Revenue Recognition On Salesforce Spring 2018 Administrator Guide

23 April 2018

Table Of Contents

You can use **Apttus Revenue Recognition** to easily manage revenue recognition schedules. Apttus Revenue Recognition is built on the Salesforce1 platform and you can generate complex revenue schedules for any combination of products, services, and subscriptions before posting final transactions on your General Ledger.

With Apttus Revenue Recognition, it's easy to take information from signed commitments and fulfilled orders to create revenue schedules that are fully compliant with accounting and corporate standards. In addition to GAAP, Apttus Revenue Recognition complies with FASB, IASB, SOP, and SEC regulations and Sarbanes-Oxley Section 404 requirements as well.

Key Benefits

With Apttus Revenue Recognition, you can

- **Increase Productivity -** eliminate manual workarounds to generate and validate revenue schedules for your orders.
- **Improve Financial Agility -** quickly model revenue schedules for new products, services, or subscriptions, and acquisitions.
- **Increase Financial Insight -** get more accurate and real-time visibility into your revenue forecasts.

Key Capabilities

With Apttus Revenue Recognition, you can

- **Integrate with Apttus Order Management -** Automatically recognize revenue as goods are delivered or services are rendered, including methods like percent complete, milestones or usage.
- **Forecast Revenue, Set Baselines, and Create Reports -** Generate revenue baselines and analyze projected revenue trends across products, regions, and divisions.
- **Revenue Consumption:** Effectively manage prepays, true-ups, and adjustments.

Overview

Revenue Recognition is the process used to align revenue with the delivery of goods and services.

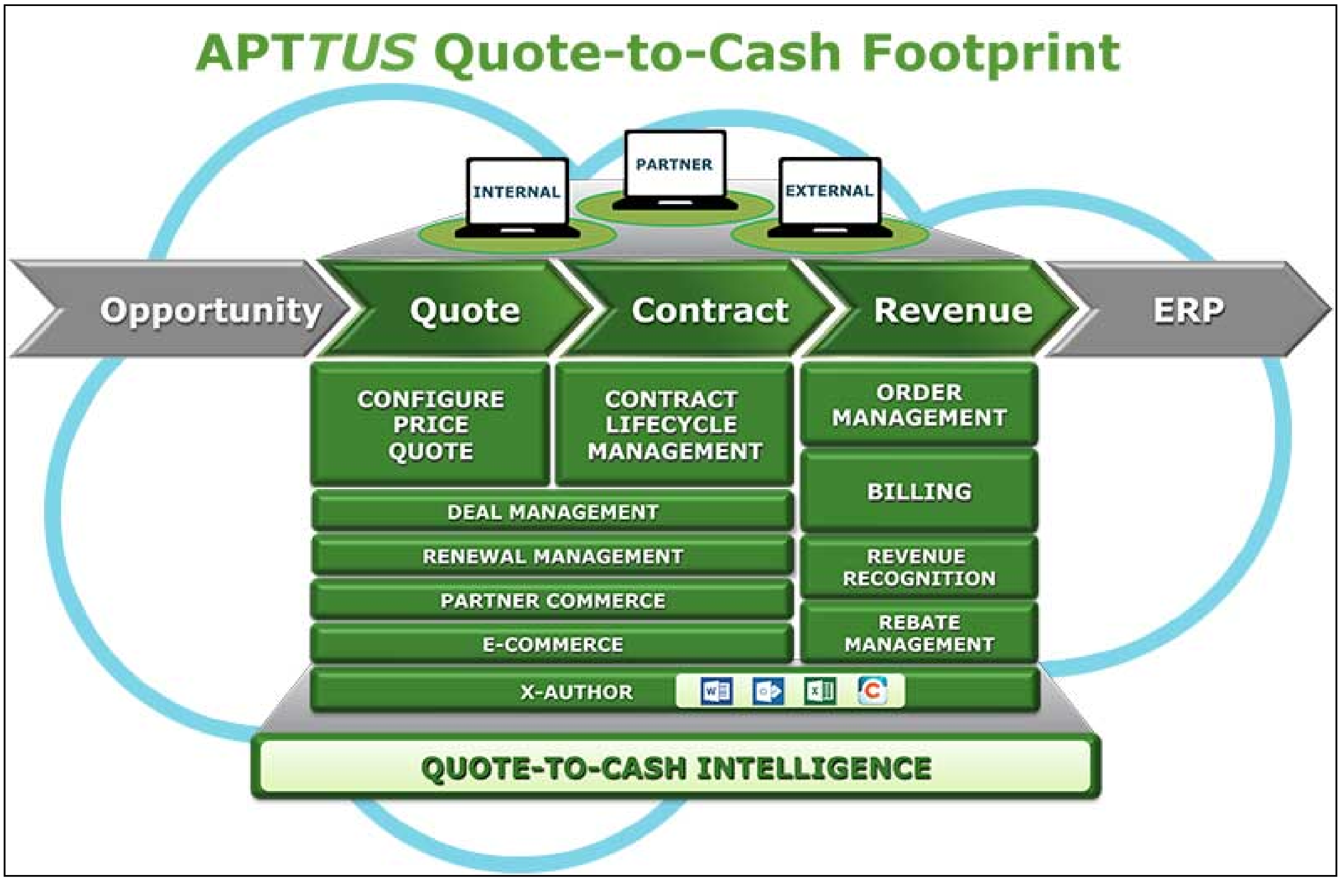
After you make a sale, you must report the revenue that results from your sales. You must consider the structure of the sale, contract terms, and the billing and fulfillment status for each deal to accurately report revenue to your investors, executives, and board members. It is vital to get this right because revenue is the most important measure of how well your company is performing.

Revenue is not the same as cash, and the two are distinctly different. Revenue is the fair market value of products or services delivered. This may differ from the invoice and payment amount for an individual service. This is especially true when sales and marketing teams offer free add-ons.

To better understand the challenges, let's look at a few transactions to see how this works in practice.

- When you buy a product for $100 at a retail store, the store recognizes $100 as revenue immediately since you received the product at the time you paid.
- When you sign up for an annual subscription to a video streaming service, the vendor will spread the revenue over 12 months since the service is delivered over the course of a year.
- When you buy a car and receive a free annual subscription for a radio service, the revenue is split between the car and the radio service. Revenue for the car is recognized immediately since you received the car at the time of purchase. Revenue for the radio is spread over 12 months as the service is delivered over time.

Complex contracts and flexible purchasing options can make managing revenue very challenging. Apttus Revenue Recognition inherits and uses information and data from other applications in the Apttus Quote-to-Cash suite. Revenue Recognition syncs seamlessly with Apttus CPQ, Apttus CLM, Apttus Billing Management, and Apttus Renewals Management. With this synchronization, you can increase your access to all the information you require, real-time, while reducing time and effort across all the teams involved.



This is why Apttus is uniquely suited to handle the Revenue Recognition process. Only Apttus with its end-to-end view of the entire Quote-to-Cash process has all of the information required without the need for messy integrations.

When using **Apttus Revenue Recognition**, you can:

- streamline Revenue recognition and Account closing processes
- gain revenue by eliminating errors, lack of detail, and duplicate entries
- decrease tracking and revenue management costs and increase your focus on analysis and process
- accelerate complex and recurring billing processes
- improve channel management, efficiency, visibility, and compliance
- view real-time reporting to make critical business decisions in a quick manner
- increase financial reporting and accuracy, all while improving customer service

Installing Revenue Recognition Application

You can install Apttus Revenue Recognition from the Apttus Customer Success Portal.

To install Revenue Recognition

You must have Apttus provided login credentials.

1. Go to https://apttusportal.force.com/community and use the credentials to log in to the portal.
2. Click **Install Center** and go to the **Revenue Recognition** tab and click **Get It Now**.
3. Choose Production or Sandbox, and click **Get It Now** on the AppExchange site.
4. Click **Login to the AppExchange** and use your Salesforce credentials to log in.
5. Select the environment to install the package and follow instructions on the page.
6. Select **I have read and agree to the terms and conditions** and click **Confirm and Install.** You may be prompted to log in using your Salesforce credentials.
7. Enter the password provided by Apttus and click Submit. Review the package items being installed. In case of an upgrade, the details of the currently installed version are also displayed.
8. Click **Continue** and click **Next.**
9. To approve package API access, click Next. These settings control the access components in this package have to standard objects through the API.
10. Select a security level and click **Next.**

    If you do not know the security settings, leave it as Grant access to admins only. You can change the security settings later.

11. To install all the components in the package, click Install.

Revenue Recognition is successfully installed in your selected environment.

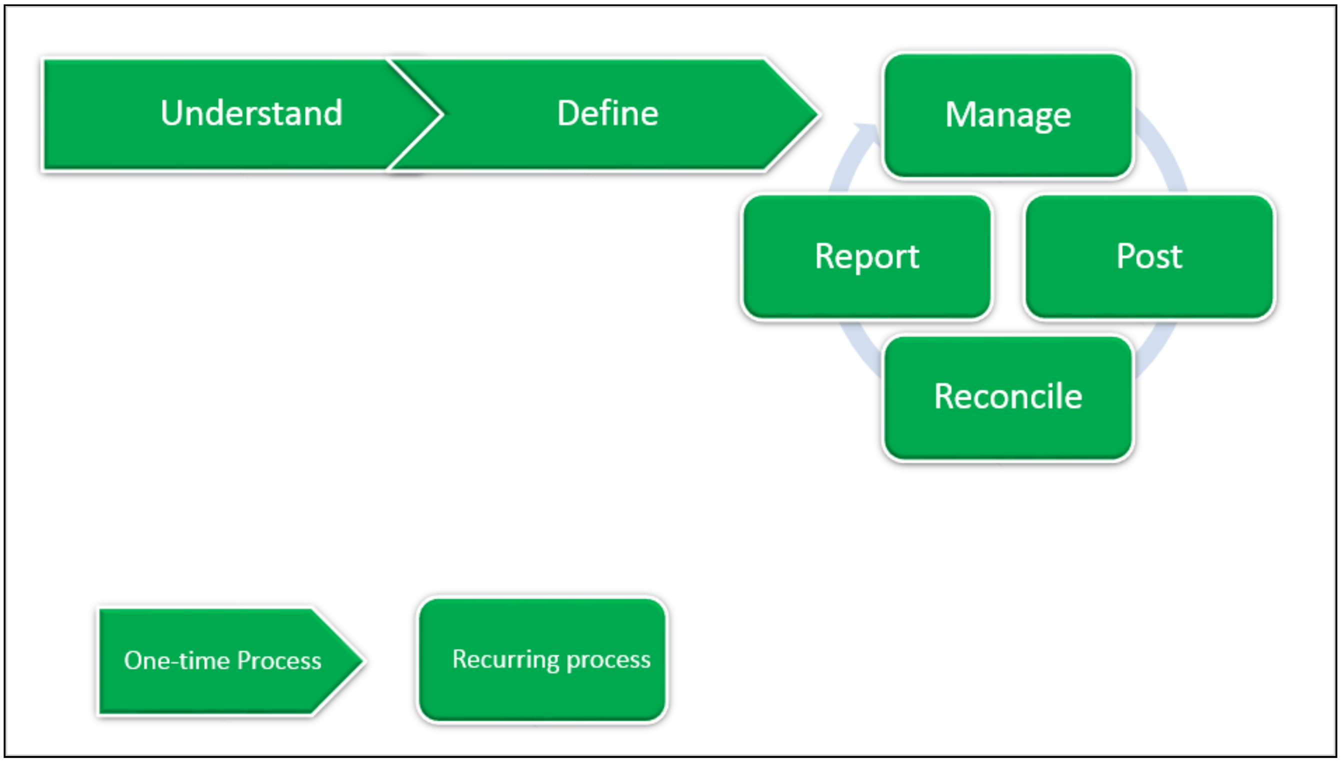
Getting Started

Apttus Revenue Recognition Life Cycle

Apttus Revenue Recognition Life cycle is the process used to manage the revenue of a deal from start to finish.

While the details of the process may vary between companies, every company follows the same major stages.

Apttus Revenue Recognition is designed around Revenue Life Cycle.



The phases of Apttus Revenue Recognition Life Cycle are:

- **Understand** - understand the essence of business transaction and determine which Revenue Recognition Policies apply to each good or service in the transaction.
- **Define** - define the specific plan including the amount of revenue to recognize and the specific accounts impacted for each good or service.
- **Manage** - manage the maintenance activities for each product or service. This includes creating adjustments to revenue, recording usage, processing up-sells or down-sells, or cancellations.
- **Post** - post the amount of revenue earned during the period to the GL during accounting close.
- **Reconcile** - reconcile your deferred revenue and revenue accounts and ensure accuracy for financial reporting.

- **Report** - report the company's financial position and plan appropriately.

Logging in to Revenue Recognition

Log in to your Salesforce.com org to access Apttus Revenue Recognition.

> **Note**
> Do not use the Back button on your browser when using Apttus Revenue Recognition.

Before you log in to Revenue Recognition, make sure you meet the following criteria:

- You have installed all of the required Revenue Recognition module packages.
- You have login credentials provided by Apttus.

To log in to Apttus Revenue Recognition

1. Go to http://www.salesforce.com.
   Or
   If your organization is using a sandbox or test environment to access Apttus Revenue Recognition (for example, if you are doing user acceptance testing), go to http://test.salesforce.com instead.
2. In the toolbar at the top of the page, click Login. The login page opens.
3. Enter your User Name and Password, and click Log in to **Salesforce**.
   You have successfully logged into Revenue Recognition.
4. Navigate to the Force.com App Menu and select Apttus Revenue Management.

Congratulations! Now you are ready to start recognizing your revenue in Apttus Revenue Management.

Key Concepts

It is essential to understand the following concepts in order to work with Apttus Revenue Recognition.

- **Revenue Agreement -** A **Revenue Agreement** is an agreement established between a customer and vendor when the vendor agrees to provide products and services and the customer agrees to receive and pay for those products and services. This agreement may be a formal document with a signed contract or it could be an informal agreement as with any subscription service provider. It is important to note that an agreement always exists; if it does not exist, no revenue exists. This agreement is represented within Apttus by a Revenue Agreement record.

- **Agreement Fee -** An **Agreement Fee** is a revenue from the goods or services provided within an agreement. As many products and services can exist under a single agreement revenue, multiple revenue fees can exist under a single revenue agreement.

- **Revenue Recognition Policy** - A **Revenue Recognition Policy** defines how to recognize revenue for an Agreement Fee. It is essential to associate the Revenue Recognition Policy with the Agreement Fee as it determines when the revenue will be recognized - revenue can be recognized at once or it can be spread over a number of periods. Each Agreement Fee may have only a single policy but many policies can be created to reflect the different ways the revenue can be recognized.

- **Revenue Schedule** - A **Revenue Schedule** is an object where the amount of revenue to be recognized in each month is stored. By applying the Revenue Recognition Policy to the Agreement Fee, a waterfall can be created to show how much revenue will be recognized in each month. Since revenue can be recognized over time, many revenue schedules can exist under a single revenue fee.

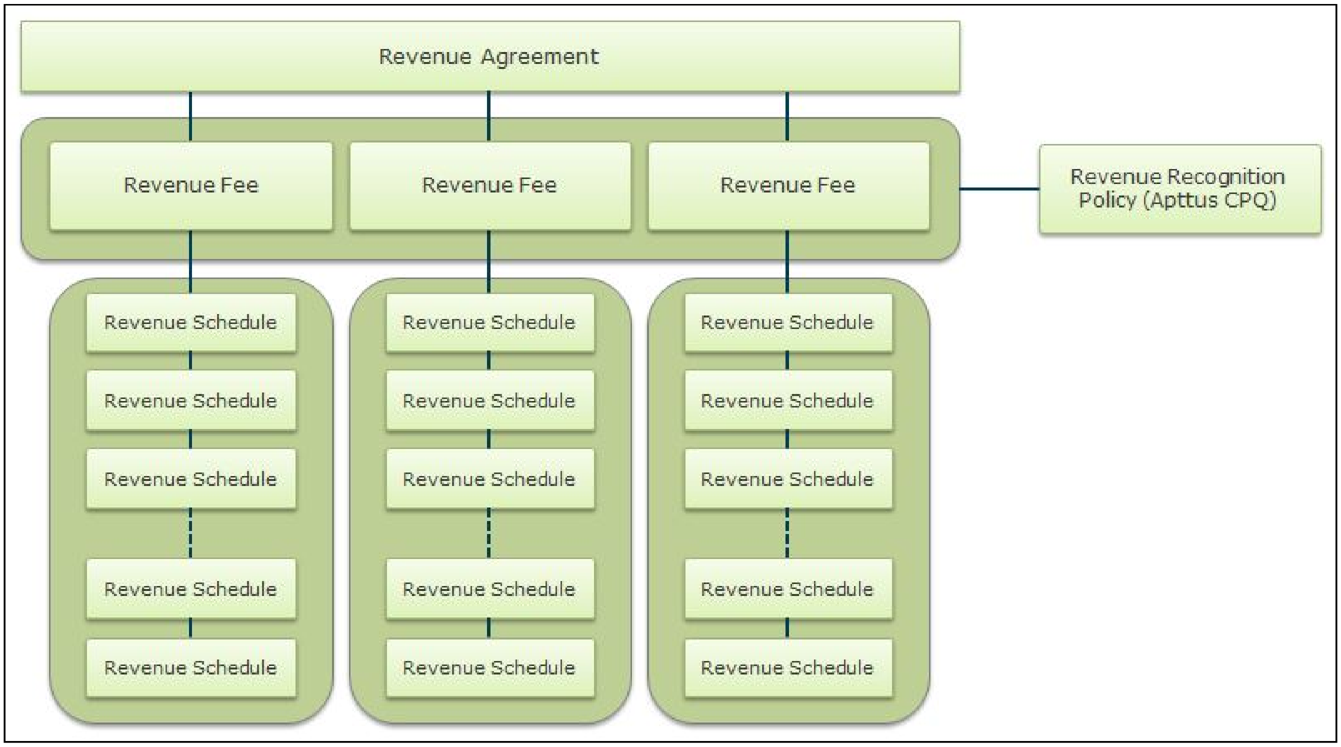
- **Performance Obligation** - Performance Obligation is a promise in a contract with a customer to transfer to the customer either:
  - A good or service (or a bundle of goods or services) that is distinct.
  - A series of distinct goods or services that are substantially the same and that have the same pattern of transfer to the customer.

- **Agreement Line** - Agreement Lines are created for each product in your agreement. Each agreement line will have net price, transaction price, and revenue allocation amount.

- **Revenue Split-Merge Policy -** Revenue Recognition Policies allows users to create rules so the system can automatically distribute revenue according to their business policies. Split or Merge policy is useful when you have single asset having multiple obligations or multiple assets that are a part of single obligation.

- **Revenue Forecast Schedules** - Revenue Forecast Schedules are created when you run the Forecast Revenue batch job. Revenue Forecast Schedules show the distribution of revenue that is yet to be recognized. They are used to predict a company's future financial state.

The diagram below depicts the revenue agreement and its components.



Understand

It is essential to analyze each transaction to identify the attributes of the transaction that impact revenue.

In order to recognize revenue from a transaction, companies must meet the following criteria.

- An Agreement exists between the company and the customer.
- Price must be fixed and determinable.
- The collection is reasonably assured.
- Proof of delivery of products or services must exist.

Before any other decisions can be made, it is essential that you analyze the transaction to determine how and when each criterion will be met. This means answering questions such as

- Which document represents the full transaction? Is the contract represented by a single order or multiple orders?
- Does this transaction modify an existing contract or is this a new agreement?
- Can each good or service be used individually or are they dependent on other goods or services being fully delivered?
- Is the price of this service fixed or does it vary based on usage?
- When will the good or service be delivered? Is it delivered all at once as a hard good? Or is it delivered over time as a service?
- Does the contract give the customer the ability to cancel at any time?
- Is this revenue recognized over time or immediately?
- What are the important dates for revenue? Will the start of revenue be delayed?
- If recognized over time, how will revenue be distributed across the revenue term?

The answers to these questions will allow you to determine how should revenue be recognized for each good or service. This allows you to identify a Revenue Recognition Policy or Rev Rec Rule to be used to recognize revenue for each good or service.

For more complex scenarios, revenue for a service may be different than the amount which has been invoiced. In these situations, it is necessary to calculate the total revenue that will be recognized for each good or service over the life of the contract. Multi-element Arrangements (MEAs) occur when we group multiple items together for the revenue recognition purposes. This is most common in the case of bundles but could also occur when multiple items on a contract are related.

For example, a sales representative could provide 2 years of service for free as part of a deal. From a billing perspective, the value of the service is zero. For revenue, we need to consider the fair value of the service and account for that as revenue over the 2 years the service is provided.

A variety of approaches exist for calculating the fair market value in MEA scenarios:

- Vendor Specific Objective Evidence (VSOE) - this method is used when a good or service is sold by the vendor as a standalone offering. When using this method, the standalone price is used.
- Third Party Evidence (TPE) - this method is used when a good or service is not sold by the vendor as standalone but similar services are offered as standalone services by other vendors in the market.

When using this method, the average price of that other vendors charge for the equivalent service is used.

- Best Estimated Selling Price (BESP) - this method is used when the good or service is not sold as a standalone offering by any vendor.

Once calculated, this value will be stored in Apttus Revenue Recognition for each revenue item.

As a result of performing these steps, the transaction will be broken down into individual goods or services. For each good or service, you will know how much revenue will be recognized as well as the rule or policy you will use to distribute that revenue.

It is essential to understand the core elements of Apttus Revenue Recognition.

- Revenue Recognition Rules
- Advanced Revenue Recognition Rules
- Revenue Recognition Policies (Apttus CPQ)

Revenue Recognition Rules

Revenue Recognition Rules determine how revenue will be recognized. Revenue Recognition Rules are the foundation rules of the policy that a company follows. Revenue Recognition Rules are assigned to each Agreement Fee.

The following table describes the three **basic rules** that Apttus supports.

Revenue Recognition Rule	Description
Immediate	Recognize the entire fee amount immediately. For example, when you buy groceries from a store, revenue is recognized immediately.

Revenue Recognition Rule	Description
Ratable	Recognize the fee amount over a period of time and calculate the deferred balance accordingly. For example, when you subscribe to a video streaming and pay for one year upfront, then revenue will be recognized every month and the deferred balance will decline as revenue is recognized. **Note** This model is useful when there is no end date to the subscribed product or service. The customer pays as per usage.
Subscription	Recognize the fee amount over a period of time but does not track deferred revenue. For example, you subscribe to any online magazine with a monthly payment, revenue will be recognized over a period of twelve months. **Note** This model is useful for monthly subscription models where a customer is free to cancel at any time.

To add a Revenue Recognition Rule to your Agreement Fee, refer to Agreement Fees in Revenue Recognition User Guide.

Note

You can opt to consider or ignore Leap year while calculating revenue from Custom Settings. Go to Setup > Build > Develop > Custom Setting and click Manage for Revenue System Properties. Click Edit for System Properties and select **Ignore Leap Year For Revenue Recognition** to ignore the leap year for revenue recognition.

Advanced Revenue Recognition Rules

Revenue Recognition Rules determine how revenue will be recognized. Apttus Revenue Recognition supports three Advanced Revenue Recognition Rules.

The following table describes three Advanced Revenue Recognition Rules.

Advanced Revenue Recognition Rule	Description
Milestone	Recognize revenue based on one or more pre-defined milestones.
Prepay	Recognize revenue ratably from a prepaid amount (prepayment amount). When usage is above the prepayment ratable schedule, revenue is increased to reflect the increased usage.
True-up	Recognize revenue from committed revenue models. Revenue is increased to meet the minimum threshold if revenue is below a specific threshold on a specific day.

Revenue Recognition Policies (Apttus CPQ)

Note

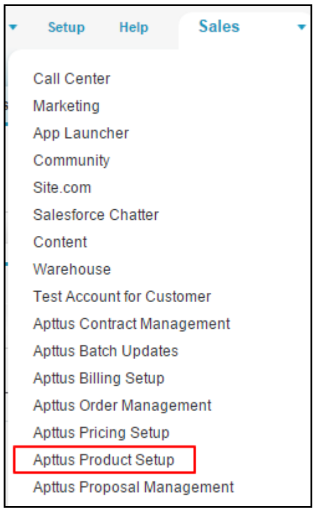
The Revenue Recognition Policies object is a part of Apttus CPQ product. After integration of Apttus Rev Rec and Apttus CPQ, Revenue Recognition Policies will impact the revenue to be generated.

Revenue Teams must be consistent in the methods they use to recognize revenue. This is done by identifying a set number of policies which are used to recognize revenue for all transactions. While exceptions may exist, the majority of their transactions will adhere to one of these policies. Revenue Recognition Policies allows users to create those rules within Apttus so the system can automatically distribute revenue according to their business policies. Revenue Recognition Policies must be assigned to Products or Price List Items to instruct the system on which rule to use for a particular transaction.

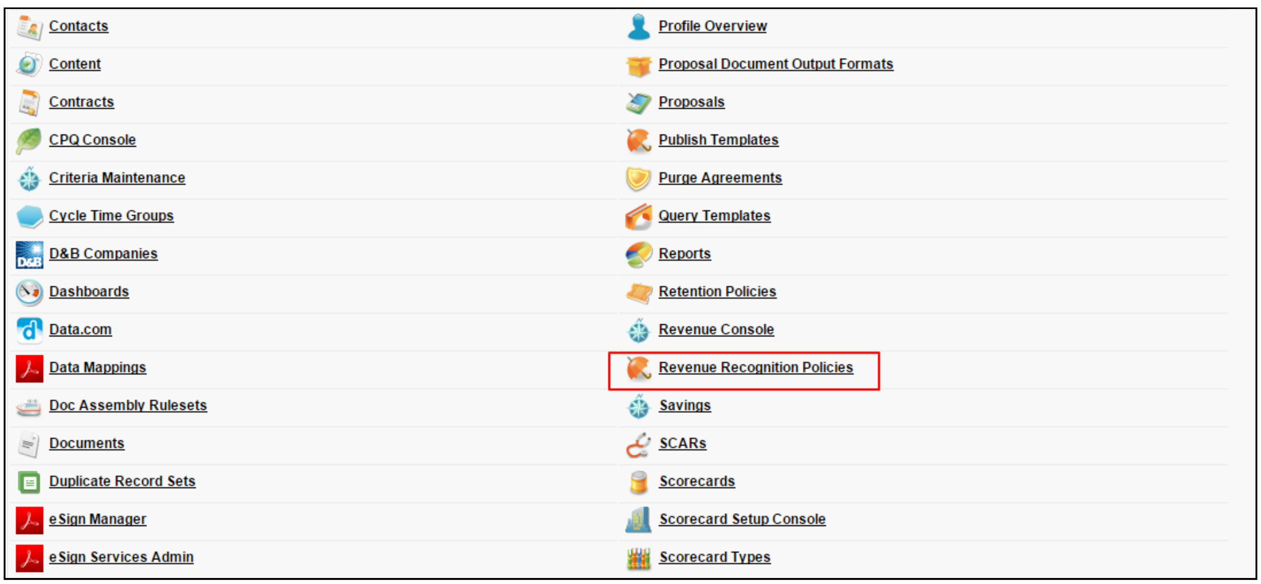
For more information on Revenue Recognition Rules, kindly refer Revenue Recognition Rules.

To create Revenue Recognition Policy

1. From Force.com drop down menu, select **Apttus Product Setup**.

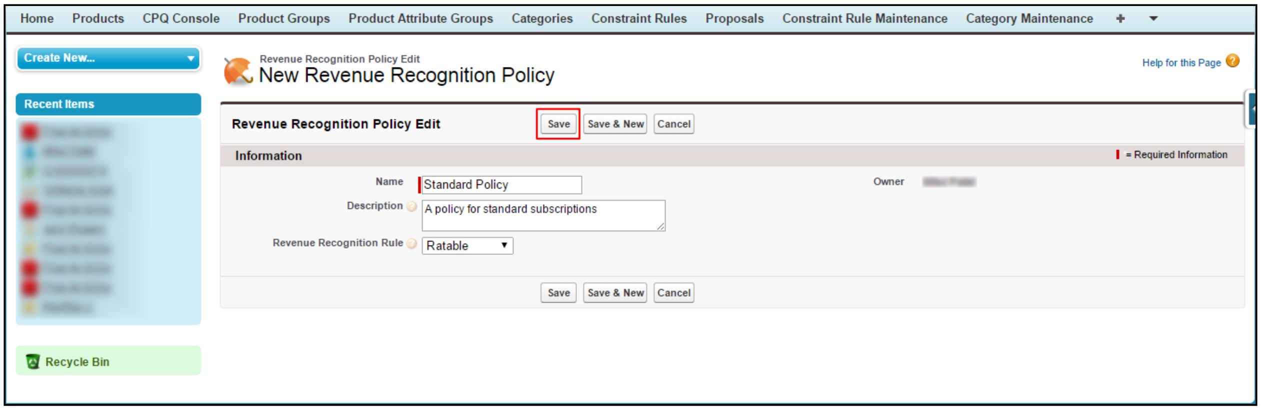


2. Click and select **Revenue Recognition Policies**.



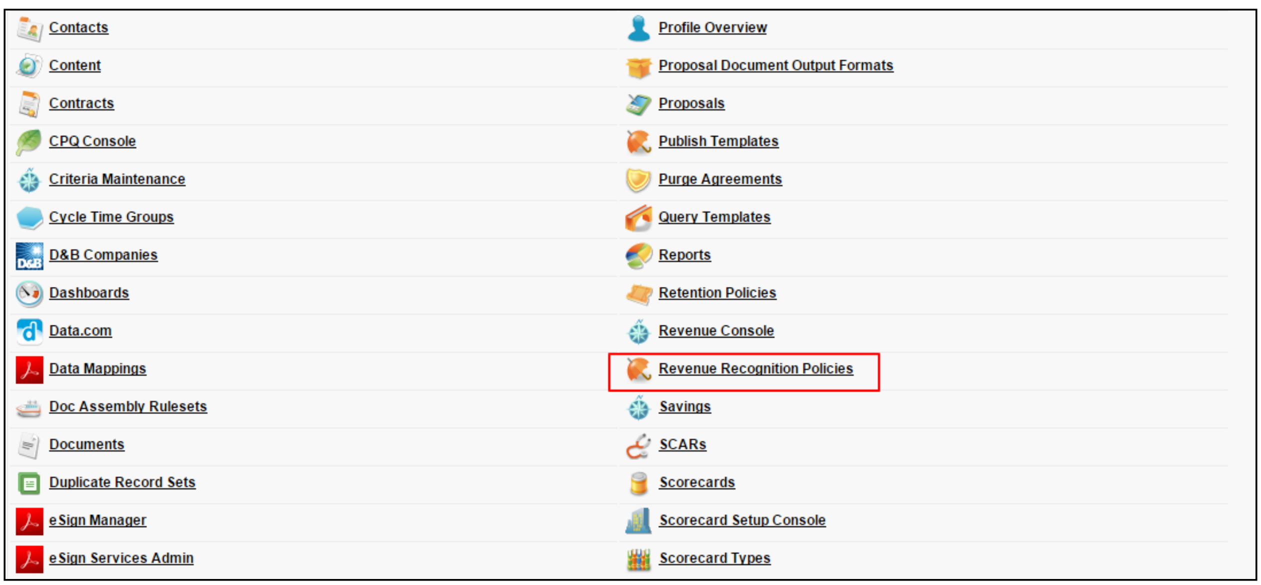
3. Click **New**.
4. Enter the **Name** and the **Description** of the Revenue Recognition Policy.
5. Select the appropriate **Revenue Recognition Rule** from the drop down menu.

6. Click **Save**.

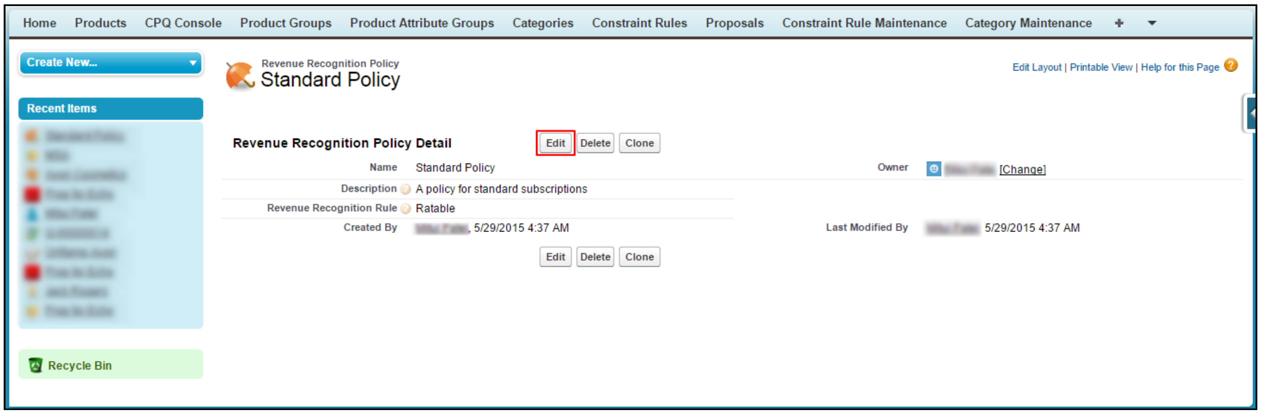


To edit Revenue Recognition Policy

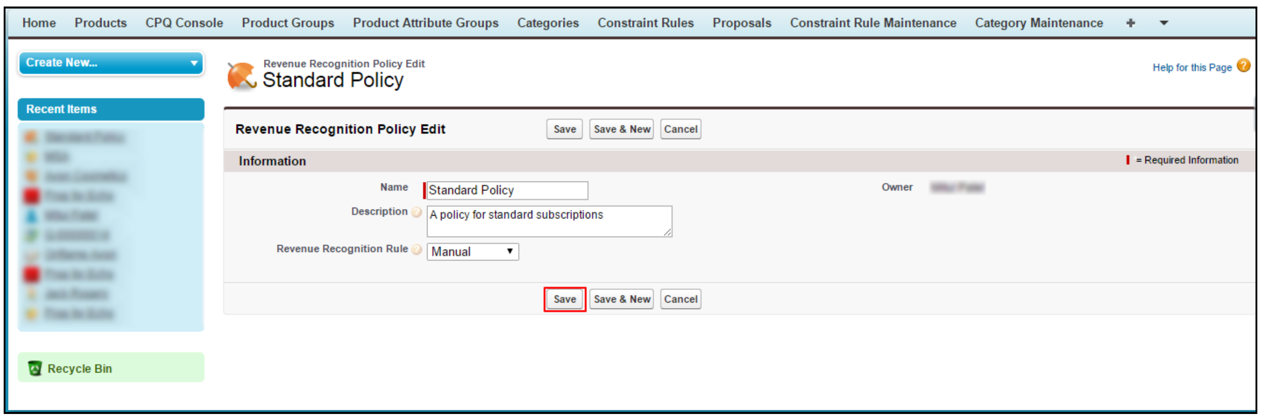
1. Click ➕ and select **Revenue Recognition Policies**.



2. Select an appropriate revenue recognition policy.
3. Click **Edit**.



4. After making appropriate changes to the policy, click **Save**.



Legal Entities

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.

To define a Legal Entity

1. Click the **Legal Entity** tab.
2. Click **New**.
3. Enter values for the fields described in the following table:

Field	Description
Legal Entity Id	Id of the Legal Entity
Company Code	The company code of the Legal Entity
Entity Name	The descriptive name of the legal entity
Is Default	Indicates whether this is a default legal entity
Legal Entity Address Line 1	The Legal Entity Address Line 1
Legal Entity Address Line 2	The Legal Entity Address Line 2
Legal Entity Address Line 3	The Legal Entity Address Line 3
Legal Entity Address Line 4	The Legal Entity Address Line 4
Legal Entity City	The Legal Entity City
Legal Entity Country Code	The Legal Entity Country Code
Legal Entity Postal Code	The Legal Entity Postal Code
Legal Entity State	The Legal Entity State
Phone Number	Contact Number of the Legal Entity
Tax ID	The Tax ID for the Legal Entity
VAT Enabled?	Indicates whether VAT is enabled for the legal entity
VAT ID	The VAT ID for the Legal Entity
Website	The Legal Entity Website Address

4. Click **Save.**

Associating a Legal Entity with an Agreement

Legal Entities are associated with an Agreement. If you do not specify a legal entity in the agreement. the default legal entity is associated to it.

To associate a Legal Entity with an Agreement

1. Click the Agreements tab and select the appropriate agreement.
2. Click Edit.

3. Associate the Legal Entity with the agreement through the Legal Entity Id lookup field.
4. Click Save.

Associating a Legal Entity with a RevRec Period

You need to create a RevRec period and associate it with a Legal Entity. Revenue for a legal entity is recognized based on the defined RevRec Period.

To associate a Legal Entity with a RevRec Period

1. Click the RevRec Period tab.
2. Select the RevRec Period to associate it with a legal entity.
3. Click Edit.
4. Enter Current Period Date and Forecast Period Date.
5. Associate the Legal Entity with the RevRec period through the Legal Entity Id lookup field.
6. Click Save.

To recognize and manage revenue for a Legal Entity, go to Revenue Console.

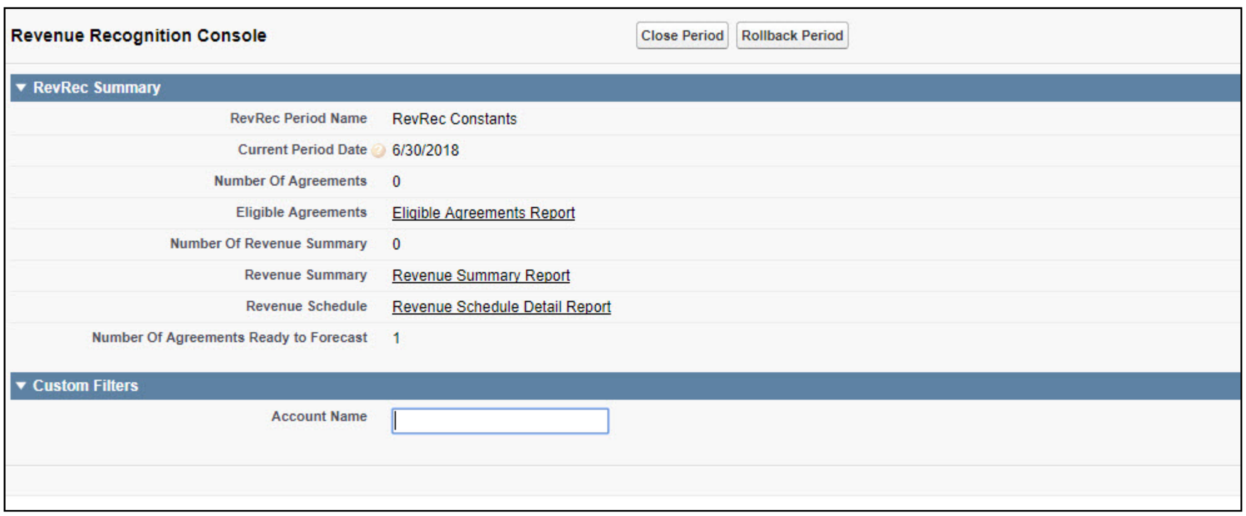
Custom Filters for Agreements

You can create custom filters to narrow down the list of agreements that you want to process. Based on the fields defined by you, you can filter out the agreements from the revenue console.

To create a Custom Filter:

1. Configure the custom settings for defining the fields for agreement filters.
   Enter your field name in the Revenue Agreement Filter Properties and select the corresponding Field Enabled check box. You can list down up to 10 fields for complex filtering.
   For more details, see Revenue Agreement Filter Properties.
2. Define the Agreement Filter Callback class.
   For details on how to add and customize the Agreement Filter Callback class, see Agreement Filter Callback Class.
3. Register your Agreement Filter Callback class in Revenue Recognition System Properties.
   For details on how to add your callback class, see Revenue Recognition System Properties.

Your custom filter is now available on the revenue console.



Use Case: Creating and Applying Custom Filters for Agreements

Description: This use case describes how to create and apply custom filters to agreements. You might use this functionality differently, depending on your business case.

Suppose you are a Finance administrator of a company and you need to configure custom filters based on the fields on which your agreements will be processed.

In the following example, you create custom filters to process agreements based on the account name.

To create a custom filter:

1. Go to **Setup > App Setup > Develop > Custom Settings** and click Manage next to Revenue Agreement Filter Properties. Click Edit next to System Properties.

2. Enter Account Name in **Field Label 1**.
3. Select **Field 1 Enabled**.
4. Click **Save**.
5. Go to **Setup > App Setup > Develop > Apex Classes.**
6. Click **New**.
7. Define your custom class and save it as TestCallBack.
8. Go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** next to **Revenue System Properties**. Click **Edit** next to **System Properties**.
9. Enter TestCallBack in **Agreement Filter Callback Class** field.
10. Click **Save**.

Result:

- Custom Filter for Account Name is created.

Applying a custom filter:

Your custom filter, **Account Name** is now available on the Revenue Recognition Console. Enter the name of the account for which you want to process your agreements in the Account Name field.

Agreements belonging to only that account are processed.

Revenue Recognition Callback Classes

This section covers the following Revenue Recognition callback classes:

- Agreement Filter Callback Class
- Obligation Allocation Callback Class
- Revenue Allocation Callback Class
- Revenue Calculation Callback Class

Agreement Filter Callback Class

A new custom callback class is supported to filter agreements to process from the revenue console. The agreements are filtered based on the fields specified by the user.

To add the custom class

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

> This is just a sample callback class to control which agreements to process for revenue management. You may change the code as per your requirements.

```
/**
  * Class used to contain one or more parameters for a given callback.
  */
global class Parameters
{
        // The Custom setting used to control which "fields" are presented to the user
        global RevenueAgreementFilterProperties__c filterProperties;

        // The values entered by the user. There should be 10 entries with null for
        // each field that is disabled. The 1st entry pertains to the 1st field, the
        // 2nd entry pertains to the 2nd field, and so on.
        global List<String> filterFieldValues;
}

global interface IAgreementFilterCallback {
```



```
        /**
          * Return an SOQL filter expression that will be appended to the "stock"
          * query used to fetch Agreements a given Revenue centric batch job.
          *
          * @param params Extra parameters for this method. Contains the
          *  "field values" entered by the user.
          */
        String getAgreementFilterExpression(Parameters params);
}
public class TestCallBackClass implements CustomClass.IAgreementFilterCallback {
    public String getAgreementFilterExpression(CustomClass.Parameters params){
            String filterQuery;
                if(params !=null){
                    filterQuery = 'Apttus__Account__r.Name = \''+params.filterFieldValues[0]+'\' ';
                    }
                    return filterQuery;
        }
    public String getAgreementRevenueSummaryFilterExpression(CustomClass.Parameters params){
    String filterQuery;
    if(params !=null){
            filterQuery = 'Agreement__r.Apttus__Account__r.Name = \''+params.filterFieldValues[0]
+'\' ';
    }
    return filterQuery;
    }

}
```

4. Click **Save.**

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

Refer to Revenue System Properties for more details.

Obligation Allocation Callback Class

A new custom callback class is supported to create Performance Obligations.

It creates performance obligations for a set of Agreement Line Items and allocates revenue to it. For each performance obligation, it calculates Allocated Amount and Transaction Price.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.

2. Click **New**.
3. Enter the sample callback class code.

> ⓘ This is just a sample callback class to allocate performance obligations. You may change the code as per your requirements.

```
global interface IPerformanceObligationAllocation
{
        /**
         * Create the Performance Obligations for the specified list of Agreement
         * Line Items. It is essential (and expected) the created Performance
         * Obligations are also "inserted" into the database and the Transaction Price
         * and Allocated Amount fields of each Obligation are initialized to 0.
         *
         * After the Performance Obligations are returned the Revenue Configuration/Integration
         * package will create/link the various Agreement Fees and compute (aggregate) and
         * set the Transaction Price and Allocation Amount fields on the returned Obligations.
         *
         * @param agreementLineItms The list of Agreement Line Items to evaluate
         *   and create the various performance Obligations for.
         * @param agreemnt The parent Agreement for the specified Agreement
         *   Line Items.
         *
         * @return the Performance Obligations that were created as a list of wrapper objects.
         *   Each wrapper object will allow the Revenue Configuration/Integration package
         *   (which calls the callbaclk) to correctly create the Agreement Fees and corrrectly
         *   set the Transaction Price and Allocated Amount fields on the newly created Performance
         *   Obligations.
         *
         *   For the case of a Revenue Split Policy there will be a "map entry" for each Revenue
         *   Split Policy Entry (because a Performance Obligation is created for each entry). In
addition,
         *   each of the Performance Obligations will reference the same Agreement Line Item.
         */
        List<PerformanceObligationInfo> createPerformanceObligations(
                List<Apttus__AgreementLineItem__c> agreementLineItms,
                Apttus__APTS_Agreement__c agreemnt);
}

/**
 * Wrapper class used to return comprehensive information for each Performance Obligation
```



```
 * returned by the "create Performance Obligations" callback. The additional information
 * will allow the Revenue Configuration/Integration package (which calls the callback)
 * to correctly create the Agreement Fees and correctly set the Transaction Price and
 * Allocated Amount fields on the newly created Performance Obligations.
 */
global class PerformanceObligationInfo
{
        // The Performance Obligation created for 1 or more Agreement Line Items
        global Apttus__Obligation__c obligationSO;

        // The Agreement Line Items that pertain to the newly created Performance Obligation.
        // Typically there is only 1 Agreement Line Item but all Agreement Line Items (of an
Agreement)
        // that have the same Revenue Merge Polcy will be grouped together.
        global List<Apttus__AgreementLineItem__c> agreementLineItems;

        // For the case where the Agreement Line Item has Revenue Split Policy this is Id of
        // Revenue Split Policy Entry that was applied to the newly created Performance Obligation,
        // otherwise this Id will be "null".
        global ID splitPolicyEntryId;
        global Id policyId;
        global Map<Id, Id> splitPolicyEntryIdbyLineItemId;
}
public class TestObligationAllocationCallback
        implements CustomClass.IPerformanceObligationAllocation {

        public List<CustomClass.PerformanceObligationInfo> createPerformanceObligations(
                        List<Apttus__AgreementLineItem__c> agreementLineItms,
                        Apttus__APTS_Agreement__c agreemnt) {

                Apttus__Obligation__c obligationSO = new Apttus__Obligation__c();
                obligationSO.Apttus__AllocationAmount__c = 0;
                obligationSO.Apttus__TransactionPrice__c = 0;

                for(Apttus__AgreementLineItem__c agreementLineItemSO :agreementLineItms) {
                        obligationSO.Apttus__AllocationAmount__c +=
agreementLineItemSO.AllocationAmount__c;
                        obligationSO.Apttus__TransactionPrice__c +=
agreementLineItemSO.TransactionPrice__c;
                        obligationSO.Apttus__StartDate__c = (obligationSO.Apttus__StartDate__c ==
null ||

obligationSO.Apttus__StartDate__c > agreementLineItemSO.Apttus_CMConfig__StartDate__c)
                                                                                                ?
agreementLineItemSO.Apttus_CMConfig__StartDate__c
                                                                                                :
obligationSO.Apttus__StartDate__c;
```



```
                        obligationSO.Apttus__EndDate__c = (obligationSO.Apttus__EndDate__c == null
 ||

obligationSO.Apttus__EndDate__c < agreementLineItemSO.Apttus_CMConfig__EndDate__c)
                                                                                                   ?

agreementLineItemSO.Apttus_CMConfig__EndDate__c
                                                                                                   :

obligationSO.Apttus__EndDate__c;

                }


                obligationSO.Apttus__RevRec_Rule__c = RevRecConstants.REVRECRULE_RATABLE;
                obligationSO.Apttus__Status__c = RevRecConstants.OBLIGATION_STATUS_ACTIVE;
                obligationSO.Apttus__Type__c = RevRecConstants.OBLIGATION_TYPE_PERFORMANCE;
                obligationSO.Apttus__Description__c = null;

                CustomClass.PerformanceObligationInfo performanceObligationInfoObj = new
 CustomClass.PerformanceObligationInfo();
                performanceObligationInfoObj.obligationSO = obligationSO;
                performanceObligationInfoObj.agreementLineItems = agreementLineItms;

                return new List<CustomClass.PerformanceObligationInfo>
{performanceObligationInfoObj};

        }
}
```

4. Click **Save.**

To register your custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Revenue System Properties**. Click **Edit** for **System Properties** and enter the name of your custom callback class in Performance Obligation Allocation Callback.

Refer to Revenue System Properties for more details.

Revenue Allocation Callback Class

A new custom callback class is supported to calculate and set the allocated revenue amount for each Agreement Line Item.

It allocates performance obligations for a set of Agreement Line Items. For each performance obligation, it calculates Allocated Amount and Transaction Price.

To add the custom class

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

> This is just a sample callback class to allocate performance obligations. You may change the code as per your requirements.

```
global interface IRevenueAllocation
        {
                /**
                 * Calculate the allocated revenue amount for each Agreement Line Item
                 * and update the "Allocation Amount" on each Agreement Line Item
                 * accordingly. There is no need to call "Upsert" as that will performed by
                 * the Revenue Configuration/Integration package.
                 *
                 * Upon return, the sum of the Transaction Price fields of the Agreement
                 * Line Items must match the sum of the Allocation Amount fields. If that
                 * is not the case, the Revenue Integration/Configuration package will throw
                 * an Exception.
                 *
                 * @param agreementLineItms The list of Agreement Line Items the dependent
                 *   charges pertain to. It is expected the Allocation Amount on each
                 *   Agreement Line Item will be updated accordingly.
                 * @param agreemnt The corresponding Agreement.
                 */
                void calculateAllocationAmount(
                        List<Apttus__AgreementLineItem__c> agreementLineItems,
                        Apttus__APTS_Agreement__c agreemnt);
        }
public with sharing class TestRevenueAllocationCallback
        implements CustomClass.IRevenueAllocation {

        @TestVisible private static Boolean specialUsecase = false;

        public void calculateAllocationAmount(List<Apttus__AgreementLineItem__c> agreementLineItems,

Apttus__APTS_Agreement__c agreemnt) {

                //get sum of all LineItems having MEA flat to true
                Decimal totalTransactionPrice = 0.0;
                Decimal totalNetPrice = 0.0;
```

```
                //initialise variables to Allocate proper value to all LineItems
                Integer iCount = 0;
                Decimal totalRemainingTransactionPrice = 0;
                for(Apttus__AgreementLineItem__c lineItemSO :agreementLineItems) {
                        totalRemainingTransactionPrice += lineItemSO.TransactionPrice__c;
                        totalTransactionPrice += lineItemSO.TransactionPrice__c;
                        totalNetPrice += lineItemSO.Apttus__NetPrice__c;
                        iCount++;
                }

                //calculate Allocation Amount for each LineItem
                for(Apttus__AgreementLineItem__c lineItemSO :agreementLineItems) {
                        if(specialUsecase) {
                                lineItemSO.AllocationAmount__c = 10;
                                continue;
                        }
                        if(iCount == 1) {
                                lineItemSO.AllocationAmount__c = totalRemainingTransactionPrice;
                        } else {
                                lineItemSO.AllocationAmount__c = ((lineItemSO.TransactionPrice__c /
totalTransactionPrice) * totalNetPrice).setScale(RuntimeContext.DEFAULT_SCALE);
                        }
                        totalRemainingTransactionPrice -= lineItemSO.AllocationAmount__c;
                        iCount--;

                }

        }

}
```

4. Click **Save.**

To register your custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Revenue System Properties**. Click **Edit** for **System Properties** and enter the name of your custom callback class in Revenue Allocation Callback.

Refer to Revenue System Properties for more details.

Revenue Calculation Callback Class

A new custom callback class is supported to calculate and set the Transaction Price for specified Agreement Line Items.

To add the custom class

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

> This is just a sample callback class to calculate Transaction Price. You may change the code as per your requirements.

```
global interface IRevenueCalculation
        {
                /**
                 * Calculate and set the Transaction Price for the specified Agreement Line Items.
                 * There is no need to call "Upsert" as that will performed by the Revenue
                 * Configuration/Integration package.
                 *
                 * @param agreementLineItms The list of Agreement Line Items to update.
                 *   It is expected the Transaction Price will be set on each Agreement Line
                 *   Item.
                 * @param agreemnt The corresponding Agreement.
                 */
                void calculateTransactionPrice(
                        List<Apttus__AgreementLineItem__c> agreementLineItems,
                        Apttus__APTS_Agreement__c agreemnt);
        }
public with sharing class TestRevenueCalculationCallback
        implements CustomClass.IRevenueCalculation {

        public void calculateTransactionPrice(List<Apttus__AgreementLineItem__c> agreementLineItems,

Apttus__APTS_Agreement__c agreemnt) {
                Decimal totalNetPrice = 0.0;
                for(Apttus__AgreementLineItem__c lineItemSO :agreementLineItems) {
                        totalNetPrice += lineItemSO.Apttus__NetPrice__c;
                        lineItemSO.TransactionPrice__c = lineItemSO.Apttus__NetPrice__c;

                }

                if(totalNetPrice > 1000) {
                        for(Apttus__AgreementLineItem__c lineItemSO :agreementLineItems) {
                                lineItemSO.TransactionPrice__c = (lineItemSO.Apttus__NetPrice__c -
(lineItemSO.Apttus__NetPrice__c / 100)).setScale(RuntimeContext.DEFAULT_SCALE);
```

```
                        }
                }
        }
}
```

4. Click **Save.**

To register your custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** beside **Revenue System Properties**. Click **Edit** for **System Properties** and enter the name of your custom callback class in Revenue Calculation Callback.

Refer to Revenue System Properties for more details.

Appendix

Appendix in this user guide includes the detailed information of the objects, system properties, and their description. Apttus and Apttus Support suggest that you view this set of information with care to ensure you fully understand how the system is working.

Note

All implementation of Apttus Revenue Management are different due to different business requirements.

Apttus Revenue Recognition Package Objects

Apttus Revenue Recognition Package Objects

The following table displays the objects of the Apttus Revenue Recognition Package and their purpose.

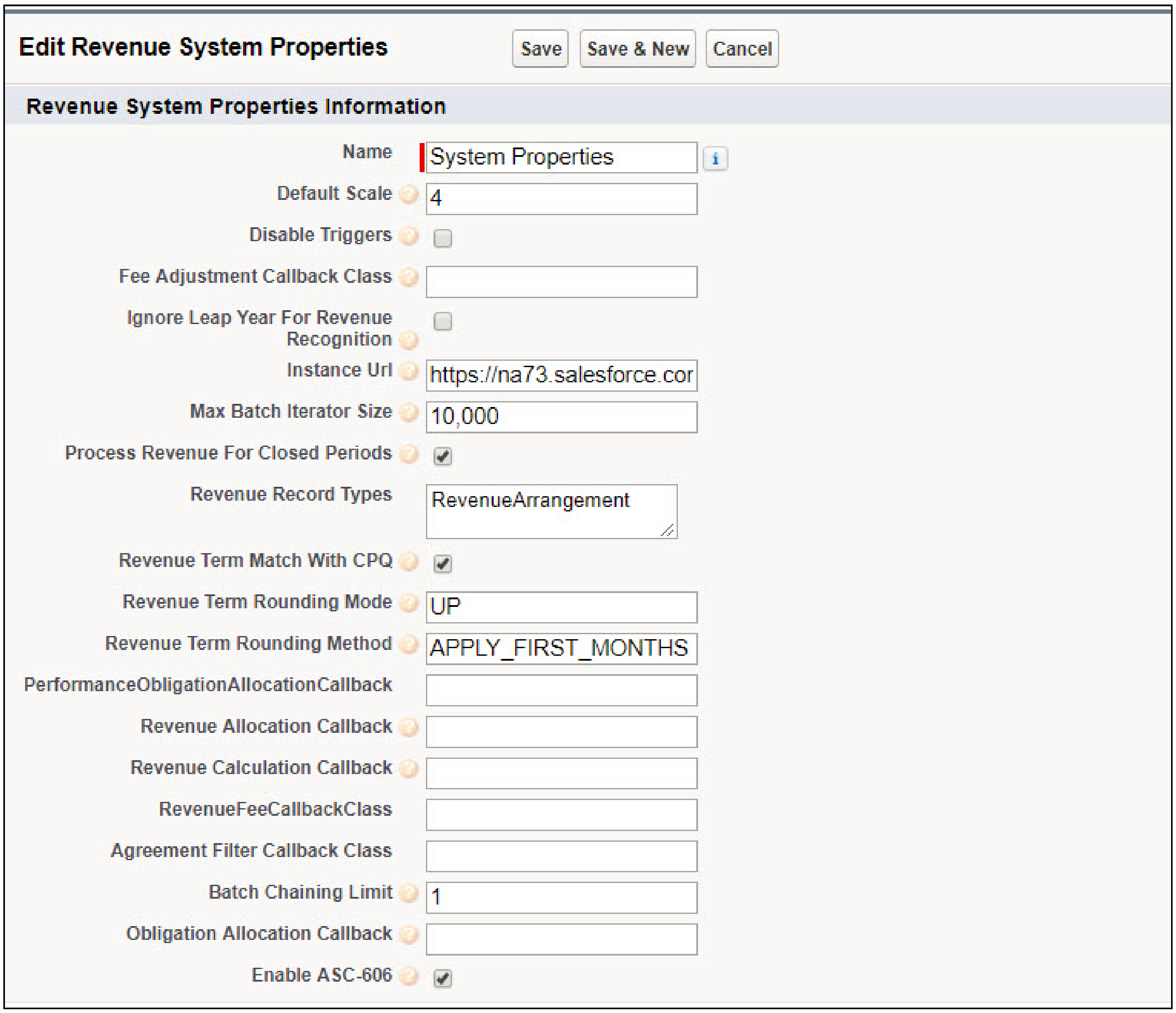
Apttus Object	Purpose of the Object
Agreement Fee	Represents a fee associated with an agreement.
Agreement Fee Adjustment	Holds the adjustments for a fee.
Agreement Fee Schedule	Holds the budget, forecast, and actual units for a fee.
Agreement Fee Tier	Holds the tier for a fee.
Agreement Fee-Prepay	Holds the many to many relationships between agreement fee and agreement prepay objects.
Agreement Fee-Product	Holds the many to many relationships between agreement fee and agreement product objects.
Agreement Fee-Trueup	Holds the many to many relationships between agreement fee and agreement trueup objects.
Agreement Prepay	Represents a prepaid amount associated with an agreement.

Apttus Object	Purpose of the Object
Agreement Prepay-Trueup	Holds the many to many relationships between agreement prepay and agreement trueup objects.
Agreement Product	Represents a product associated with an agreement.
Agreement Revenue Adjustment Schedule	Holds the revenue adjustments for an agreement.
Agreement Revenue Forecast Snapshot	Holds revenue forecast snapshot for an agreement.
Agreement Revenue Schedule	Holds a revenue schedule for an agreement.
Agreement Revenue Schedule Forecast	Holds a revenue schedule forecast for an agreement.
Agreement Revenue Snapshot	Holds a revenue snapshot for an agreement.
Agreement Revenue Summary	Holds a revenue summary for an agreement.
Agreement Trueup	Represents guaranteed units or amount for an agreement.
BatchJob	Holds batch jobs.
Milestone	Represents milestones associated with other business objects.
Related Agreement Fee	Holds the many to many relationships between agreement fee objects.
RevRec Period	Holds revenue recognition period constants.

Apttus Revenue Recognition System Properties

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

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



The following table lists the available fields and their significance.

Field	Description
Agreement Filter Callback Class	The name of the custom class that implements the _ Agreement Filter Callback_ interface.
Batch Chaining Limit	A System Setting to control the Number of chained Batch Jobs. Default value is 1.
Default Scale	The number of decimal places for computed revenue.
Disable Triggers	Indicates whether managed triggers are disabled. The application would not work correctly when triggers are turned off. ⚠ **Note** Please use this only in special data load situations like data migration.
Enable ASC-606	Controls which Revenue Recognition Standard to apply (ASC-605 or ASC-606).
Fee Adjustment Callback Class	The fee adjustment callback class name. The class should implement the CustomClass.IFeeAdjustmentCallback interface.
Ignore Leap Year For Revenue Recognition	Indicates whether leap year should be ignored for revenue recognition
Instance URL	The URL of your Salesforce instance. The instance URL will be used to create an absolute URL for your custom pages.
Max Batch Iterator Size	The maximum batch records to process in a single iteration.
Obligation Allocation Callback	The name of the class that implements the IPerformanceObligationAllocation interface.
Process Revenue For Closed Periods	Indicates whether revenue would be recognized for fees in closed periods. This is to support business transactions backdated under a variety of circumstances.
Revenue Allocation Callback	The name of the class that implements the IRevenueAllocation interface.
Revenue Calculation Callback	The name of the class that implements the IRevenueCalculation interface.
Revenue Record Types	Indicates the record type to be used to recognize the revenue, generate forecasts, and rollback revenue.
Revenue Term Match With CPQ	This setting allows RevRec term calculation to match with CPQ.

Field	Description
Revenue Term Rounding Mode	The rounding mode for computing the revenue term. Valid values are UP, DOWN, HALF_UP, and HALF_DOWN.
Revenue Term Rounding Method	The rounding method for computing the revenue term. Valid values are SPREAD_DEFERRED-BALANCE, APPLY_FIRST, APPLY_LAST, APPLY_FIRST_MONTHS and APPLY_LAST_MONTHS.

Revenue Agreement Filter Properties

You can specify custom fields on which you want to filter the agreements to process.

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

The following table lists the available fields and their significance.

Field	Description
Console Revenue Summary Count	Stores the Rev Summary Count, which is populated in the Revenue Console. Revenue Summary Count is calculated by the batch job, CalulateRevConsoleSummaryValueBatchJob, which starts running in an asynchronous mode every time you load the Revenue Console.
Field1 Label, **Field 2 Label,** **.....,** **Field10 Label**	The name of the 1st field that is shown to the user in the Revenue Console.
Field 1 Enabled, **Field 2 Enabled,** **......,** **Field 10 Enabled**	If it is selected, the 1st field is shown to the user in the Revenue Console.

Apttus Copyright Disclaimer

For additional resources and support, please visit https://apttusportal.force.com/community.

