 1st
- Billing Frequency = Yearly
- Net Price = $1200.00 per year
  The Billing Schedule is Invoiced.

Schedule	Period Start	Period End	Status	Fee Amount	Superseded
BS1	1/1/2016	12/31/2016	Invoiced	1200.00	

After Amendment

- Date of Amendment = 4/16
- Net Price = Reduced to $600.00 per year and invoiced.

Schedule	Period Start	Period End	Status	Fee Amount	Superseded
BS1	1/1/2016	12/31/2016	Invoiced	1200.00	**Yes**

Schedule	Period Start	Period End	Status	Fee Amount	Superseded
BS2	**4/16/2016**	**12/31/2016**	**Pending Billing**	**-850.00**	
BS3	**4/16/2016**	**12/31/2016**	**Pending Billing**	**425.00**	

Since the duration of Billing Schedule (BS1) is reduced and it has already been invoiced, it will be superseded and augmented with 2 new Billing Schedules.

Cancel mid-cycle with all Pending Billing Schedules

Before Cancellation

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	1/1/2015	1/31/2015	Pending Billing	100.00		
BS2	2/1/2015	2/28/2015	Pending Billing	100.00		
BS3	3/1/2015	3/31/2015	Pending Billing	100.00		
BS4	4/1/2015	4/30/2015	Pending Invoiced	100.00		

After Cancellation (Cancelled on 2/14 – next day cancellation goes into effect one day later)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	1/1/2015	1/31/2015	Pending Billing	100.00		
BS2	2/1/2015	2/28/2015	**Superseded**	100.00	**Yes**	
BS5	**2/1/2015**	**2/14/2015**	**Pending Billing**	**50.00**		
BS6	**2/15/2015**	**2/28/2015**	**Cancelled**	**50.00**		

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS3	3/1/2015	3/31/2015	Cancelled	100.00		
BS4	4/1/2015	4/30/2015	Cancelled	100.00		
BS1	1/1/2015	1/31/2015	Pending Billing	100.00		

1. Since the Cancellation was mid-cycle the original Billing Schedule (BS2) is superseded and replaced by two new Billing Schedules:

- The 1st Billing Schedule (BS5) is for the partial period that is still active and has not been billed.
- The 2nd Billing Schedule (BS6) is for the partial period that has been cancelled.

2. Since the Billing Schedules for April and March (BS3, BS4) are unbilled their status will be set to Cancelled.

Usage Schedule Cancellation

Cancel mid-cycle with all Pending Billing Schedules that have Usage

Before Cancellation

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	1/1/2015	1/31/2015	Pending Billing	88.00		
BS2	2/1/2015	2/28/2015	Pending Billing	72.00		
BS3	3/1/2015	3/31/2015	Pending Billing	94.00		
BS4	4/1/2015	4/30/2015	Pending Billing	0.00		

Corresponding Usage Schedule

Usage Schedule	Period Start	Period End	Status	Billing Schedule ID	Quantity	Superseded
US1	1/1/2015	1/31/2015	Pending Billing	BS1	30	
US2	2/1/2015	2/28/2015	Pending Billing	BS2	26	
US3	3/1/2015	3/31/2015	Pending Billing	BS3	34	
US4	4/1/2015	4/30/2015	Pending Billing	BS4	0	

After Cancellation (Cancelled on 2/21 – cancellation goes into effect 1 day later for next day option)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	1/1/2015	1/31/2015	Pending Billing	88.00		
BS2	2/1/2015	2/28/2015	**Superseded**	72.00	**Yes**	
BS5	**2/1/2015**	**2/21/2015**	**Pending Billing**	**52.50**		
BS6	**2/22/2015**	**2/28/2015**	**Cancelled**	**19.50**		
BS3	3/1/2015	3/31/2015	**Cancelled**	94.00		
BS4	4/1/2015	4/30/2015	**Cancelled**	0.00		

Usage Schedule	Period Start	Period End	Status	Billing Schedule ID	Quantity	Superseded
US1	1/1/2015	1/31/2015	Pending Billing	BS1	30	
US2	2/1/2015	2/28/2015	**Superseded**	BS2	26	**Yes**

Usage Schedule	Period Start	Period End	Status	Billing Schedule ID	Quantity	Superseded
US5	**2/1/2015**	**2/21/2015**	**Pending Billing**	**BS5**	**17**	
US6	**2/22/2015**	**2/28/2015**	**Cancelled**	**BS6**	**9**	
US3	3/1/2015	3/31/2015	**Cancelled**	BS3	34	
US4	4/1/2015	4/30/2015	**Cancelled**	BS4	0	

1. Since the cancellation was mid-cycle and the February Billing and Usage Schedules are unbilled they will be marked as Superseded and augmented be with two new Billing and Usage Schedules.

- The 1st Billing Schedule is for the partial period that is still active and has not been billed. It reflects the aggregate amount of the rated Usage Inputs that have a date greater than or equal to 2/1 and less than or equal 2/21.
- The 1st Usage Schedule is for the partial period that is still active and has not been billed. It reflects the aggregate quantity of the rated Usage Inputs that have a date greater than or equal to 2/1 and less than or equal 2/21.
- The 2nd Billing Schedule is for the partial period that has been canceled. It reflects the aggregate amount of the rated Usage Inputs that have a date greater than or equal to 2/22 and less than or equal 2/28.
- The 2nd Usage Schedule is for the partial period that has been canceled. It reflects the aggregate quantity of the rated Usage Inputs that have a date greater than or equal to 2/22 and less than or equal 2/28.

  Since the Billing and Usage Schedules for March and April are unbilled their status will be set to Cancelled.

Cancel mid-cycle with Invoiced and Pending Billing Schedules that have Usage

Before Cancellation

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	1/1/2015	1/31/2015	Invoiced	88.00		
BS2	2/1/2015	2/28/2015	Invoiced	72.00		

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS3	3/1/2015	3/31/2015	Invoiced	78.00		
BS4	4/1/2015	4/30/2015	Pending Billing	66.00		

Corresponding Usage Schedule

Usage Schedule	Period Start	Period End	Status	Billing Schedule ID	Quantity	Superseded
US1	1/1/2015	1/31/2015	Invoiced	BS1	30	
US2	2/1/2015	2/28/2015	Invoiced	BS2	26	
US3	3/1/2015	3/31/2015	Invoiced	BS3	31	
US4	4/1/2015	4/30/2015	Pending Billing	BS4	24	

After Cancellation (Cancelled on 2/21 – cancellation goes into effect 1 day later)

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS1	1/1/2015	1/31/2015	Invoiced	88.00		
BS2	2/1/2015	2/28/2015	Invoiced	72.00	**Yes**	
BS5	**2/1/2015**	**2/28/2015**	**Pending Billing**	**-72.00**		BS2
BS6	**2/1/2015**	**2/22/2015**	**Pending Billing**	**52.50**		
BS7	**2/22/2015**	**2/28/2015**	**Cancelled**	**19.50**		
BS3	3/1/2015	3/31/2015	Invoiced	78.00	**Yes**	
BS8	**3/1/2015**	**3/31/2015**	**Pending Billing**	**-78.00**		BS3

Schedule	Period Start	Period End	Status	Fee Amount	Superseded	Debit Schedule
BS4	4/1/2015	4/30/2015	Cancelled	66.00		

Usage Schedule	Period Start	Period End	Status	Billing Schedule ID	Quantity	Superseded
US1	1/1/2015	1/31/2015	Invoiced	BS1	30	
US2	2/1/2015	2/28/2015	Invoiced	BS2	26	Yes
US5	2/1/2015	2/21/2015	Pending Billing	BS6	17	
US6	2/22/2015	2/28/2015	Cancelled	BS7	9	
US3	3/1/2015	3/31/2015	Invoiced	BS3	31	
US4	4/1/2015	4/30/2015	Cancelled	BS4	24	

Since the cancellation was *mid-cycle* and the February Billing and Usage Schedules are *invoiced* they will remain as be *invoiced* and will be augmented with three new Billing and Usage Schedules.

- The 1st Billing Schedule is a credit for the amount that was *invoiced*.
- The 2nd Billing Schedule is for the partial period that was not canceled and is used to charge for the portion of the invoiced amount that was billed. It reflects the aggregate amount of the *rated* Usage Inputs that have a date greater than or equal to 2/1 and less than or equal 2/21.
- The 1st new Usage Schedule is for the partial period that was not canceled and is used to record (audit) that portion of the quantity amount was billed. It reflects the aggregate quantity of the *rated* Usage Inputs that have a date greater than or equal to 2/1 and less than or equal 2/21.
- The 3rd Billing Schedule is for the partial period that was canceled and the amount the Account will not be charged for. It reflects the aggregate amount of the *rated* Usage Inputs that have a date greater than or equal to 2/22 and less than or equal 2/28.
- The 2nd new Usage Schedule is for the partial period that has been canceled. It reflects the aggregate quantity of the *rated* Usage Inputs that have a date greater than or equal to 2/22 and less than or equal 2/28.

Because the Billing and Usage Schedules for March have been invoiced, they will remain as invoiced and one new Billing Schedule will be created.The new Billing Schedule is a credit for the amount that was *invoiced*.
Since the Billing and Usage Schedule for April are unbilled their status will be set to *Cancelled*.

Understanding Proration

The literal meaning of Proration is to divide or distribute proportionally. For Billing, proration means the way a billed amount is distributed over partial billing cycles.
Let us take a look at the following scenarios to understand what is proration and why it is required.

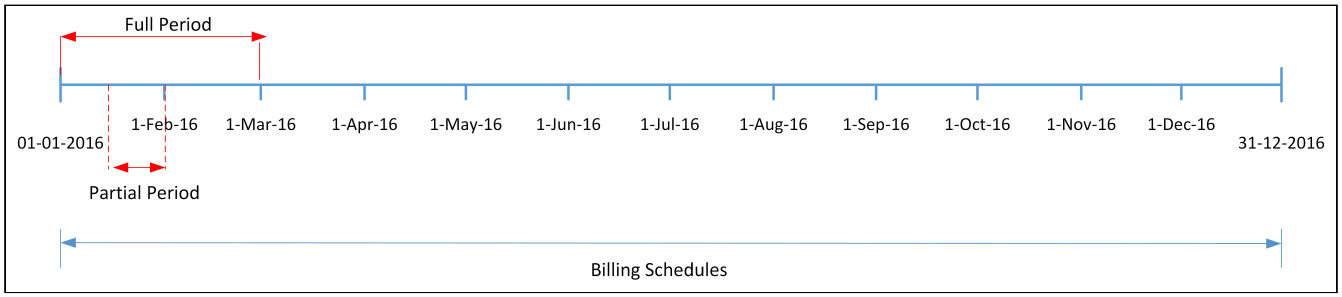
Scenario 1

Start Date: 1/1/2016 **End Date:** 12/31/2016
The customer has opted for a monthly subscription of a product, where he starts using the services on 1st of January 2016 and wants the invoice on 1st of each month.
Here, the customer is consuming the product for a full month. This is a non-prorated scenario because Billing Schedules generated for this purchase will be even and uniformly distributed.

Scenario 2

Start Date: 1/20/2016 **End Date:** 1/19/2017

This scenario is similar to Scenario 1, except the change in Start Date. Customer starts using the services on 20th of January 2016 and wants the invoice on 1st of each month.
In this case, you would want to charge customers only for the duration (11 days) they have used the services rather than charging them for the entire month. To account for this partial period you'll need the charge applicable for just this period which is the prorated amount. This prorated amount must be distributed proportionally between the resulting Billing Schedules.



Now there are different ways through which you can choose to distribute the prorated amount in Billing Schedules and thereby in the invoices,
a) You could account for only 11 days, and send an invoice on 1st of the month,
b) You could combine the 11 days in the first full month and then invoice the customer or
c) You might want to account for these 11 days in the last Billing Schedule of the Billing Cycle, thereby invoicing customer at the end of the contract.

Apttus Billing Management offers the following **Proration Treatment** to comply with your business needs and practices:

- Separate Period

Proration, first of all, derives the prorated amount and second, identifies in which Billing Schedules it should distribute this amount. This is based on your selection of the Proration Treatment, each of which is explained in the following sections.

Separate Period

This is the **default** Proration Treatment. The prorated amount is separated out completely in *two partial periods* at the start and end of Billing Schedules.

Start Date	End Date	Ready for Billing Date	Fee Amount
1/15/2015	1/14/2016	2/1/2015	$100

Schedule	Period Start Date	Period End Date	Fee Amount
BS1	1/15/2015	1/31/2015	50
BS2	2/1/2015	2/28/2015	100
BS12	12/1/2015	12/31/2015	100
BS13	1/1/2016	1/14/2016	50

Combine with the First Period

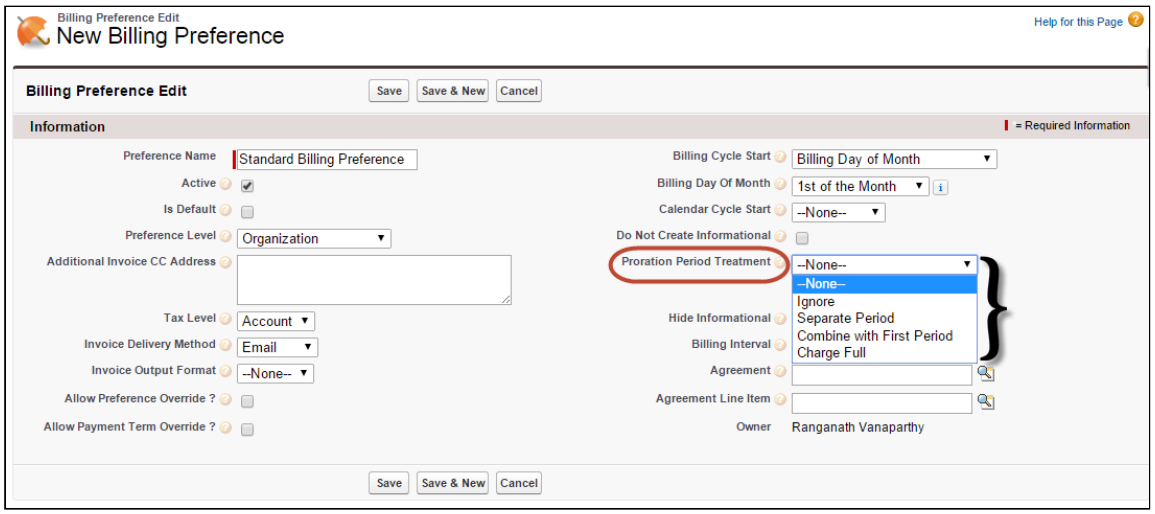
Combine with the first period allows you to invoice the first partial billing schedule in the next full billing schedule. For the prorated billing schedule, the ready for invoice date is set as the invoice date of the next complete billing schedule.

Schedule	Period Start Date	Period End Date	Ready for Invoice Date	Fee Amount
BS1	1/1/2018	3/31/2018	4/1/2018	200
BS2	4/1/2018	6/30/2018	4/1/2018	300
BS3	7/1/2018	9/30/2018	7/1/2018	300
BS4	10/1/2018	12/31/2018	10/1/2018	300

The Apttus Billing Management Proration:

- Identifies if the period is a full period or a partial period.
- Derives the number of partial days.
- Calculates the prorated amount for partial days.
- Picks the applicable **Proration Period Treatment** based on the number of days for each partial period.
- Distributes the prorated amount in rest of the Billing Schedules proportionally.

To apply any one of the Proration behaviors, go to **Billing Preferences** and select a **Proration Period Treatment** as Separate Period. For information on Billing Preferences, please refer to Billing Preferences.



> **Note**
> Proration period Treatment is always **Separate Period** whenever you Add, Remove, Amend or Terminate services in the middle of the Billing Cycle.

Usage-based Billing

Traditionally One-time, Recurring, and Usage are the three types of billing models. With the advent of service based industries, usage-based billing model is quite popular because customers wish to pay only for what they consume.

For example, a high-tech company may offer a monthly cloud-based data storage product. For this product, the pricing is set up so the price per unit decreases as the total usage increases. This pricing model promotes increased usage volume per customer which in turn creates higher recurring revenue. Usage inputs are totaled monthly and rated using a pricing matrix.

The pricing matrix can be tiered as follows:

- 1 GB to 999 GB: $10 per GB
- 1000 GB to 1999 GB: $9 per GB
- 2000 GB to 2999 GB: $8 per GB

In this scenario, you need to use usage-based billing to successfully rate the usage inputs and invoice your customers the correct rated amount.

> ⓘ Flat or rated amount can be set by a matrix, and it cannot be set by tier.

If a product has the **Price Type** set to *Usage*, it is a usage-based product and hence qualifies for the Usage-based billing. When an order containing a usage-based product is activated, Apttus Billing Management generates a Usage Schedule in addition to the Billing Schedule. You can input the usage volume/quantity through Usage Inputs and Apttus Billing Management will derive the usage amount for that input. The net amount due at the time you generate an invoice then depends on the quantity of the units consumed and the per unit price of the product or service.

You must provide a Usage Input to record the usage quantity consumed within a specific duration.

The process flow:

1. Create an order with a usage-based product. The product which is now an activated asset will have a **Usage** and a **Billing Schedule** generated automatically.
2. Enter **Usage Input** to record the quantity consumed over a period.
3. This quantity is reflected in the **Usage Schedules** for the effective period.
4. The rated amount derived from the Usage Input is reflected in the **Billing Schedules** for the effective period.

In a nutshell, Usage Schedules contain the usage quantity entered through Usage Inputs. Whereas Billing Schedules contain the usage amount corresponding to the usage quantity.

Usage Billing Scenario

You must follow 3 major steps to bill a usage-based product:

- Define Product PLI (Price List Item) for Usage
- Apply Price Matrices
- Process Usage Input

With the help of few scenarios, let's see how configuration and processing of a usage-based product are achieved.

Scenario 1: Flat Price with Discrete when Enable Usage Indexing is un-checked

The customer is billed according to the flat rate defined for each fix quantity for a product 'StarKit'. A flat rate is defined for quantity as per the following:

- 10 units = $120
- 20 units = $150
- 30 units= $275
- 40 units = $500

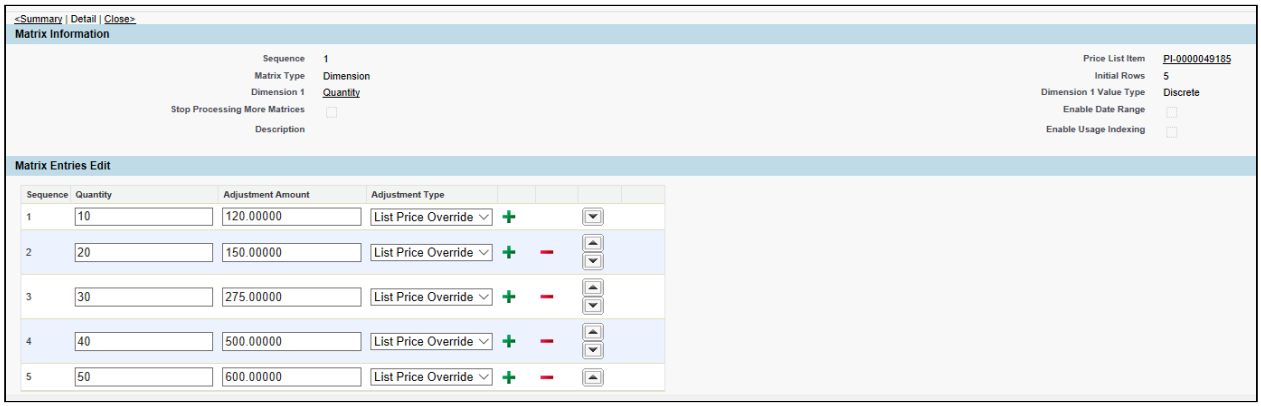
Steps:

PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Flat Price
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Discrete.*
3. Go to Matrix Detail and list the entries as



4. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> ⓘ A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

Process Usage Input

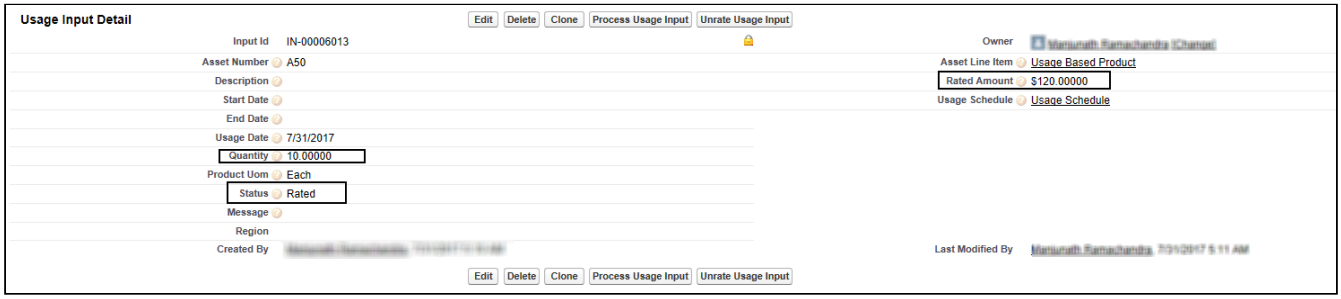
Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.

5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 10, you will see the **Rated Amount** as $ 120.



Scenario 2: Flat Price with Range when Enable Usage Indexing is un-checked

The customer is billed at a per unit rate based on the volume consumed for a product 'StarKit'. A flat rate is defined for each volume slab as per the following:

- 1 to 10 units = $120
- 11 to 20 units = $150
- 21 to 30 units= $275
- 31+ units = $500

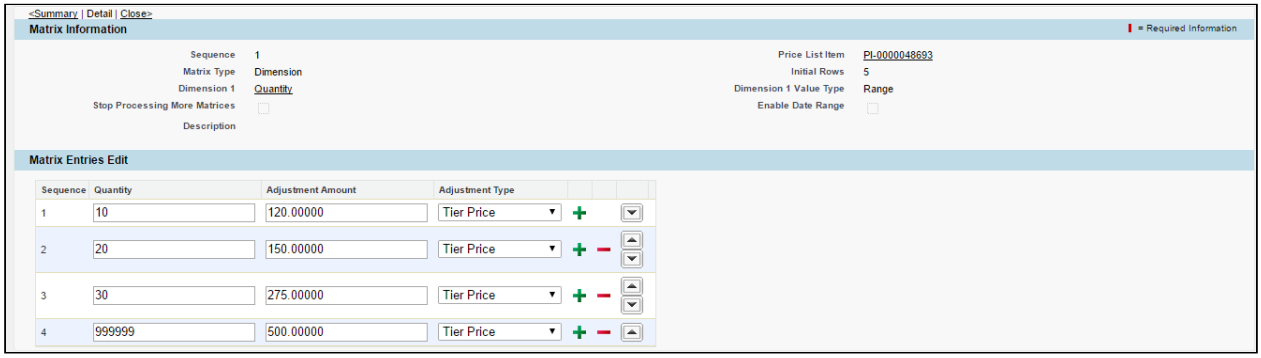
Steps:

PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Flat Price
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Range.*
3. Go to Matrix Detail and list the entries as



4. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

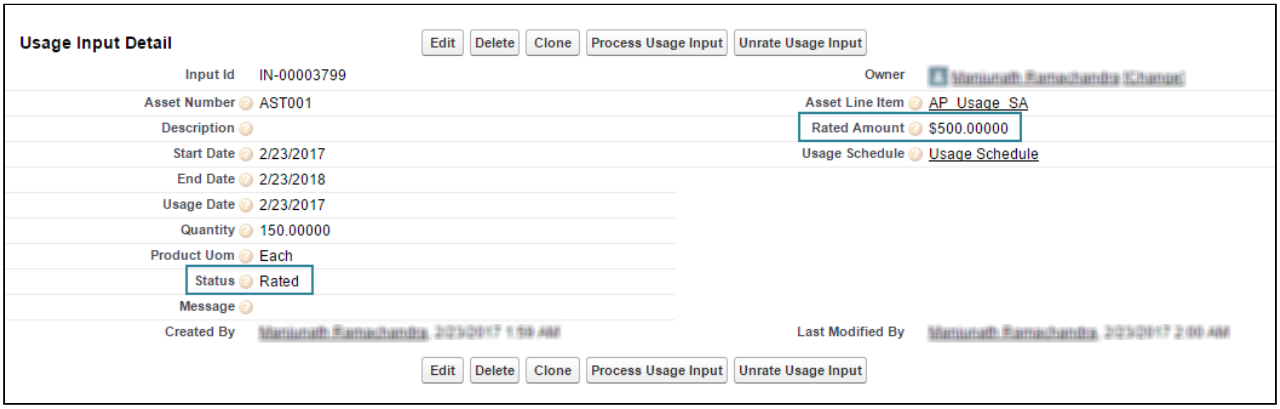
Process Usage Input

Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 150, you will see the **Rated Amount** as $ 500.



Scenario 3: Per Unit with Range when Enable Usage Indexing is un-checked

Scenario 3 - Per Unit with Range when Enable Usage Indexing is un-checked

The customer is billed at a per unit rate based on the volume consumed for a product 'StarKit'. A price range is defined for each volume slab as per the following:

- 1 to 10 units = $120
- 11 to 20 units = $150
- 21 to 30 units= $275
- 31+ units = $500

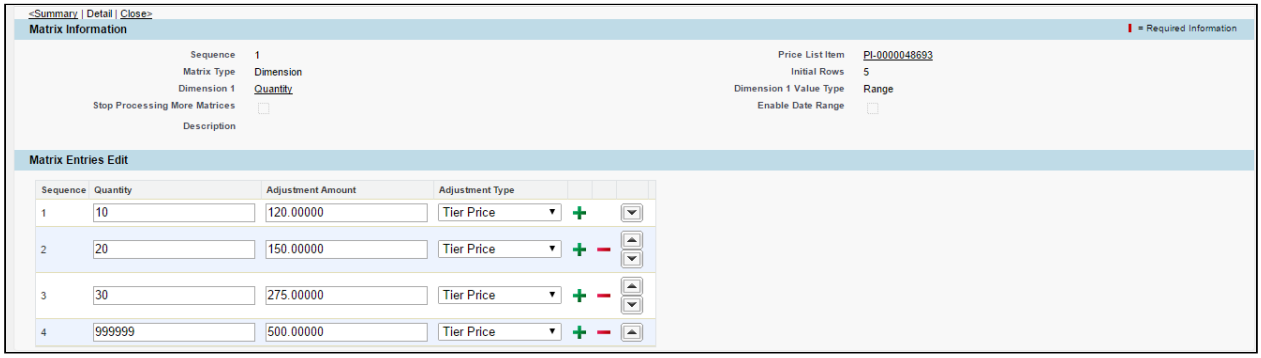
Steps:

PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Per Unit
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Range.*
3. Go to Matrix Detail and list the entries as



4. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

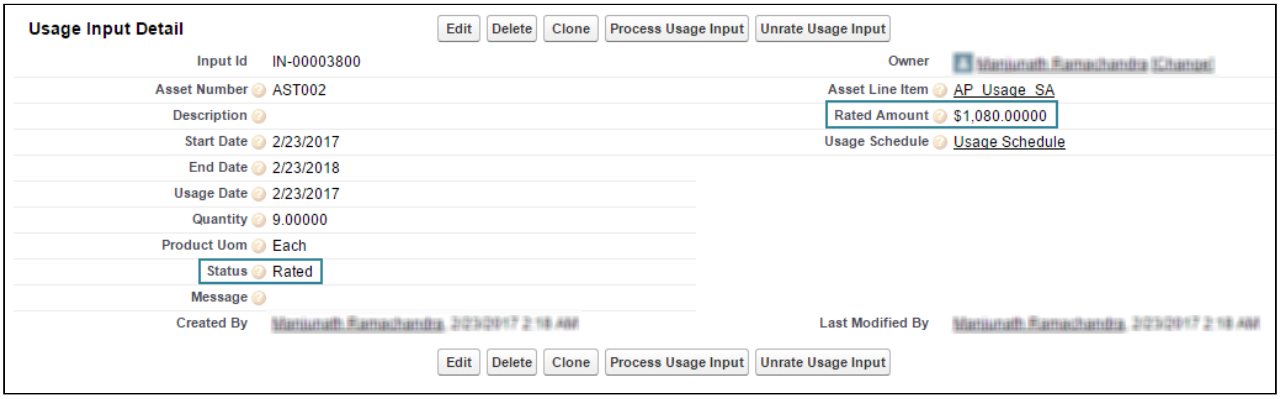
Process Usage Input

Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 9, you will see the **Rated Amount** as $ 1080 (9 * 120).



- Let's take another example. If you specify **Quantity** as 15, you will see the **Rated Amount** as 2250 (15 * 150)

Scenario 4: Per Unit with Cumulative Range when Enable Usage Indexing is un-checked

Scenario 4 - Per Unit with Cumulative Range when Enable Usage Indexing is un-checked

The customer is billed cumulatively based on the volume consumed for a product 'StarKit'. A price range is defined for each volume slab as per the following:

- 1 to 10 units = $120
- 11 to 20 units = $150
- 21 to 30 units= $275
- 31+ units = $500

Steps:

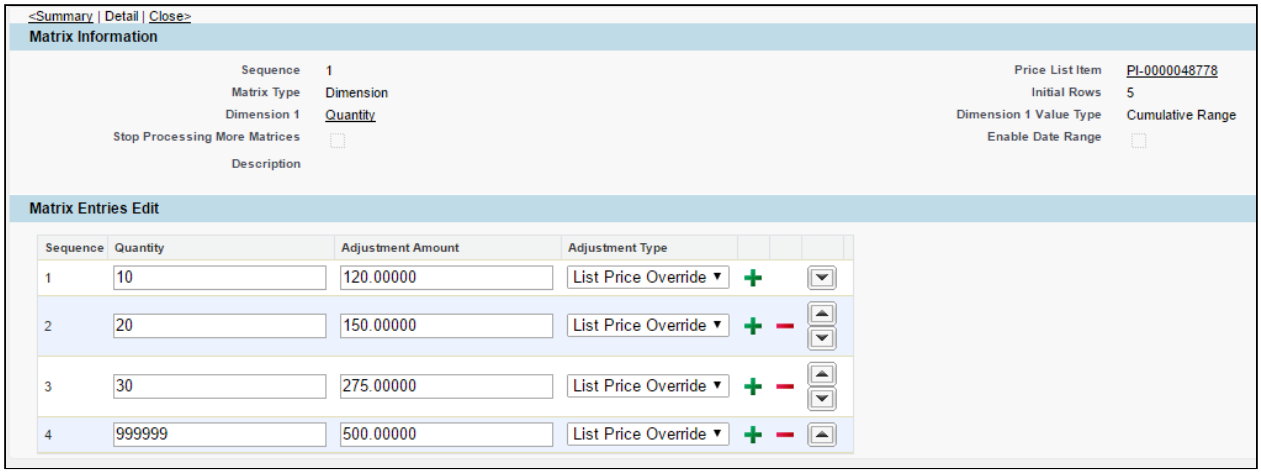
PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Per Unit
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Cumulative Range.*

3. Go to Matrix Detail and list the entries as



> **Note**
> For **Pricing Method** = *Per Unit* and **Dimension1 Value Type** = *Cumulative Range*, you must specify the **Adjustment Type** other than *Tiered Pricing*.

4. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

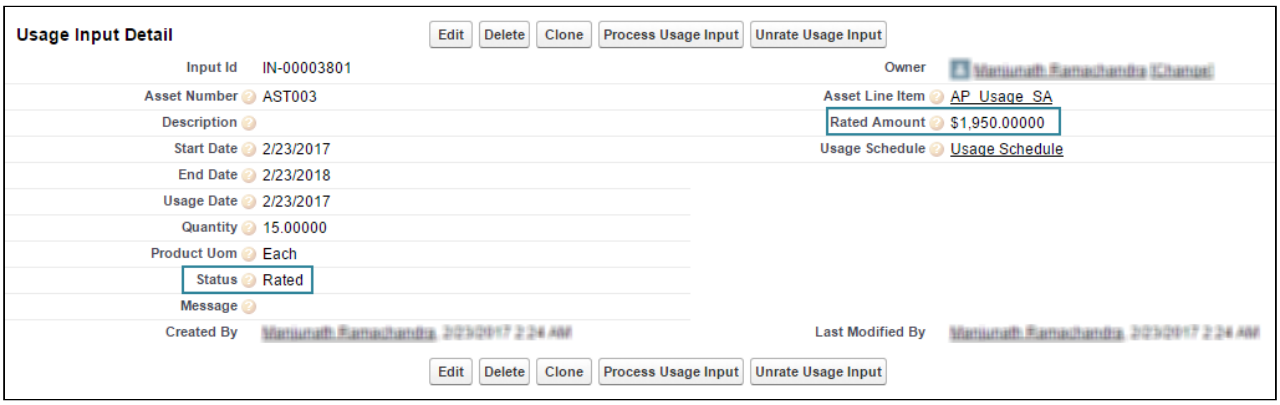
Process Usage Input

Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** that corresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 15, you will see the **Rated Amount** as $ 1950 ((10 * 120) + (5 * 150)).



- Let's take another example. If you specify **Quantity** as 25, you will see the **Rated Amount** as 4075 ((10 * 120) + (10 * 150) + (5 * 275)).

Scenario 5: Flat Price with Cumulative Range when Enable Usage Indexing is un-checked

Scenario 5 - Flat Price with Cumulative Range when Enable Usage Indexing is un-checked

The customer is billed cumulatively based on the volume consumed for a product 'StarKit'. A flat price range is defined for each volume slab as per the following:

- 1 to 10 units = $120
- 11 to 20 units = $150
- 21 to 30 units= $275
- 31+ units = $500

Steps:

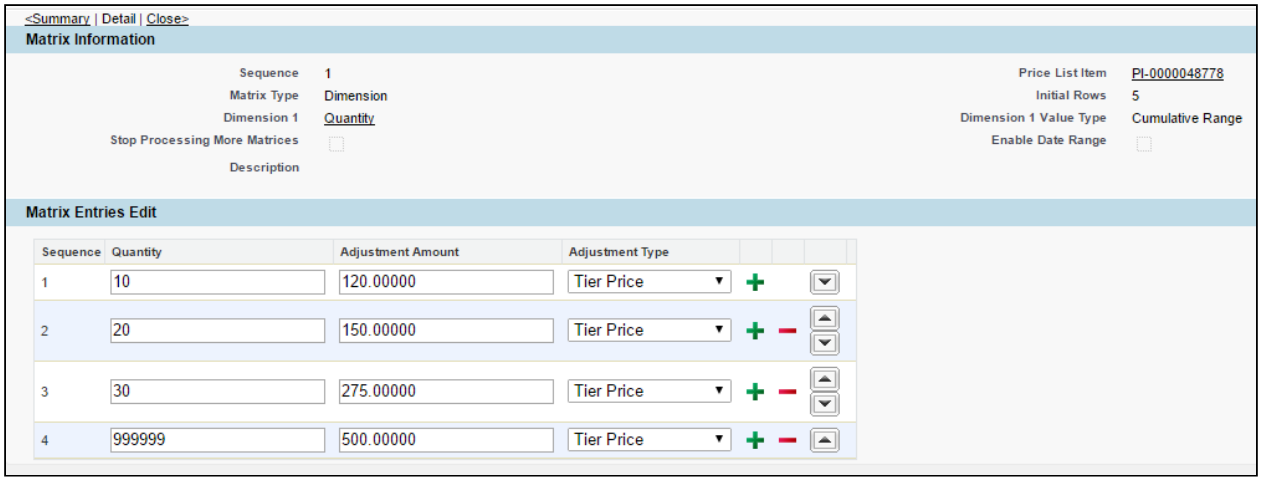
PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Flat Price
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Cumulative Range.*

3. Go to Matrix Detail and list the entries as



4. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

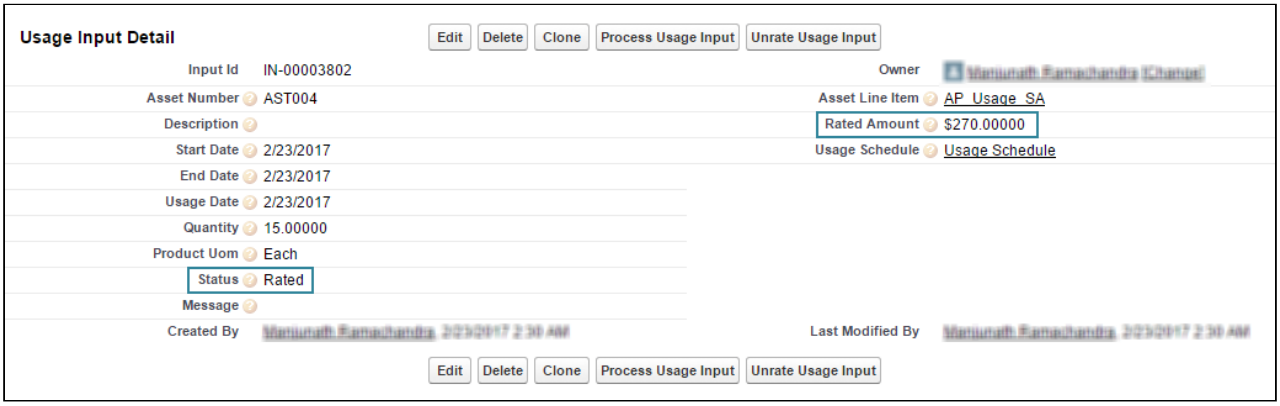
Process Usage Input

Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 15, you will see the **Rated Amount** as $ 270 (120 + 150).



- Let's take another example. If you specify **Quantity** as 25, you will see the **Rated Amount** as 545 (120 + 150 + 275).

Scenario 6: Flat Price with Range when Enable Usage Indexing is checked

Scenario 6 - Flat Price with Range when Enable Usage Indexing is checked

The customer is billed at a per unit rate based on the volume consumed for a product 'StarKit'. A flat rate is defined for each volume slab as per the following:

- 1 to 10 units = $120
- 11 to 20 units = $150
- 21 to 30 units= $275
- 31+ units = $500

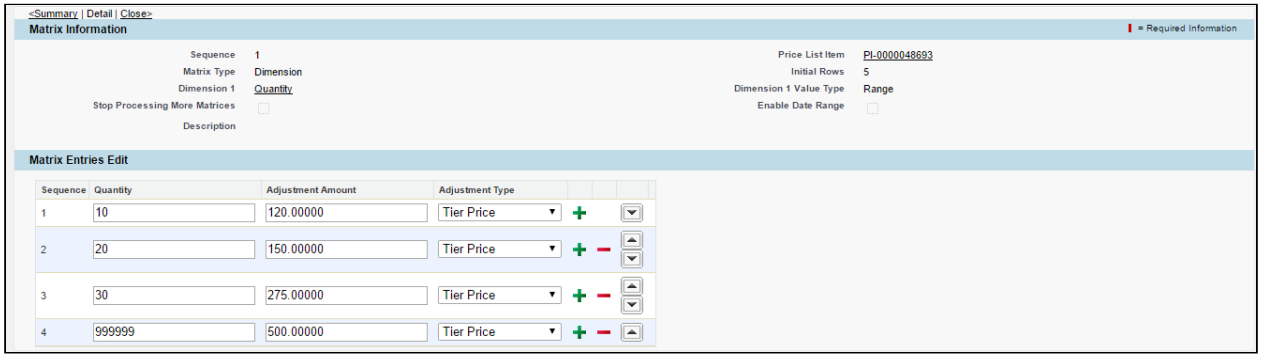
Steps:

PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Flat Price
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Range.*
3. Select **Enable Usage Indexing**.
4. Go to Matrix Detail and list the entries as



5. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

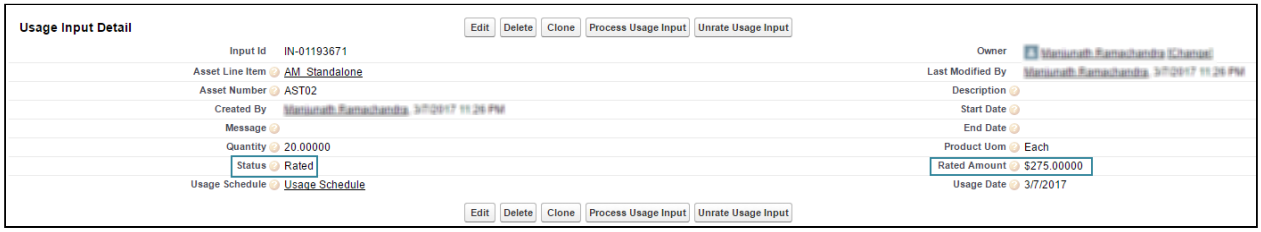
Process Usage Input

Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 5, you will see the **Rated Amount** as 120, since this quantity falls under the first price tier.
- When creating another usageinput with **Quantity** as 20, you will see the Rated Amount as 275. Note that as Usage Indexing is enabled, the total quantity consumed thus far is 25, which falls under the third price tier.



Scenario 7: Per Unit with Range when Enable Usage Indexing is checked

Scenario 7 - Per Unit with Range when Enable Usage Indexing is checked

The customer is billed at a per unit rate based on the volume consumed for a product 'StarKit'. A price range is defined for each volume slab as per the following:

- 1 to 10 units = $120
- 11 to 20 units = $150
- 21 to 30 units= $275
- 31+ units = $500

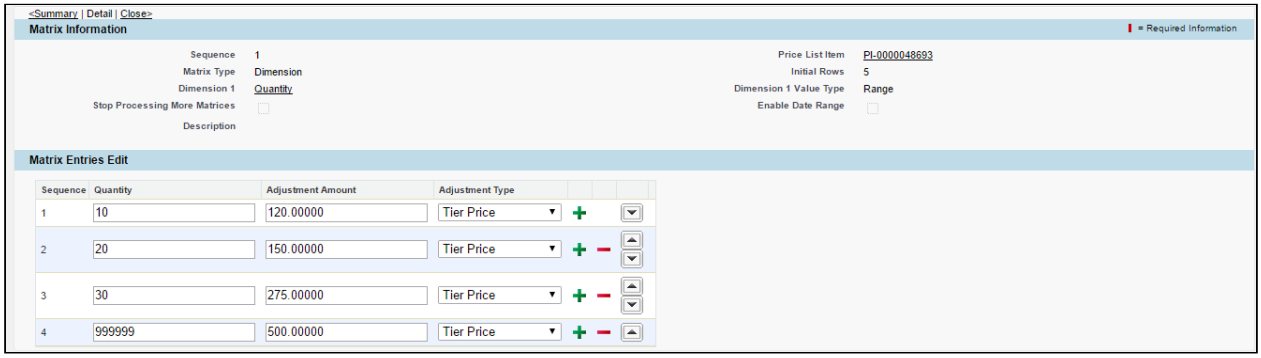
Steps:

PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Per Unit
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Range.*
3. Select **Enable Usage Indexing**.
4. Go to Matrix Detail and list the entries as



5. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

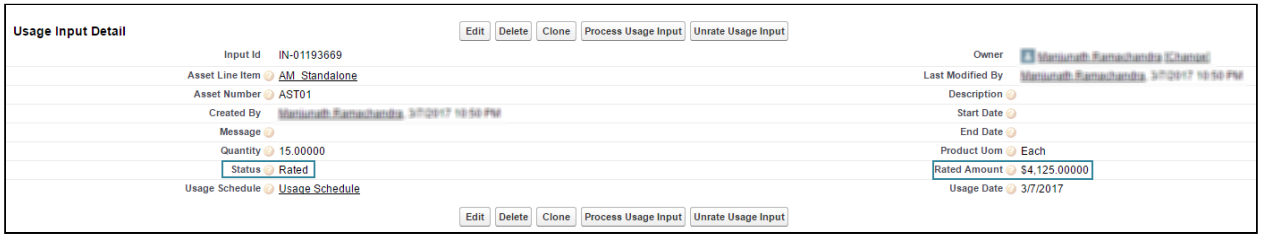
Process Usage Input

Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 5, you will see the **Rated Amount** as $ 600 (5 * 120).
- When creating another usageinput with **Quantity** as 5, you will see the **Rated Amount** as $ 600 (5 * 120). Note that as Usage Indexing is enabled, the total quantity consumed thus far is 10, which falls under the first price tier.
- If you further create another usageinput with **Quantity** as 15, you will see the **Rated Amount** as $ 4125 (15 * 275). Note that as Usage Indexing is enabled, the total quantity consumed thus far is 25, which falls under the third price tier.



Scenario 8: Per Unit with Cumulative Range when Enable Usage Indexing is checked

Scenario 8 - Per Unit with Cumulative Range when Enable Usage Indexing is checked

The customer is billed cumulatively based on the volume consumed for a product 'StarKit'. A price range is defined for each volume slab as per the following:

- 1 to 10 units = $120
- 11 to 20 units = $150
- 21 to 30 units= $275
- 31+ units = $500

Steps:

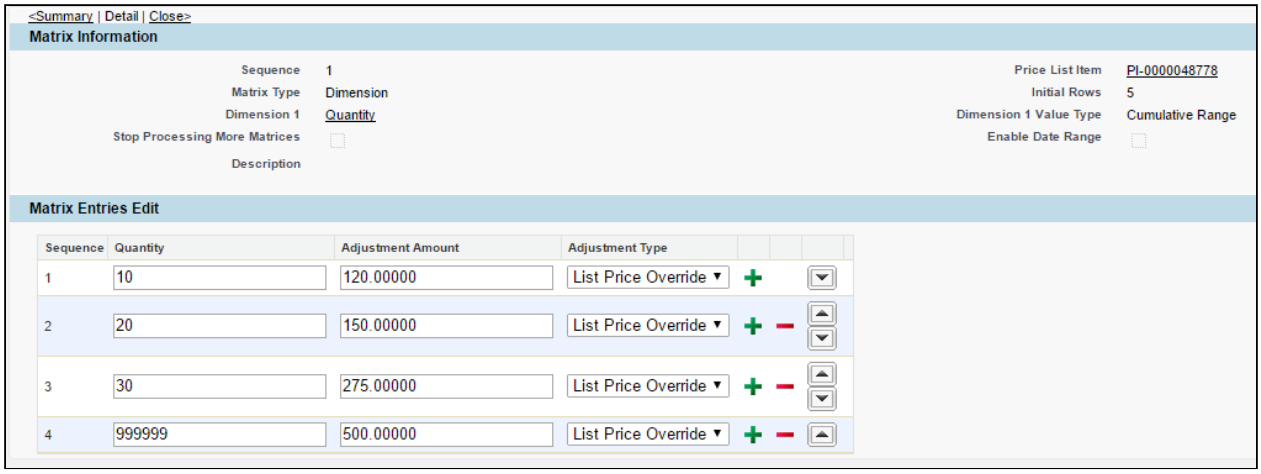
PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Per Unit
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Cumulative Range.*
3. Select **Enable Usage Indexing**.

4. Go to Matrix Detail and list the entries as



> **Note**
> For **Pricing Method** = *Per Unit* and **Dimension1 Value Type** = *Cumulative Range*, you must specify the **Adjustment Type** other than *Tiered Pricing*.

5. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

Process Usage Input

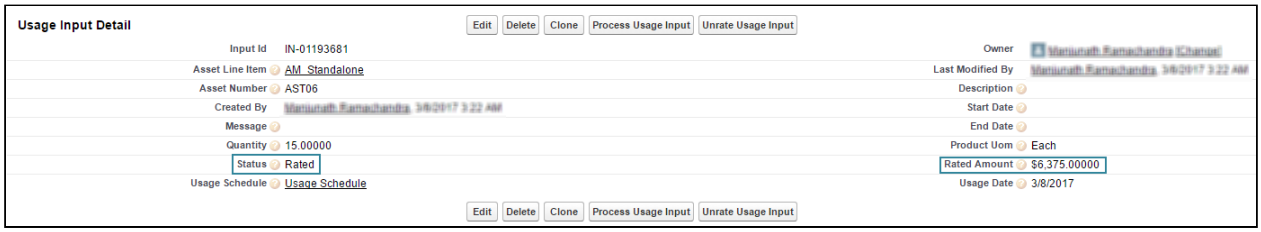
Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 5, you will see the **Rated Amount** as $ 600 (5 * 120).

- When creating another usage input with **Quantity** as 20, you will see the **Rated Amount** as $ 3475 ((5 * 120) + (10 * 150) + (5 * 275)). Note that as Usage Indexing is enabled, the total quantity consumed thus far is 25. Since it is a cumulative range, the **Rated Amount** is the total amount of first, second, and third cumulative price tiers amount calculated for per unit consumption.
- If you further create another usage input with **Quantity** as 15, you will see the **Rated Amount** as $ 6375 ((5 * 275) + (10 * 500)). Note that as Usage Indexing is enabled, the total quantity consumed thus far is 40. Since it is a cumulative range, the **Rated Amount** is the total amount of third and fourth cumulative price tiers amount calculated for per unit consumption.



Scenario 9: Flat Price with Cumulative Range when Enable Usage Indexing is checked

Scenario 9 - Flat Price with Cumulative Range when Enable Usage Indexing is checked

The customer is billed cumulatively based on the volume consumed for a product 'StarKit'. A flat price range is defined for each volume slab as per the following:

- 1 to 10 units = $120
- 11 to 20 units = $150
- 21 to 30 units= $275
- 31+ units = $500

Steps:

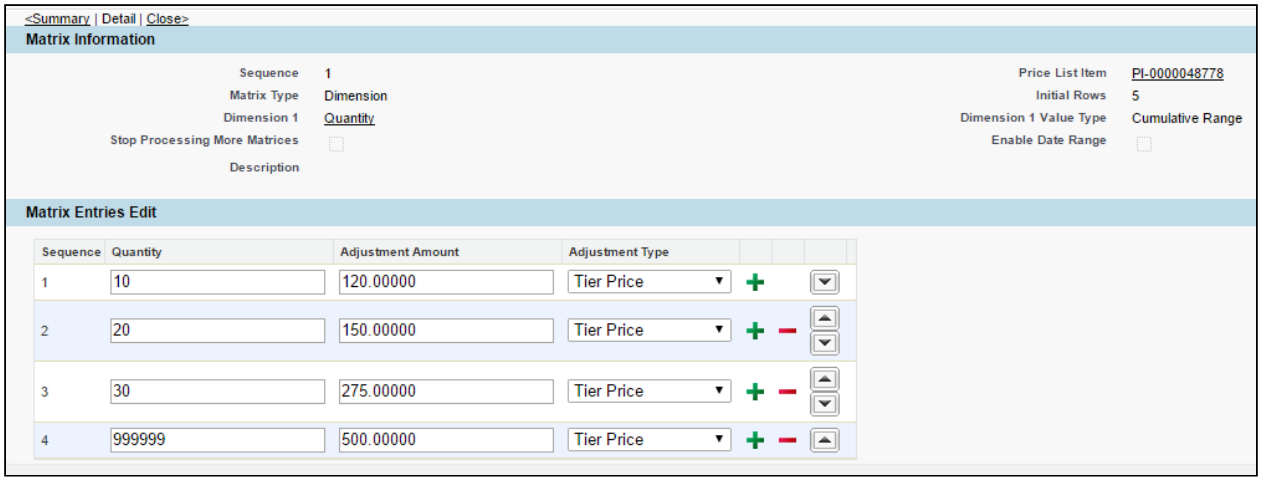
PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Flat Price
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Cumulative Range.*
3. Select **Enable Usage Indexing**.

4. Go to Matrix Detail and list the entries as



5. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

Process Usage Input

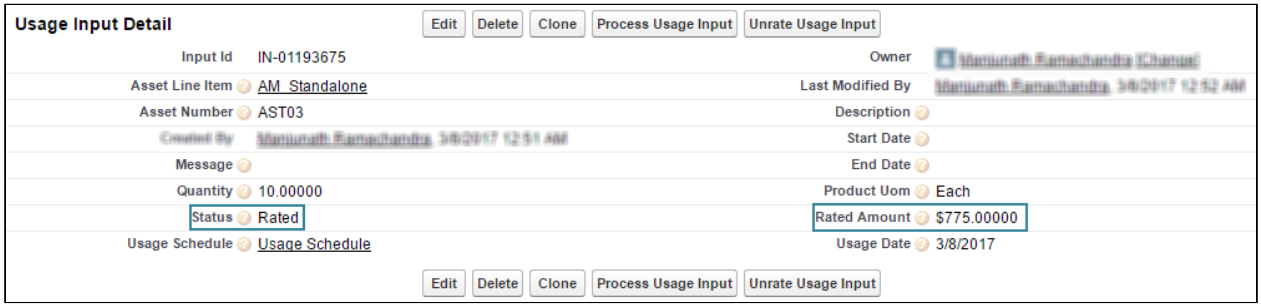
Go to **Usage Inputs** object.

1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 5, you will see the **Rated Amount** as $ 120.
- When creating another usageinput with **Quantity** as 20, you will see the **Rated Amount** as $ 545 (120 + 150 + 275). Note that as Usage Indexing is enabled, the total quantity consumed thus far is 25, which falls under the third price tier. Since it is a cumulative range, the **Rated Amount** is the total amount of first, second, and third cumulative price tiers amount.

- If you further create another usage input with **Quantity** as 10, you will see the **Rated Amount** as $ 775 (275 + 500). Note that as Usage Indexing is enabled, the total quantity consumed thus far is 35, which falls under the fourth price tier. Since it is a cumulative range, the **Rated Amount** is the total amount of third and fourth cumulative price tiers amount.



Scenario 10: Usage Calculation using Multi-dimensional Price Matrix

Scenario 10 - Usage Calculation using Multi-dimensional Price Matrix

In all the above-mentioned scenarios, we have considered only one dimension that is quantity. However, it is possible to configure the PLI to have multiple dimensions.

Consider a case where you need to bill your customer based on the volume of the quantity consumed. The price for each volume tier varies as per the customer rating. In such a scenario, you need two dimensions, quantity and customer rating.

For example, the customer is billed at a flat rate based on the volume consumed for a product 'StarKit'. As per the customer rating, different flat rates are defined for each volume slab as per the following:

Units	Customer Rating	
	Gold	Silver
1 - 10	100	120
11 - 20	180	200
21 - 30	255	275
31 +	480	500

Ensure that following prerequisites are satisfied:

1. A field named **Customer Rating** on the **Account** object.
2. New price dimension called **Customer Rating** created on the basis of the above field.

3. A field named **Customer Rating** on the **Usage Input** object

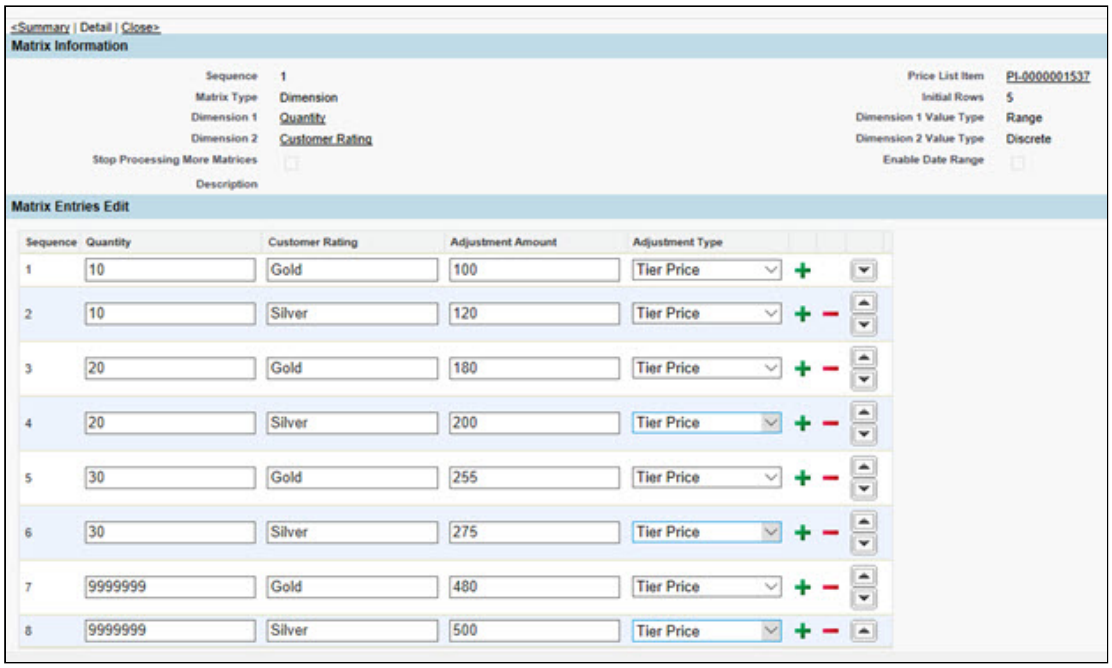
Steps:

PLI Configuration

1. The first step is to configure the PLI for the Starkit. Set following values for the given fields.
   Price Type= Usage
   Frequency= Monthly
   Price Method= Flat Price
   Price UOM= Each
2. To enable a user to modify the usage tier, select **Is Usage Tier Modifiable**.

Apply Price Matrix

1. From the Price List Item tabs, open the **Matrices** tab and click **New** to add a new Price Matrix.
2. Select **Matrix Type** as *Dimension.*
   On the Detail page, **Dimension1** as *Quantity* and **Dimension1Value Type** as *Range.* **Dimension2** as *Customer Rating*, and **Dimension2Value Type** as *Discrete.*
3. Go to Matrix Detail and list the entries as



4. The tiered pricing for this product is inherited from the Price Matrix. After defining the tiered rates, click **Save**. The PLI configuration for a Usage product is complete.

> **Note**
> A usage product is always billed in Arrears. It is recommended that you set the **Billing Rule** to **Bill In Arrears**.

Create an order with this usage-based product. The product which is now an activated asset will have a Usage and a Billing Schedule generated automatically. Next, you should provide a Usage Input to record the usage quantity consumed within a specific duration.

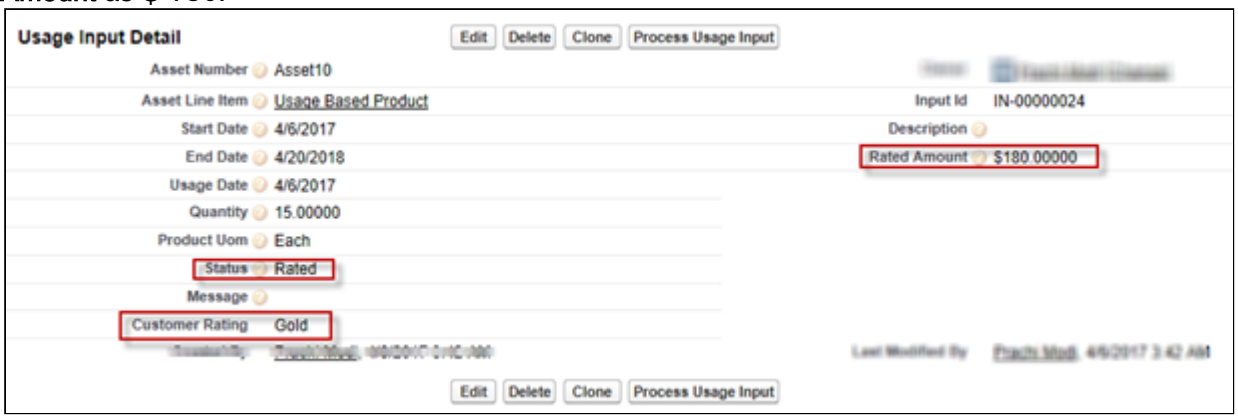
Process Usage Input

Go to **Usage Inputs** object.

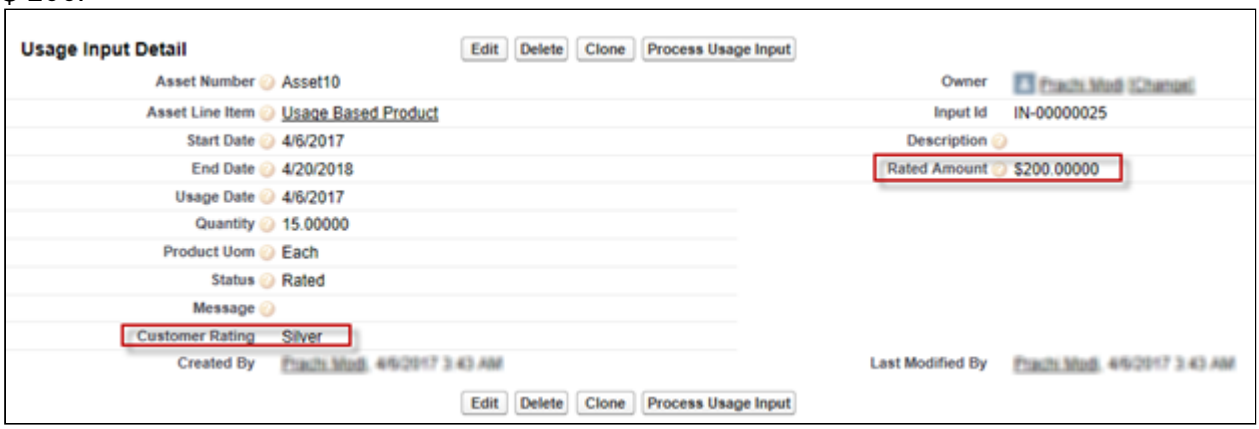
1. Enter the **Asset Number**. This asset number must match with the corresponding asset number that you have specified in the **Asset Line Item Detail** page.
2. Enter a **Usage Date** thatcorresponds with the Billing Schedule.
3. Enter the **Product UOM**. This value should match the value provided on the Product PLI.
4. Ensure that **Status** is *Loaded*.
5. Enter the **Quantity** and click **Save**.

After saving the input, click **Process Usage Input** to process the amount against the entered quantity. When the usage is processed successfully, the **Status** will change to **Rated** and you will see the amount under **Rated Amount**.

- For example, if you specify **Quantity** as 15 and **Customer Rating** as Gold, you will see the **Rated Amount** as $ 180.



- And if you specify **Quantity** as 15 and **Customer Rating** as Silver, you will see the **Rated Amount** as $ 200.



For details about configuring and processing usage input, refer to the Usage Input.

Usage Input

Usage input taps the raw, unrated usage in the system based on the quantity consumption. Usage price tiers and ratings are also associated with the Usage input object.

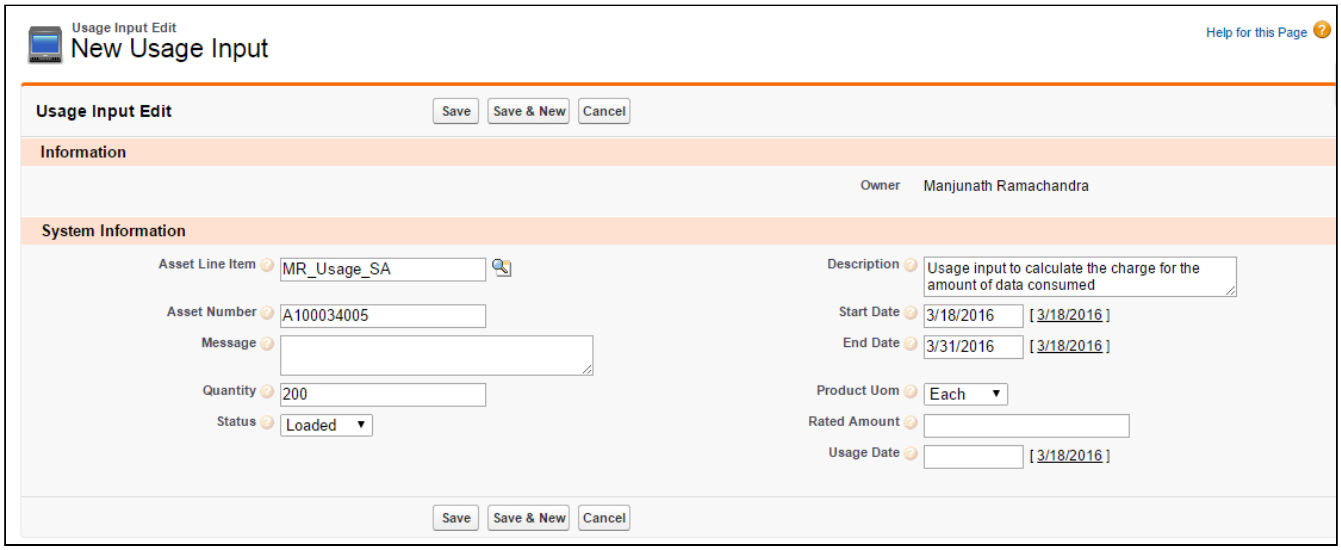
You can,

- Enter the quantity through Usage Input
- Process it to rate or
- Unrate the usage.

You could also run a batch job to process or unrate the usage automatically.

For example, you are an executive in the Billing department of an Internet Service Provider. You must capture and process the amount of data your customers have used so that you can calculate the amount that you will generate an invoice for. In the Apttus Billing Management, you would create a *new* **Usage Input** and process it in order to rate that usage.

With a scheduled batch job, Apttus Billing Management processes the Usage Input to rate the usage and automatically updates the rated amount to the associated Billing Schedule and quantity to Usage Schedule.



To create a New Usage Input

1. Click **Usage Input** > **New**.
2. Enter values for the fields described in the following table:

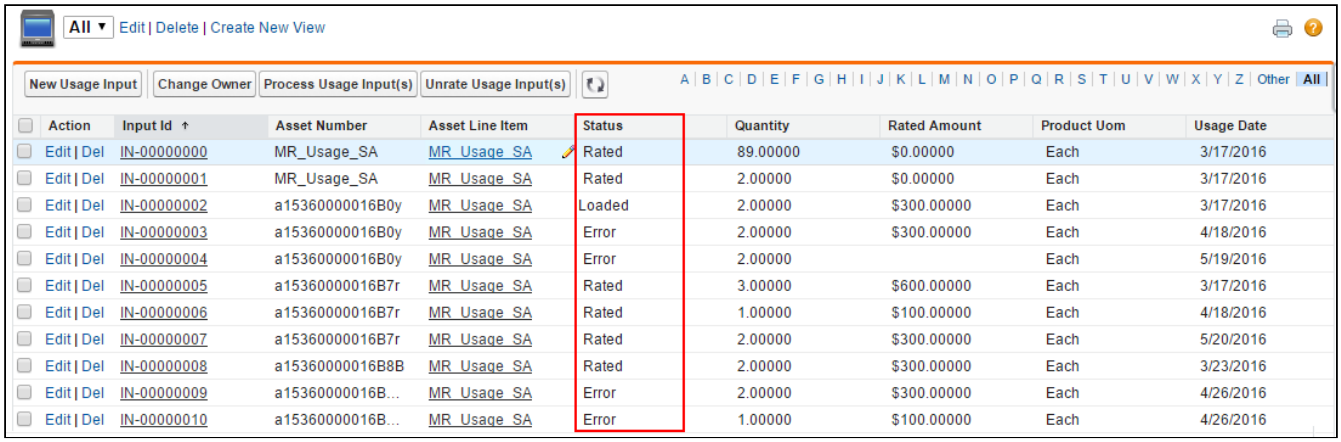
Field	Description
Asset Line Item	Lookup and select the Asset Line Item for which you want to process the Usage Input.
Asset Number	Enter the Asset Number you for which you want to process the Usage Input.
Description	Describe the purpose of this Usage Input.
Message	Auto-generated after the input processing. The message corresponds to the Processing Status. Please refer Usage Input Status.
Status	The stage of the usage input processing. It can be any one of the following options: • **Loaded** - If this Usage Input is ready to be processed. • **Assigned** - If this Usage Input is assigned to a user for processing. • **Ready** - If this Usage Input is ready to be rated. • **Rated** - If this Usage Input is rated and ready to be loaded. • **Error** - If some parameters to process this usage input are incorrect or unavailable. For more details, please refer Usage Input Status.
Usage Date	From the calendar, select the date when you want to calculate the Usage Input.
Start Date	From the calendar, select the date from when you want to calculate the start of the usage.
End Date	From the calendar, select the date until when you want to calculate the end of the usage.
Quantity	Enter the number of units of the asset for which you want to process the Usage Input.
Product UoM	Select any one of the following options to calculate the **Product Unit of Measure**: • **Each** • **Hour** • **Day** • **Month** • **Year** • **Quarter** • **Case** • **Gallon**
Rated Amount	This field is automatically populated with the amount based on Quantity after you process the Usage Input.

3. Click **Save**.

Processing Usage Input

Usage Input processing is a batch process run by the system that derives rated amount from a given quantity, updates the Usage and Billing Schedules and sets the **Input Status** to *Rated* if the input data is accurate.

You can process multiple usage inputs in one go from the **Usage Inputs** Homepage. At the top of the page, next to **View**, Click **Go** to see all Usage Inputs entered in the system.



Action	Input Id	Asset Number	Asset Line Item	Status	Quantity	Rated Amount	Product Uom	Usage Date
Edit | Del	IN-00000000	MR_Usage_SA	MR_Usage_SA	Rated	89.00000	$0.00000	Each	3/17/2016
Edit | Del	IN-00000001	MR_Usage_SA	MR_Usage_SA	Rated	2.00000	$0.00000	Each	3/17/2016
Edit | Del	IN-00000002	a15360000016B0y	MR_Usage_SA	Loaded	2.00000	$300.00000	Each	3/17/2016
Edit | Del	IN-00000003	a15360000016B0y	MR_Usage_SA	Error	2.00000	$300.00000	Each	4/18/2016
Edit | Del	IN-00000004	a15360000016B0y	MR_Usage_SA	Error	2.00000		Each	5/19/2016
Edit | Del	IN-00000005	a15360000016B7r	MR_Usage_SA	Rated	3.00000	$600.00000	Each	3/17/2016
Edit | Del	IN-00000006	a15360000016B7r	MR_Usage_SA	Rated	1.00000	$100.00000	Each	4/18/2016
Edit | Del	IN-00000007	a15360000016B7r	MR_Usage_SA	Rated	2.00000	$300.00000	Each	5/20/2016
Edit | Del	IN-00000008	a15360000016B8B	MR_Usage_SA	Rated	2.00000	$300.00000	Each	3/23/2016
Edit | Del	IN-00000009	a15360000016B...	MR_Usage_SA	Error	2.00000	$300.00000	Each	4/26/2016
Edit | Del	IN-00000010	a15360000016B...	MR_Usage_SA	Error	1.00000	$100.00000	Each	4/26/2016

Select the check box for those Usage Inputs which you want to process and click **Process Usage Input(s)**.

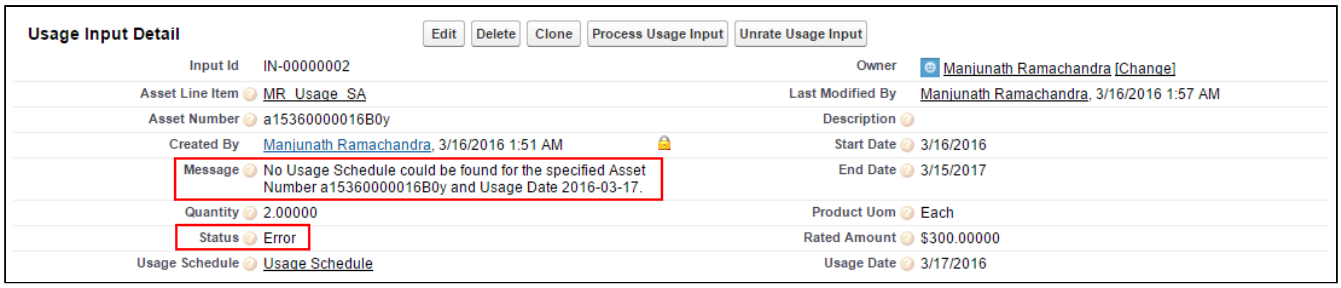
> **Note**
> You can process usage only for those inputs which have the **Status** as *Loaded*.

Similarly, to unrate the processed inputs in bulk, select the check box for multiple inputs and click **Unrate Usage Input(s)**.

The processing is unsuccessful in case,

- The Asset Line Item does not exist for the given Asset Number.
- There is no usage schedule found for the Asset as mentioned on the Usage Date.

In both cases, the Usage Input **Status** will be set to *Error*. Go to **Usage Input Detail** page to view the **Message** to understand the cause of error.



You can also Unrate a processed usage Input from the **Usage Input Detail** view if inaccurate quantity or pricing has been processed. Click **Unrate Usage Input**.

With the Apttus Billing Management application, when you process usage inputs, you will receive an email notification at the conclusion of the process with the following information:

- Number of Usage Inputs processed: X
- Number of Usage Inputs processed successfully: X
- Number of Usage Inputs processed with errors: X

Check out Usage Schedules for the successfully *Rated* Usage Inputs.

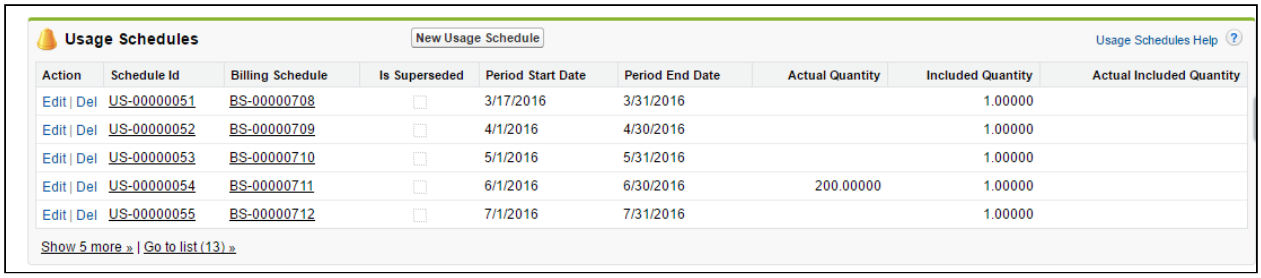
Usage Input Status

Action	Status (Pre)	Message	Status (Post)
Process Usage Input	Loaded	-	Rated
	Error	• *No Usage Schedule could be found for the specified Asset Number and Usage Date* (Verify the Start Date and End Date) • *Invalid Asset Number* (Confirm if the Asset Number is valid) • *Invalid Usage Date* (The Usage Start and End Date should lie between the Billing Schedules)	No change
	Rated	Usage Input with Status as Loaded can only be processed.	No change
Unrate Usage Input	Loaded	Usage Input with Status as Rated can only be unrated.	No change
	Error	Usage Input with Status as Rated can only be unrated.	No change
	Rated	Usage Input has been unrated.	Error

Usage Schedules

On the creation of a usage-based asset, Usage Schedules are automatically generated. Usage Schedules record the quantity consumed for each Billing Schedule.

1. On the **Account** page, go to **Assets** to view the **Usage Schedules** under the Asset related lists.



Field Description

Field	Description
Billing Schedule	The corresponding Billing Schedule for this Usage Schedule.
Is Superseded	Checked if the usage schedule is superseded due to amendments.
Period Start Date	Select the date from which you want to calculate your customer's usage.
Period End Date	Select the date until which you want to calculate your customer's usage.
Actual Quantity	The quantity included at no charge in the billing schedule.
Included Quantity	The number of units of the product you want to include for billing your customer.

You must enter the usage or metering data associated with an order and then link this usage rating to a Usage Schedule. You can do so by entering a Usage Input.

Related A/R Transactions

Apttus Billing Management communicates with third-party secure payment gateways to receive the payment information on invoices.
When customer receives an invoice and pays using any third-party payment solution, the payment detail such as amount, mode, date and other information is passed on to the Apttus Billing Management System and registered as a related A/R transaction corresponding to that transaction type. Transactions can be of different types like payment, invoice, credit memo, refunds etc. In order to handle a variety of transactions, A/R Transaction object is categorized into

- Source Related A/R transactions
- Destination Related A/R Transactions

The aim is to have many to one relationship between the Source and the Destination transactions. For example, there can be multiple sources like Credit Memo or Payment which are issued against an invoice which is the destination object. Therefore, whenever a new invoice is generated, a **Destination Related A/R Transaction** (under **Invoice** related list) is created.

For payment against an invoice, a **Source Related A/R** of the type *Payment* is added under the invoice related list. If you issue a Credit Memo against an invoice, a **Source Related A/R** of the type *Credit Memo* is added under the invoice related list.

Invoice Creation

Invoice ID- INV101

Source A/R

Txn Type	Starting Balance	Ending Balance	Source Invoice ID	Destination Invoice ID	Source Payment ID	Destination Payment ID	Source Credit Memo ID	Destination Credit Memo ID

Destination A/R

Txn Type	Starting Balance	Ending Balance	Source Invoice ID	Destination Invoice ID	Source Payment ID	Destination Payment ID	Source Credit Memo ID	Destination Credit Memo ID
Invoice Creation	$0	$1200		INV-101				

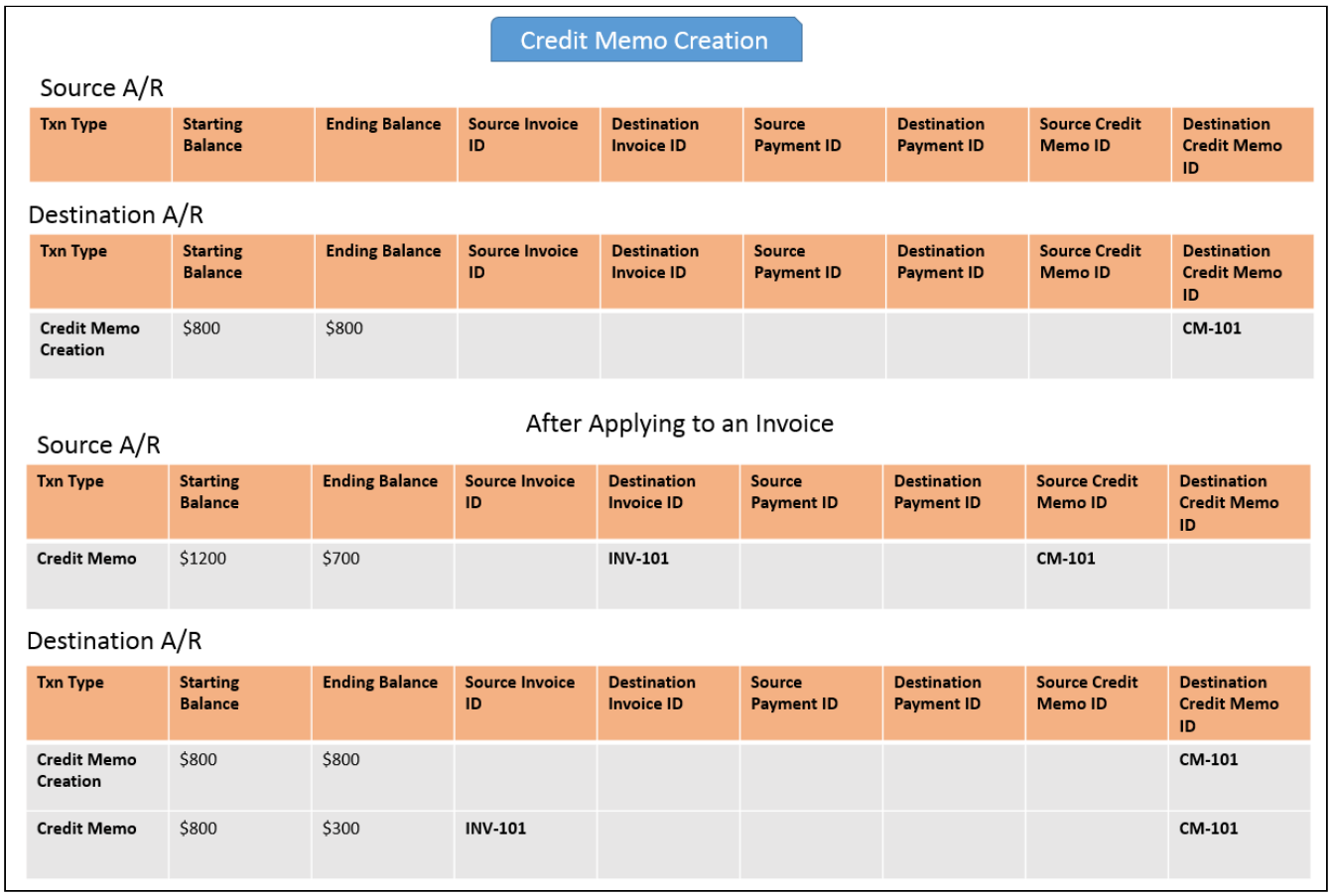
After Applying a Credit Memo of $500

Source A/R

Txn Type	Starting Balance	Ending Balance	Source Invoice ID	Destination Invoice ID	Source Payment ID	Destination Payment ID	Source Credit Memo ID	Destination Credit Memo ID
Credit Memo	$800	$300	INV-101					CM1-01

Destination A/R

Txn Type	Starting Balance	Ending Balance	Source Invoice ID	Destination Invoice ID	Source Payment ID	Destination Payment ID	Source Credit Memo ID	Destination Credit Memo ID
Invoice Creation	$0	$1200		INV-101				
Credit Memo	$1200	$700		INV-101			CM-101	



Credit Memo Creation

Source A/R

Txn Type	Starting Balance	Ending Balance	Source Invoice ID	Destination Invoice ID	Source Payment ID	Destination Payment ID	Source Credit Memo ID	Destination Credit Memo ID

Destination A/R

Txn Type	Starting Balance	Ending Balance	Source Invoice ID	Destination Invoice ID	Source Payment ID	Destination Payment ID	Source Credit Memo ID	Destination Credit Memo ID
Credit Memo Creation	$800	$800						CM-101

After Applying to an Invoice

Source A/R

Txn Type	Starting Balance	Ending Balance	Source Invoice ID	Destination Invoice ID	Source Payment ID	Destination Payment ID	Source Credit Memo ID	Destination Credit Memo ID
Credit Memo	$1200	$700		INV-101			CM-101	

Destination A/R

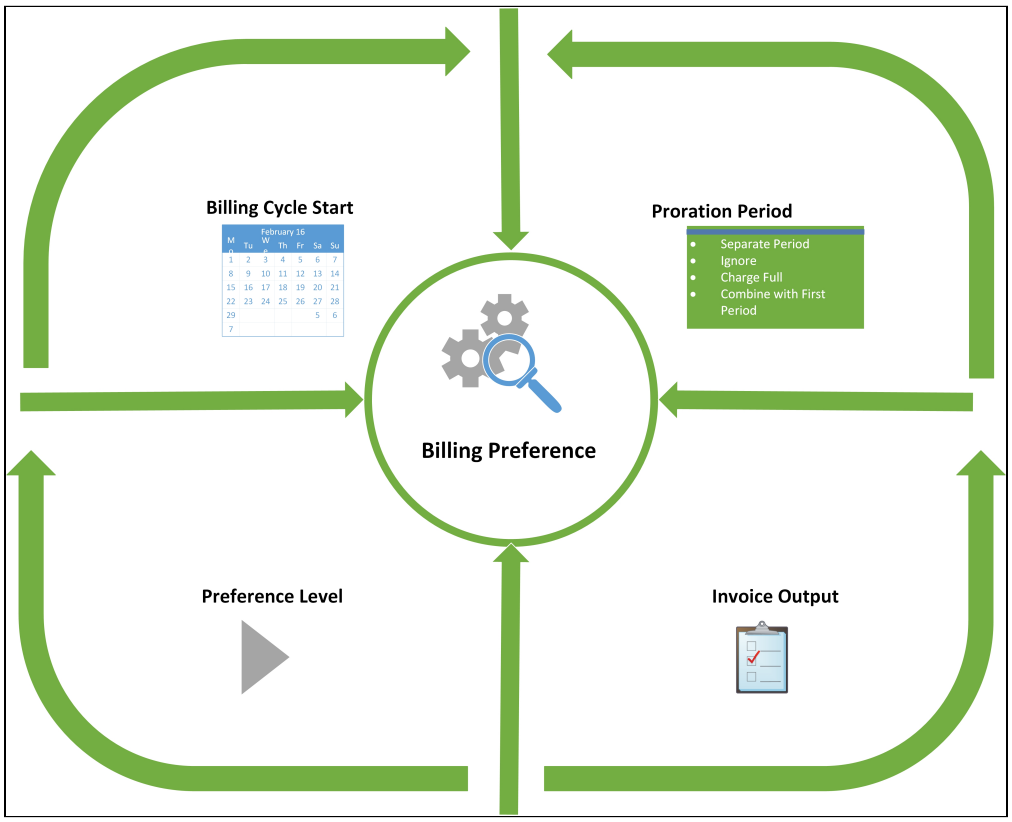
Txn Type	Starting Balance	Ending Balance	Source Invoice ID	Destination Invoice ID	Source Payment ID	Destination Payment ID	Source Credit Memo ID	Destination Credit Memo ID
Credit Memo Creation	$800	$800						CM-101
Credit Memo	$800	$300	INV-101					CM-101

Billing Preferences

As a billing system admin, you have to perform following configurations- for different accounts, generate bills on different days of the month (1st of the month for one account, 15th of the month for other accounts and so forth). For some accounts, the invoices should be generated in a pdf format and for some, in a word format. Few accounts want the bills to be generated every month and the rest every quarter.

Each of these requirements apparently involves different configuration at account level- making this a tedious task. The best approach would be to group all these settings under one roof and apply them in one go. Billing Preferences does all this; it is a container which holds many key settings that are necessary to drive your Billing. You can feed in your choices for various fields in the Billing Preference and then associate Billing Preferences to an entire Org, Account or a Quote/Proposal or Quote/Proposal Line. You can use this functionality to define each customer's billing preference to suit their requirements and convenience.

Billing Preferences drive the alignment of Billing Schedules. Later if you want to make any changes to the preference, you can update the Billing Preference through an asset-based order.



You can create several billing preferences for one customer and set each preference at a different level. The Billing Preference that you define, determines the

- Billing Date and Month,
- Billing frequency,
- Applicable Taxes, and
- Method and format of Invoice delivery.

With this feature, you can select a preference for every new invoice that you generate for that customer. If you do not select a Billing Preference, a default preference is applied when you generate an invoice. You can generate an Invoice for an account even if the Billing Preference you select is inactive. You must activate only one Billing Preference at a time.

To create a Billing Preference

1. Click **Billing Preference > New**.
2. Enter values for the fields described in the following table.

Field	Description
Preference Name	Enter a unique name for the Billing Preference you want to create. It is recommended that you keep an easily identifiable name.
Preference Level	Select the level at which you want to apply this Billing Preference.
Billing day of the month	Day of the month when you want to bill your customer. For example, to set '10' as the Billing Day, select **10th of the month** from the picklist. If you select **End of the month**, the Billing Day is set to the **last day** of each month as and when the schedule advances. For example, Billing Day of Month will be '31' for March and '30' for April. **Note:** This field holds significance only if you set 'Billing Cycle Start' to 'Billing Day of Month'.
Billing Cycle Start	Billing Cycle Start can be any day on which you want to bill customers. Billing Schedules are aligned on the basis of Billing Cycle Start. Billing Schedule for an asset takes the **Expected Start Date** mentioned on the Proposal Line and Order Line as the starting point for generating the schedules. To align rest of the duration, it looks up to the value provided in **Billing Cycle Start**. Please refer this example. Select one of the following options: • **Billing Day of Month** - value specified in the **Billing Day of Month** field. • **Ready for Billing Date** - value from **Ready for Billing Date** field on the accepted Quote/Proposal. • **Period Start Date** - value from **Expected start date** field on Asset Line Item. • **Order Date** - value from **Order Date** field in the activated order. • **Account Billing Day of Month** - value from **Billing Day of Month** field on Accounts. Select this option, if you want to set a common billing day for all orders in an account.
Calendar Cycle Start	Select the month to align your customer's billing to a specific business calendar year. To set calendar cycle start from an account, set **Calendar Cycle Start** to *Account Calendar Cycle Start.* Next, select a month from **Calendar Cycle Start** on the Account.
Tax Level	Select the object to which you want to apply taxes, from one of the following options: • **Account** • **Product**
Invoice Delivery Method	Select the preference for Invoice Delivery from Email or Print.
Invoice Output format	Select the format from one of the following options: • **PDF** • **DOC** • **DOCX** • **RTF**

Field	Description
Allow Payment Term Override	Select this check box if you want to override the default Payment Term.
Allow Preference Override	Select this check box if you want to override this Billing Preference.
Do not Create Informational	Select this option to suppress the creation of all Informational Billing Schedules. For Bundle products, if the **Bundle Invoice Level** is set to **Bundle**, Billing Schedules for Bundle Options will not be created.
Hide Informational	Select this option to suppress the visibility of Bundle Option details on the Invoice. On selecting thischeck box, invoice line items will not be created for products with Informational Billing Schedules.
Is Default	Select this option to make this Billing Preference the default preference for thisorg.
Proration Period Treatment	To define how you want to distribute the billing amount proportionally between the first and last Billing Schedules, select from one of the following picklist options: • **Separate Period** - To create a separate Billing Schedule for the outstanding amount. • **Ignore** - To levy no charge for the Initial partial period. • **Charge Full** - To levy no charge for the last partial period. • **Combine with First Period** - To charge each partial period along with the next full period. To see the impact Proration has on Billing Schedules, please refer Proration.
Active	Select the check box to activate this Billing Preference.
Ready for Invoice Date Offset	Enter the number of days by which you want to offset the ready for invoice date. The number of days are added to the Ready for Invoice Date determined by the billing rule.

3. Click **Save**.

Associating Billing Preferences

One way of associating Billing Preferences to Accounts, Quote or Orders is by setting the **Preference Level** at the line level. You can also associate a Billing Preference to an Account or a Proposal header from the **Billing Preference Lookup** () field.

The lowest level billing preference will take precedence. Also making changes to a Billing Preference will only affect future orders.

Billing Schedules

If the **Expected Start Date** on Quote/Proposal is '1/20/2016' and **Billing Cycle Start** is set to **Billing Day of the Month** (10th of the month), the Billing Schedules will be:

Period Start Date	Period End Date
1/20/2016	2/09/2016
2/10/2016	3/09/2016
3/10/2016	4/09/2016

And so on.
You have the flexibility to choose Billing Day from other objects such as Account, Quote/Proposal, and Order. Please see the other values for Billing Cycle Start in the above table.

> **Note**
> To generate Billing Schedules, you must provide **Expected Start Date** and **Expected End Date** on the Quote/Proposal page.

Let us take a look at how Billing Schedules are aligned for 3 different products based on the **Billing Rule** and **Bill Cycle Start Date**.

Example 1

List Price of the product Ace is $100.

Product Name	Start Date	End Date	Billing Rule	Billing Frequency	Bill Cycle Start Date	Proration Period Treatment	Billing Day of the month
Ace	1/1/2016	12/31/2016	Arrears	Monthly	Billing Day of the month	Separate Period	1st of the month

For Billing Rule set to **Arrears**, **Ready for Invoice Date** is a day after the Period End Date. Billing Schedules for Ace will be,

Period Start Date	Period End Date	Ready for Invoice Date	Amount
1/1/2016	1/31/2016	2/1/2016	$100
2/1/2016	2/29/2016	3/1/2016	$100
3/1/2016	3/31/2016	4/1/2016 and so on	$100

If you set **Ready For Invoice Date Offset** as 10, Ready for Invoice Date will be,

Period Start Date	Period End Date	Ready for Invoice Date	Amount
1/1/2016	1/31/2016	**2/11/2016**	$100
2/1/2016	2/29/2016	**3/11/2016**	$100
3/1/2016	3/31/2016	**4/11/2016 and so on**	$100

Example 2

List Price of the product DigiX is $200.

Product Name	Start Date	End Date	Billing Rule	Billing Frequency	Bill Cycle Start Date	Proration Period Treatment	Billing Day of the month
DigiX	1/15/2016	1/14/2017	Advance	Monthly	Ready for Billing Date	Separate Period	

For Billing Rule set to **Advance**, **Ready for Invoice Date** is the Period Start Date for a schedule. Billing Schedules for **DigiX** if **Ready for Billing Date** is 1/9/2016,

Period Start Date	Period End Date	Ready for Invoice Date	Amount
1/15/2016	2/8/2016	1/15/2016	$160
2/9/2016	3/8/2016	2/9/2016	$200
3/9/2016	4/8/2016 and so on.	3/9/2016	$200

Example 3

List Price of the product MyShot is $300.

Product Name	Start Date	End Date	Billing Rule	Billing Frequency	Bill Cycle Start Date	Proration Period Treatment	Billing Day of the month
MyShot	1/1/2016	12/31/2016	Bill on Ready for Billing Date	Monthly	Order Date	Separate Period	

For Billing Rule set to Bill on Ready for Billing Date, Ready for Invoice Date takes the value of Ready for Billing Date on the **Quote/Proposal** for all schedules. Billing Schedules for MyShot, if the **Order Date** is 1/15/2015,

Period Start Date	Period End Date	Ready for Invoice Date	Amount
1/1/2016	1/14/2016	2/1/2016	$145.162
1/15/2016	2/14/2016	2/1/2016	$300
2/15/2016	3/14/2016 and so on.	2/1/2016	$300

To see the impact Proration has on Billing Schedules, please refer Proration.

Impact of Calendar Cycle Start on Billing Schedules

Calendar Cycle Start is used to align the Billing Schedules to a defined calendar month. For example, Billing Frequency is **Quarterly** and Calendar Cycle Start is **None**.

Product Name	Start Date	End Date	Billing Rule	Billing Frequency	Bill Cycle Start Date	Proration Period Treatment	Billing Day of the month	Calendar Cycle Start
Ace	4/1/2016	12/31/2016	Arrears	Quarterly	Billing Day of the month	Separate Period	1st of the month	None

The Billing Cyle start is 4/1/2016, so the Billing Schedules will align on the Start Day and **first Quarter** will be 4/1 to 7/31.

Period Start Date	Period End Date	Ready for Invoice Date	Amount
4/1/2016	7/31/2016	8/1/2016	$300
8/1/2016	11/31/2016	12/1/2016	$300
12/1/2016	12/31/2016	1/1/2017	$100

If Calendar Cyle Start is set to **June**

Product Name	Start Date	End Date	Billing Rule	Billing Frequency	Bill Cycle Start Date	Proration Period Treatment	Billing Day of the month	Calendar Cycle Start
Ace	4/1/2016	12/31/2016	Arrears	Quarterly	Billing Day of the month	Separate Period	1st of the month	June

The Billing Schedule for the first period will be 4/16 to 5/31. After this, quarters will align with 6/1 (June) and hence the period will be 6/1 to 8/31.

Period Start Date	Period End Date	Ready for Invoice Date	Amount
4/1/2016	5/31/2016	8/1/2016	$100
6/1/2016	8/31/2016	2/1/2016	$300
9/1/2016	11/31/2016 and so on.	2/1/2016	$300

Note
Billing Schedule alignment based on Calendar Cycle Start is applicable only for Products with Billing Frequency set to Quarterly, Half-yearly, and Yearly.

To create a Billing Preference

1. Click **Billing Preference > New**.
2. Enter values for the fields described in the following table.

Field	Description
Preference Name	Enter a unique name for the Billing Preference you want to create. It is recommended that you keep an easily identifiable name.
Preference Level	Select the level at which you want to apply this Billing Preference.

Field	Description
Billing day of the month	Day of the month when you want to bill your customer. For example, to set '10' as the Billing Day, select **10th of the month** from the picklist. If you select **End of the month**, the Billing Day is set to the **last day** of each month as and when the schedule advances. For example, Billing Day of Month will be '31' for March and '30' for April. **Note:** This field holds significance only if you set 'Billing Cycle Start' to 'Billing Day of Month'.
Billing Cycle Start	Billing Cycle Start can be any day on which you want to bill customers. Billing Schedules are aligned on the basis of Billing Cycle Start. Billing Schedule for an asset takes the **Expected Start Date** mentioned on the Proposal Line and Order Line as the starting point for generating the schedules. To align rest of the duration, it looks up to the value provided in **Billing Cycle Start**. Please refer this example. Select one of the following options: • **Billing Day of Month** - value specified in the **Billing Day of Month** field. • **Ready for Billing Date** - value from **Ready for Billing Date** field on the accepted Quote/Proposal. • **Period Start Date** - value from **Expected start date** field on Asset Line Item. • **Order Date** - value from **Order Date** field in the activated order. • **Account Billing Day of Month** - value from **Billing Day of Month** field on Accounts. Select this option, if you want to set a common billing day for all orders in an account.
Calendar Cycle Start	Select the month to align your customer's billing to a specific business calendar year. To set calendar cycle start from an account, set **Calendar Cycle Start** to *Account Calendar Cycle Start.* Next, select a month from **Calendar Cycle Start** on the Account.
Tax Level	Select the object to which you want to apply taxes, from one of the following options: • **Account** • **Product**
Invoice Delivery Method	Select the preference for Invoice Delivery from Email or Print.
Invoice Output format	Select the format from one of the following options: • **PDF** • **DOC** • **DOCX** • **RTF**
Allow Payment Term Override	Select this check box if you want to override the default Payment Term.
Allow Preference Override	Select this check box if you want to override this Billing Preference.

Field	Description
Do not Create Informational	Select this option to suppress the creation of all Informational Billing Schedules. For Bundle products, if the **Bundle Invoice Level** is set to **Bundle**, Billing Schedules for Bundle Options will not be created.
Hide Informational	Select this option to suppress the visibility of Bundle Option details on the Invoice. On selecting this check box, invoice line items will not be created for products with Informational Billing Schedules.
Is Default	Select this option to make this Billing Preference the default preference for this org.
Proration Period Treatment	To define how you want to distribute the billing amount proportionally between the first and last Billing Schedules, select from one of the following picklist options: • **Separate Period** - To create a separate Billing Schedule for the outstanding amount. • **Ignore** - To levy no charge for the Initial partial period. • **Charge Full** - To levy no charge for the last partial period. • **Combine with First Period** - To charge each partial period along with the next full period. To see the impact Proration has on Billing Schedules, please refer Proration.
Active	Select the check box to activate this Billing Preference.

3. Click **Save**.

Credit Memos

During financial transactions, you might need to provide credit to the customers for accommodating situations where goods are returned, there is a pricing dispute or where the buyer is not required to pay the full amount of the invoice. In such cases, you can issue a Credit Memo to store the credit balance and offset it against a transaction.

A Credit Memo is a legal document that states the credit balance. The Credit Memos can be issued to counterbalance current invoices or apply to future invoices.

> **Note**
> You can create a Credit Memo only when you have already invoiced your customer. Also when issuing a credit memo against an invoice, you must ensure that you issue it for the same amount or amount lower than the total amount of all the invoices in the billing schedule for an asset.

Let us take a couple of scenarios to understand the concept of Credit Memos.

Scenario 1: The credit amount is less than or equal to the invoiced amount

If the credit amount is less than the invoiced amount, the credit is calculated from the same invoice.

For example, your customer purchased a product 'CloudStream' with a base price $100 for 6 months. Therefore each Billing Schedule would carry an amount of $100 spread across 6 months. You've invoiced the customer for $600 in advance.

Now due to a price downgrade ($10) effective from the 3rd month you have to amend the product price. Post-amendment, the existing Billing Schedules are revised to $90 from 3rd month onward. On comparing the old and new Billing Schedules, you can deduce that $40 needs to be credited back to the customer's account for the last 4 months owing to the downgrade. So you must issue a Credit Memo document which has a total amount of $40.

Scenario 2: The credit amount is greater than the invoiced amount

If the credit amount exceeds the invoiced amount for which it was credited, the maximum possible amount is credited from the corresponding invoice. The remaining amount is credited from all the invoices in the billing schedule, starting from the first invoice.

For example, your customer purchased a product 'CloudStream' with a base price $100 for 3 months. Therefore each Billing Schedule would carry an amount of $100 spread across 3 months. You've invoiced the customer for $300 in advance. Now, let's assume that for the billing schedule BS1, a direct credit memo of $65 was created and for the billing schedule BS2, a direct credit memo of $80 was created. After credit memos are generated and approved, the amount remaining in BS1 is $35 and in BS2 is $20.

Billing Schedule	Start Date	End Date	Fee Amount	Status	Is Superseded?	Debit Schedule	Available Credit
BS1	3/1/2017	3/31/2017	100.00	Invoiced			35.00
BS2	4/1/2017	4/30/2017	100.00	Invoiced			20.00
BS3	5/1/2017	5/31/2017	100.00	Invoiced			100.00

Now if you amend the contract and reduce the rate of the asset to $70 per month, you will have to provide your customer with a $30 credit for each billing schedules. After the contract is amended and the renewed quote/proposal is accepted, a new order is created. Once the order is activated, new billing schedules are generated as mentioned in the following table:

Billing Schedule	Start Date	End Date	Fee Amount	Status	Is Superseded?	Debit Schedule	Available Credit
BS1	3/1/2017	3/31/2017	100.00	Invoiced	Yes		0.00
BS4	3/1/2017	3/31/2017	-30.00	Pending Billing		BS1	
BS2	4/1/2017	4/30/2017	100.00	Invoiced	Yes		0.00
BS5	4/1/2017	4/30/2017	-20.00	Pending Billing		BS2	
BS6	4/1/2017	4/30/2017	-5.00	Pending Billing		BS1	
BS7	4/1/2017	4/30/2017	-5.00	Pending Billing		BS3	
BS3	5/1/2017	5/31/2017	100.00	Invoiced	Yes		65.00
BS8	5/1/2017	5/31/2017	-30.00	Pending Billing		BS3	

You need to invoice the billing schedules, for the credit memos to get generated.

For BS4, $30 credit is calculated from BS1 which had $35 amount. For BS5, only $20 credit is calculated from BS2 as only $20 is available in this billing schedule. For the remaining $10, the system starts calculating from the first billing schedule, which is BS1. As BS1 has only $5 left, a billing schedule BS6 is created with an amount of $5 with BS1 as the debit schedule. The remaining $5 credit is calculated from BS3. For BS8, $30 credit is calculated from BS3.

There are two ways of generating a Credit Memo:

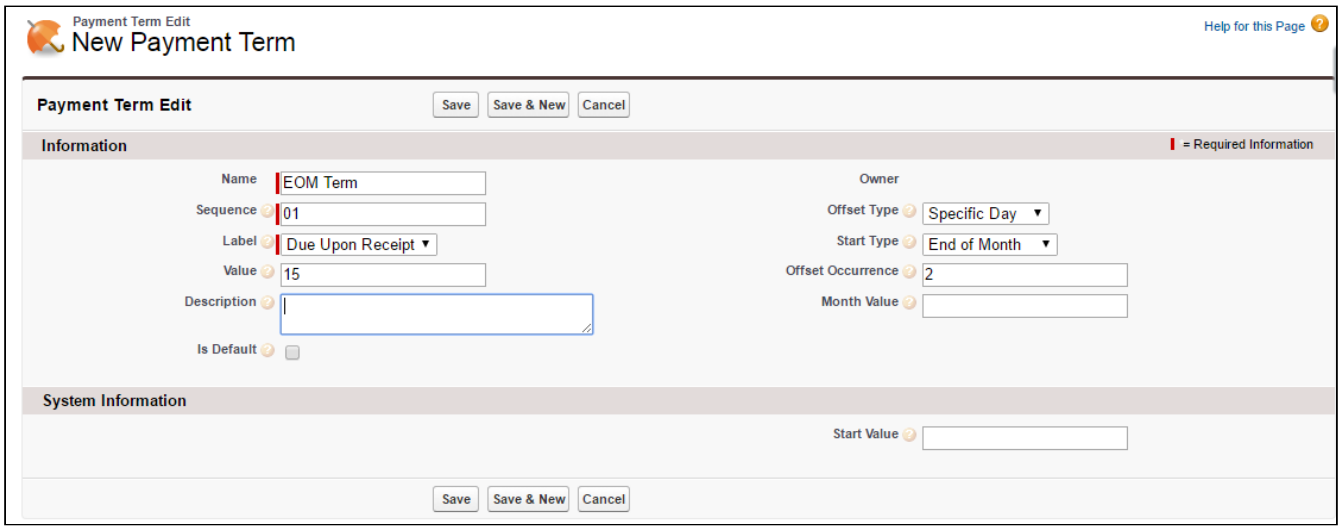
- From an Invoice
- Through Invoice Runs

Payment Term

A payment term is a condition or guideline under which you can make or receive a payment. You must include the Payment Term on the Invoice so your customers know that they must pay outstanding charges

within a certain time. You can create multiple payment terms and even define the sequence in which each term is displayed on the Invoice.

With Apttus Billing Management you can define multiple payment terms. Click the **Payment Terms** tab on the console to see a list of predefined payment terms.



To define a Payment Term

1. Click the **Payment Terms** tab.
2. Click **New**.
3. Enter values for the fields described in the following table.

Field	Description
Name	Enter a unique name for this payment term so that others in your organization can identify it with ease.
Sequence	Enter a number to define the sequence in which this term is displayed on the invoice.
Label	You can select one label from the following options. • Net 30 days • Net 45 days • Net 60 days • Net 90 days • Due upon Receipt You can define more options in the Label object pick list. For more information see <*Salesforce documentation*>.

Field	Description
Offset Type	Select the unit of a payment term, either day or month. If you set *day* as the **Offset Type** and enter *20* as the **Value**, the **Payment Term** will be set to 20 days. If you set *month* as the Offset Type and enter *1* as the **Value**, the **Payment Term** will be set to 1 month.
Value	The number of days or months in this payment term as selected in the **Offset Type**.
Start Type	The date, month or period, Payment Term is calculated from.
Start Value	This defines the start date for a **Start Type**. Specify the date if you chose *Month*, Period or Quarter as the **Start Type**.
Offset Occurrence	Frequency associated with the Offset Type. For example, if an invoice is due on the second occurrence of 15th, **Offset Type** should be Specific Day and **Offset Occurrence** should be 2.
Description	Enter a brief description of the Payment Term.
Is Default	Select the check box to make this payment term the default for invoices for this account.

4. Click **Save.**

Let us take a look at a couple of scenarios to understand how you can achieve different configurations with the Payment Term fields.

Use Case 1
Requirement -Any invoice billed after the 20th of the month (21st to End of Month) will be due on 20th of the second month.
For example: Invoice dated Apr 11, should have the due date of June 20th

Start Type=Specific Day
Start Value=20
Offset Type=Month
Offset Value=2
Offset Occurrence= null

Use Case 2
Requirement -Any invoice billed in a month will be due End of Month, two months later.
For example: Invoice dated Jan 20, should have the due date of March 31st.

Start Type=End of Month
Start Value= null
Offset Type=Month
Offset Value=2
Offset Occurrence= null

Use Case 3
Requirement -All invoices billed in a Quarter (i.e. Jan - Mar) will be due 20 days from the End of the Quarter.
For example: The due date for all invoices in the period Jan 1st to March 31st will be April 20th.

Start Type=End of Quarter
Start Value= null
Offset Type=Day
Offset Value=20
Offset Occurrence= null

Invoice Run

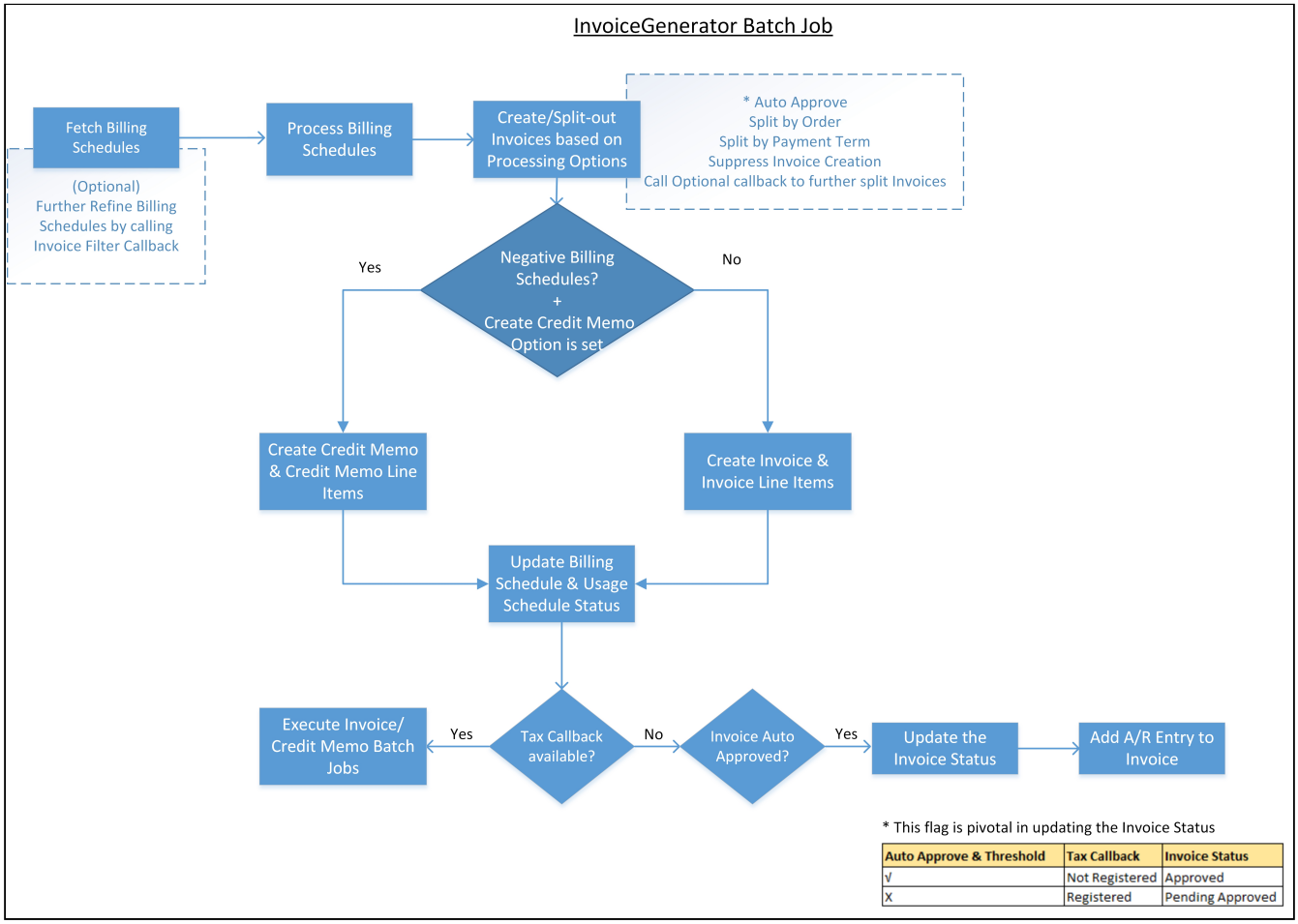
Invoice Run is a batch job for generating invoices in bulk. You can generate invoices ad hoc or schedule it for the future. Four batch jobs execute behind the scenes to generate the invoices. Let us take a look at each.

Pre-requisite

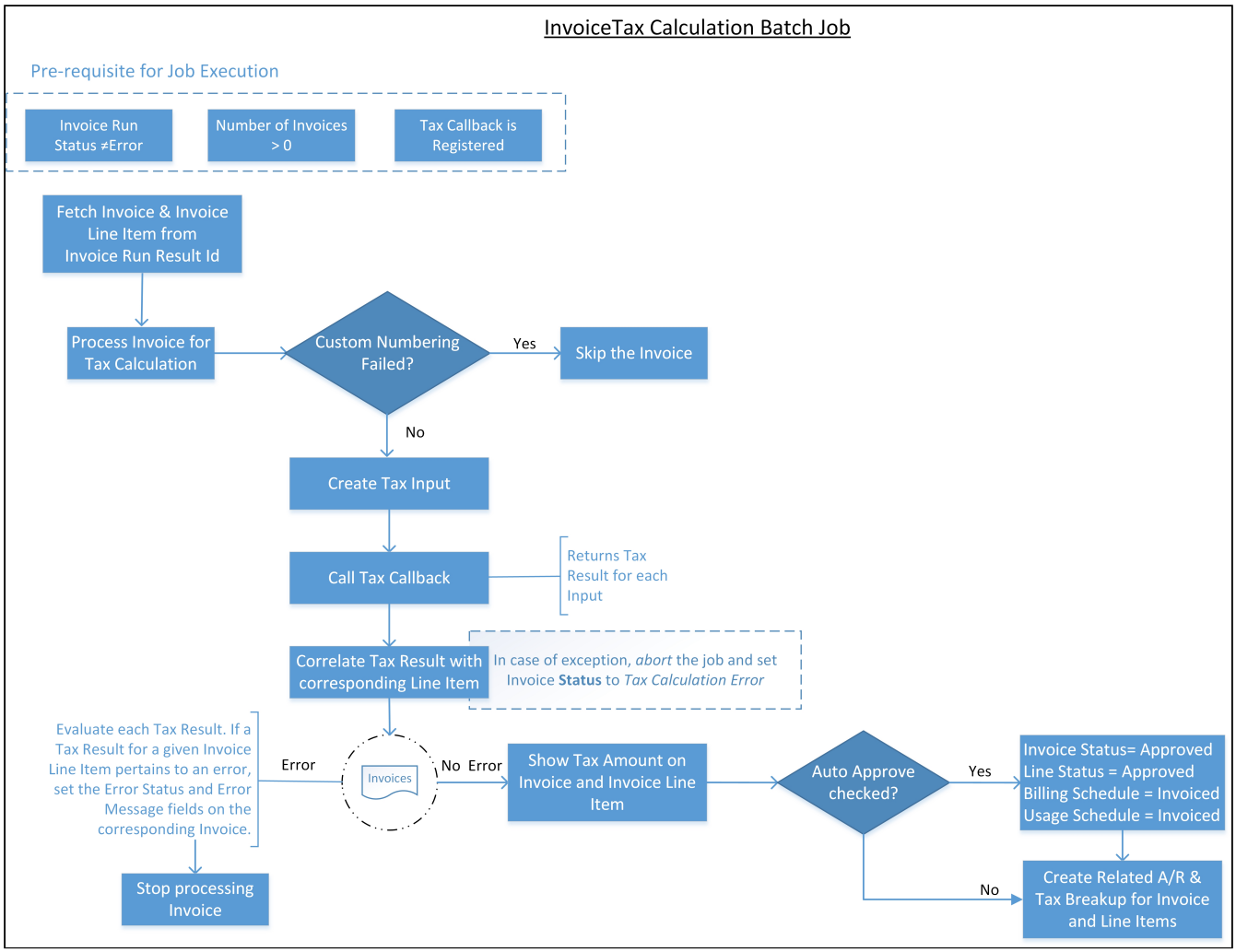
You must override the standard Invoice Run page with a Visualforce page to drive the Invoice Run behavior as required.

Go to **Invoice Runs** object. Under **Buttons, Links, and Actions** section, edit **New** label to select an VisualForce page, *InvoiceRunCreate[InvoiceRunCreate].*

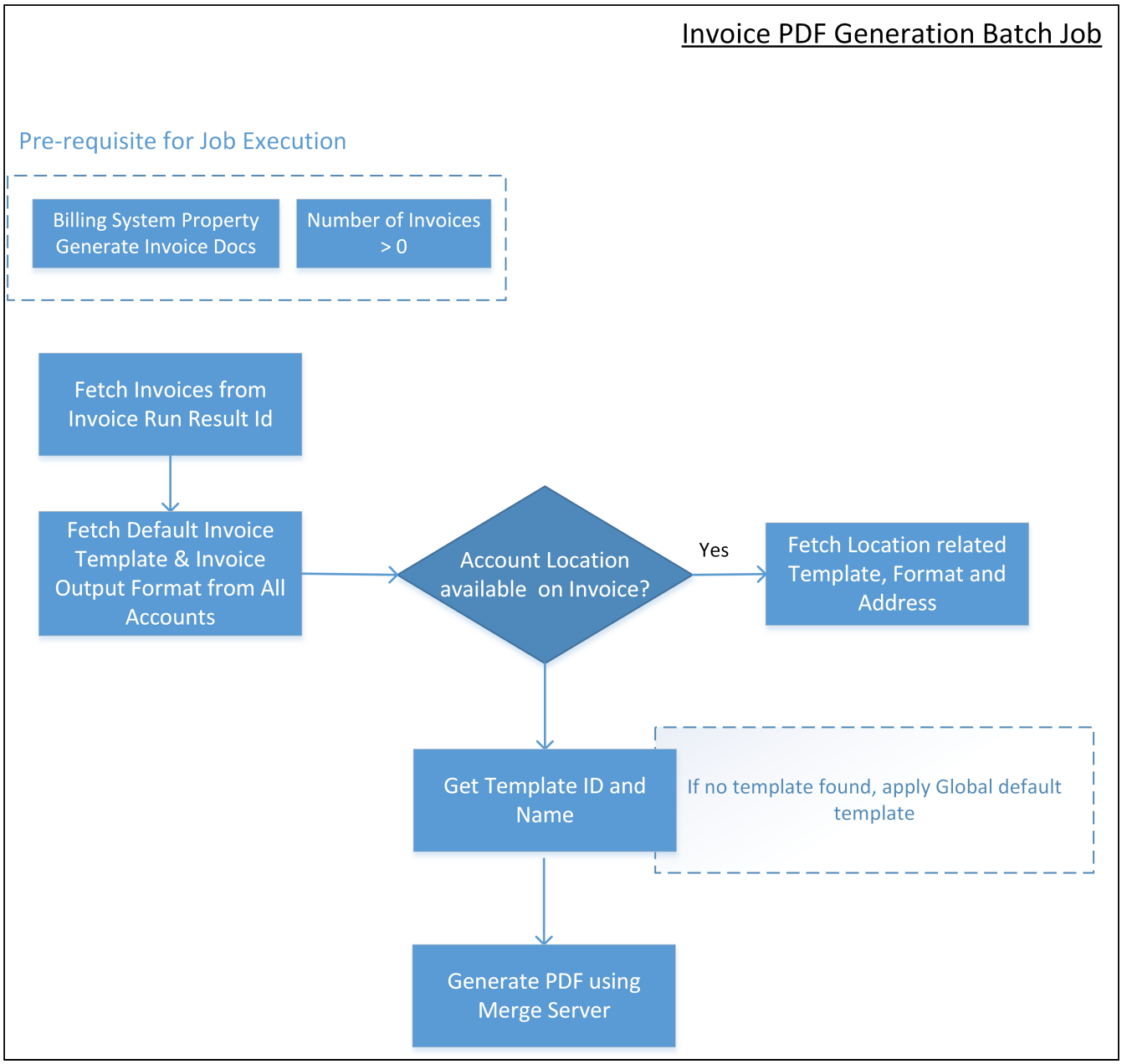
Invoice Generator Batch Job



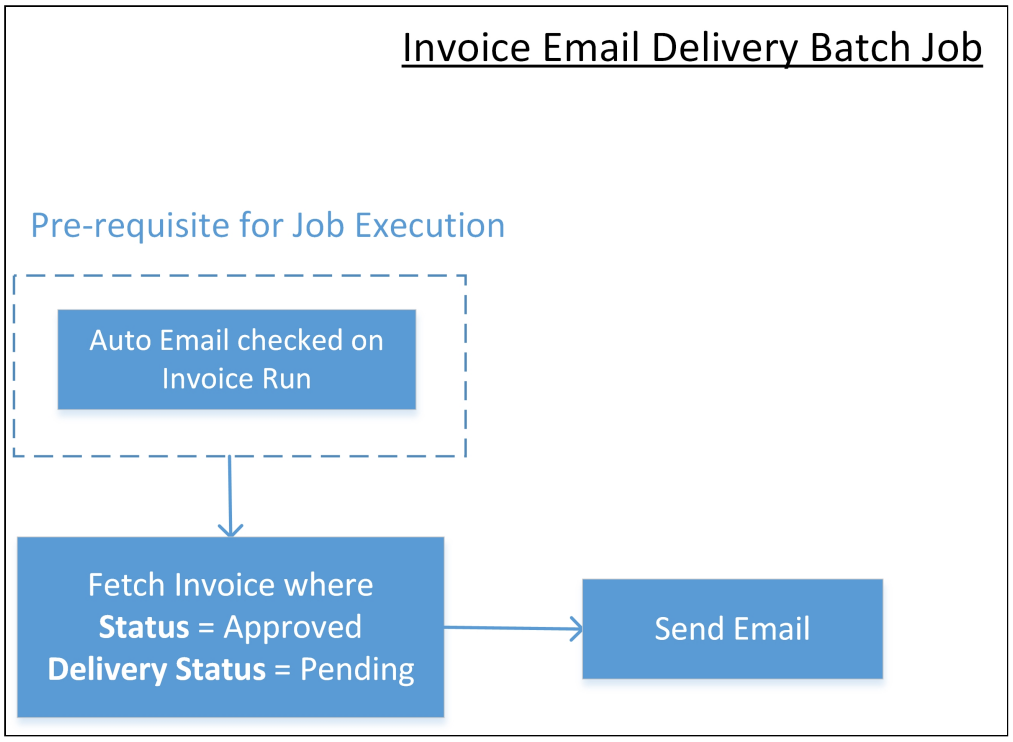
Invoice Tax Calculation Batch Job



Invoice PDF Generation Batch Job



Invoice Email Delivery Batch Job



Wallets

Many business practices require a prepayment from the customers. Prepayments can be used for future product invoices and for usage-based services. Wallets in Apttus Billing Management facilitate that. Wallets are created in the same manner the products are created.
Wallet supports two types of prepayments.

- Prepayments made against a product
- Prepayments made for usage-based services

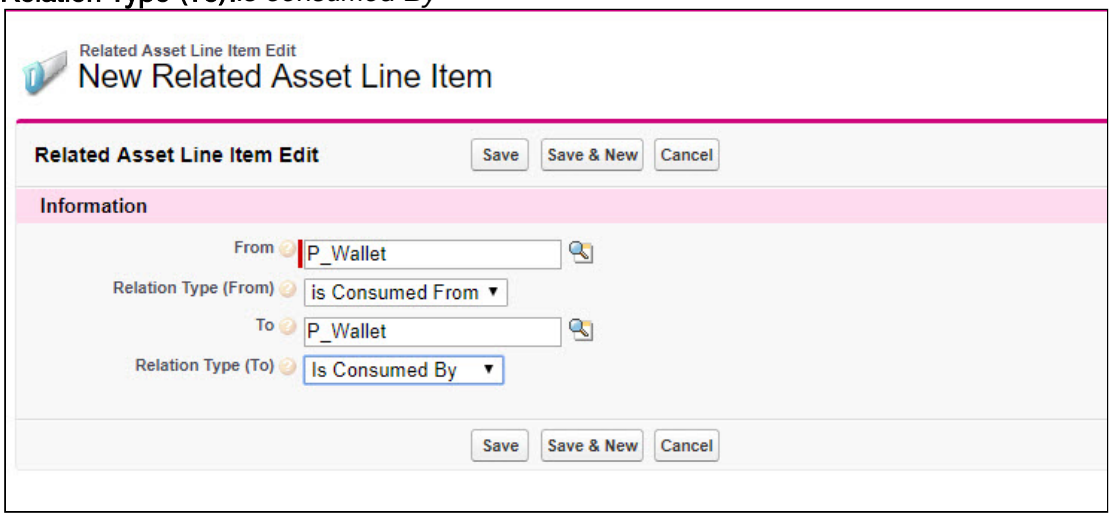
Creating a wallet

Wallets are created in the same way as Assets. For example, let us create a Wallet with Balance Amount as $1000. First, configure a product with a *Product Type* of Wallet and with List Price as 1000. Add the wallet to your cart and accept the quote/proposal. Create and activate an order to create an asset. Invoice the wallet to make it available for consumption.

Associating a Wallet with an Asset

Wallets are linked with products by creating a Related Asset Line Item record. Perform the following steps to link a wallet to a product.

1. Go to the Asset Line Item Detail page of your wallet.
2. Click New Related Asset Line Item in the Related Asset Line Item (From) Related List.
3. Create the record as follows:
   From: *Wallet*
   Relation Type (From):*is consumed from*
   To:*Product*
   Relation Type (To):*Is consumed By*



4. Click Save.

You can link multiple wallets to a product by creating multiple Related Asset Line Item records. Multiple wallets are consumed in the order in which they were linked to the product.

Wallet Drawdown

Wallet drawdown is created when any transaction is done using wallet. It contains look-up to the Asset line Item and the Billing Schedules for which the wallet is used.

Field	Description
Amount	Amount consumed from the wallet
Asset Line Item	Asset Line Item links the drawdown to the asset line item that consumes from the wallet.
Backout	Flag to indicate if the drawdown is backed out due to ABO actions
Billing Schedule	Lookup to corresponding Billing Schedule
Wallet	Lookup to corresponding wallet Asset Line Item

ⓘ Informational Billing Schedules do not create wallet drawdowns.

Amount field shows the amount consumed from the wallet. The drawdown amount is calculated as follows:

- The fee amount of the billing schedule is equal to the available balance of the wallet

If the fee amount of the billing schedule is equal to the available balance of the wallet, the drawdown amount is set as the Fee Amount of the Billing Schedule. The available balance of the wallet is set to $0 and the next billing schedule is processed.

- The fee amount of the billing schedule is less than the available balance of the wallet

If the fee amount of the billing schedule is less than the available balance of the wallet, the drawdown amount is set as the Fee Amount of the Billing Schedule. The available balance of the wallet is updated after deducting the Fee amount and the next billing schedule is processed.

- The fee amount of the billing schedule is more than the available balance of the wallet

If the fee amount of the billing schedule is more than the available balance of the wallet, the drawdown amount is set as the available balance of the wallet. The available balance of the wallet is set to $0 and the billing system checks for the next wallet.

Dunning Policy

Dunning is the process of methodically communicating with customers to ensure the collection of accounts receivable. It is a formalized notification process to follow up and collect the money owed by the customer. You can send multiple reminders to the customer with different email templates depending on your organization's dunning policy. For example, you can set a Dunning Policy which will auto send 3

emails to your customer if they missed the Invoice Due Date. You can also include the invoice PDF in your email.

Dunning Policy batch job can be scheduled to run on daily basis. It scans all the invoices present in your Salesforce Org to identify the invoices that qualify for dunning process.

An invoice is picked for dunning policy if it fulfills all the three condition:

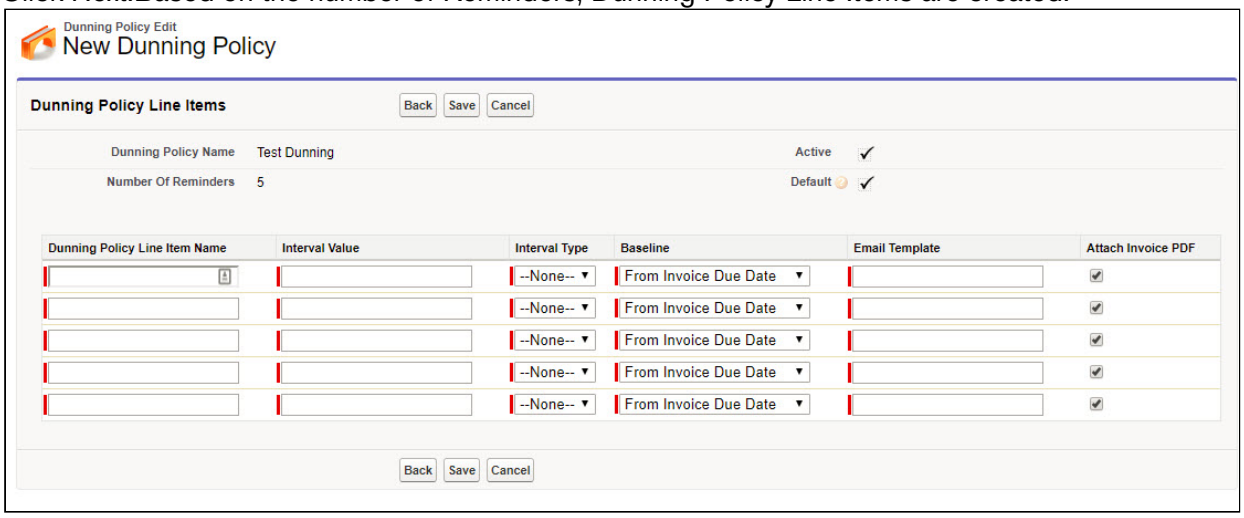
- Invoice Due Date is less than system date
- Payment status of the invoice is unpaid or partially paid
- The invoice is approved

Setting up a Dunning Policy

1. From the Billing Console, Click Dunning Policies under Invoices.
2. Click New.
3. Enter Values for the following fields.

Fields	Description
Dunning Policy Name	Name of your Dunning Policy
No. of Reminders	The number of email reminders to send if the payment is delayed.
Active	Whether the Dunning Policy is Active or not. By default, it is set to true.
Default	Select the checkbox to make it the default dunning policy.

4. Click Next.Based on the number of Reminders, Dunning Policy Line Items are created.



5. Enter values for the following fields:

Field	Description
Dunning Policy Line Item Name	The name of the Dunning Policy Line Item
Interval Value	A numerical value depicting the number of days. For the first dunning line item policy, it is the numbers of days to wait before the first email reminder. For the remaining dunning policy line items, it is the number of days to the wait between each reminder or the number of days to wait after the Invoice Due Date.
Interval Type	Select Day(s) from the picklist.
Base Line	The baseline for sending the reminder emails. You can select from the following values: • From Invoice Due Date: Email reminder is sent when the system date crosses the invoice due date by the number of days mentioned in the interval value. • From Previous Reminder: This value is not supported for the first dunning policy line item. An email reminder is sent when the system date crosses the previous reminder date by the number of days mentioned in the interval value.
Email Template	The API name of the email template. For more details on Dunning Invoice Email Templates, refer to Dunning Policy Templates.
Attach Invoice PDF	Select this check box to attach the invoice in a PDF format in the reminder email.

6. Click Save.

Scheduling the Dunning Policy Batch Job

You can schedule the Dunning Policy Batch Job to runs daily.

To schedule the Dunning Policy Batch Job,

1. Go to Setup → Apex Classes.
2. Click **Schedule Apex.**
3. Enter a Job Name and Select **DunningPolicyExecutionSchedular** from the Apex Class lookup.
4. Set Weekly frequency and select all days to run the dunning policy daily.
5. Enter Start Date, End Date and Preferred Start Time.
6. Click Save.

Your dunning policy is now scheduled to run daily.

⚠ To manually run the dunning policy batch job, execute the following apex code from your developer console:

```
Apttus_Billing.DunningPolicyExecutionBatchJob obj = new
 Apttus_Billing.DunningPolicyExecutionBatchJob();
database.executeBatch(obj);
```

Associating a Dunning Policy

The administrator can create a dunning policy and mark it as default. This default dunning policy will be used for all customer accounts and account locations.

Users can also override the Dunning Policy on the Account or Account Location. If dunning policies associated with the Account and Account Location are different, the dunning policy set at the Account Location takes precedence. If dunning policy is not associated with the account or account location, then the default dunning policy is used.

You can also exclude a specific account or an account location from dunning notifications. To disable dunning for a specific account or an account location, select **Exclude from Dunning** field on the account or account location object.

Dunning Reminder Log

Each time a dunning notification is sent to a customer, an entry is created in the **Dunning Reminder Log** object.

Dunning Reminder Log captures the following information:

- Dunning Reminder Log Id
- Reminder Sequence
- Reminder Date
- Link to the Invoice
- Link to the Dunning Policy Line Item

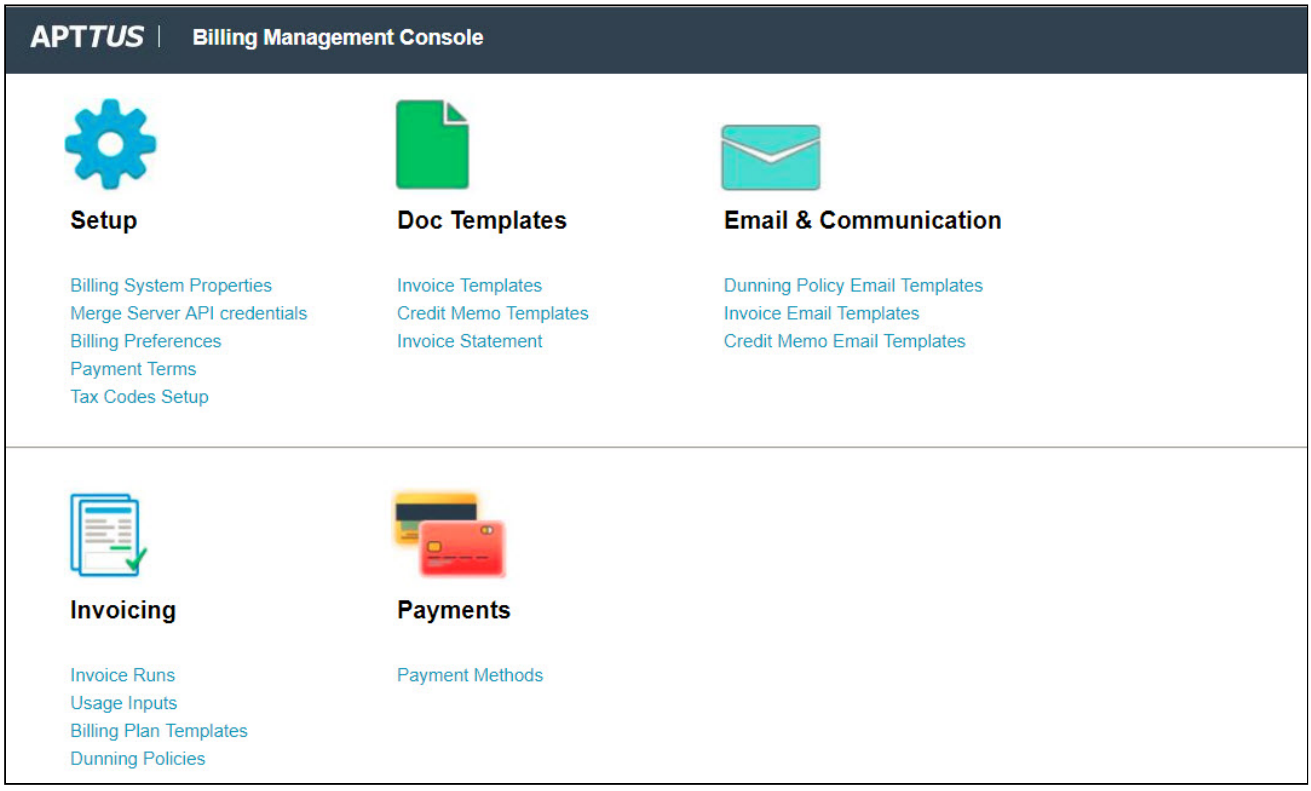
Billing Settings

You must configure following objects before billing a customer's purchase:

1. Settings on the Account and Account Location for Billing.
2. Define Billing Preference.
3. Configure Products.
4. On the **Proposals** page, you can select products, adjust pricing, adjust billing settings, change bill preference, initiate approvals processes and you can automate the process to activate an order, create a Bill, and create revenue.
5. Set Payment Terms.
6. Add an Invoice Template.
7. Verify Billing System Properties.

Billing Console

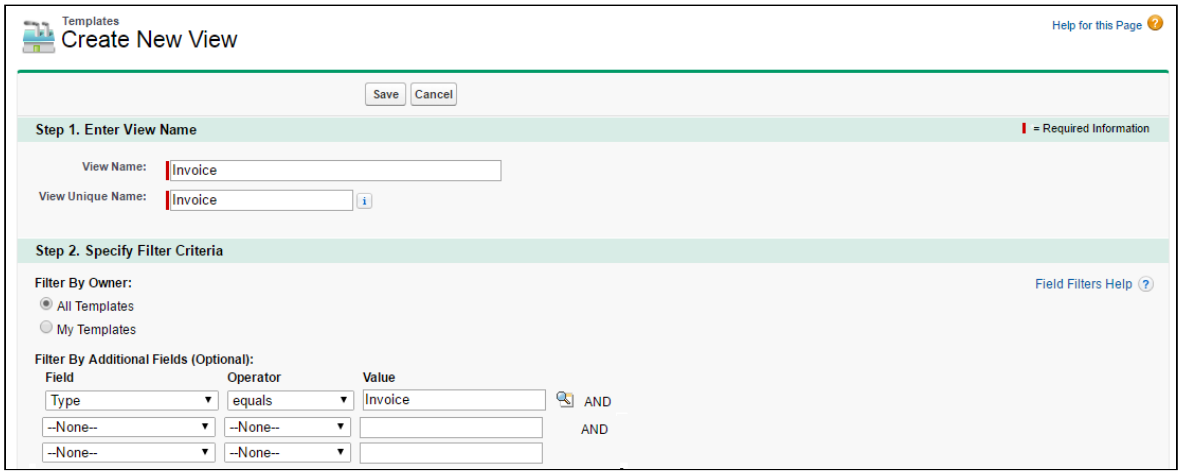
You can navigate to all key pages of configuration and execution from the Billing Console.
From the Force.com app menu, select **Apttus Billing Management** and click **Billing Console** from the header.



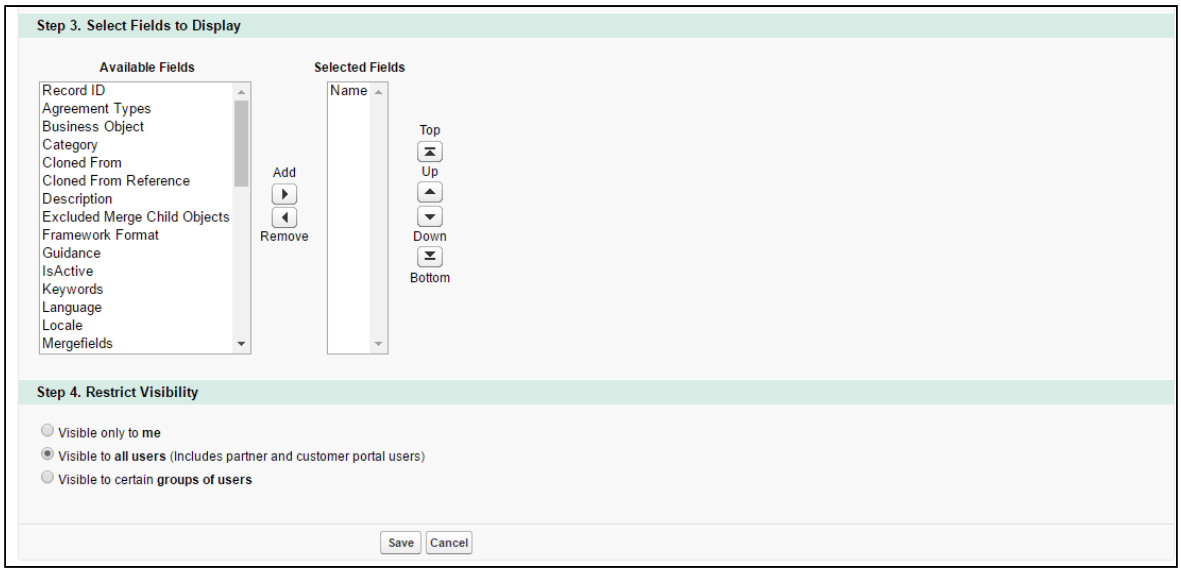
Pre-requisite

1. To view Templates tile on the Console, you need to create *List Views* on the **Template** page.

a. Go to **All Tabs > Templates** and click **Create New View**.
b. Enter the **View Name** and include **Invoice**, **Credit Memo** and **Invoice Statement** in the Filter Criteria.



c. Select Fields to Display and select **Visible to all users** radio button.



d. Click **Save**.
e. Repeat **steps a to d** for **Invoice Statement** and **Credit Memo**.

Billing Console is broken down into 6 basic functions:

- Billing Setup
- Doc Templates
- Email and Communication
- Invoicing and

- Payments

To get started, you must define billing and invoice related information on the Account page.

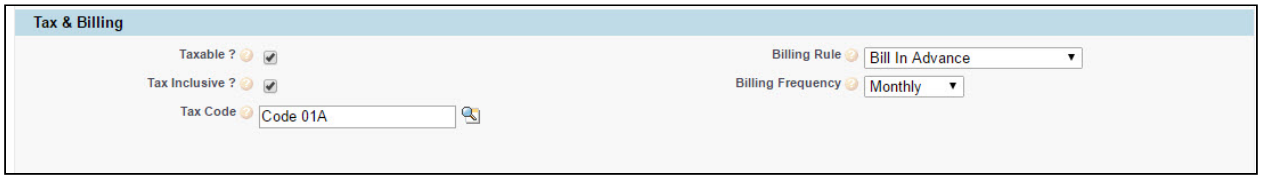
Product Configuration

In order to provide the flexibility to apply different Billing Rule and Billing Frequency to diverse products, these configurations are done at the PLI level of a product. You can,

- Define Taxes & Billing
- Set Auto Renewal

To define Taxes, Billing Rule, and Billing Frequency

1. Select a Product. Go to **Product Console** and **Edit** Price List Item.
2. Navigate to **Tax and Billing** tab.



You can configure the following fields:

Field	Description
Taxable?	Select this check box if the order is a taxable product or service.
Tax Inclusive?	Select the check box if the price of the product is inclusive of taxes.
Tax Code	Lookup to select the relevant tax code.
Billing Rule	Select one of the following options: • **Bill in Advance** - Enables you to invoice the receivables at the beginning of the schedule. • **Bill in Arrears** - Enables you to invoice the receivables at the end of the schedule. • **Bill on Ready for Billing Date** - Enables you to bill your customer with a consolidated invoice, on a day of their choice.

Field	Description
Billing Frequency	Select one of the following options: • **Monthly** - To generate a bill once every month • **Quarterly** - To generate a bill once every three months • **Half-yearly** - To generate a bill once in the middle of a calendar or financial year. • **Yearly** - To generate a bill once every year • **Single Period** - To generate a single bill for the entire amount even if the product **Price Type** is **Recurring**. Please note that Single Period Frequency cannot be used for a product if its Auto Renewal Type is set to Evergreen. • **Usage** - To generate a bill based on usage or consumption For products to be billed as one-time, please select the frequency as **One Time**. For products with recurring billing subscriptions, you can select frequencies such as monthly, quarterly, half-yearly and yearly.

3. Click **Save**.

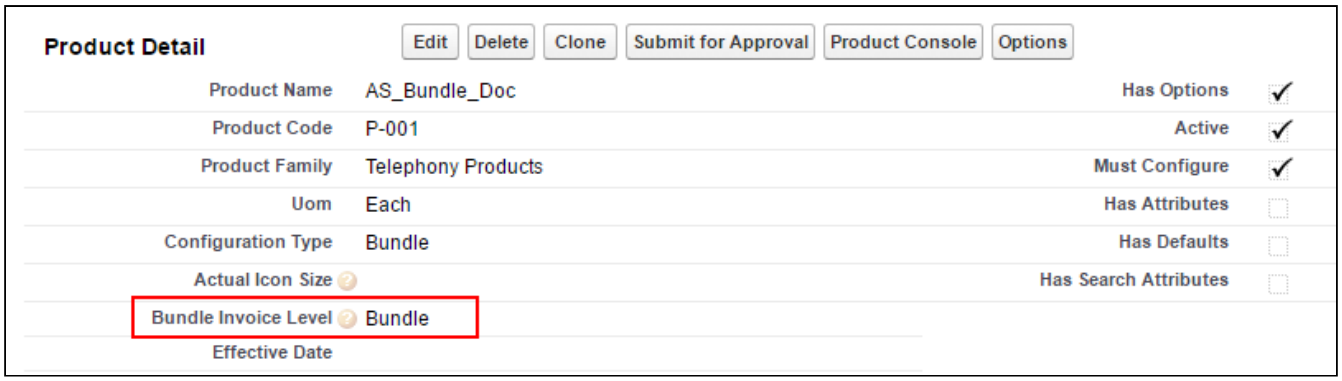
Configuration for Bundle Products

For standalone products, there are no fields that you need to exclusively set for Billing. But you can manage the display of bundle products on an invoice from a field on **Product Detail** page as explained in this section.

You can invoice a bundle product in one of the following ways:

- Invoice Bundle product and Bundle Options separately (Configuration 1)
- Invoice Bundle product including all Options rolled to the bundle product (Configuration 2)
- Invoice Each Bundle Option Item separately (Configuration 3)

On **Product Detail** page, you can specify a value for a **Bundle Invoice Level** field which defines the type of Billing Schedules generated for it and eventually its representation on the Invoice.



Bundle Invoice Level along with the **Line Type** field on **Asset Line Item** determines the type of Billing Schedule generated for bundle and options. You can have two types of Billing Schedules: **Contracted** and **Informational**.

> **Note**
> **Contracted** Billing Schedules are included in the Invoice total whereas **Informational** Billing Schedules are for informational purposes only. For more information, please refer Billing Schedules in Billing Management User Guide.

Following table lists the combination of different values for both Bundle Invoice Level and Line Type resulting in a Billing Schedule Type.

Bundle Invoice Level	Line Type	Billing Schedule Type
Bundle	Option	Informational
Bundle	Product/Service	Contracted
Detail	Product/Service	Informational
Detail	Option	Contracted

Configuration 1

To Invoice Bundle product and Options separately,

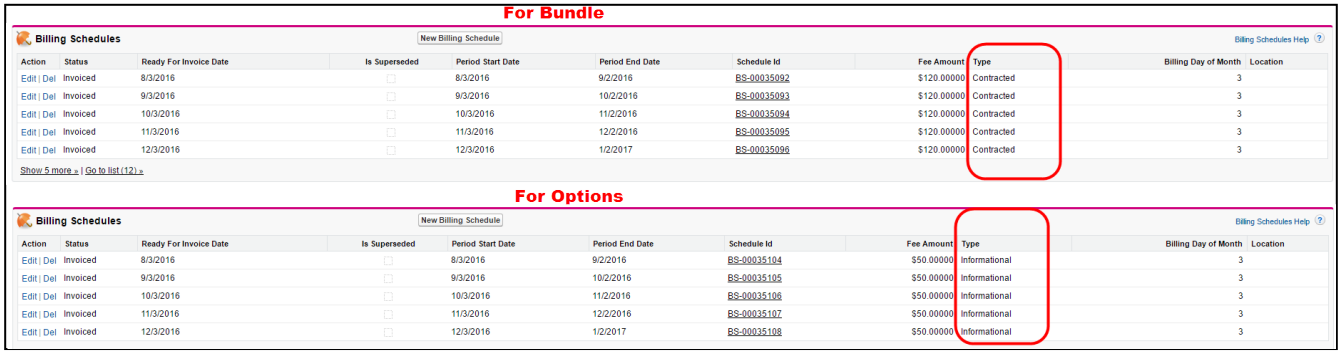
Type	Bundle Invoice Level
Product	Bundle
Options	Detail

Configuration 2

To roll up all Options to the Bundle,

Type	Bundle Invoice Level
Product	Bundle
Options	Bundle

This will create a contracted billing schedule for the header and informational billing schedule for all the options.

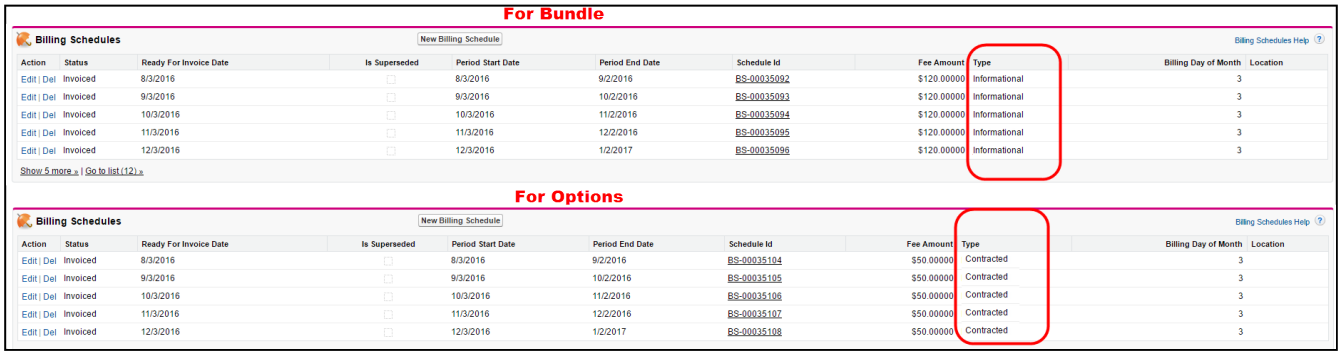


Configuration 3

To invoice Options and not the Bundle,

Type	Bundle Invoice Level
Product	Detail
Options	Detail

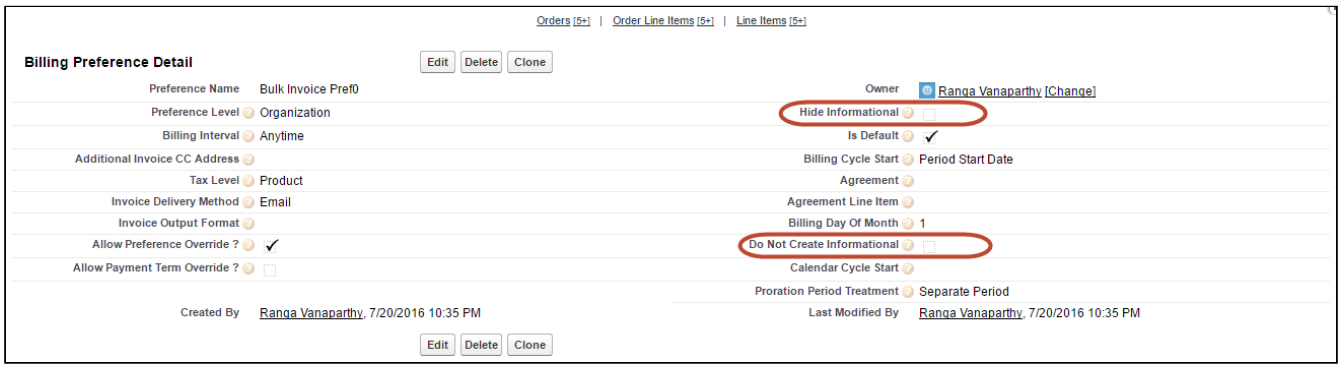
This will create an informational billing schedule for the header and contracted billing schedule for all the options.



Display on Invoice

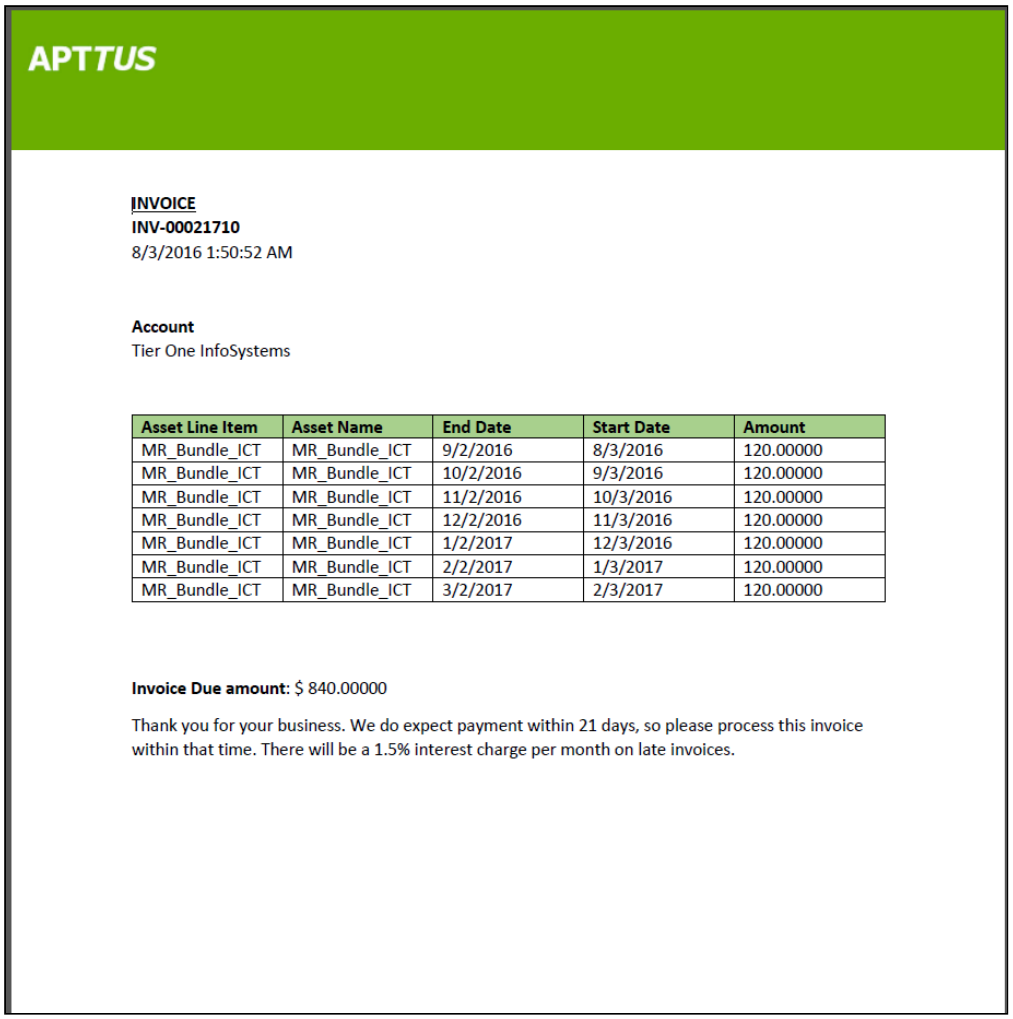
On the Invoice Template, you can control the display of informational Billing Schedules using **Hide Informational** and **Do not Create Informational** fields. These fields are available on **Billing Preferences**.

- **Hide Informational** - If you select this option, you will not see any Invoice Line Item for *Informational* Billing Schedules.
- **Do not Create Informational** - If you select this option, Billing Schedules are not created for *Informational* Billing Schedules.



Generate Invoices

To hide informational billing schedules on the invoice or not create information schedules at all, configure **Hide Informational** and **Do not Create Informational** fields respectively on Billing Preferences. If you opt for **Hide Informational,** your Invoice would contain just the bundle details as shown on the following invoice.



APTTUS

INVOICE
INV-00021710
8/3/2016 1:50:52 AM

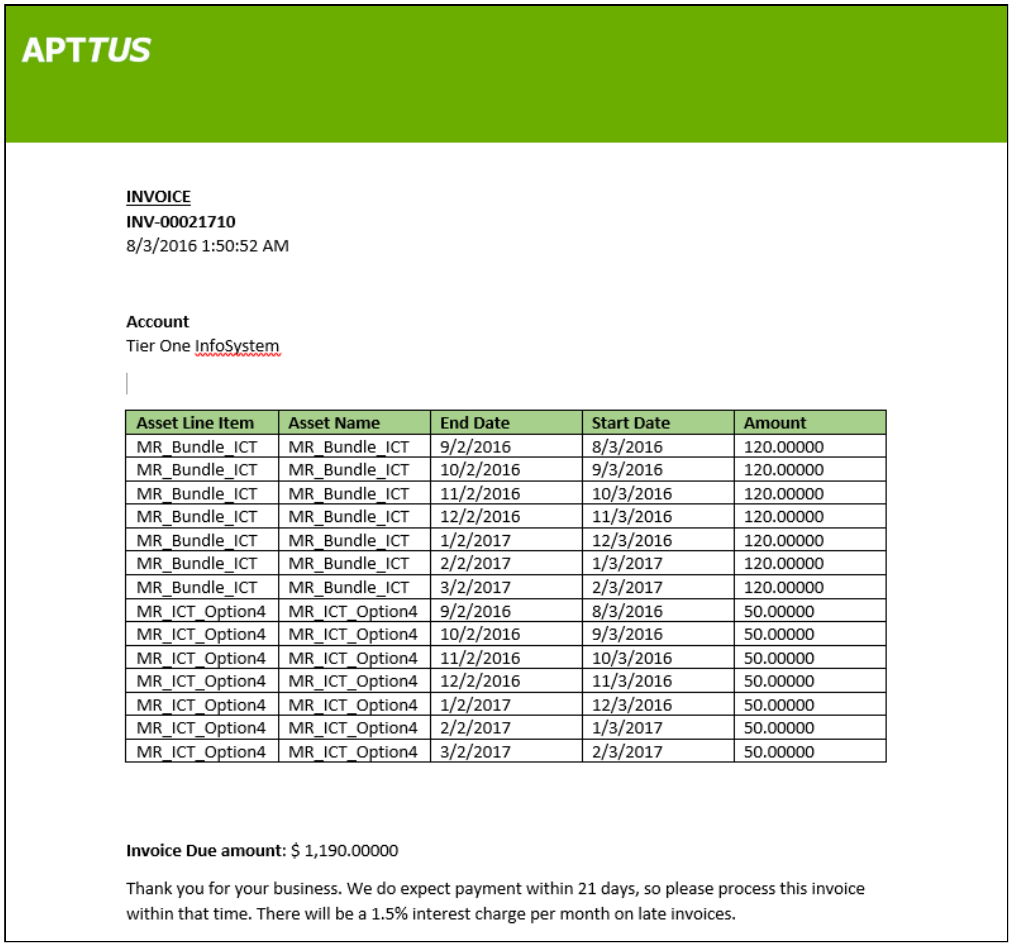
Account
Tier One InfoSystems

Asset Line Item	Asset Name	End Date	Start Date	Amount
MR_Bundle_ICT	MR_Bundle_ICT	9/2/2016	8/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	10/2/2016	9/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	11/2/2016	10/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	12/2/2016	11/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	1/2/2017	12/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	2/2/2017	1/3/2017	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	3/2/2017	2/3/2017	120.00000

Invoice Due amount: $ 840.00000

Thank you for your business. We do expect payment within 21 days, so please process this invoice within that time. There will be a 1.5% interest charge per month on late invoices.

The following template shows both Bundles and Options.



APTTUS

INVOICE
INV-00021710
8/3/2016 1:50:52 AM

Account
Tier One InfoSystem

Asset Line Item	Asset Name	End Date	Start Date	Amount
MR_Bundle_ICT	MR_Bundle_ICT	9/2/2016	8/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	10/2/2016	9/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	11/2/2016	10/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	12/2/2016	11/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	1/2/2017	12/3/2016	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	2/2/2017	1/3/2017	120.00000
MR_Bundle_ICT	MR_Bundle_ICT	3/2/2017	2/3/2017	120.00000
MR_ICT_Option4	MR_ICT_Option4	9/2/2016	8/3/2016	50.00000
MR_ICT_Option4	MR_ICT_Option4	10/2/2016	9/3/2016	50.00000
MR_ICT_Option4	MR_ICT_Option4	11/2/2016	10/3/2016	50.00000
MR_ICT_Option4	MR_ICT_Option4	12/2/2016	11/3/2016	50.00000
MR_ICT_Option4	MR_ICT_Option4	1/2/2017	12/3/2016	50.00000
MR_ICT_Option4	MR_ICT_Option4	2/2/2017	1/3/2017	50.00000
MR_ICT_Option4	MR_ICT_Option4	3/2/2017	2/3/2017	50.00000

Invoice Due amount: $ 1,190.00000

Thank you for your business. We do expect payment within 21 days, so please process this invoice within that time. There will be a 1.5% interest charge per month on late invoices.

For information on configuring templates, please refer to Managing Templates section in X-Author Contracts User Guide.

Wallet Manager Configuration

Wallet manager allows users to manage funds in a wallet without going through the quoting process. The wallet manager flow is similar to the Asset manager flow however, the wallet manager is applicable only for wallet product types.

To enable the wallet manager flow, admins will have to complete the below configurations.

To Configure Wallet Manager

1. Create two custom fields on the object **Line Item** as follows:

Field Label	Data Type	Return Type	Formula
Total Balance	Formula	Currency	Apttus_Config2__AssetLineItemId__r.Apttus_Config2__TotalBalance__c
Available Balance	Formula	Currency	Apttus_Config2__AssetLineItemId__r.Apttus_Config2__AvailableBalance__c

2. Set up the Wallet Manager Flow.
   Go to **All Tabs** → **Config Settings** → **Flow Settings.** Use the following values to create the WalletManager Flow.

Name	WalletManager
Assets Page	Apttus_Config2__Cart#assetsgrid
Cart Page	Apttus_Config2__Cart#cartgrid
Catalog Page	Apttus_Config2__Cart
Options Page	Apttus_Config2__Cart
Product Atrribute Detail Page	Apttus_Config2__Cart

3. Set Display Column Properties.
   Go to All Tabs → Config Settings → **Display Column Settings.** Select **Flow** as ***WalletManager.***

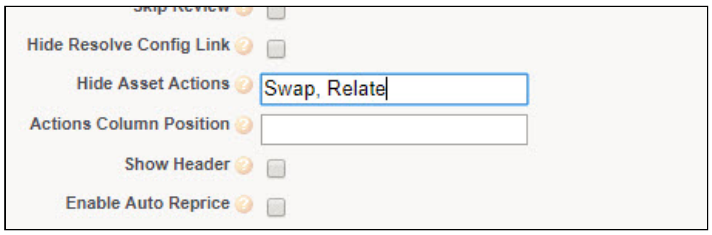
   a. For *Display Type* as **Cart Line Item**, Add **Adjustment Type** and **Adjustment Amount** fields. Set *Is Editable* as True.

   > ⚠ Adjustment Type and Adjustment Amount should be the only editable fields in the WalletManager Flow. Changes to other fields in the wallet manager flow will result in unexpected behavior.

   b. For *Display Type* as **Installed Product**, Add **Total Balance** and **Available Balance** fields.

4. Set Billing System Properties.
   In Billing System Properties, set the value for **Wallet Flow** as *WalletManager*.

5. Create a Config System Property.
   Go to **Custom Settings** → **Config System Properties.**
   a. Click Manage and create a new config system property.
   b. Name the Property as **WalletManager.**
   c. For the **Hide Asset Actions** settings, Enter *Swap, Relate.*



6. Define an Asset Line Item Filter Callback Class. For details on the classback class, refer to Asset Line Item Filter Callback Class.
7. Register the Asset Line Item Filter Callback Class in Config Custom Classes.
8. Define an Asset Line Item Validation Callback Class. For details on the classback class, refer to Asset Line Item Validation Callback Class.
9. Register the Asset Line Item Validation Callback Class in Config Custom Classes.
10. Add the Wallet Manager Button on the Account UI.
    Wallet Manager is a simple formula field that enables you to modify the Wallet which is customer's assets without a proposal.
    a. Click **Setup > Create > Objects**.
    b. Select the **Account** object and go to the **Custom Fields & Relationships** related list.
    c. Create a new formula field with the label **Wallet Manager** and *Text* as **Return Type**.
    d. In the advanced formula editor, enter the query string with the following parameters.

```
HYPERLINK("/apex/Apttus_Config2__AccountOrderCreate?id=" & Id &
"&method=csrFlow&priceListId=<pricelistid>&flow=WalletManager&launchState=ass
ets&activateOrder=true&retId=" & Id , IMAGE("/resource/WalletManager",
"Wallet Manager"),"_self")
```

    In the above formula, replace **<pricelistid>** with the *ID* of your desired Price List.
    e. Click **Save**.

Wallet Manager is configured and ready to use.

Setup Billing Preference

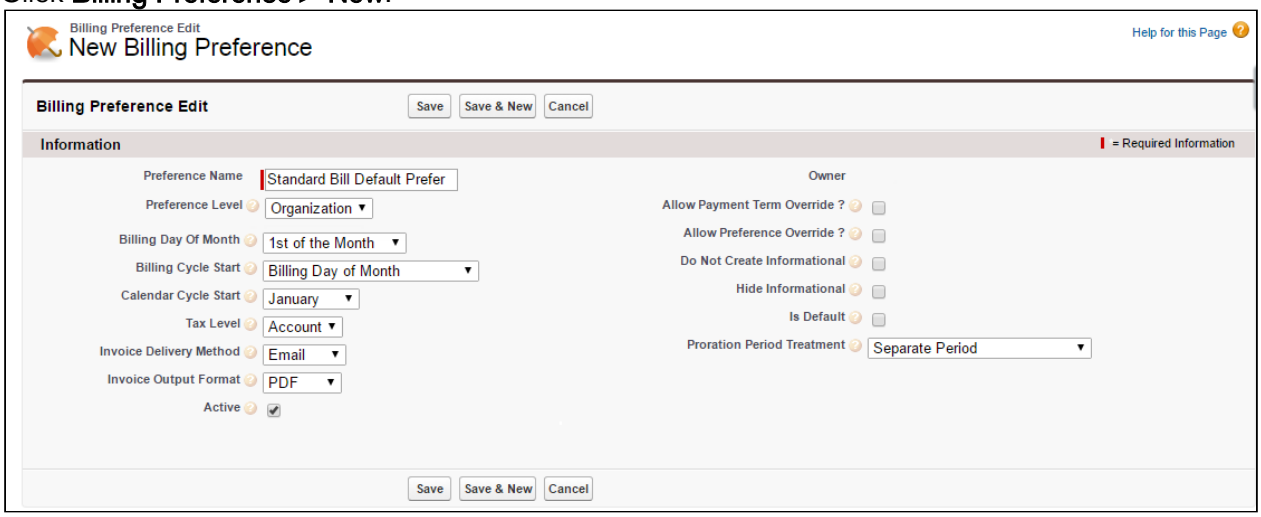
Billing Preferences drive the alignment of Billing Schedules. The Billing Preference that you define, determines the

- Billing Date and Month,
- Billing frequency,
- Applicable Taxes, and

- Method and format of Invoice delivery.

To create a Billing Preference

1. Click **Billing Preference > New**.



2. Enter values for the fields described in the following table.

Field	Description
Preference Name	Enter a unique name for the Billing Preference you want to create. It is recommended that you keep an easily identifiable name.
Preference Level	Select the level at which you want to apply this Billing Preference. If you select Organization, this Billing Preference will become the default preference for your org.
Billing day of the month	Day of the month when you want to bill your customer. For example, to set '10' as the Billing Day, select **10th of the month** from the picklist. If you select **End of the month**, the Billing Day is set to the **last day** of each month as and when the schedule advances. For example, Billing Day of Month will be '31' for March and '30' for April. **Note:** This field holds significance only if you set 'Billing Cycle Start' to 'Billing Day of Month'.

Field	Description
Billing Cycle Start	Billing Cycle Start can be any day on which you want to bill customers. Billing Schedules are aligned on the basis of Billing Cycle Start. Billing Schedule for an asset takes the **Expected Start Date** mentioned on the Proposal Line and Order Line as the starting point for generating the schedules. To align rest of the duration, it looks up to the value provided in **Billing Cycle Start**. Select one of the following options: • **Billing Day of the Month** - value specified in the **Billing Day of Month** field. • **Ready for Billing Date** - value from **Ready for Billing Date** field on the accepted Quote/Proposal. • **Period Start Date** - value from **Expected start date** field on Asset Line Item. • **Order Date** - value from **Order Date** field in the activated order. • **Account Billing Day of the Month** - value from **Billing Day of Month** field on Accounts. Select this option, if you want to set a common billing day for all orders in an account. If you see one of the options as **Account Billing Day**, please update it to **Account Billing Day of Month**. The option **Account Billing Day** is invalid and will throw an exception when activating the Order.
Calendar Cycle Start	Select the month to align your customer's billing to a specific business calendar year.
Tax Level	Select the object to which you want to apply taxes, from one of the following options: • **Account** • **Product**
Invoice Delivery Method	Select the preference for Invoice Delivery from Email or Print.
Invoice Output format	Select the format from one of the following options: • **PDF** • **DOC** • **DOCX** • **RTF**
Allow Payment Term Override	Select this check box if you want to override the default Payment Term.
Allow Preference Override	Select this check box if you want to override this Billing Preference.
Do not Create Informational	Select this option to suppress the creation of all Informational Billing Schedules. For Bundle products, if the **Bundle Invoice Level** is set to **Bundle**, Billing Schedules for Bundle Options will not be created.

Field	Description
Hide Informational	Select this option to suppress the visibility of Bundle Option details on the Invoice. On selecting this check box, invoice line items will not be created for products with Informational Billing Schedules.
Is Default	Select this option to make this Billing Preference the default preference for this org.
Proration Period Treatment	To define how you want to distribute the billing amount proportionally between the first and last Billing Schedules, select the following picklist option: • **Separate Period** - To create a separate Billing Schedule for the outstanding amount. • **Combine with first period** - To invoice the first partial period with the next complete period. ⚠ Please note that Separate Period and Combine with the first period are the only Proration Treatment options currently supported.
Active	Select the check box to activate this Billing Preference.
Early Termination Rule	Select No credit on termination, to charge your customers a penalty fee if they terminate the services mid-contract.

3. Click **Save**.

Account Setup

You will always generate an invoice for your customers' account.

Account is a standard Salesforce object from which all Apttus applications inherit values. Apttus Billing Management uses values from the fields that you define in the Account object.

> ⓘ **Note**
> Please ensure you have added custom fields to the Account object Layout. Refer the section on Accounts to know how you can add fields to your layout.

The information specific to billing and invoicing that you must define on Accounts page is described in the following table.

Field	Description
Billing Preference	Select a predefined billing preference. For details, see Billing Preference.
Billing Day of the Month	The day you want to generate a bill for the account. To apply this configuration, please set **Billing Cycle Start** to *Account Billing Day of Month* on **Billing Preference** associated with this account.

Field	Description
Calendar Cycle Start	Select a month to align the account's billing to a specific calendar year. To apply this configuration, please set **Calendar Cycle Start** to *Account Calendar Cycle Start* on **Billing Preference** associated with this account.
Payment Term	Select a predefined payment term. For details, see Payment Terms.
Tax Certificate	Select the certificate you want to apply to this account. For details, see Associating a Tax exemption certificate in Billing Management User Guide.
Tax Exempt?	Select from one of the following options. • Yes • No
Tax Exempt Status	Select from one of the following options. • Processing • Rejected • Approved
Billing Address	Enter the address where your customer wants to receive the invoice.
Shipping Address	Enter the address where your customer wants to receive the products.
Dunning Policy	Select a Dunning Policy to override the default dunning policy with an account specific dunning policy. For details, refer to Dunning Policy
Exclude from Dunning	Select this field to disable dunning for this account.

It is recommended to add **Template** details after you **Save** rest of the Account information. You should see action buttons that will help you select a template of your choice.

Field	Description
Set Invoice Email Template	Set an invoice template that will be applied to the email body for sending invoice emails. Click **Set Invoice Email Template** and select a template. For information on creating an email template, refer Templates.
Set Billing Contacts	Set filter criteria to pick a Billing Contact where you want to email the automatically generated invoices. You might have multiple contacts linked to an account. Click **Set Billing Contacts** and add filter criteria. The criteria help you select the contacts from Account Contacts that you want to email the invoices to.

Field	Description
Set Default Credit Memo Template	Choose a template on which all Credit Memos you generate for this Account will be based. This will be an attachment to the emails sent for all Credit Memos from the account.
Set Credit Memo Email Template	Choose an email template that will be applied to the email body for sending Credit Memo emails from an Account. Click **Set Credit Memo Email Template**, select an email template from the picklist and Save. For information on creating a custom email template for Credit Memos, please refer Credit Memo Templates in Billing Management User Guide.
Payment Email Template	Enter the API name of the email template to be used when sending the payment email template.

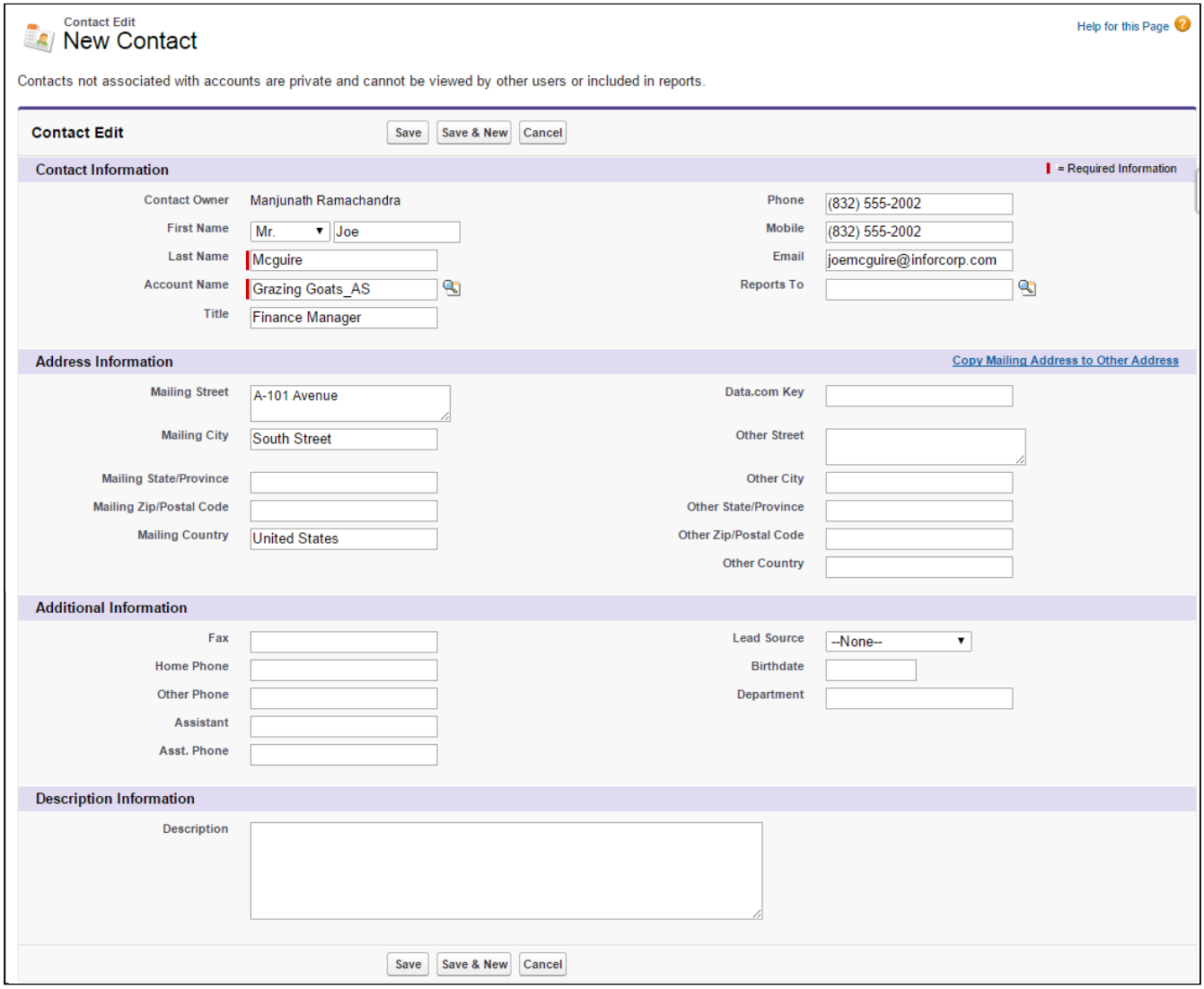
> **Note**
> For different Account Locations, you can add a different Invoice Template and Billing Contact.

After you save information on the Accounts page, you must define the Account Location.

Account Contacts

Account Contact holds the information on persons looking after the account. For any communication, like sending agreements, invoices, emails etc contact persons added under the Account Contacts can be approached.

Go to **Accounts**, under the **Contacts** related list, click **New Contact**.



After you are done with adding all contact related information, click **Save**.

Account Location

You can associate one or more locations with each customer account. You can use different account locations as shipping address, billing address, and tax invoice address.

You can generate an invoice separately, for purchases made from each account location. Each Account Location may also have a separate tax policy or tax rate.

For example, you are an executive in the billing department of an online university. One of your customers want to add their new location in Miami to their account. With Apttus Billing Management, you can add a new location to your customer's account, and generate a separate invoice for products and services that you bill to this new location.

To add a new location

1. Click the **Accounts** tab.
2. Select the account you want to add a new location for.
3. Go the **Account Locations** related list and click **New Account Location**.
4. Enter values for the fields described in the following table.

Field	Description
Location Name	Enter a name that is unique to this location so other users can distinguish it from other locations for the same account.
Type	Describe if the location is a billing address, shipping address, or both.
Address (Street, City, State, Postal Code)	Enter the detailed address with specific values for each field.
Tax Exempt	Select **Yes** if the purchases made for this location are exempted from state and federal taxes.
Tax Exempt Status	Select the Tax Exempt status- Processing, Approved or Rejected.
Tax Certificate	Select the Tax certificate that you want to apply to this account.
Is Default	Select the check box if you want to make this address the default address for Billing.
Account	Select the Account (mandatory) for which you want to add this location.
Payment Term	Select the Payment Term you want to apply to invoices you generate for this location.
Billing Preference	Select the Billing Preference you want to apply for this location.
Billing Day of Month	Select the day of the month when you want to generate an invoice for this location of the account.
Invoice Separately?	Select **Yes** if you want to generate a separate invoice for purchases made for this location.
Default Invoice Template	Select an Invoice template that you want to apply as default for invoices generated for this location. For more information on Invoice Templates, please refer Invoice Templates.

Field	Description
Default Invoice Statement Template	Enter the name of an Invoice Statement template that you want to apply when generating Invoice statement for this Account Location. An **Invoice Statement** incorporates amount owed from past invoices till the specified date.
Invoice Email Template	Enter the name of an Invoice Template which you want to send as an attachment when the invoice is automatically emailed via Invoice Runs.
Dunning Policy	Select a Dunning Policy to override the default dunning policy with an account location specific dunning policy. For details, refer to Dunning Policy
Exclude from Dunning	Select this field to disable dunning for this account location.

5. Click **Save**.

You can also define filters for an Invoice Run based on the Account or Account Location object and view the specific filter applied on the Invoice Run detail page.

After you have defined your customers' account and account location, you can define your customers' Billing Preference.

Set Billing Contacts

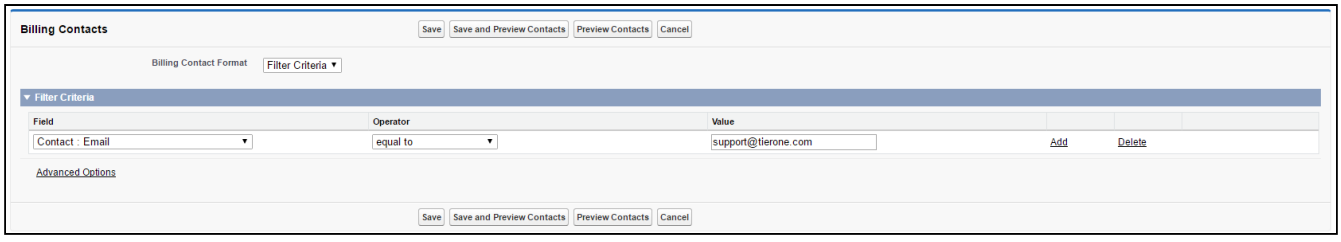
You might have multiple contacts linked to an account. For sending emails to a few selected contacts from the Account Contacts, click **Set Billing Contacts** on the Account.

You can add filter criteria for picking out contacts you wish to email Invoice or Credit Memo documents.

For example, you wish to add all support contacts on an Account to your mailing list, your filter condition can be

```
<Filter>  Contact: Email
<Operator> Equal to
<Value> support@tierone.com
```

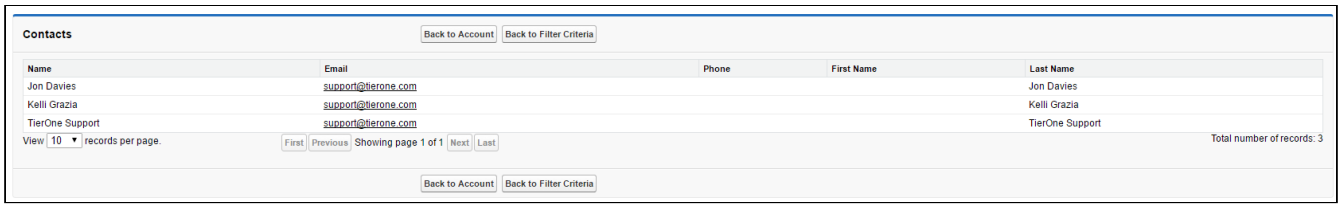
.



After adding the filter conditions, you can **Save and Preview** to save the filter criteria and preview added contacts.

If you have already saved the filter criteria and wish to see the included contacts, click **Preview Contacts**. If you make any modifications to the filter criteria and click Preview, you will not get the accurate preview.

> **Note**
> To see the Preview with recent results, you must save the Filter Criteria. Therefore it is recommended that you **Save and Preview** after updating the filter conditions.



For the given example, all contact emails matching the given value will be added to the mailing list.

Billing Custom Classes

The following custom classes are available in Billing Management:

- Creating Custom Filters for Invoice Runs using Filter Callback
- Separate Invoicing using Grouping Callback
- Custom Numbering for Invoice using Invoice Numbering Callback
- Email Invoice Attachment Callback Class
- Email Credit Memo Attachment Callback Class

Creating Custom Filters for Invoice Runs using Filter Callback

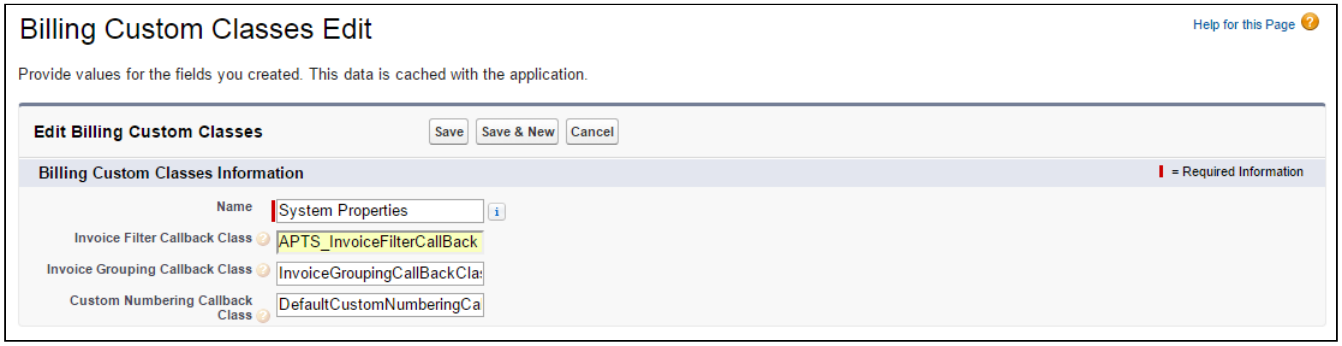
Apart from the three filters, Account, Account Location, and Product, you can create Custom Filters to sort and process Invoices.

For example, you are a Billing executive in a retail chain and you want to generate only those invoices for a custom object that is local to your organization and is not a standard object in Apttus CPQ or Apttus Billing Management. You can define **Custom Settings** to call the custom Object in your implementation of Apttus Billing Management.

> **Note**
> The **Invoice Filter Callback Class** value in the above illustration is a class name. Consult Apttus Professional Services or your Apttus Implementation Engineer for the value you must use in your instance of Apttus Billing Management.

To define the Custom Settings,

1. Click **Setup** > **Custom Settings** > **Manage** (to the left of **Billing Custom Classes)** > **System Properties** > **Edit**.
2. Enter a value for **Invoice Filter Callback Class** and click **Save.**



When you define a value for the Invoice Filter Callback Class, the Invoice Run User interface is updated with the Custom Options section that has 2 fields, **Category,** and **Sub Category**.

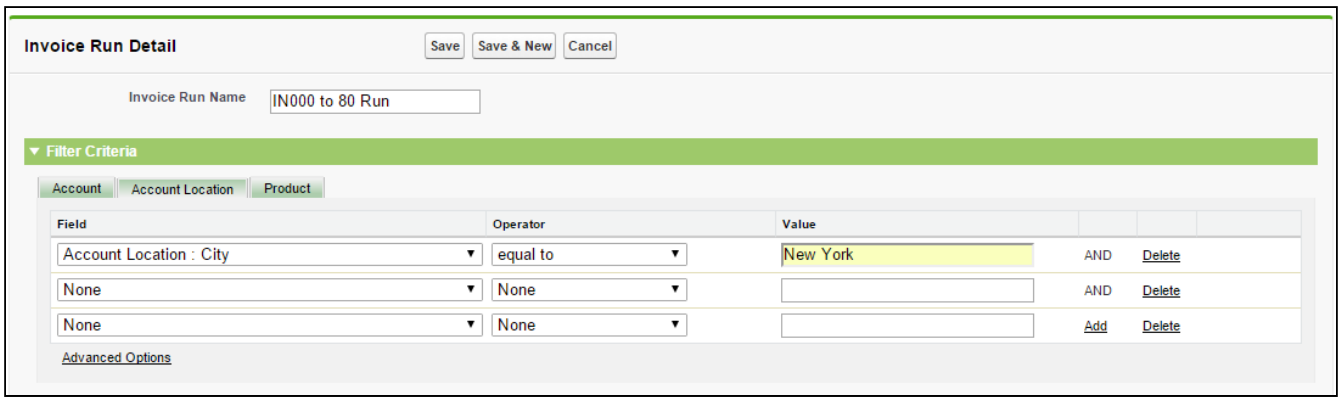
Select the values with the from each field with the picklist you have defined and the resulting Invoice Run is processed based on the fields you define here. For example, if a customer wants to only apply the **Invoice Run** to one or more Business Partner countries (a custom field added to the Asset Line Item) then the **Category** pick list should be updated to contain all countries applicable to their application: Germany, France, Italy, and others. Based on the country selected in **Category**, selected Billing Schedules will be picked by Invoice Run. If further granular filtering is needed then you can leverage the **Sub Category** pick list.

Separate Invoicing using Grouping Callback

Apttus Billing Management, by default, consolidates assets onto a single invoice. Therefore, if you add products or make any amendments over time, those will be included a single invoice.

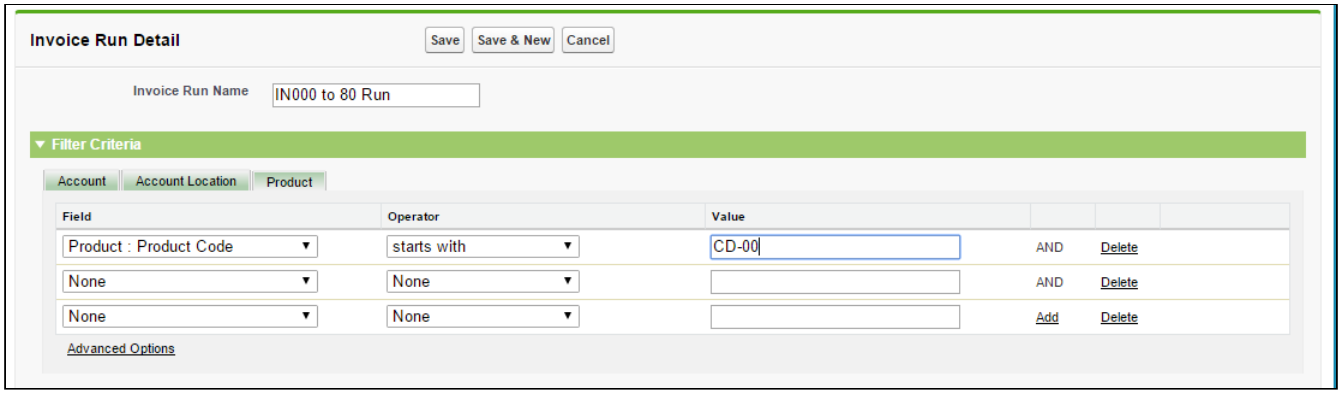
However, you have to option to generate separate invoices for each of the Billing Schedules.

Assume that for an asset, Ready for Invoice Date for 4 Billing schedules is 3/20/2016, 4/20/216, 5/20/2016 and 6/20/216. Create a new **Invoice Run** and set **Process Through Date** as 6/30/2016; only 1 invoice will be generated.



Now go to **Custom Settings> Billing Custom Class** and set **Invoice Grouping Callback Class** to *testinvoicegroupingcallback*. After this configuration, the invoices will be segregated on the basis of **Ready for Invoice Date** for Billing Schedules. Consult Apttus Professional Services or your Apttus Implementation Engineer for the value you must use for the class name.

For the same asset, now 4 invoices will be generated.



Also, you can separate invoices by order, or by a location based on the filter options set on the Invoice Run. Please refer Defining Filter Criteria in Billing Management User Guide for more information.

> **Note**
> If **Billing Rule** is set to **Ready For Billing Date**, only one invoice will be generated irrespective of this configuration.

Custom Numbering for Invoice using Invoice Numbering Callback

You can stamp custom number on the invoices by adding a callback class for Invoice Numbering.

Email Invoice Attachment Callback Class

You can now email invoice attachments to a customized contact list. Refer to Email Invoice Attachment Callback Class for more details.

Email Credit Memo Attachment Callback Class

You can now email credit memo attachments to a customized contact list. Refer to Email Credit Memo Attachment Callback Class for more details.

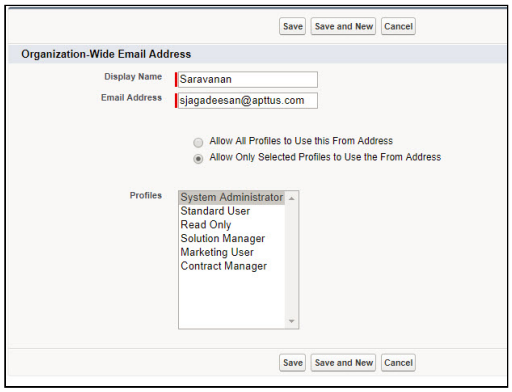
Billing System Properties

You can drive the functioning of certain features in the application from the fields available in Billing System Properties.

From **Setup**, enter **Custom Settings** in the *Quick Find* box and search for Billing System Properties. Then select **Billing System Properties** and click **Manage** to see **System properties**. **Edit** System Properties to provide value for the fields.

⚠ You must name the System Properties record as **System Properties**.

The following table lists the fields available for edit, their significance and the functionalities they impact.

Field	Description	Pre-requisite for Functionalities
Allow Billing Preference Override	A default Billing Preference is applicable to all account org-wide. Select this check box to allow *overriding* the default Billing Preference.	
Billing Org-Wide Email Address	Enter the email address which you want to use for sending invoice emails. You need to create a record for the email address in **Organization-Wide Email Addresses** Object.  If you do not provide a default email address name here, the system will send emails using the email address of the user who is generating invoices. If you see sender's email address as no-reply@salesforce.com on invoice emails, go to **Setup** > **Administration Setup** > **Email Administration** > **Deliverability** and clear **Enable Sender ID compliance** check box. This way you can send emails on behalf of the email address you set on **Billing Org-Wide Email Address**.	Invoice Email Delivery
Create redundant Billing Schedules	For a product with invoiced Billing Schedules, if you perform a mid-cycle amend without changing the amount, redundant credit and debit Billing Schedules that cancel each other are created. Select the check box to enable the creation of such redundant Billing Schedules.	
Currency Decimal Places	Enter the number of decimal places to scale the currency fields.	

Field	Description	Pre-requisite for Functionalities
Custom Billing Plan Item Percent API Name	Enter the API name of the field used to calculate the percentage amount of the Billing Plan Items and Billing Plan Template Items. To leverage this functionality for existing billing plans and billing plan templates, refer to Post-Installation Configurations.	
Generate Invoice Documents	Select this option to generate an invoice document in the required format (DOC, DOCX, RTF or PDF) during the Invoice Generation Batch job. You can set the format from the field **Invoice Output Format** on the **Billing Preference**.	Invoice Email Delivery
Invoice Email Delivery Period	This period accounts for invoices generated within the specified *number of days* from the day of Invoice Run. For example, if the Invoice Email Delivery period is set to 5, the invoices generated in the past 5 days (with Status *Approved* and Delivery Status *Pending*) from the day of Invoice Run will qualify for the email delivery.	
Invoice Generation Batch Size	Enter the number of accounts you want the Invoice Runs to process in one batch for generating invoices.	Invoice Runs
Tax Level	Enter the *object name* to which you want to apply taxes by default. You can mention either **Account** or **Product** object for the default tax applicability.	
Generate Credit Memo Documents	Select this option to generate a document for Credit Memo in the required format (DOC, DOCX, RTF or PDF) as part of the Invoice Run. You can set the format from the field **Invoice Output Format** on the **Billing Preference**.	Credit Memo
Same Day Cancellation	Select this option if you want the Asset Cancellation applicable on the same day. For example, while performing Termination on an Asset you enter the Termination Date as 5/11/2016. If you select **Same Day Cancellation**, the cancellation will be effective from 5/11/2016. Otherwise, the cancellation will be in effect a day later which is 5/12/2016. Same Day Cancellation is applicable by default. Clear the checkbox to apply cancellations after a day.	

Field	Description	Pre-requisite for Functionalities
Create Activity for Billing Emails	Select this option to create an email activity record for all Invoice or Credit Memo emails sent to the Account Contacts. You can see the records under **Activity History** related list on the related Invoice/Credit Memo detail page. For details, refer Credit Memo Detail.	
Usage Rating Batch Size	Enter the numeric value to specify the number of Usage Inputs to process per batch when rating the usage. The default value is 200.	Usage Rating
Credit Memo Display Fields	Comma separated API names of invoice line item fields that should be displayed on the credit memo page	
Wallet Flow	Name of the wallet manager flow,	
Delete Forecast	Select this option to delete the forecast schedules when the actual billing schedules are created. **Note:** Do not select this option if you want to retain the forecast billing schedules linked to the proposal document.	
No Proration on Renewal	When performing asset-based operations, select this option to consider the start date of the asset renewal for alignment of the billing schedules instead of the original start date set at the Quote/Proposal.	

Third-party Integrations

Follow the Steps given in this section for third-party integrations.

Choose the Right API

Select any one approach from REST or SOAP for interfacing to the third-party web services. You will need end-points or URL for both.

Authentication Protocol

Get the Username and password to access the third-party database.

Authentication Response

Once the authentication request is validated, the License key and certificate is passed along with it.

Validation Structure

Confirm the duration of certificate and License key validity.

Considering above specifics, after you pick one from the REST and SOAP API, take a detailed approach.

If you pick REST, identify the services provided. All requests will be in the form of a URL (HTTP/ HTTPS). Pass request parameters such as Product, Price, Account Location etc and the Authentication Token as part of the URL.

If you opt for SOAP the communication will be XML based. Get the class file from third-party vendor using which you can generate Apex classes. You'll also need Request and Response parameters.Invoke the methods from these Apex classes.

You can also combine REST and SOAP approaches. You can perform authentication using REST and call services using SOAP.

Post Integration

- After integration you should architecture the Error mechanism. It is necessary to let the client know about the success and failure of an API call with the help of good error messages.
- Identify the mode of notification to the admin or technical team about the errors. For example, you can configure an auto-email on failure of a job.

Template Configuration

You require a template to define how you present information on an Invoice and Credit Memo.

You can use **X-Author for Contracts** to create Invoice, Invoice Statement template, and Credit Memo.

Take a look at how you can create Invoice and Credit Memo Email templates in the subsequent sections.

Credit Memo

Credit Memo templates can be created from the Template object located in the header.

Pre-requisite

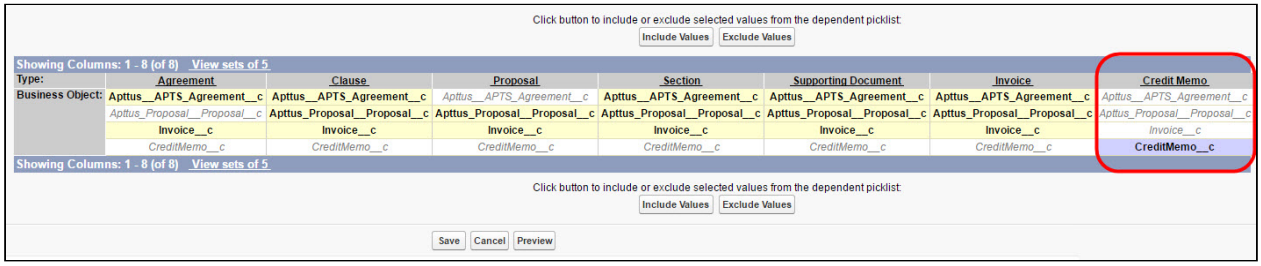
Before creating a template of the type Credit Memo, you need to add this type as a picklist value on **Template** object.

Follow the steps to add a Credit Memo type picklist:

1. Go to the **Template** object, under **Custom Fields & Relationships** click the **Business Object** Field Label.
2. In the **Picklist Values** section, click **New** to add a new value. Enter *CreditMemo__c* as the new picklist value and **Save**.

> **Note**
> If you are adding this value after installing the Billing Package, the picklist value must be a fully qualified object name like Apttus_Billing__CreditMemo__c.

3. Go back to the **Custom Fields & Relationships** and click the **Type** Field label.
4. In the **Picklist Values** section, click **New** to add a new value. Enter *Credit Memo* as the new picklist value and **Save**.
   Now that you have Credit Memo picklist added to both the fields, next step is to associate the values.
5. In the **Type** field, under **Field Dependencies** section click **Edit** next to the **Business Object** field.



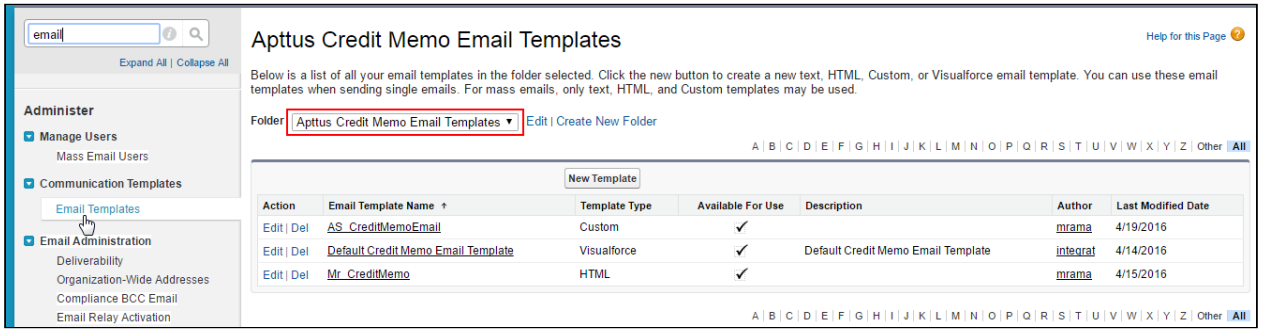
6. Under **Credit Memo** column, select **CreditMemo__C** and click **Include Values** button to include the picklist values. Click **Save**.

You should see **Credit Memo** added to the Template **Type** field.

To create a Credit Memo template using Microsoft Word, refer to Managing Templates and Cloning a Template sections in the X-Author Contracts User Guide.

Creating a Credit Memo Email Template

1. Go to **Setup** and type email in the *quick find* box**.** Alternately, open **Email Templates** under **Communication Templates**.
2. Select *Apttus Credit Memo Email Templates* as the **Folder**. Click **New Template** and select a template type from Text, HTML, Custom or Visualforce.



3. Provide template information such as Email Template Name, Subject, and Email Body.

The templates you create will be added to the **Apttus Credit Memo Email Templates** folder.

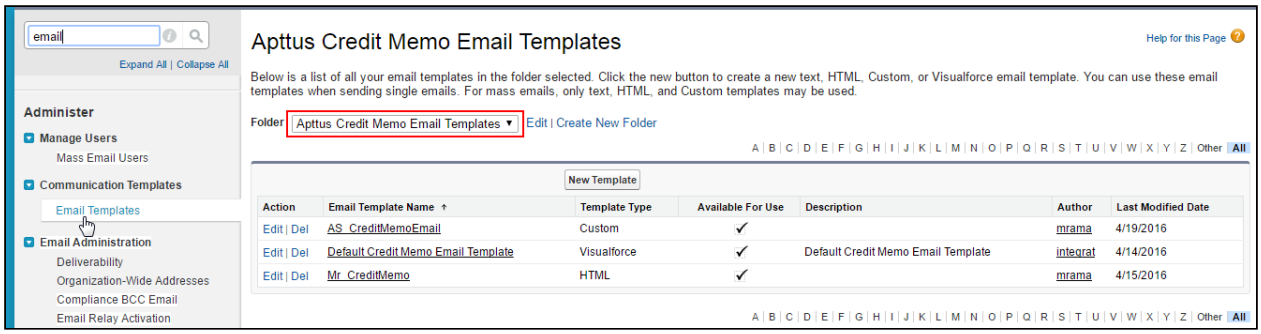
After creating a Credit Memo template and a Credit Memo Email Template, you can apply them at the Account or Account Location level.

Credit Memo Email Template

You need to create a credit memo email template and apply it to the account or account location.

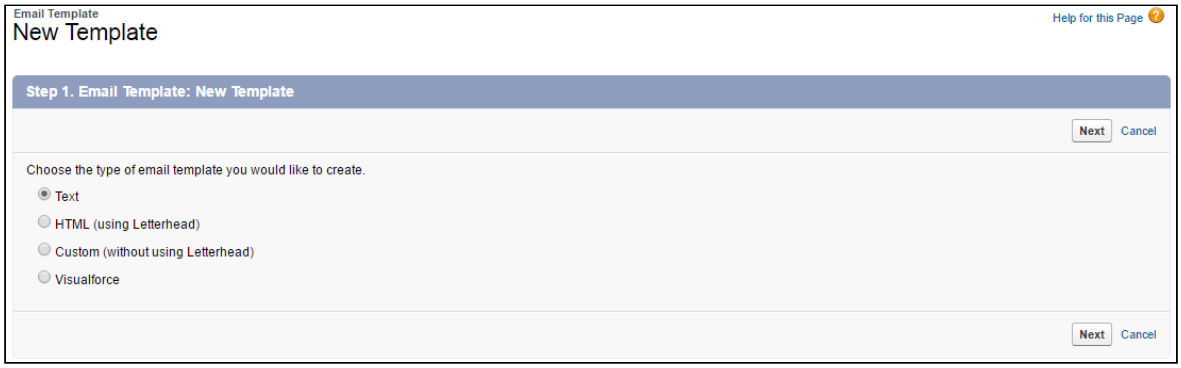
Creating a Credit Memo Template

1. Go to **Setup** and type email in the *quick find* box**.** Alternately, open **Email Templates** under **Communication Templates**.
2. Select *Apttus Credit Memo Email Templates* as the **Folder**. Click **New Template** and enter values for the fields described in the following table:



3. Click **New Template** and select a template type from the list of options.
   a. Select **Text** to use merge fields to personalize your email content.
   b. Select **HTML** to use Salesforce's pre-defined Email layout options to create your own HTML template.
   c. Select **Custom** to create your own HTML layout from scratch.

d. Select **VisualForce** to use Visualforce for creating email templates.Developers and administrators can use this option to perform advanced operations on data that'll be sent to recipients.

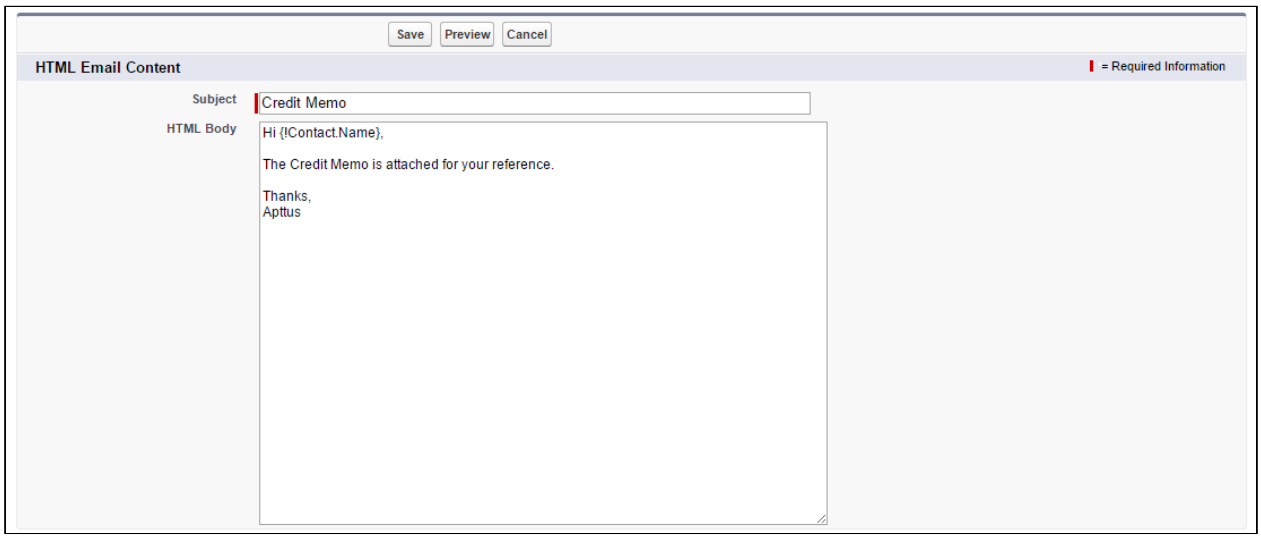


4. Enter the template information such as **Folder**, **Template Name, Email Layout** and **Description**.

> **Note**
> You must select **Available For Use** option to make the template visible and available for selection.

5. Using the **Merge Field Options**, you can add dynamic fields to your email body. As shown in the following screen, on selecting the *Name* field from the **Select Field** picklist, you get a **Copy Merge Field Value** *'{!Contact.Name}'*.

Paste this value in the **Email body** to display the contact name in the email.



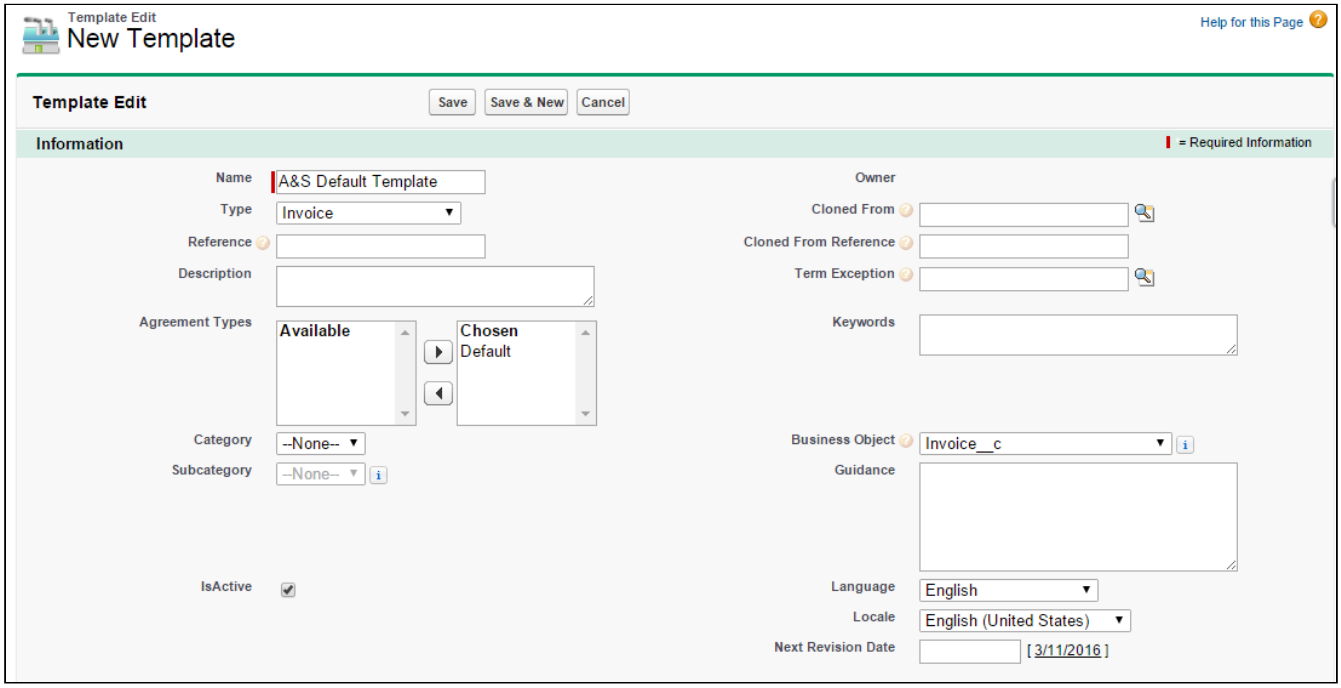
6. Click **Save**.

The templates you create will be added to the *Apttus Credit Memo Email Templates* folder. You can add multiple templates for Credit Memos and specify one for applicability at the Account or Account Location level.

Invoice

You require a template to define how you present information on an Invoice. You can use **X-Author for Contracts** to create an Invoice and an Invoice Statement template.

To learn about templates, Term exceptions, merge fields and child objects, see the X-Author Contracts User Guide. To know more about creating a template using Microsoft Word, see Managing Templates and Cloning a Template sections in the X-Author Contracts User Guide.



To create a new template,

1. Click **+** > **Templates** > **New**.
2. Enter values for the fields described in the following table:

Field	Description
Information section	
Name	Give this template a unique name to help you and others in your organization identify it with ease.
Type	Select from one of the following options. • Invoice - Select this option if you want to create a template for an Invoice • Invoice Statement - Select this option if you want to create a template for an Invoice Statement
Reference	Enter a unique external reference identifier to help you and others in your organization identify it with ease.
Description	Enter a brief description of the functions and purpose of this template for the benefit of other in your organization who may want to use this template.
Agreement type	Select Default.
Category	Select Default.

Field	Description
Subcategory	Select Default.
Is Active?	Select the check box to activate the template and make it available for use.
Cloned from?	If you are cloning this template, lookup and select the source you are using to clone this template.
Cloned from Reference	Enter the unique external reference identifier of the source you are using to clone this template.
Term Exception	Lookup and select the Id for the term exception associated with the clause or section.
Keywords	enter the keywords that will help other users effectively search for this template.
Language	Select the language you want to present your invoice in.
Locale	Select the geographical region where you are presenting this invoice.
Next Revision Date	Select a date when you want to revise this template.
Publishing section	
Needs Publishing?	Select the check box to publish this template.
Published Date	Select a date to publish this template.
Published Doc ID	Enter the Document ID of this template.
Content and Merge fields section	
Text Content	Enter the verbiage that you want to include in the template. For more information, contact your manager or the Legal team in your organization.
Merge fields	Enter the names of the merge fields and clauses you want to include in the template.
Excluded Merge Child Objects	Enter the names of the child objects that you want to reference in the template.

3. Click **Save**.

Now, you can apply this template to generate an Invoice.

Overriding an Invoice Template

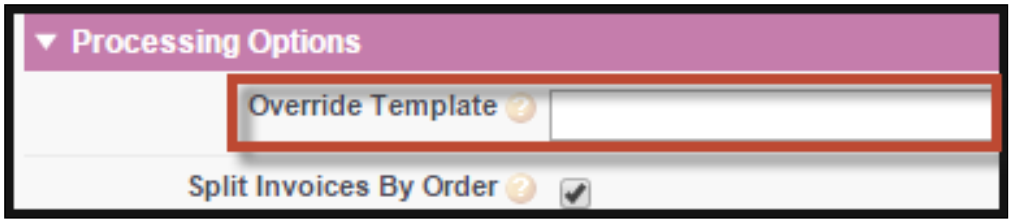
With the Apttus Billing Management application, you can now override the Invoice Template you use.

You can create several templates on which your invoices are based. You can select the Invoice template at the account or account location that you want to use. However, if you want to override this account wide choice and switch to a different template, you can do so with this newly added Invoice Run page.

For example, you work in billing operations at a software company. You want to generate an invoice in a different format only for one invoice run, but do not want to change the default template at the account level. Input the name of the template you wish to override with and run the invoice run to create invoices using a new template.

To Override an Invoice Template,

1. Create a new Invoice run.
2. Enter appropriate values for required fields.
3. In the Processing Options section, enter the name of the Invoice template you want to use.
4. Click **Save**.

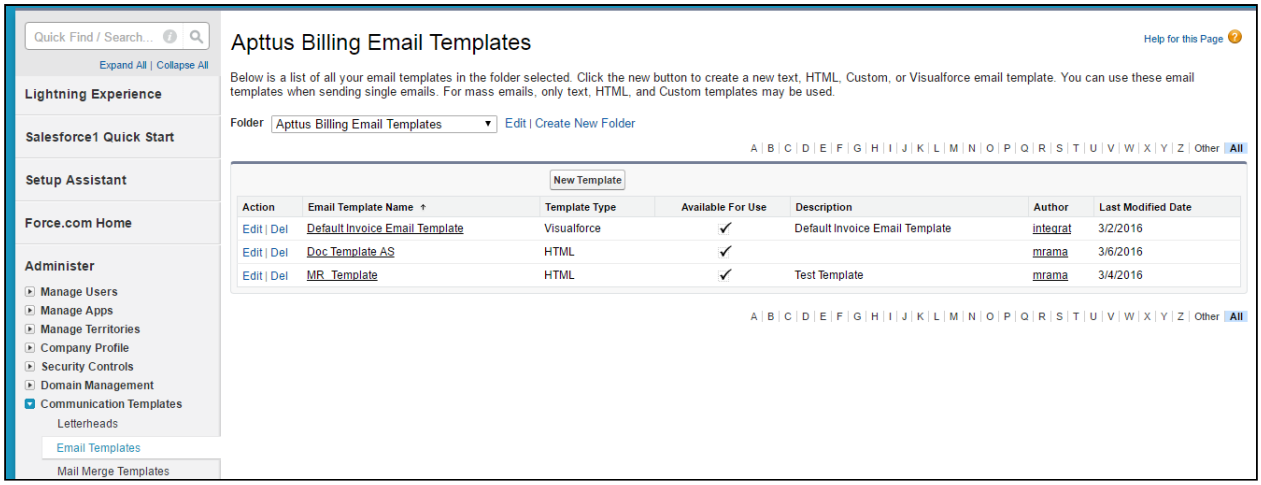


Now that you have a template ready to base the invoice on, you can start to configure fields and define values for these fields to present on the invoice that you send to your customer.

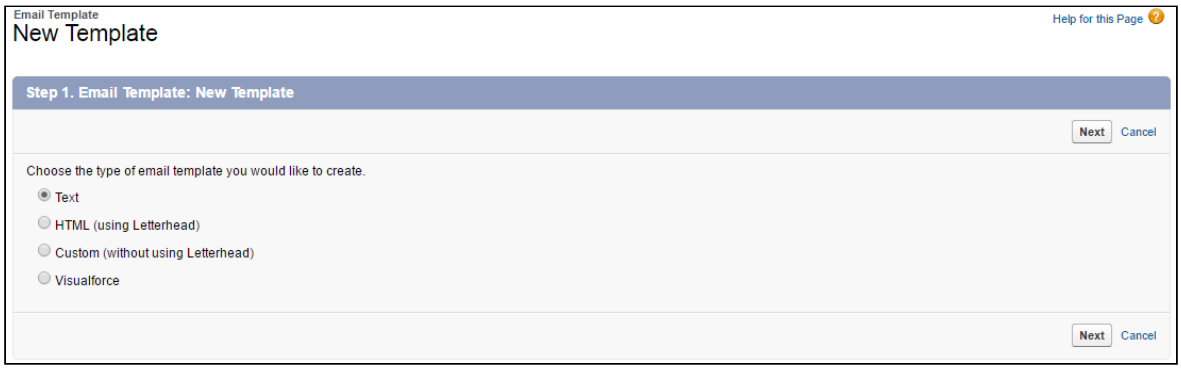
Invoice Email Template

Creating an Invoice Email Template

1. Go to **Setup** and type email in the *quick find* box**.** Alternately, open **Email Templates** under **Communication Templates**.
2. Select *Apttus Billing Email Templates* as the **Folder**.



3. Click **New Template** and select a template type from the list of options.
    a. Select **Text** to use merge fields to personalize your email content.
    b. Select **HTML** to use Salesforce's pre-defined Email layout options to create your own HTML template.
    c. Select **Custom** to create your own HTML layout from scratch.
    d. Select **VisualForce** to use Visualforce for creating email templates.Developers and administrators can use this option to perform advanced operations on data that'll be sent to recipients.

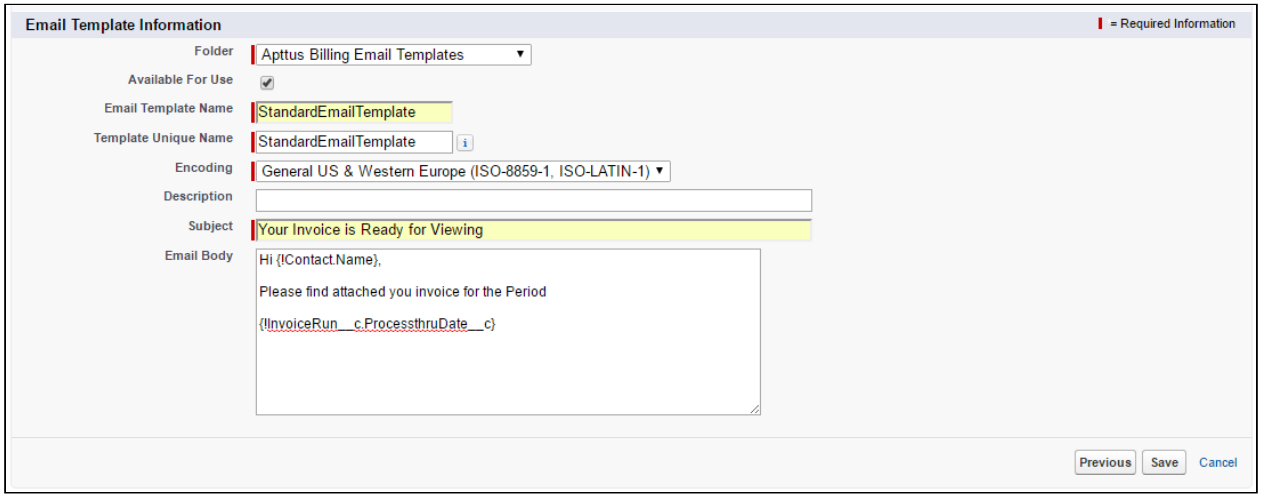


4. Enter the template information such as **Folder**, **Template Name, Email Layout** and **Description**.

> **Note**
> You must select **Available For Use** option to make the template visible and available for selection.

5. Using the **Merge Field Options**, you can add dynamic fields to your email body. As shown in the following screen, on selecting the *Name* field from the **Select Field** picklist, you get a **Copy Merge Field Value** *'{!Contact.Name}'*.

Paste this value in the **Email body** to display the contact name in the email.



6. Enter the **Subject** and click **Save**.

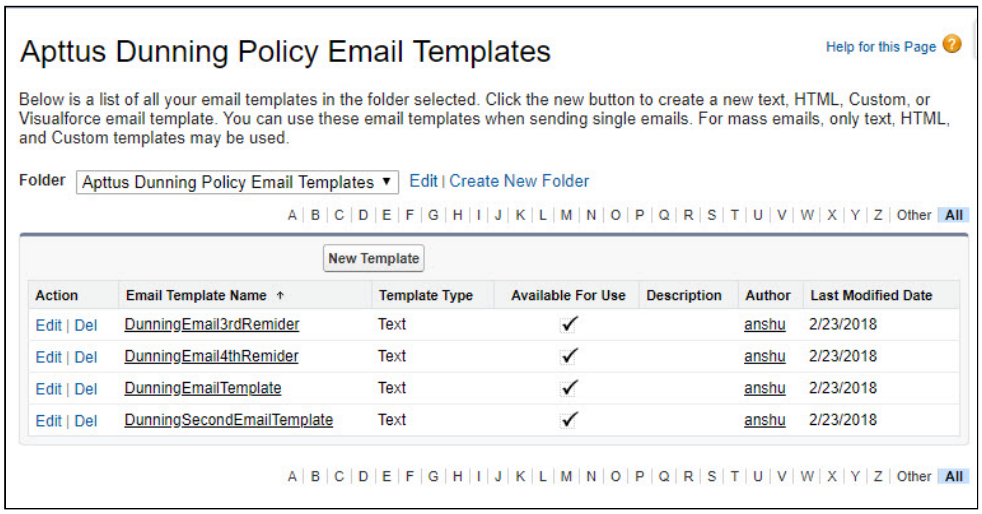
The templates you create will be added to the *Apttus Billing Email Templates* folder. You can add multiple templates for Invoice and specify one for applicability at the Account or Account Location level.

Dunning Policy Templates

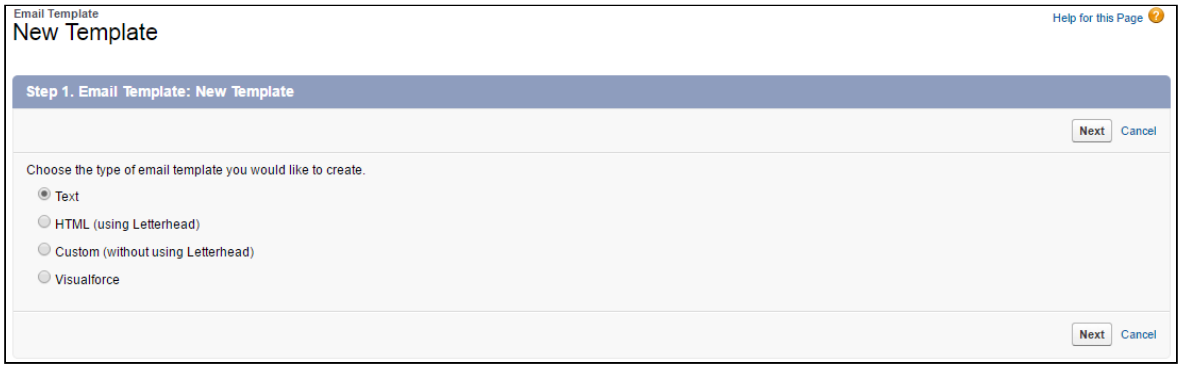
You can create different types of email templates for your dunning policy.

Creating a Dunning Policy Email Template

1. Go to Billing Management Console and Click **Dunning Policy Email Templates** under *Email & Communication*. Alternetively,Go to **Setup** and type email in the *quick find* box. Open **Email Templates** under **Communication Templates**.
2. Select *Apttus Dunning Policy Email Templates* as the **Folder**.



3. Click **New Template** and select a template type from the list of options.
    a. Select **Text** to use merge fields to personalize your email content.
    b. Select **HTML** to use Salesforce's pre-defined Email layout options to create your own HTML template.
    c. Select **Custom** to create your own HTML layout from scratch.
    d. Select **VisualForce** to use Visualforce for creating email templates.Developers and administrators can use this option to perform advanced operations on data that'll be sent to recipients.



4. Enter the template information such as **Folder**, **Template Name, Email Layout** and **Description**.

> **Note**
> You must select **Available For Use** option to make the template visible and available for selection.

5. Using the **Merge Field Options**, you can add dynamic fields to your email body. As shown in the following screen, on selecting the *Name* field from the **Select Field** picklist, you get a **Copy Merge Field Value** *'{!Contact.Name}'*.

Paste this value in the **Email body** to display the contact name in the email.

6. Enter the **Subject** and click **Save**.

The templates you create will be added to the *Apttus Dunning Policy Email Templates* folder. You can add multiple templates for different reminders Invoice and specify one for applicability in the Dunning Policy Line Item.

Legal Entity based Template Configuration

Legal Entities are individual business units within an organization. They are responsible for conducting business transactions with customers across geographies. For example, your organization can have a US Legal Entity and a UK Legal Entity. The US Legal entity is responsible for conducting business transactions with customers in North America. The UK legal entity is responsible for conducting business transactions in the EMEA Region.

For more information on legal entities, refer to **Legal Entities** in *Revenue Recognition on Salesforce Administrator Guide.*

You can have different document templates for different Legal Entities. Link the legal entity to your Quote/ Proposal to generate a document with a legal entity based template.

To set-up a document template for a Legal Entity:

1. From All Tabs, click **Legal Entity Template Settings Entries.**
2. Enter values for the following field:

Field	Description
Document Template Name	Enter the name of the template to use for PDF generation
Delivery Type	Select the delivery method for the document. **Note: Currently we do not support this field.**

Field	Description
Document Type	Select the document type from one of the following: • Invoice • Split Invoice • Credit Memo • Wallet Credit • Dunning
Email Template Name	Enter the name of the Email Template to use while sending email notifications **Note: Currently we do not support this field.**
Legal Entity	Enter the name of the Legal Entity

3. Click **Save.**

You can use this functionality to have legal entity based templates for the following documents:

- Invoice
- Split Invoice
- Credit Memo
- Wallet Credit
- Dunning

Template Preferences

Apttus Billing Management follows the following order of preference when assigning a document template:

1. Template specified in the Invocie Run if Template Over Ride is set to true
2. Account Location Template
3. Account Template
4. Legal Entity Template

If you do not specify a template in any of the four objects, the default template is used.

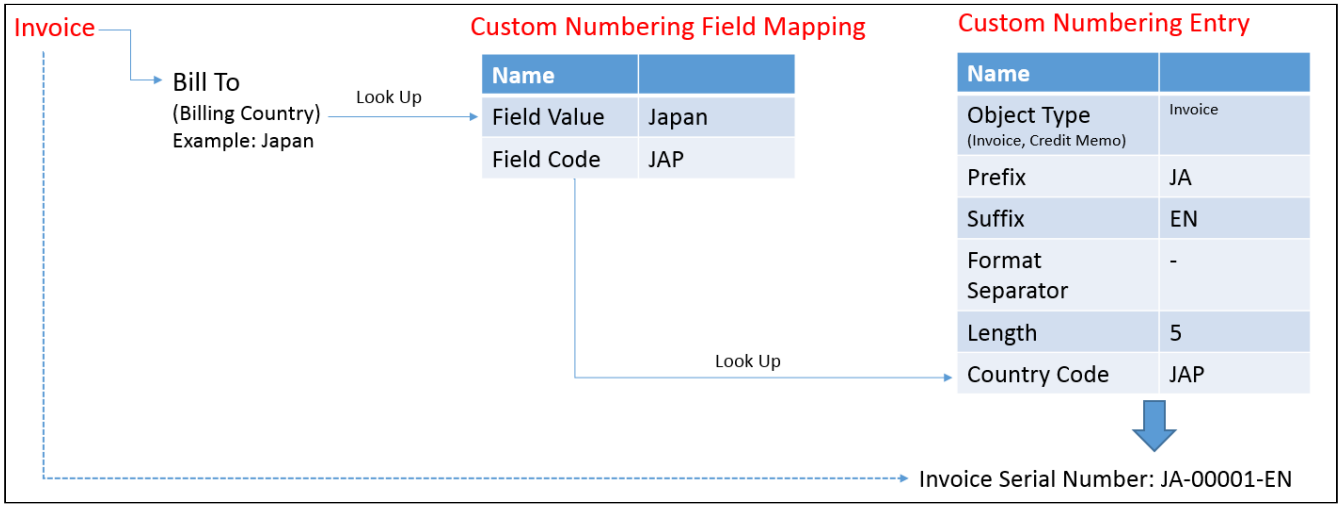
Custom Invoice Numbering

Invoices are numbered according to a pattern formed by adding dynamic prefixes and suffixes. Invoice Numbers usually contain the country code and other numeric or alphabetical identifiers that make an Invoice easy to track for different states or regions. Invoice numbering also helps in conforming to a legal or auditing requirement. With a couple of configurations, you can apply a sequential pattern of Invoice Numbering on all approved invoices.

> **Note**
> Custom Number is not assigned to an Invoice on Creation. Only after the invoice is 'Approved' or an existing invoice transitions to the 'Approved' or 'Pending Approved' state, Custom Invoice Numbering is triggered.

Overview

When an Invoice is approved, value in the field **Billing Country** (which is derived from the **Bill To** address on Account) is matched with the value of country fields stored in a mapping table (Custom Numbering Field Mapping). Each country value has a corresponding code. The equivalent code is looked up in the Custom Numbering Entry table and the same format is applied for numbering the Invoice.



> **Note**
> The evaluation of a mapping is case-insensitive. For example, if the mapping name is 'United States' and the Billing Country on an Invoice is 'united states' there will still be a match because the comparison is *not* case sensitive.

Use Case
Requirement:
Generate Invoices of the Format *STG-xxxx-MT* for Account 'OneTier' when the **Billing Country** in Billing Address is **USA.** Here, xxxx stands for a sequential numeric value which should start from 10.
Procedure:

1. Go to **Setup** > **Custom numbering Field Mappings** and make an entry for the country code. Set the **Field Value** as USA and **Field Code** as US.

2. Go to **Custom Numbering Entry Detail**. Make following entries to form the numbering pattern:
   Field1 = US
   Object Type = Invoice
   Number Length = 4
   Prefix = STG
   Format Separator = -
   Suffix = MT
   Starting Number= 10

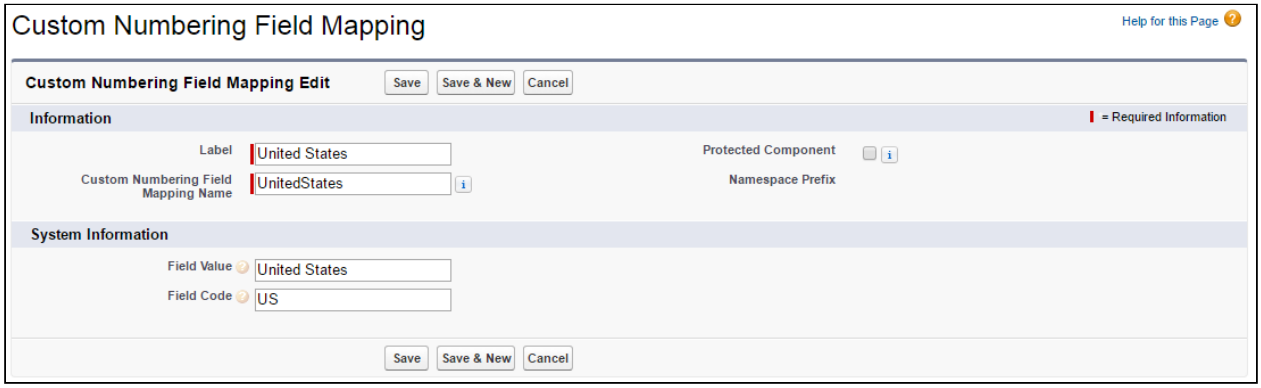
3. Go to **Custom Numbering Configuration Setting Detail** and select **Custom Numbering Enabled** and **Field One**.

4. Go to Invoice Runs, if you want to auto approve invoice, select auto approval. Alternatively you can go to Invoice detail and approve it manually. As soon as the Invoice status changes to 'Approved', you will see the **Invoice Number**.

Configurations

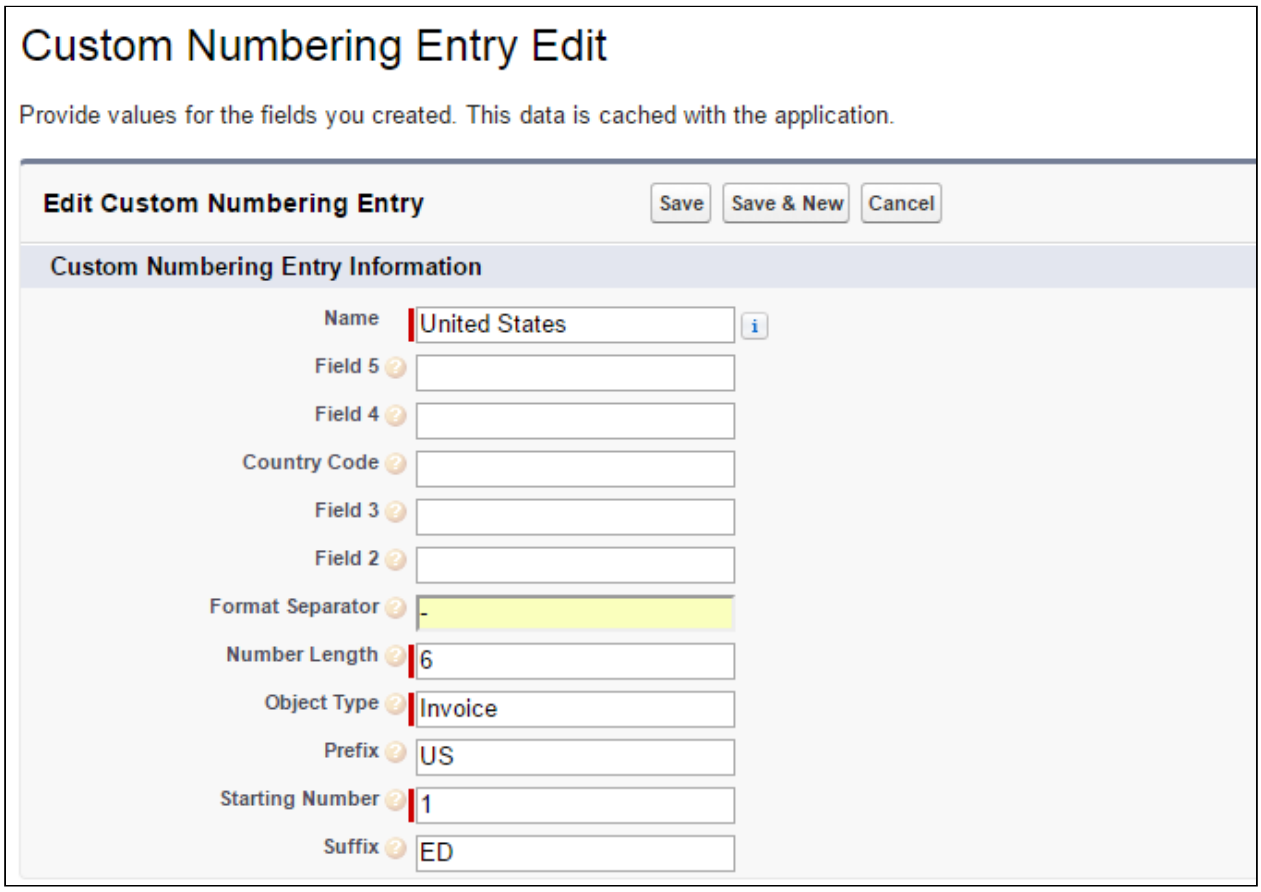
1. Add a **Custom Numbering Callback** class.
   This callback class defines the actions to execute for Custom Numbering implementation.
   Go to **Setup**> **Custom Settings** > **Billing Custom Classes** and click **Edit** to specify the **API Name** of the callback class.

2. Add optional Country Codes and Names under **Custom Numbering Field Mapping**.
   Go to **Setup**> **Develop** > **Custom Metadata Types** and click **Manage Records** for Custom Numbering Field Mapping.



Enter the **Field Value** and **Field Code** and **Save** the mapping. Please ensure the **Field Value** specified here matches the Bill To **Country Name** on Account.

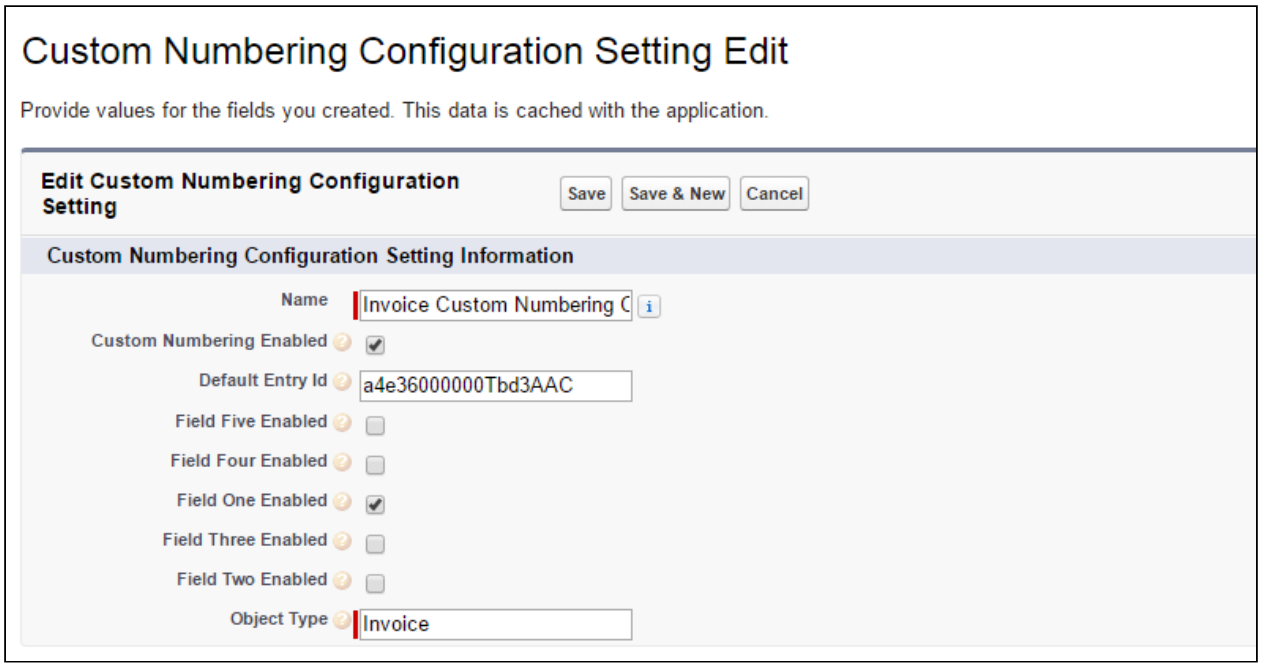
3. Configure **Custom Numbering Entry**.
   This is where you can provide various numbering formats for Object Types such as Invoice, Credit Memos and Payments.
   Go to **Setup> Custom Settings** , search for **Custom Numbering Entry** and click **Manage**. Click **New** to add a numbering entry.



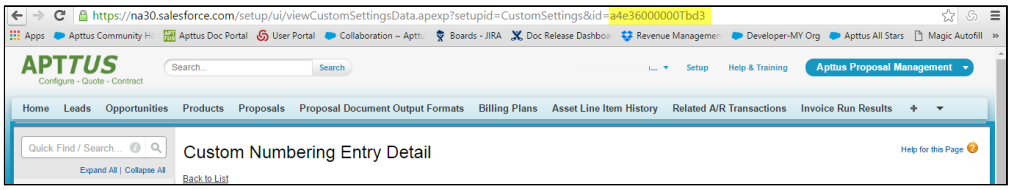
Field	Description
Name	Enter a name for this entry for easy identification.
Format Separator	Enter a symbol (hash, colon, or any other) to separate the Prefix, Number and Suffix. The separator helps to demarcate the dynamic and static fields of the format.
Number Length	Enter the maximum length of an Invoice Number. This value accounts only for the number of digits in the Invoice Number sequence and not the prefix or suffix. The maximum length for invoice number is 99.
Object Type	Enter the Object name which this entry pertains to. The valid values are - Invoice, Credit Memo, Refund and Payment.
Prefix	Enter a Prefix that will be applied at the beginning of the number.
Starting Number	Enter a digit from where you want to start the numbering. If you enter 5 and the Number Length is 3, the numbering shall begin from 005.

Field	Description
Suffix	Enter a Prefix that will be applied at the end of the number.
Field 1	Enter a Country Code which should match the **Field Code** you have defined in the **Custom Numbering Field Mapping**. You can provide value for other fields

4. Configure **Custom Numbering Configuration Setting**.
   This setting is used to enable or disable the custom numbering feature and fields that are verified for applying the numbering.



Field	Description
Name	Enter a name for this configuration.
Custom Numbering Enabled	Select the checkbox to enable Custom Numbering.

Field	Description
Default Entry Id	Out of various numbering formats (Custom Numbering Entries), you can specify one format that will be applicable by default for numbering invoices. Open a Custom Numbering Entry that you want to apply by default, copy its Id from the URL and paste it here.  **Note** If you do not specify a Default Entry Id for numbering and Country on the Invoice does not match with any of the entries, Invoice will not be approved. You will see the following error on Invoice: **Error Status**: *Custom Numbering Invoice Error* **Error Message**: *Cannot create a Custom Invoice Number because no Custom Entry was matched and there is no default.*
Field One Enabled	Select the checkbox to enable Field One on the **Custom Numbering Entry**. When enabled, the value of Field One on the Entry will be matched with the value of **Field Code** on **Custom Numbering Field Mapping**.
Object Type	Enter the Object name which this setting pertains to. The valid values are - Invoice, Credit Memo, Refund and Payment.

5. Add a new field **Invoice Serial Number** on the Invoice object by Editing the Layout.

Note
In case you do not see a custom number on the Invoice in spite of the performing above mentioned configurations, check out **Error Status** and **Error Message** fields on the Invoice to know what went wrong.

Legal Entity based Custom Numbering

You can create custom invoice numbers based on Legal Entity. Custom Numbering functionality is available for Invoice, Split Invoice, Credit Memos, and Wallet Credit records.

Custom Numbering is applied to an invoice when an invoice is generated for an account related to the Legal Entity. All the credit memos and split invoices generated from that invoice also have custom numbers.

To apply custom numbers based on the legal entity,

1. From the legal entity, click **New Legal Entity Custom Numbering Entry** from the Legal Entity Custom Numbering Entries related list.
2. Enter values for the following fields.

Field	Description
Legal Entity	Select the Legal Entity from the lookup field
Numbering Type	Select the object to apply the custom numbering settings
Field Separator	Enter the special character to use as a separator between fields.
Custom Number Creation Option	Select one of the following options: **Approved** - Custom Numbering is only applied if the status of the invoice or the credit memo is Approved **Draft** - Custom Numbering is also applied on draft invoices and credit memos
Initial Sequence Number	The initial number for the custom numbering sequence
Number of Leading Zeros	Enter the number of zeros to precede in the custom number
Static Prefix	Enter the text to use as a prefix
Field Based Prefix	Use this field to prepend values from the legal entity fields for the custom numbering. To enter multiple fields, use comma separated values.
Static Suffix	Enter the text to use as a suffix
Field Based Suffix	Use this field to append values from the legal entity fields for the custom numbering. To enter multiple fields, use comma separated values.

3. Click **Save.**

After each use of the record, the next sequence number is auto-incremented.

Use Case: Creating Custom Numbers for all approved invoices generated for the UK Legal Entity.

Description: This use case describes how to configure custom invoice numbering based on legal entity.

Suppose for the UK legal entity, you want to generate invoice numbers in the following format: INV-CompanyCode-00001-APP

Pre-requisite: Make sure you have a legal entity named UK associated with your quote/proposal.

Procedure:

1. Go to UK Legal Entity. Click **New Legal Entity Custom Numbering Entry.**
2. Make following entries to form the numbering pattern:
   Numbering Type: Invoice

Field Separator: -
Custom Number Creation Option: Approved
Initial Sequence Number: 1
Number of Leading Zeros: 5
Static Prefix: INV
Field Based Prefix: Apttus_Config2__CompanyCode__c
Static Suffix: APP

3. Invoice the billing Schedules.

The invoice is created with the Invoice Number as INV-CODE01-00001-APP. After the invoice is generated, the Next Sequence Number is auto-incremented to 2.

The next generated invoice will have the Invoice Number as INV-CODE01-00002-APP.

Advanced Settings

This section describes some of the advanced settings that will help you monitor the system closely. It includes the following:

- Administering Jobs
- Monitoring Log Files

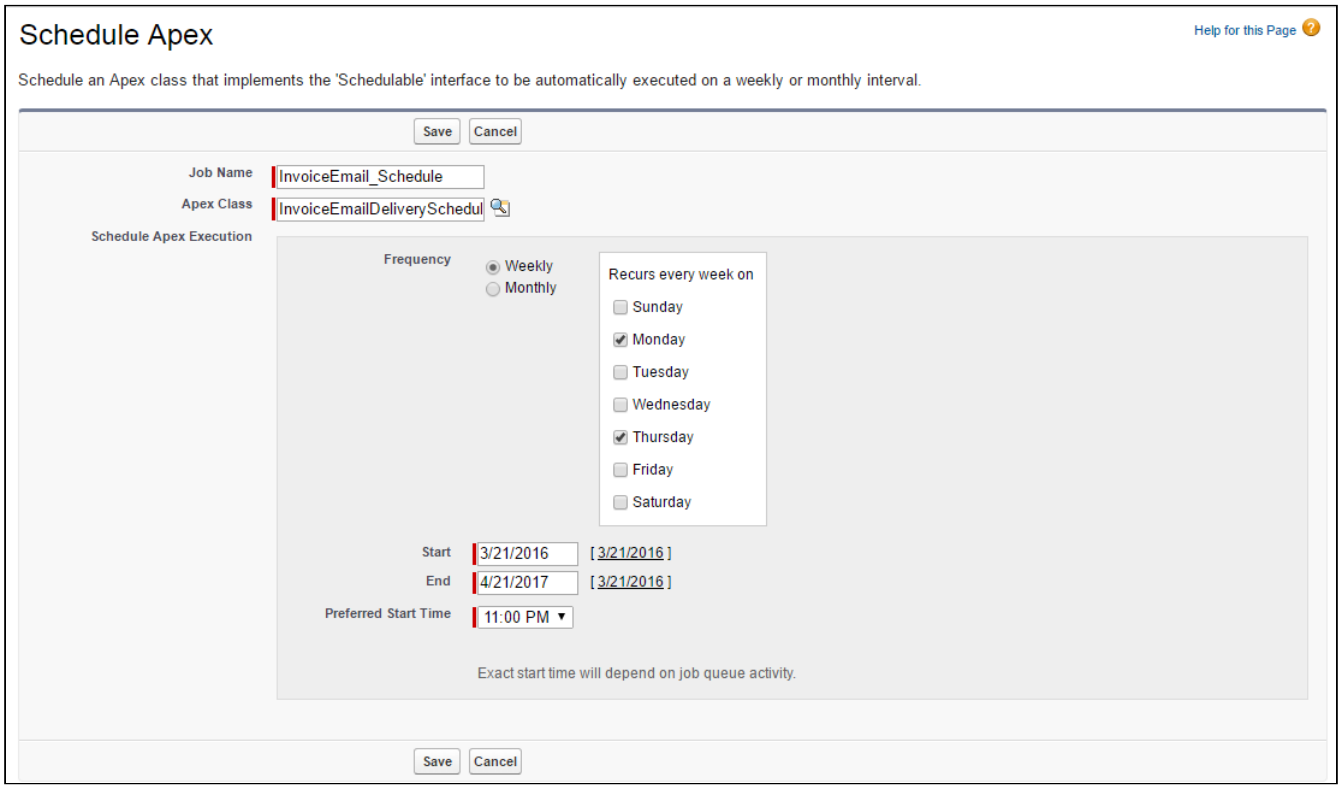
Administering Jobs

This section contains information about the following Jobs:

- Schedule Jobs
- Monitoring Jobs

Schedule Jobs

You can use the Apex Scheduler to schedule execution of *Apex* classes.



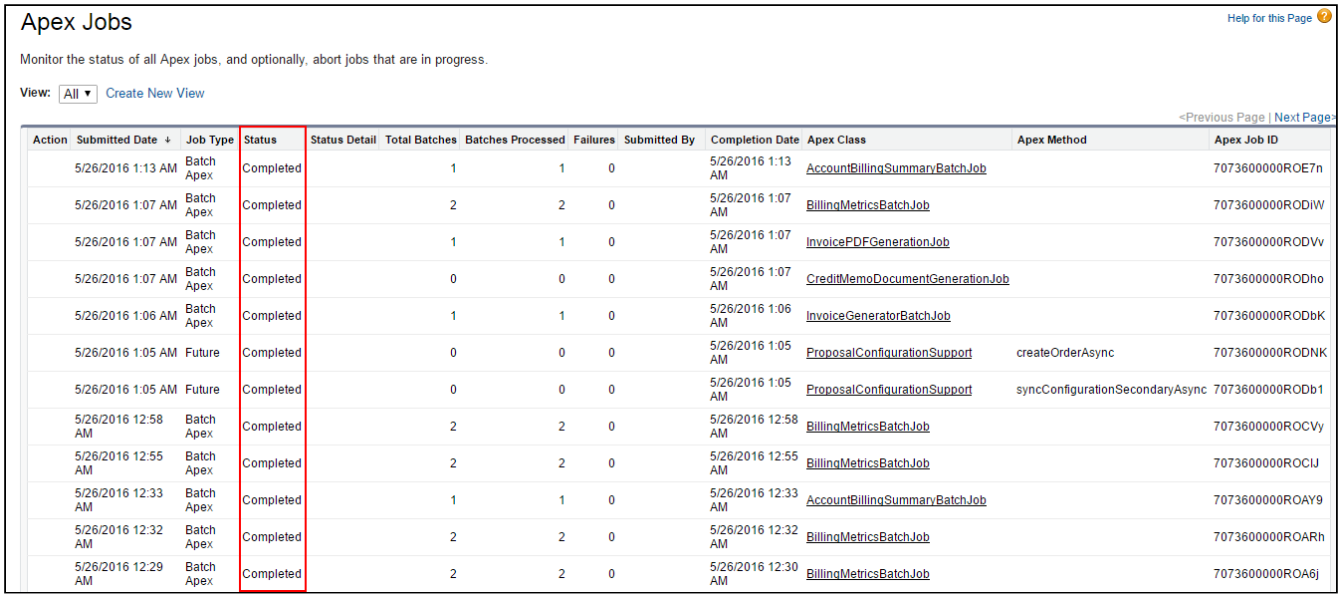
1. Go to **Setup**, enter **Apex Classes** in the *Quick Find box*, select Apex Classes (under **Develop**), and then click **Schedule Apex** button located at the top.

AccountBillingSummaryScheduler
BillingMetricsScheduler
CreditMemoEmailDeliveryScheduler
EvergreenScheduleGenerationScheduler
InvoiceEmailDeliveryScheduler
InvoiceRunScheduler

2. Enter **Job Name**.
3. From the **Apex Class** lookup, select the class you want to set this schedule for.
4. Under Schedule Apex Execution, specify how often the Apex class should run.
    a. For **Weekly**–Select one or more days of the week for the job to run (such as Monday and Thursday).
    b. For **Monthly**–Select either date or day of the job to run (such as second Monday of every month).
5. Enter the **Start** and **End** dates for the Apex scheduled class.
6. Specify a **Preferred Start Time**.
7. Click **Save**.

Monitoring Jobs

A lot of requests such as Invoice Creation, document PDF generation, Billing Summary and few others are submitted as Apex Jobs for execution. To know the status on execution of jobs, go to **Setup** > **Jobs** > **Apex Jobs**.



Apex Jobs

Help for this Page

Monitor the status of all Apex jobs, and optionally, abort jobs that are in progress.

View: All ▾ Create New View

<Previous Page | Next Page>

Action	Submitted Date ↓	Job Type	Status	Status Detail	Total Batches	Batches Processed	Failures	Submitted By	Completion Date	Apex Class	Apex Method	Apex Job ID
	5/26/2016 1:13 AM	Batch Apex	Completed		1	1	0		5/26/2016 1:13 AM	AccountBillingSummaryBatchJob		7073600000ROE7n
	5/26/2016 1:07 AM	Batch Apex	Completed		2	2	0		5/26/2016 1:07 AM	BillingMetricsBatchJob		7073600000RODiW
	5/26/2016 1:07 AM	Batch Apex	Completed		1	1	0		5/26/2016 1:07 AM	InvoicePDFGenerationJob		7073600000RODVv
	5/26/2016 1:07 AM	Batch Apex	Completed		0	0	0		5/26/2016 1:07 AM	CreditMemoDocumentGenerationJob		7073600000RODho
	5/26/2016 1:06 AM	Batch Apex	Completed		1	1	0		5/26/2016 1:06 AM	InvoiceGeneratorBatchJob		7073600000RODbK
	5/26/2016 1:05 AM	Future	Completed		0	0	0		5/26/2016 1:05 AM	ProposalConfigurationSupport	createOrderAsync	7073600000RODNK
	5/26/2016 1:05 AM	Future	Completed		0	0	0		5/26/2016 1:05 AM	ProposalConfigurationSupport	syncConfigurationSecondaryAsync	7073600000RODb1
	5/26/2016 12:58 AM	Batch Apex	Completed		2	2	0		5/26/2016 12:58 AM	BillingMetricsBatchJob		7073600000ROCVy
	5/26/2016 12:55 AM	Batch Apex	Completed		2	2	0		5/26/2016 12:55 AM	BillingMetricsBatchJob		7073600000ROCIJ
	5/26/2016 12:33 AM	Batch Apex	Completed		1	1	0		5/26/2016 12:33 AM	AccountBillingSummaryBatchJob		7073600000ROAY9
	5/26/2016 12:32 AM	Batch Apex	Completed		2	2	0		5/26/2016 12:32 AM	BillingMetricsBatchJob		7073600000ROARh
	5/26/2016 12:29 AM	Batch Apex	Completed		2	2	0		5/26/2016 12:30 AM	BillingMetricsBatchJob		7073600000ROA6j

The **Status** of a job will be **Completed** if it was executed successfully without any failures. If you see the status as **Failed**, the **Status Detail** will show the error message to help you identify what went wrong.

Monitoring Log Files

A debug log records all operations that occur when executing a transaction or running tests. The activities are classified as:

- Database Operations
- System Processes
- Errors

Billing Package Objects

This page lists the objects and its description.

Object Name	Description
Billing Metric	Represents a billing metric for a product or service.
Billing Schedule	Represents a billing schedule for a product or service.
Credit Memo Line Item Tax Breakup	Represents the tax break up for a Credit Memo Line Item (which is usually based on an Order).
Credit Memo Line Item	Represents a single item in a credit memo.
Credit Memo Tax Breakup	Represents the tax break up for a Credit Memo (which is an aggregation of the Credit Memo Line Item Tax Breakups).
Credit Memo	Represents a credit memo associated with an account.
Gateway Communication	Represents request and responses during a communication with a 3rd party Gateway.
Gateway Transaction	Used to store the data from a specific payment/refund transaction (ex: authorization request or capture request and result).
Invoice Line Item Tax Breakup	Represents the tax breakup for an Invoice Line Item (which is usually based on an Order).
Invoice Line Item	Represents a single item in an invoice.
Invoice Run Result	Shows the list of invoices created by an associated invoice run.
Invoice Run	Represents an Immediate or Scheduled invoice creation batch job.
Invoice Tax Breakup	Represents the tax breakup for an Invoice (which is an aggregation of the Invoice Line Item Tax Breakups).

Object Name	Description
Invoice	Represents an invoice associated with an account.
Payment Method	Stores the information needed to process a payment.
Payment	Represents a payment.
Refund	Represents Refunds given to the customer.
Related A/Rtransaction 2	
Related A/R Transaction	Represents a single AR transaction related to the Invoice, Credit Memo, Payment, Refund etc..
Temporal Input	Object used to contain input fields required by pages that need to leverage *apex*:inputfield when gathering input for user driven actions (like creating Invoices)
Usage Input	The usage input for a product or service.
Usage Schedule	Represents a usage schedule for a product or service.

Email Invoice Attachment Callback Class

This callback interface allows you to deliver Invoice emails along with attachments to a custom set of contacts. This interface allows the caller to deliver Invoice emails to contacts other than the ones specified at the Account or Account Location level.

To add the custom class

1. Go to Setup > App Setup > Develop > Apex Classes.
2. Click New.
3. Enter the sample callback class code.
4. Click Save.

> This is just a sample callback class. You may change the code per your requirements.

This sample callback interface comprises the following object:

EmailInvoiceAttachmentInput		
Field	**Type**	**Description**
commaSeparatedEmailAddresses	String	Email Address of the Contact
List<Contacts>	List	List of Contacts to send the Invoice Attachment
Invoice__c invoice	Apttus Object	The invoice to which the attachment is related
invoiceEmailTemplateId	Id	Idof the invoice Email Template to apply

ⓘ In order to implement the interface, prefix it with Apttus_Billing.CustomClass.<Interface Name>

The sample callback class enables the caller to customize the email delivery for a given invoice attachment. By default, Apttus Billing Management uses the contacts corresponding to the Account Location or the Account. If an Invoice has an Account Location, which has Invoice Separately? selected, the contracts are retrieved from the Account Location. Else, contacts are retrieved from the Account.
With this interface, you can override the contacts retrieved from the Account Location or Account.

ⓘ Only the latest created attachment is sent in the email.

```
/**
 * Information used to customize the email delivery for a given
 * Invoice attachment.  This will allow each Invoice attachment
 * to be delivered to a custom set of "contacts" with a custom
 * Email Template Id.
 */
global class EmailInvoiceAttachmentInput
{
        // The Invoice the Invoice attachment pertains to.
        global Invoice__c invoice { get; private set; }

        // The list of "contacts" to the send the Invoice attachment to. By default
Billing
        // will try to get the contacts that correspond to affiliate Account Location.
If an
        // Account Location does not exist or it is not applicable, Billing will fetch
the
        // "contacts" of the Bill-to Account.
        //
        // If relevant, overwrite this list with a list of "contacts" based on
        // custom business logic (like retrieve Contacts from corresponding Proposal).
        global List<Contact> contacts { get; set; }

        // The optional list of comma separated Email Addresses.
        //
        // If relevant, overwrite this list with a list of comma separated Email
Addresses
        // based on  custom business logic.
         global String commaSeparatedEmailAddresses;

        // The Id of the Invoice Email Template to apply. By default Billing will
        // try to use the Invoice Email Template of the affiliate Account Location.
        // If the Account Location does not exist or is not applicable, Billing will get
        // the Invoice Email template Id of the affiliate Bill-To Account.
        //
        // If relevant, overwrite this Id with the Template Id according to the custom
        // business logic.
        global ID invoiceEmailTemplateId { get; set; }
        global string defaultEmailBody {get; set; }
}

/** Implement this interface in order to customize the list of contacts and template Id
  * to "apply" when emailing attachments for Invoices and Credit Memos.
  */
global interface IUpdateEmailAttachmentInputs
{
        /**
         * Evaluate the list of Email Invoice Attachment objects and modify
         * (in place) if the list of "contacts" and/or the Email Template Id
         * need to be altered for a given Invoice.
```

```
         *
         * If the list of "Contacts" is emptied or set to "null" for a given Invoice
         * then no Email (and attachment) will be sent for that Invoice.
         *
         * @param deliveryInputs The list of "inputs" used to customize Email
         *  delivery for a set Invoice attachments.
         *
         * @return The list of "delivery inputs" that was passed in and customized
         *  accordingly.
         *
         * For a given Email Invoice Attachment Input object, Billing will "consume"
         * the object as follows.
         *
         * If (invoiceEmailTemplateId  != null) Then
         *          Use Contact list in conjunction with the Email Template
         * Else
         * If (Comma Separated List of Email Addresses IS NOT Null or Empty)
         *          Use Comma Separated List of Email Addresses with default Email body
         * Else If (Contact List IS NOT Null or Empty)
         *         Get Email Addresses from Contact List and use with default Email body
         */
        List<EmailInvoiceAttachmentInput> updateEmailAttachmentInputs(
                final List<EmailInvoiceAttachmentInput> deliveryInputs);
}
public class TestEmailAttachmentInputCallback implements
 Apttus_Billing.CustomClass.IUpdateEmailAttachmentInputs
{
   /**
         * Implement interface method to set
    * different attributes of email delivery inputs to verify different scenarios
    * for Invoice
     */
    public

List<CustomClass.EmailInvoiceAttachmentInput>updateEmailAttachmentInputs(List<CustomClas
s.EmailInvoiceAttachmentInput>deliveryInputs){
       for(CustomClass.EmailInvoiceAttachmentInput emailInputObj : deliveryInputs){
           //Customization for EmailInputs
        }

        return deliveryInputs;
}
```

If you are using the **Send Invoice Email** button from the Invoice Detail page, it is mandatory for the invoice to have an attachment.

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Billing Custom Classes.** Click **Edit** for **System Properties** and enter the name of your custom callback class in *Email Invoice Attachment Callback Class.*

Email Credit Memo Attachment Callback Class

This is a callback interface allows you to deliver credit memo emails along with attachments to a custom set of contacts. This interface allows the caller to deliver credit memo emails to contacts other than the ones specified at the Account or Account Location level.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes.**
2. Click **New.**
3. Enter the sample callback class code.
4. Click **Save.**

ⓘ This is just a sample callback class. You may change the code as per your requirements.

This sample callback interface comprises the following object:

EmailCreditMemoAttachmentInput		
Field	**Type**	**Description**
commaSeparatedEmailAddresses	String	Email Address of the Contact
List<Contacts>	List	List of Contacts to send the Credit Memo Attachment
CreditMemo__ccreditmemo	Apttus Object	The credit memo to which the attachment is related
creditMemoEmailTemplateId	Id	Idof the credit memo email Template to apply

⚠ In order to implement the interface, prefix it with Apttus_Billing.CustomClass.<Interface Name>

The sample callback class enables the caller to customize the email delivery for a given credit memo attachment. By default, Apttus Billing Management uses the contacts corresponding to the Account Location or the Account. If a credit memo has an Account Location, which has **Invoice Separately?** selected, the contracts are retrieved from the Account Location. Else, contacts are retrieved from the Account.
With this interface, you can override the contacts retrieved from the Account Location or Account.

Only the latest created attachment is sent in the email.

```
/**
 * Information used to customize the email delivery for a given
 * Credit Memo attachment.  This will allow each Credit Memo  attachment
 * to be delivered to a custom set of "contacts" with a custom
 * Email Template Id.
 */
global class EmailCreditMemoAttachmentInput
{
        // The Credit Memo the attachment pertains to.
        global CreditMemo__c creditMemo{ get; private set; }

        // The list of "contacts" to the send the Credit Memo attachment to. By default
Billing
        // will try to get the contacts that correspond to affiliate Account Location.
If an
        // Account Location does not exist or it is not applicable, Billing will fetch
the
        // "contacts" of the Bill-to Account.
        //
        // If relevant, overwrite this list with a list of "contacts" based on
        // custom business logic (like retrieve Contacts from corresponding Proposal).
        global List<Contact> contacts { get; set; }

        // The optional list of comma separated Email Addresses.
        //
        // If relevant, overwrite this list with a list of comma separated Email
Addresses
        // based on  custom business logic.
         global String commaSeparatedEmailAddresses;

        // The Id of the Credit Memo Email Template to apply. By default Billing will
        // try to use the Invoice Email Template of the affiliate Account Location.
        // If the Account Location does not exist or is not applicable, Billing will get
        // the Invoice Email template Id of the affiliate Bill-To Account.
        //
        // If relevant, overwrite this Id with the Template Id according to the custom
        // business logic.
        global ID creditMemoEmailTemplateId { get; set; }
}

/** Implement this interface in order to customize the list of contacts and template Id
  * to "apply" when emailing attachments for Invoices and Credit Memos.
  */
global interface IUpdateEmailAttachmentInputs
{
        /**
         * Evaluate the list of Email Credit Memo Attachment objects and modify
         * (in place) if the list of "contacts" and/or the Email Template Id
         * need to be altered for a given Invoice.
         *
```

```
         * If the list of "Contacts" is emptied or set to "null" for a given Invoice
         * then no Email (and attachment) will be sent for that Invoice.
         *
         * @param deliveryInputs The list of "inputs" used to customize Email
         *  delivery for a set Credit Memo attachments.
         *
         * @return The list of "delivery inputs" that was passed in and customized
         *  accordingly.
         *
         * For a given Email Credit Memo Attachment Input object, Billing will "consume"
         * the object as follows.
         *
         * If (creditMemoEmailTemplateId  != null) Then
         *          Use Contact list in conjunction with the Email Template
         * Else
         * If (Comma Separated List of Email Addresses IS NOT Null or Empty)
         *          Use Comma Separated List of Email Addresses with default Email body
         * Else If (Contact List IS NOT Null or Empty)
         *         Get Email Addresses from Contact List and use with default Email body
         */
        List<EmailCreditMemoAttachmentInput> updateEmailAttachmentInputs(
                final List<EmailCreditMemoAttachmentInput> deliveryInputs);
}
public class TestEmailAttachmentInputCallback implements
 Apttus_Billing.CustomClass.IUpdateEmailAttachmentInputs
{
   /**
         * Implement interface method to set
    * different attributes of email delivery inputs to verify different scenarios
    * for Credit Memos
     */
    public

List<CustomClass.EmailCreditMemoAttachmentInput>updateEmailAttachmentInputs(List<CustomC
lass.EmailCreditMemoAttachmentInput>deliveryInputs){
       for(CustomClass.EmailCreditMemoAttachmentInput emailInputObj : deliveryInputs){
           //Customization for EmailInputs
        }

        return deliveryInputs;
}
```

If you are using the **Send Email** button from the Credit Memo detail page, it is mandatory for the credit memo to have an attachment.

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Billing Custom Classes.** Click **Edit** for **System Properties** and enter the name of your custom callback class in *Email Credit Memo Attachment Callback Class*.

Asset Line Item Filter Callback Class

Implement the Asset Line Item Filter Callback interface if you are enabling Wallet Manager Flow. This callback interface filters all the regular asset line items and displays only the active wallet asset line items on the Installed Products pages. All wallets with asset end date older than the system date are considered as expired and are filtered out from the Installed Products page.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.
2. Click **New**.
3. Enter the sample callback class code.
4. Click **Save.**

ⓘ This is just a sample callback class. You may change the code per your requirements.

⚠ In order to implement the interface, prefix it with Apttus_Billing.CustomClass.<Interface Name>

```
global interface IAssetLineItemCallback2{
        **
     * Method to initialize variables and thereby start the callback.
     **/

 global void start(Apttus_Config2.ProductConfiguration configuration, String
assetSearchFilter, List<String> assetSearchScope) {

    }
         /**
     * Method to get query filter.
     **/
    global String getQueryFilter(ID accountId) {

    }
         /**
     * Method to get string expression to put along with where clause to filter asset
line items.
     **/
    global String getFilterExpr(Apttus_Config2.CustomClass.ActionParams params) {


    }

    /**
     * Methog to get list of scopes to search asset.
     **/
    global List<String> getAssetSearchScope() {

    }

    /**
     * Method to perform finalization before ending callback execution.
     **/
    global void finish() {

    }

}
global with sharing class AssetLineItemFilterCallback implements
 Apttus_Config2.CustomClass.IAssetLineItemCallback2 {

    // Apttus_Config2__ProductType__c
    public static final String PRODUCT_TYPE_WALLET = 'Wallet';

    // System properties
    public static final String PROP_SYSTEM_PROPERTIES = 'System Properties';
```

```
    //WalletFlow Field Value from Billing System Properties
    private String walletFlowCustomFieldValues;

    /**
     * Method to initialize variables and thereby start the callback.
     **/
    global void start(Apttus_Config2.ProductConfiguration configuration, String
assetSearchFilter, List<String> assetSearchScope) {

        Apttus_Billing__BillingSystemProperties__c prop =
Apttus_Billing__BillingSystemProperties__c.getInstance(PROP_SYSTEM_PROPERTIES);
        walletFlowCustomFieldValues = prop.WalletFlow__c;
    }

    /**
     * Method to get query filter.
     **/
    global String getQueryFilter(ID accountId) {
        return '';
    }

    /**
     * Method to get string expression to put along with where clause to filter asset
line items.
     **/
    global String getFilterExpr(Apttus_Config2.CustomClass.ActionParams params) {

        String conditionExpression = '';

        if(params.Flow == walletFlowCustomFieldValues) {
            conditionExpression = 'AccountId__c = \'' + params.AccountId + '\' ';
            conditionExpression += ' AND ProductType__c = \'' + PRODUCT_TYPE_WALLET +
'\'';
            conditionExpression += ' AND EndDate__c >= Today';
        }else{
            conditionExpression = 'AccountId__c = \'' + params.AccountId + '\' ';
            conditionExpression += ' AND ProductType__c != \'' + PRODUCT_TYPE_WALLET +
'\'';
        }
        return conditionExpression;
    }

    /**
     * Methog to get list of scopes to search asset.
     **/
    global List<String> getAssetSearchScope() {
        return new List<String>();
    }

    /**
     * Method to perform finalization before ending callback execution.
```

```
     **/
    global void finish() {

    }

}
```

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Config Custom Classes.** Click **Edit** for **System Properties** and enter the name of your custom callback class in *Asset Line Item Callback Class.*

Asset Line Item Validation Callback Class

This callback interface allows you to validate all the asset-based amendments performed for a wallet asset on the cart page.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.
2. Click **New**.
3. Enter the sample callback class code.
4. Click **Save.**

> ⚠ In order to implement the interface, prefix it with Apttus_Billing.CustomClass.<Interface Name>

The sample callback class performs the following validations to prevent the user from removing funds more than the available balance:

- Discount % should be between 1- 100 %
- Discount amount cannot be more than 100%
- Discount amount cannot be more than the available amount
- Price Override adjustment amount cannot be more than the available balance
- Base Price Override adjustment amount cannot be more than the available balance

> ⓘ This is just a sample callback class. You may change the code per your requirements.

```
global interface IValidationCallback2 {
        global Apttus_Config2.CustomClass.ValidationResult
validateCart(Apttus_Config2.ProductConfiguration configuration) {
        }
        global Apttus_Config2.CustomClass.ValidationResult
validateAssetItems(Apttus_Config2.ProductConfiguration cart,
List<Apttus_Config2__TempRenew__c> assetItems) {

        }

    global Apttus_Config2.CustomClass.ValidationResult
validateRampLineItems(Apttus_Config2.ProductConfiguration cart,
List<Apttus_Config2.LineItem> rampLineItems) {

    }

    private static List<Apttus_Config2__LineItem__C>
getLineItems(List<Apttus_Config2.LineItem> allLines) {

    }
}
global with sharing class AssetlineItemValidationCallBack implements
 Apttus_Config2.CustomClass.IValidationCallback2 {

    // system properties
    public static final String PROP_SYSTEM_PROPERTIES = 'System Properties';

    //Error Message display on cart
    public static final String DISCOUNT_VALIDATION_ERROR_MESSAGE = 'You cannot remove
funds more than the available balance. Please enter a discount percentage between 1 and
100';
    public static final String AMOUNT_VALIDATION_ERROR_MESSAGE = 'You cannot remove
funds more than the available balance. Please enter an amount less than (or) equal to
the available balance';


    global Apttus_Config2.CustomClass.ValidationResult
validateCart(Apttus_Config2.ProductConfiguration configuration) {
        Apttus_Config2.CustomClass.ValidationResult result = new
 Apttus_Config2.CustomClass.ValidationResult(true);

        Boolean flag = false;
        String errorMessage;

        //WalletFlow Field Value from Billing System Properties
        String walletFlowname;

        try {
```

```
            Apttus_Billing__BillingSystemProperties__c prop =
Apttus_Billing__BillingSystemProperties__c.getInstance(PROP_SYSTEM_PROPERTIES);
            walletFlowname = prop.WalletFlow__c;

            List<Apttus_Config2__TempObject__c> tempData = [SELECT
Id,Apttus_Config2__Data__c
                                                            FROM
Apttus_Config2__TempObject__c
                                                            WHERE
Apttus_Config2__ConfigurationId__c = :configuration.getConfigSO().Id
                                                            LIMIT 1];
            if(tempData != null && !tempData.isEmpty()) {
                String xml = tempData[0].Apttus_Config2__Data__c;
                Dom.Document doc = new Dom.Document();
                doc.load(xml);
                Dom.XMLNode xroot = doc.getRootElement();
                List<Dom.XmlNode> param = new List<Dom.XmlNode>();
                param = xroot.getChildElement('Params', null).getChildElements();
                for(Dom.XmlNode child : param) {
                    if(child.getChildElement('Value',null).getText() == walletFlowname)
{
                        flag=true;
                        break;
                    }
                }
            }

            if(flag == false) {
                return result;
            }

            if(flag == true) {
                List<Apttus_Config2.LineItem> allLines = configuration.getLineItems();
                if (allLines.size() == 0) {
                    return result;
                }

                list<Apttus_Config2__LineItem__c> lineItemList = getLineItems(allLines);

                for(Apttus_Config2__LineItem__c lineItem : lineItemList) {

                    if((lineItem.Apttus_Config2__AdjustmentType__c == '% Discount' ||
lineItem.Apttus_Config2__AdjustmentType__c == '% Discount Off List')
                        && lineItem.Apttus_Config2__AdjustmentAmount__c > 100) {
                        errorMessage = DISCOUNT_VALIDATION_ERROR_MESSAGE;
                    }
```

```
                    if((lineItem.Apttus_Config2__AdjustmentType__c == 'Discount Amount'
 || lineItem.Apttus_Config2__AdjustmentType__c == 'Price Override' ||
lineItem.Apttus_Config2__AdjustmentType__c == 'Base Price Override')
                        && lineItem.Apttus_Config2__AdjustmentAmount__c >
lineItem.AvailableBalance__c) {
                        errorMessage = AMOUNT_VALIDATION_ERROR_MESSAGE;
                    }
                }

                if(!String.isEmpty(errorMessage)) {
                    result.isSuccess = false;
                    result.Messages.add(new apexpages.Message(Apexpages.Severity.Error,
'' + errorMessage));
                }
            }
        }
        catch (Exception ex) {
            errorMessage = 'Validation callback Exception: ' + ex.getMessage();
            System.debug('Validation callback Exception: ' + ex.getMessage());
            result.isSuccess = false;
            result.Messages.add(new apexpages.Message(Apexpages.Severity.Error, '' +
errorMessage));
        }

        return result;
    }

    global Apttus_Config2.CustomClass.ValidationResult
validateAssetItems(Apttus_Config2.ProductConfiguration cart,
List<Apttus_Config2__TempRenew__c> assetItems) {
        Apttus_Config2.CustomClass.ValidationResult result;
        return result;
    }

    global Apttus_Config2.CustomClass.ValidationResult
validateRampLineItems(Apttus_Config2.ProductConfiguration cart,
List<Apttus_Config2.LineItem> rampLineItems) {
        Apttus_Config2.CustomClass.ValidationResult result;
        return result;
    }

    private static List<Apttus_Config2__LineItem__C>
getLineItems(List<Apttus_Config2.LineItem> allLines) {
        List<Apttus_Config2__LineItem__C> lineItems = new
 List<Apttus_Config2__LineItem__C>();
        for (Apttus_Config2.LineItem lineItemMO : allLines) {
            lineItems.add(lineItemMO.getLineItemSO());
        }
        return lineItems;
    }
}
```

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Config Custom Classes.** Click **Edit** for **System Properties** and enter the name of your custom callback class in V*alidation Callback Class.*

Invoice Filter Callback Class

Apart from the three filters, Account, Account Location, and Product, you can create Custom Filters to sort and process Invoices.

For example, you are a Billing executive in a retail chain and you want to generate only those invoices for a custom object that is local to your organization and is not a standard object in Apttus CPQ or Apttus Billing Management. You can define Custom Settings to call the custom Object in your implementation of Apttus Billing Management.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.
2. Click **New**.
3. Enter the sample callback class code.
4. Click **Save.**

> This is just a sample callback class. You may change the code per your requirements.

> In order to implement the interface, prefix it with Apttus_Billing.CustomClass.<Interface Name>

```
 /*    Apttus Billing
* TestInvoiceFilterCallback
 */
global class TestInvoiceFilterCallback implements CustomClass.IInvoiceFilterCallback {
global String getBillingScheduleFilterExpression(CustomClass.Parameters params) {
String filter = 'AssetLineItemId__r.Apttus_Config2__BusinessLineItemId__c = \'' +
params.invoiceRunSO.Category__c + '\''; return filter;
        }
}
```

.

When you define a value for the Invoice Filter Callback Class, the Invoice Run User interface is updated with the Custom Options section that has 2 fields, **Category**, and **Sub Category**.

Select the values with the from each field with the picklist you have defined and the resulting Invoice Run is processed based on the fields you define here. For example, if a customer wants to only apply the **Invoice Run** to one or more Business Partner countries (a custom field added to the Asset Line Item) then the Category pick list should be updated to contain all countries applicable to their application: Germany, France, Italy, and others. Based on the country selected in Category, selected Billing Schedules will be picked by Invoice Run. If further granular filtering is needed then you can leverage the Sub Category pick list.

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Billing Custom Classes**. Click **Edit** for **System Properties** and enter the name of your custom callback class in *Invoice Filter Callback Class.*

Invoice Grouping Callback Class

Apttus Billing Management, by default, consolidates assets onto a single invoice. Therefore, if you add products or make any amendments over time, those will be included a single invoice.

However, you have an option to generate separate invoices for each of the Billing Schedules.

Assume that for an asset, Ready for Invoice Date for 4 Billing schedules is 3/20/2016, 4/20/216, 5/20/2016 and 6/20/216 respectively. Create a new **Invoice Run** and set **Process Through Date** as 6/30/2016; only 1 invoice will be generated.

You can use Invoice Grouping Callback Class to create 4 invoices.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.
2. Click **New**.
3. Enter the sample callback class code.
4. Click **Save.**

> ⚠ This is just a sample callback class. You may change the code per your requirements.

```
global class TestInvoiceGroupingCallback
      implements CustomClass.IInvoiceGroupingCallback {

      global String getKeyComponentForInvoiceGrouping(CustomClass.Parameters params) {

            String key =
string.valueOf(params.billingScheduleSO.ReadyForInvoiceDate__c);
            return key;
      }
}
```

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Billing Custom Classes.** Click **Edit** for **System Properties** and enter the name of your custom callback class in *Invoice Grouping Callback Class.*

> ⚠ If **Billing Rule** is set to **Ready For Billing Date**, only one invoice is generated irrespective of this configuration.

Custom Numbering Callback Class

You can stamp custom number on the invoices by adding a callback class for Invoice Numbering.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.
2. Click **New**.
3. Enter the sample callback class code.
4. Click **Save.**

> ⓘ This is just a sample callback class. You may change the code per your requirements.

> ⚠ In order to implement the interface, prefix it with Apttus_Billing.CustomClass.<Interface Name>

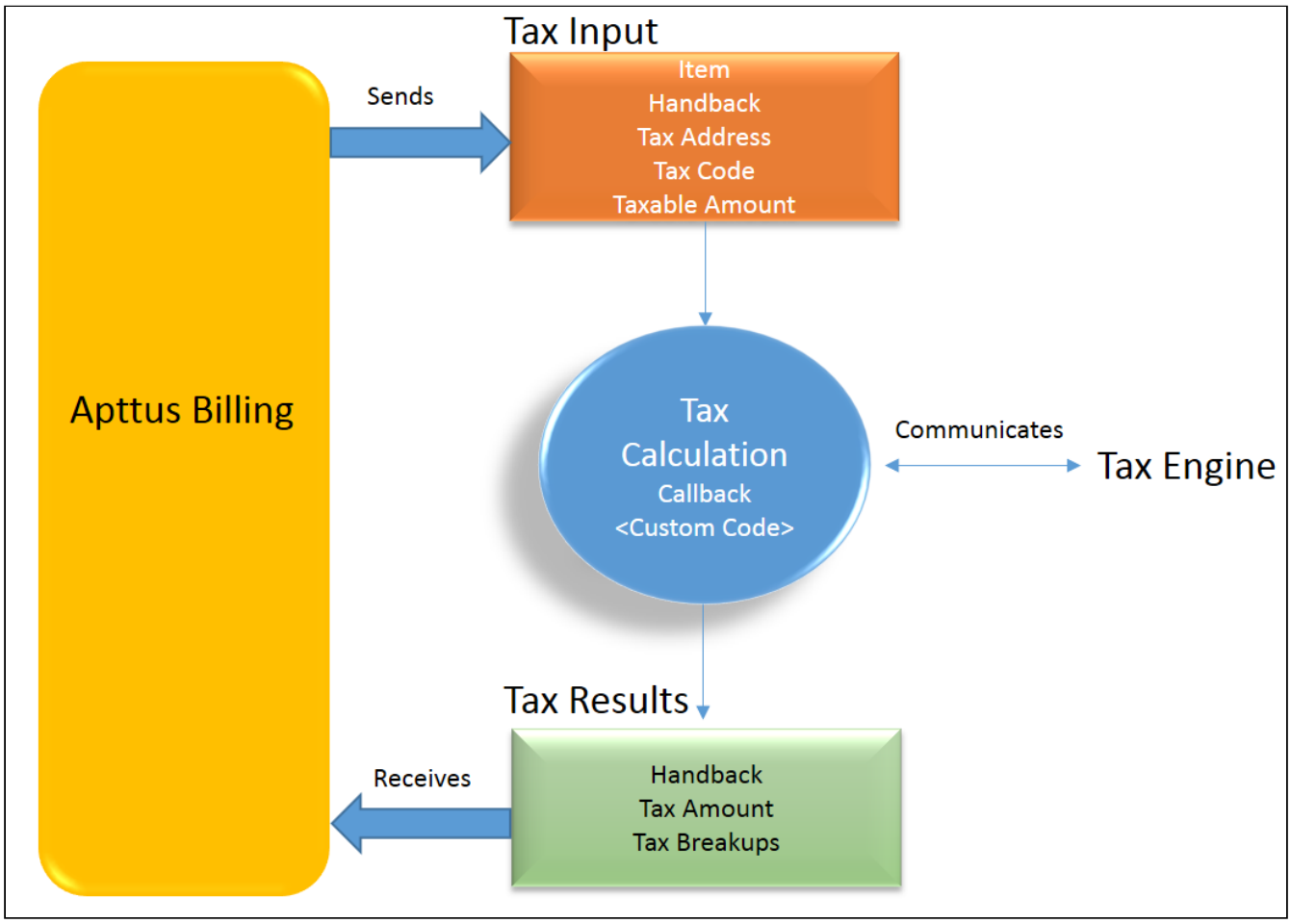
```
/**
*  Apttus Billing
*  DefaultCustomNumberingCallback
*
 *  @2011-2016 Apttus Inc. All rights reserved.
*/
public with sharing class DefaultCustomNumberingCallback
      implements CustomClass.IFacilitateCustomTransactionNumbers {

      /**
      * Evaluate the "object type" of the specified numbering configuration settings and
determine
      * if the settings are valid for your custom implementation.  For example, the
default  Apttus
      * implementation of this interface for the "Invoice" object type expects that only
"field 1" has
      * been checked.  This is critical because the user interface for the custom number
entries
      * uses the numbering configuration setting object to determine which fields to
force the
      * user to enter.  Any field unchecked will be hidden from the user.
      *
      * @param customNumberingConfigSetting The custom number configuration setting to
evaluate.
      *
       * @return true if the "custom numbering configuration settings" is valid for its
underlying "object type",
      *   otherwise return false.
      */
      public boolean isValidConfiguration(CustomNumberingConfigurationSetting__c
numberingConfigSettings) {

            if

(numberingConfigSettings.ObjectType__c.equals(CustomClass.CUSTOM_NUMBERING_OBJECT_TYPE_I
NVOICE)) {

                  if (true == numberingConfigSettings.FieldOneEnabled__c &&
                        false == numberingConfigSettings.FieldtwoEnabled__c &&
                        false == numberingConfigSettings.FieldThreeEnabled__c &&
                        false == numberingConfigSettings.FieldFourEnabled__c &&
                        false == numberingConfigSettings.FieldFiveEnabled__c) {

                        return true;
                  }
            }

            return false;
      }
```

```
    /**
    * Build the list of "lookup key" field lists for the each of the
    * specified Invoices.
    *
    * @param invoices The list of Invoices to evaluate and to find the
    *  corresponding list of lookup key field lists for.
    * @param customNumberingConfigSetting The custom number configuration
    *  setting for Invoices.
    *
    * @return the list of lists.  Each element of the returned list must
    *  contain the "lookup key" field list for the corresponding Invoice.
    *  The number of lists returned must match the number of Invoices passed in
    *  and the ith list must correspond to the ith Invoice.  For example, if
    *  2 Invoices are passed in, then 2 "field lists" should be returned and
    *  the 1st field list must correspond to the 1st Invoice and the 2nd
    *  field list must correspond to the 2nd Invoice.
    */
    public List<List<String>> fetchInvoiceCustomNumberingFieldLists(
          List<Invoice__c> invoices,
          CustomNumberingConfigurationSetting__c customNumberingConfigSetting)
    {

          List<List<String>> fieldLists = new List<List<String>>();
          for (Invoice__c inv : invoices) {

                List<String> fieldList = new List<String>();
                fieldList.add(inv.BillingCountry__c);
                fieldLists.add(fieldList);
          }
          return fieldLists;
    }

}
```

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Billing Custom Classes.** Click **Edit** for **System Properties** and enter the name of your custom callback class in *Custom Invoice Numbering Callback Class.*

Tax Callback Class

You can communicate with a Tax Engine of your choice with the help of a Tax Callback class. For this, you should know the Request Fields that are sent to Callback and further to the Tax Engine. Following diagram gives an overview of how information is sent from Product to the Tax Engine and received back after processing.



Tax input is a container which holds the following fields:

- **Item:** Contains an Invoice Line Item or a Credit Memo Line Item
- **Handback**: A generic wrapper class that can be used to pass an additional field value. Set the value for this field to *TaxInputRelatedObjects*. This class will contain the parent Invoice or Credit Memo based on whether the item contains an Invoice Line Item or a Credit Memo Line Item.
- **Tax Address**: The address specified as the Shipping Address of the Ship To account. If there is no Shipping Address mentioned in the Ship To account, then the Billing Address of the Ship To account is used.

  > **Note**
  > If both the address of the Ship To account are missing, you need to cancel the invoice. Recreate the invoice after setting the proper address in the Ship To account.

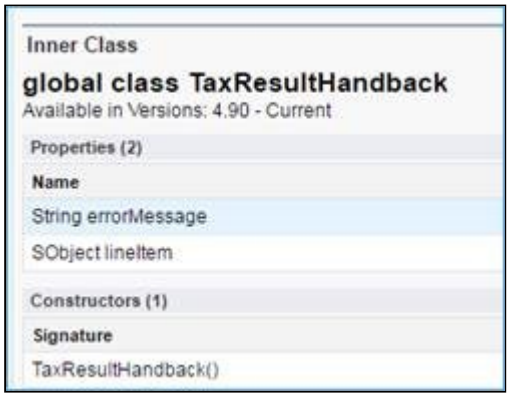
- **Tax Code** : This value is taken from the product PLI.
- **Taxable Amount**: This is the amount to apply the tax to.

You can use *Invoice Number* as a *Document Id* when calling a Tax Engine. After processing, the Tax Results are sent to Apttus Billing. *Handback* field holds the line item and error message (if any).

> **Note**
> If there is any error during the tax calculation because of which the Invoice is not approved, check out **Error Status** and **Error Message** fields on the Invoice to find out the reason behind the failure.

When communicating with a Tax Engine, you must note that:

- The implementation of the Tax Callback must return a *Apttus_Billing.CustomClass.TaxResultHandback* object in the *Handback* field of a Tax Result.
- The implementation of the Tax Callback must determine *commit mode* by checking if the status of the Invoice is *Approved Pending.*


Inner Class

global class TaxResultHandback

Available in Versions: 4.90 - Current

Properties (2)

Name
String errorMessage
SObject lineItem

Constructors (1)

Signature
TaxResultHandback()

You must register a **Tax Callback class** which is called for tax calculation on Invoice generation from Custom Settings.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.
2. Click **New**.
3. Enter the sample callback class code.
4. Click **Save.**

> This is just a sample callback class. You may change the code per your requirements.

```
/**
*  Apttus Billing
*  TestTaxCallback
*
 *  @2013-2018 Apttus Inc. All rights reserved.
*/
global with sharing class TestTaxCallback
      implements Apttus_Config2.CustomClass.ITaxCallback,
Apttus_Config2.CustomClass.ITaxCallback2 {

      public static final Decimal TAX_RATE_DEFAULT = 0.10;
      public static final Decimal TAX_RATE_CA = 0.0875;
      public static final Decimal TAX_RATE_NV = 0.0;
      public static final Decimal TAX_RATE_NY = 0.0925;
      public static final Decimal TAX_RATE_TX = 0.0;
      public static final Decimal TAX_RATE_WA = 0.075;

      //Note :: If we Introduce any new state in this callback then we need to update
      //            the "AbstractInvoiceSupportTest" class as well.
      //            We are validating the Line Item Tax Breakups so for new State values
we
      //            need to add logic and assert values in
"validateInvoiceLineItemTaxBreakups" method.
      public static final String STATE_CA = 'CA';
      public static final String STATE_NV = 'NV';
      public static final String STATE_NY = 'NY';
      public static final String STATE_TX = 'TX';
      public static final String STATE_WA = 'WA';

      public static final String STATE_FL = 'FL';
      public static final String STATE_OK = 'OK';

      public static final Decimal OK_COUNTY_PERCENTAGE = 0.70;
      public static final Decimal OK_DISTRICT_PERCENTAGE = 0.30;

      public static final Decimal FL_CITY_PERCENTAGE = 0.10;
      public static final Decimal FL_COUNTY_PERCENTAGE = 0.20;
      public static final Decimal FL_DISTRICT_PERCENTAGE = 0.20;
      public static final Decimal FL_STATE_PERCENTAGE = 0.50;

      private static final Map<String, Decimal> stateTaxRateMap =
            new Map<String, Decimal> {
                  STATE_CA => TAX_RATE_CA,
                  STATE_NV => TAX_RATE_NV,
                  STATE_NY => TAX_RATE_NY,
                  STATE_TX => TAX_RATE_TX,
                  STATE_WA => TAX_RATE_WA
            };

      /**
```

```
      * Callback invoked to compute tax based on the given input
      * @param input the tax input
       * @return the tax result
      */
      global Apttus_Config2.CustomClass.TaxResult
computeTax(Apttus_Config2.CustomClass.TaxInput input) {

            List<Apttus_Config2.CustomClass.TaxInput> inputs = new
 List<Apttus_Config2.CustomClass.TaxInput>{input};
            List<Apttus_Config2.CustomClass.TaxResult> results =
computeTaxMultiple(inputs);
            return (null == results || 1 > results.size()) ? null : results[0];
      }

      /**
      * Callback invoked to compute tax based on the given list of inputs
      * @param inputs the list of tax inputs
      * @return the list of tax results
      */
      global List<Apttus_Config2.CustomClass.TaxResult> computeTaxMultiple(
            List<Apttus_Config2.CustomClass.TaxInput> inputs) {
        //System.debug('From compute Multiple Tax>>>');
            // Create list of Tax Results
            List<Apttus_Config2.CustomClass.TaxResult> results = new
 List<Apttus_Config2.CustomClass.TaxResult>();

            // create mock result
            for (Integer i = 0; i < inputs.size(); i++) {

                  Apttus_Config2.CustomClass.TaxInput input = inputs[i];

                  System.assertNotEquals(null, input, 'A Tax Input cannot be null!');
                  System.assertNotEquals(null, input.item, 'The Item contained in a Tax
Input cannot be null!');
                  System.assertNotEquals(null, input.item.Id, 'The Item Id contained in
a Tax Input cannot be null!');
                  // Create a Tax Result and populate
                  Decimal taxRate = getTaxRate(input);
                  Apttus_Config2.CustomClass.TaxResult result = new
 Apttus_Config2.CustomClass.TaxResult();
                  CustomClass.TaxResulthandBack taxResulthandBackObj = new
 CustomClass.TaxResulthandBack();
                  if(null != input.item) {
                        taxResulthandBackObj.lineItem = input.item;
                        taxResulthandBackObj.errorMessage = null;

                  }
                  result.Handback = taxResulthandBackObj;
```

```
                    //System.debug('From Multiple Tax>>> taxabeamount' +
input.TaxableAmount + '#taxRate' + taxRate);
                    result.TaxAmount = input.TaxableAmount * taxRate;
                    // Create Test Tax Breakups
                    addTestTaxBreakups(input, result, taxRate);
                    Decimal aggregateTaxAmount = 0.00;

                    Integer currencyDecimalPlacesToSet
=CurrencyTypeSupport.getCurrencyDecimalPlaces(input.item);

                    //Integer currencyDecimalPlacesToSet =
SystemUtil.getCurrencyDecimalPlaces();
                    for(integer
iBreakupCount=0;iBreakupCount<result.TaxBreakups.size();iBreakupCount++)
                    {

                      Apttus_Config2__TaxBreakup__c taxBreakup =
result.TaxBreakups.get(iBreakupCount);
                      aggregateTaxAmount
+=taxBreakup.Apttus_Config2__TaxAmount__c.setScale(currencyDecimalPlacesToSet);
                    }
                    //System.debug('TaxAmount aggregateTaxAmount' + aggregateTaxAmount);
                    result.TaxAmount =aggregateTaxAmount;

                    //BIL-1080 :: Check the Tax Address state, If it contains value from
(V1,V2,V3,V4) then null the expected properties from tax Result
                    String state = (null != input && null != input.TaxAddress && null !=
input.TaxAddress.State)
                    ? input.TaxAddress.State.trim().toUppercase() : null;


                    //    result.TaxAmount =
aggregateTaxAmountFromTestTaxBreakups(result);
                    // Add the Tax Result to the return list
                    results.add(result);

            }

            //System.debug('From compute Multiple Tax>>>' + results);
      return results;
    }


      /** Get a tax rate based on the "address" in the specified Tax input. If
        * there is a rate for the "state code" then return that rate, otherwise
        * return the "default rate" (10%).
        * 10%.
        *
        * @param input The Tax input to extract State Code from.
        * @return A rate that pertains to the State Code, or the "default rate"
        *  if there is there no such rate exists.
```

```
        */
      public Decimal getTaxRate(Apttus_Config2.CustomClass.TaxInput input) {

            String state = (null != input && null != input.TaxAddress && null !=
input.TaxAddress.State)
                  ? input.TaxAddress.State.trim().toUppercase() : null;
            Decimal rate = stateTaxRateMap.get(state);

            return (null == rate) ? TAX_RATE_DEFAULT : rate;
      }

      private void addTestTaxBreakups(
            Apttus_Config2.CustomClass.TaxInput input,
            Apttus_Config2.CustomClass.TaxResult result,
            Decimal taxRate) {

            String state = (null != input.TaxAddress && null != input.TaxAddress.State)
                  ? input.TaxAddress.State : '';

            //System.debug('From compute Multiple Tax State>>>' + state);

            if (STATE_OK.equals(state)) {

                  // Create/add two Tax Breakups (County and District)
                  result.TaxBreakups.add(
                        new Apttus_Config2__TaxBreakup__c(
                              Apttus_Config2__Sequence__c = 1,
                              Apttus_Config2__LineItemId__c = input.Item.Id,
                              Apttus_Config2__BreakupType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TYPE_DETAIL,
                              Apttus_Config2__TaxType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TAXTYPE_COUNTY_TAX,
                              Apttus_Config2__TaxRate__c = taxRate,
                              Apttus_Config2__TaxAppliesTo__c =
InvoiceLineItemTaxBreakup.TAX_APPLIESTO_NET_PRICE,
                              Apttus_Config2__TaxAmount__c = (result.TaxAmount *
OK_COUNTY_PERCENTAGE)
                        )
                  );

                  result.TaxBreakups.add(
                        new Apttus_Config2__TaxBreakup__c(
                              Apttus_Config2__Sequence__c = 2,
                              Apttus_Config2__LineItemId__c = input.Item.Id,
                              Apttus_Config2__BreakupType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TYPE_DETAIL,
                              Apttus_Config2__TaxType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TAXTYPE_DISTRICT_TAX,
                              Apttus_Config2__TaxRate__c = taxRate,
```

```
                              Apttus_Config2__TaxAppliesTo__c =
InvoiceLineItemTaxBreakup.TAX_APPLIESTO_NET_PRICE,
                              Apttus_Config2__TaxAmount__c = (result.TaxAmount *
OK_DISTRICT_PERCENTAGE)
                        )
                  );

            } else if (STATE_FL.equals(state)) {

                  // Create/add four Tax Breakups (City, County, District, and State)
                  result.TaxBreakups.add(
                        new Apttus_Config2__TaxBreakup__c(
                              Apttus_Config2__Sequence__c = 1,
                              Apttus_Config2__LineItemId__c = input.Item.Id,
                              Apttus_Config2__BreakupType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TYPE_DETAIL,
                              Apttus_Config2__TaxType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TAXTYPE_CITY_TAX,
                              Apttus_Config2__TaxRate__c = taxRate,
                              Apttus_Config2__TaxAppliesTo__c =
InvoiceLineItemTaxBreakup.TAX_APPLIESTO_NET_PRICE,
                              Apttus_Config2__TaxAmount__c = (result.TaxAmount *
FL_CITY_PERCENTAGE)
                        )
                  );

                  result.TaxBreakups.add(
                        new Apttus_Config2__TaxBreakup__c(
                              Apttus_Config2__Sequence__c = 2,
                              Apttus_Config2__LineItemId__c = input.Item.Id,
                              Apttus_Config2__BreakupType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TYPE_DETAIL,
                              Apttus_Config2__TaxType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TAXTYPE_COUNTY_TAX,
                              Apttus_Config2__TaxRate__c = taxRate,
                              Apttus_Config2__TaxAppliesTo__c =
InvoiceLineItemTaxBreakup.TAX_APPLIESTO_NET_PRICE,
                              Apttus_Config2__TaxAmount__c = (result.TaxAmount *
FL_COUNTY_PERCENTAGE)
                        )
                  );

                  result.TaxBreakups.add(
                        new Apttus_Config2__TaxBreakup__c(
                              Apttus_Config2__Sequence__c = 3,
                              Apttus_Config2__LineItemId__c = input.Item.Id,
                              Apttus_Config2__BreakupType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TYPE_DETAIL,
                              Apttus_Config2__TaxType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TAXTYPE_DISTRICT_TAX,
                              Apttus_Config2__TaxRate__c = taxRate,
```

```
                              Apttus_Config2__TaxAppliesTo__c =
InvoiceLineItemTaxBreakup.TAX_APPLIESTO_NET_PRICE,
                              Apttus_Config2__TaxAmount__c = (result.TaxAmount *
FL_DISTRICT_PERCENTAGE)
                        )
                  );

                  result.TaxBreakups.add(
                        new Apttus_Config2__TaxBreakup__c(
                              Apttus_Config2__Sequence__c = 4,
                              Apttus_Config2__LineItemId__c = input.Item.Id,
                              Apttus_Config2__BreakupType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TYPE_DETAIL,
                              Apttus_Config2__TaxType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TAXTYPE_STATE_TAX,
                              Apttus_Config2__TaxRate__c = taxRate,
                              Apttus_Config2__TaxAppliesTo__c =
InvoiceLineItemTaxBreakup.TAX_APPLIESTO_NET_PRICE,
                              Apttus_Config2__TaxAmount__c = (result.TaxAmount *
FL_STATE_PERCENTAGE)
                        )
                  );

            } else {

                  // Create/add one Tax Breakup
                  result.TaxBreakups.add(
                        new Apttus_Config2__TaxBreakup__c(
                              Apttus_Config2__Sequence__c = 1,
                              Apttus_Config2__LineItemId__c = input.Item.Id,
                              Apttus_Config2__BreakupType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TYPE_DETAIL,
                              Apttus_Config2__TaxType__c =
InvoiceLineItemTaxBreakup.BREAKUP_TAXTYPE_STATE_TAX,
                              Apttus_Config2__TaxRate__c = taxRate,
                              Apttus_Config2__TaxAppliesTo__c =
InvoiceLineItemTaxBreakup.TAX_APPLIESTO_NET_PRICE,
                              Apttus_Config2__TaxAmount__c = result.TaxAmount
                        )
                  );
            }
      }

}
```

Registering the Tax Callback class

From **Setup,** enter **Custom Settings** in the Quick Find box and look for **Config System Classes**. Click **Manage** to see System properties. Edit **System Properties** to add a **Tax Callback Class.**

Apttus Copyright Disclaimer

For additional resources and support, please visit https://community.apttus.com.

DOC ID: BMSFWIN18AG20181206

