Billing and Revenue Recognition Solution Documentation

1. Introduction

This document outlines the implementation of a revenue recognition calculation and invoice automation solution in Salesforce. The solution leverages the standard Salesforce data model (Opportunity, OpportunityLineItem, and Invoice) along with custom logic to adhere to ASC 606 revenue recognition standards. It also provides a user interface component via Lightning Web Components (LWC) to display the calculated revenue milestones directly on the Opportunity record page.

Note on Invoice Object:
Only if Salesforce Billing is installed can you generate the invoice using the standard Invoice object. Hence, a custom Invoice object has been created in this solution. For more information on Salesforce Billing and its Invoice object, see the Salesforce Billing Overview.

---

2. Architecture Overview

a. Invoice Object

Standard vs. Custom:

- **Standard Invoice Object:** Available with Salesforce Billing, allowing native invoice generation and processing.
- **Custom Invoice Object:** A custom Invoice object (with a Master-Detail relationship with Opportunity) is created and linked with the Opportunity object to track invoice details.

b. EC_RevenueRecognitionController Class

Purpose:
This Apex class calculates revenue recognition milestones based on a two-stage process:

- **Milestone-Based Recognition:** 40% of total revenue is recognized upon product delivery.
- **Subscription-Based Recognition:** The remaining 60% is recognized evenly over a 12-month subscription period following delivery.

Additional Responsibilities:
The controller also retrieves and displays invoice status (open, paid, or not generated), the invoiced amount, and the total recognized revenue.

c. Trigger Handler on Opportunity

Trigger Functionality:
A trigger handler is implemented on the Opportunity object.

- **When Opportunity Stage is Inserted or Updated:**
  - When an Opportunity is Closed, the handler automatically updates the closed date on the Opportunity record.
  - When an Opportunity is moved to the Closed Won stage, the handler automatically creates an associated Invoice record.

d. Scheduled Batch Process

Batch Job Schedule:
A scheduled Apex batch runs daily at 12 AM.

Batch Responsibilities:

- It checks if one month has elapsed since the last invoice date.
- If the condition is met, it creates a new Invoice record in an "Open" status.

Purpose:
This ensures that subscription-based revenue is invoiced on a recurring basis until the full amount is recognized.

e. Lightning Web Component (LWC)

Component Features:

- Displays revenue recognition milestones (dates and amounts) for each Opportunity.
- Shows invoice statuses and provides a summary of invoiced amounts versus total recognized revenue.
- Dynamically renders data in a tabular format and is embedded on the Opportunity record page.
- Also displays products (Opportunity Line Items) using the Lightning Tree Grid component.

Visibility:
The component is visible only on Opportunities with a Closed Won status.

3. Business Logic & Assumptions

Revenue Recognition Split:

- **40% upon Product Delivery:** Recognized immediately when the product is delivered.
- **60% over Subscription Period:** Spread evenly across a 12-month period post-delivery.

Opportunity and Invoice Integration:

- The Opportunity's CloseDate is used as the revenue recognition start date.
- Related OpportunityLineItems provide the detailed breakdown of revenue, if needed.
- Invoice status (open, paid) helps in reconciling actual payments against recognized revenue.

Trigger & Batch Processes:

- The Opportunity trigger creates initial invoices when an Opportunity is marked as Closed Won.
- A scheduled batch process ensures that invoices are generated regularly (monthly) to support ongoing revenue recognition.

---

4. Error Handling Strategy

A robust error handling strategy is critical to ensure data integrity and reliability. The following potential issues and corresponding handling strategies have been identified:

a. Data Inconsistencies

- **Scenario:** Missing or null values in key fields such as Opportunity Amount or CloseDate.
- **Handling:**
    - Validate data before performing calculations.
    - The system will return a clear error message.
    - Use default values or skip processing for records that do not meet minimum data criteria.

b. DML Exceptions

- **Scenario:** Failures during record creation or updates (e.g., invoice creation in the trigger or batch).
- **Handling:**
    - Use try-catch blocks in Apex to capture and throw exceptions.

c. Batch Processing Failures

- **Scenario:** Errors occurring during scheduled batch execution (e.g., issues with invoice creation).
- **Handling:**
    - Implement error logging within the batch job.
    - Configure monitoring and alerts to notify administrators in the event of repeated failures.

---

5. Production Deployment (PD) Steps

To deploy this solution successfully, follow these production deployment steps:

1. **Assign Permission Set:**
    - **Permission Set Name:** `EC_RevenueRecognitionPermissionSet`
    - **Purpose:** Grants users the necessary permissions to access and interact with the revenue recognition functionality and related components.
2. **Activate Custom Setting:**
    - **Custom Setting Name:** `EC_IsTriggerEnable__c`
    - **Field:** `EC_Is_Active_Opportunity_Trigger`
    - **Purpose:** Enables the Opportunity trigger. Ensure this setting is active to allow the trigger to operate.
3. **Schedule Batch Process:**
    - **Batch Job Name:** `EC_CreateOpportuniyInvoiceBatch`
    - **Schedule:** Configure the scheduled batch job to run every day at 12 AM.
    - **Purpose:** Automatically checks if one month has elapsed since the last invoice date and, if so, creates a new Invoice record in Open status.

---

6. Future Improvements & Enhancements

a. Invoice Line Items

- **Enhancement:** In the future, implement invoice line items to break down each Invoice record into detailed components.
- **Benefit:** Enhances the transparency of invoicing and allows more detailed tracking of revenue against specific products or services.

b. Refactoring to Utility Classes

- **Enhancement:** Identify common or reusable logic within the Apex classes (e.g., calculation methods, data formatting, or error logging) and refactor them into dedicated utility classes.
- **Benefit:** Improves code maintainability and reusability, reduces code duplication, and simplifies testing.

c. Lazy Loading

- **Enhancement:** Implement lazy loading to efficiently handle large datasets.
- **Benefit:** Reduces initial load time and resource consumption, improving performance and scalability when processing and displaying large amounts of data.

d. Implementation of a Security Library

- **Enhancement:** Implement a security library that centralizes security-related functions such as input validation, field-level security enforcement, and protection against common vulnerabilities.
- **Benefit:** Enhances the overall security posture of the application by ensuring that data access and modifications follow best practices and compliance standards.

---

7. GitHub Repository

For version control, collaboration, and further reference, the solution is maintained in the following GitHub repository:
Repository Link: https://github.com/anshulsimlote/Salesforce-Milestone-Revenue

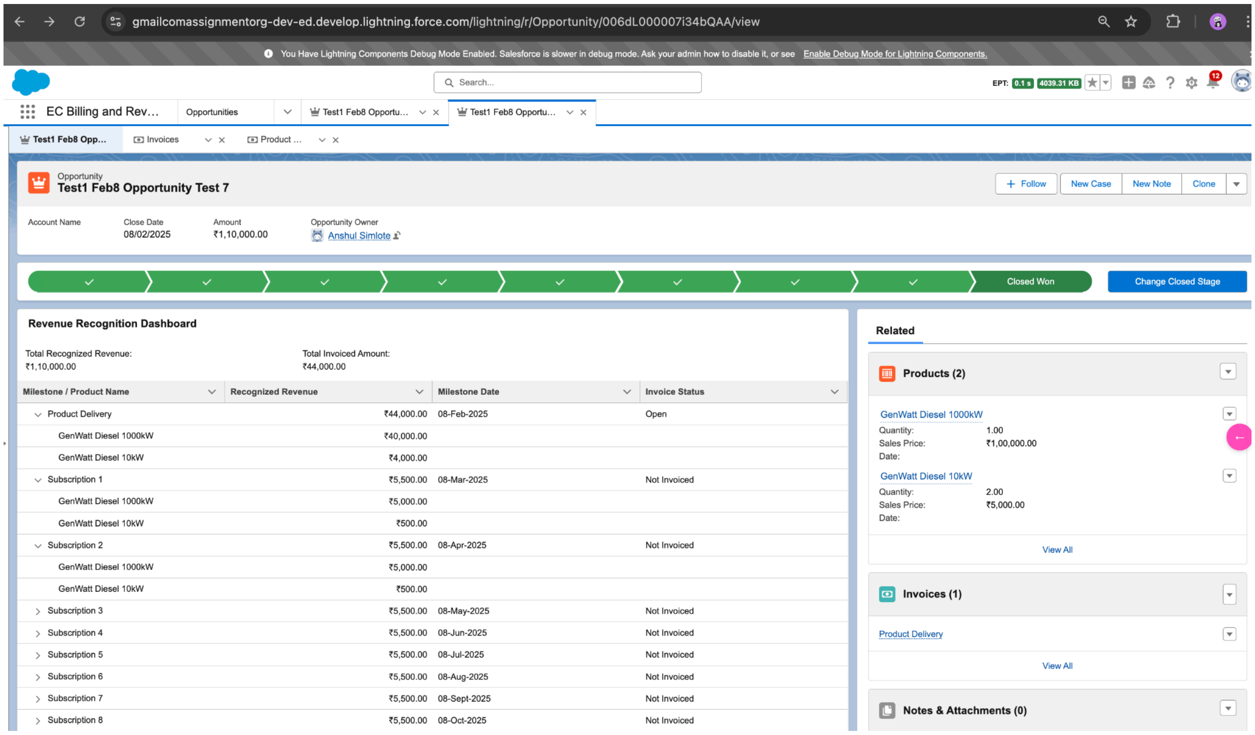
---

8. Manifest

Component Type	**Members**
ApexClass	EC_RevenueRecognitionController, EC_RevenueRecognitionMilestoneWrapper, EC_OpportunityTriggerHandler, EC_CreateOpportuniyInvoiceBatch, EC_Constants, EC_Exception
ApexTrigger	EC_OpportunityTrigger
CustomApplication	EC_Billing_and_Revenue_Recognition

Component Type	**Members**
CustomField	EC_Invoice__c.EC_Amount__c, EC_Invoice__c.EC_Invoice_Date__c, EC_Invoice__c.EC_Opportunity__c, EC_Invoice__c.EC_Status__c, EC_IsTriggerEnable__c.EC_Is_Active_Opportunity_Trigger__c, Opportunity.EC_Invoiced_Amount__c, Opportunity.EC_Remaning_Amount__c
CustomObject	EC_Invoice__c, EC_IsTriggerEnable__c
FlexiPage	EC_Billing_and_Revenue_Recognition_UtilityBar, Opportunity_Record_Page
Layout	EC_Invoice__c-Invoice Layout, Opportunity-Opportunity Layout
LightningComponentBundle	ecrevenueRecognitionDashboard
PermissionSet	EC_RevenueRecognitionPermissionSet

8. Screenshots


gmailcomassignmentorg-dev-ed.develop.lightning.force.com/lightning/r/Opportunity/006dL000007i34bQAA/view
You Have Lightning Components Debug Mode Enabled. Salesforce is slower in debug mode. Ask your admin how to disable it, or see Enable Debug Mode for Lightning Components.
Search...
EPT: 0.1 s 4039.31 KB
12
EC Billing and Rev...
Opportunities
Test1 Feb8 Opportu...
Test1 Feb8 Opportu...
Test1 Feb8 Opp...
Invoices
Product ...
+ Follow
New Case
New Note
Clone
Closed Won
Change Closed Stage
Related
View All
View All

Opportunity
Test1 Feb8 Opportunity Test 7

Account Name

Close Date
08/02/2025

Amount
₹1,10,000.00

Opportunity Owner
Anshul Simlote

Revenue Recognition Dashboard

Total Recognized Revenue:
₹1,10,000.00

Total Invoiced Amount:
₹44,000.00

Milestone / Product Name	Recognized Revenue	Milestone Date	Invoice Status
Product Delivery	₹44,000.00	08-Feb-2025	Open
GenWatt Diesel 1000kW	₹40,000.00		
GenWatt Diesel 10kW	₹4,000.00		
Subscription 1	₹5,500.00	08-Mar-2025	Not Invoiced
GenWatt Diesel 1000kW	₹5,000.00		
GenWatt Diesel 10kW	₹500.00		
Subscription 2	₹5,500.00	08-Apr-2025	Not Invoiced
GenWatt Diesel 1000kW	₹5,000.00		
GenWatt Diesel 10kW	₹500.00		
Subscription 3	₹5,500.00	08-May-2025	Not Invoiced
Subscription 4	₹5,500.00	08-Jun-2025	Not Invoiced
Subscription 5	₹5,500.00	08-Jul-2025	Not Invoiced
Subscription 6	₹5,500.00	08-Aug-2025	Not Invoiced
Subscription 7	₹5,500.00	08-Sept-2025	Not Invoiced
Subscription 8	₹5,500.00	08-Oct-2025	Not Invoiced

Products (2)

GenWatt Diesel 1000kW
Quantity: 1.00
Sales Price: ₹1,00,000.00
Date:

GenWatt Diesel 10kW
Quantity: 2.00
Sales Price: ₹5,000.00
Date:

Invoices (1)

Product Delivery

Notes & Attachments (0)

9. Conclusion

This solution demonstrates a comprehensive approach to automating revenue recognition and invoicing in Salesforce while aligning with ASC 606 standards. By integrating standard objects, custom logic, triggers, scheduled batches, and a user-friendly LWC. Additionally, the roadmap for future enhancements—including per-product revenue recognition, refactoring into utility classes, and the implementation of a security library—positions the system well for future growth and improved maintainability.