Billing Management on Salesforce

Summer 2020 API Reference Guide

Table of Contents

About this Guide ... 4
What's New ... 5
Overview ... 8
API Supported Packages ... 8
API Standards and Development Platforms ... 9
Standards ... 9
Development Platforms ... 9
Field Types ... 9
Integrating Apttus with External Systems ... 11
Generating the Apttus Web Services WSDL ... 12
Connecting to Apttus ... 13
Billing APIs ... 15
Billing Services ... 15
Creating Credit Memo Documents ... 16
Creating Invoice Documents ... 17
Creating Invoices ... 19
Creating Invoices for Orders ... 24
Previewing Pending Usage Inputs ... 29
Processing Pending Usage Inputs ... 31
Processing Rated Usage Inputs ... 33
Updating Tax Calculations And Breakups on Credit Memos ... 34
Updating Tax Calculations and Breakups on Invoice ... 35
Forecasting Billing Schedules ... 36
Creating Credit and Rebill ... 38
Creating Billing Plans ... 40
Forecasting Billing Schedules and Billing Summaries ... 42
Creating Direct Credit Memos ... 45
Creating Billing Plan with Product Configuration ... 49
Forecasting Billing Schedules for Smart Cart ... 51
Applying a Late Fee ... 53
Cancelling a Late Fee ... 55
Creating Direct Credit Memos Asynchronously ... 56
Retrieving Session Key ... 60
Use Case for Usage-Based Billing ... 61

WSDL Services 62
Applying Credit Memos to Invoices 62
Applying Payments to Invoices 68
Adding a A/R Transaction - Deprecated 74
Adding Multiple A/R Transactions - Deprecated 76
REST Services 78
Billing Management REST APIs 78
Apttus Copyright Disclaimer 79

About this Guide

Apttus provides Application Programming Interfaces (API) for you to extend the features offered by Apttus Billing Management. These extensions add more functionality to the features available through configuration on Salesforce.

The Apttus Billing Management API Reference Guide describes the APIs provided to work with payments, usage inputs, credit memos, and invoice calculations.

What's New

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

Document	Topic	Description
Summer 2020	Integrating Apttus with External Systems	New topic for external integrations using Apttus Billing Management APIs.
	API Supported Packages	Updated the topic with the Summer 2020 package details.
	Creating Credit Memo Documents	Updated topic to include API signature.
	Creating Invoice Documents	Updated topic to include API signature.
	Creating Invoices	Updated topic to include API signature.
	Creating Invoices for Orders	Updated topic to include API signature.
	Previewing Pending Usage Inputs	Updated topic to include API signature.
	Processing Pending Usage Inputs	Updated topic to include API signature.
	Processing Rated Usage Inputs	Updated topic to include API signature.
	Updating Tax Calculations And Breakups on Credit Memos	Updated topic to include API signature.
	Updating Tax Calculations and Breakups on Invoice	Updated topic to include API signature.
	Forecasting Billing Schedules	Updated topic to include API signature.
	Creating Credit and Rebill	Updated topic to include API signature.

Document	Topic	Description
	Creating Billing Plans	Updated topic to include API signature.
	Forecasting Billing Schedules and Billing Summaries	Updated topic to include API signature.
	Creating Direct Credit Memos	Modified topic. Updated the API to include newly introduced isFullCredit parameter and API signature.
	Creating Billing Plan with Product Configuration	Updated topic to include API signature.
	Forecasting Billing Schedules for Smart Cart	Updated topic to include API signature.
	Applying a Late Fee	Updated topic to include API signature.
	Cancelling a Late Fee	Updated topic to include API signature.
	Creating Direct Credit Memos Asynchronously	New Topic. New API introduced in this release.
	Retrieving Session Key	New Topic. New API introduced in this release.
	Use Case for Usage-Based Billing	New topic. Use case to explain usage-based scenario using API execution.
	Creating Invoices for Orders - REST Service	Deleted topic. Code sample added in Billing Management REST APIs.
	Creating Invoices - REST Service	Deleted topic. Code sample added in Billing Management REST APIs.
	Processing Pending Usage Inputs - REST Service	Deleted topic. Code sample added in Billing Management REST APIs.
	Billing Management REST APIs	New topic. Added code sample in JSON format.

Document	Topic	Description
	Applying Credit Memos to Invoices	Updated topic to include API signature and SOAP Response/ Request XML for API integration.
	Applying Payments to Invoices	Updated topic to include API signature and SOAP Response/ Request XML for API integration.
Winter 2019	Creating Billing Plan with Product Configuration	New Topic. New API introduced in this release.
	Applying a Late Fee	New Topic. New API introduced in this release.
	Cancelling a Late Fee	New Topic. New API introduced in this release.
	Forecasting Billing Schedules for Smart Cart	New Topic. New API introduced in this release.
Summer 2019	Creating Direct Credit Memos	New Topic. New API introduced in this release.
Spring 2019	Forecasting Billing Schedules and Billing Summaries	New Topic. New API introduced in this release.
Winter 2018	Creating Billing Plans	New Topic. New API introduced in this release.
Summer 2018	Forecasting Billing Schedules	New Topic. New API introduced in this release.
	Creating Credit and Rebill	New Topic. New API introduced in this release.

Overview

- API Supported Packages
  The package names and version numbers required for the APIs to work seamlessly.
- API Standards and Development Platforms
  Apttus APIs are based on Salesforce APIs and use the same standards and platforms.
- Field Types
  Apttus APIs use a subset of the supported data and field types on Salesforce.
- Integrating Apttus with External Systems
  Information about integrating Apttus APIs with external systems.

API Supported Packages

The following packages and dependent packages are required for Billing Management APIs:

Product	Package Version (Name | Number)
Apttus Billing Management	6.2.238 | 6.238
Apttus Base Library	1.1.93 | 1.93
Apttus Quote/Proposal-Configuration Integration	12.1.0328 | 12.328
Apttus Configuration & Pricing	12.1.1787 | 12.1787
Apttus CPQ API	1.0.0080 | 10.80
Apttus Quote/Proposal-Asset Integration	6.5.0014 | 6.14
Apttus Proposal Management	10.1.0221 | 10.221
Apttus Contract Management	11.1.0543 | 11.543
Apttus Contract-Configuration Integration	12.1.0132 |12.132
Apttus Quote/Proposal-Contract Integration	9.1.0062 | 9.62

API Standards and Development Platforms

Apttus APIs are based on Salesforce APIs and use the same standards and platforms.

Standards

Name	Reference
Simple Object Access Protocol (SOAP) 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508
Web Service Description Language (WSDL) 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315
WS-I Basic Profile 1.1	http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html

Development Platforms

Apttus SOAP API works with standard SOAP development environments. For a list of compatible development platforms, see Salesforce Developer Force API details.

Field Types

Apttus APIs use a subset of the supported data and field types on Salesforce.

The following table lists the field types that Apttus supports. For a comprehensive list of all field types supported by Salesforce, see Salesforce Data Types.

Type	Description
Boolean	The **Boolean** field has a `true`(or 1) or `false`(or 0) value.
Data object	The **Data Object** field is an ID type and is represented by CPQ.nnDO in this document.

Type	Description
Date	The **Date** field contains date values only and do not contain relevant time values. Time in a date field is always set to midnight in the UTC time zone. If you want a timestamp you must use a **dateTime** field.
Decimal	The **Decimal** field provides an exact numeric value and you can arbitrarily size the precision and scale of the value.
ID	The **ID** field is an alphanumeric field that acts as the primary key for a specific record associated with an object. The ID value includes a three-character code that identifies which object the record is associated with. The ID for a specific record does not change. For some objects, this field may also be a reference type value, which contains the ID value for a related record. They are identified by field names ending in 'Id', such as priceListId. The ID field acts like foreign keys and their values can be changed using an update() call.
Integer	The **Integer** field contains whole numbers only. There are no digits after the decimal.
List	The **List** field includes a fixed set of values from which you must select a single value. Picklists are available as drop-down lists. If a picklist is unrestricted, the API does not limit entries to only currently active values.
String	The **String** field contains text and may have different length restrictions based on the data you store in the specific field. For instance, City may be limited to 50 characters, while AddressLine1 is limited to 255 characters.

Integrating Apttus with External Systems

Additional steps are required when you choose to integrate Billing Management on Salesforce with external applications, customer portals, or other critical business systems. Because Billing Management Web Services are hosted on Salesforce, you should familiarize yourself with the Salesforce SOAP API and processes surrounding integration and best practices detailed here: https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/sforce_api_quickstart_intro.htm

> **Best Practices**
>
> Apttus recommends that you work with Apttus Professional Services to design and implement your integration. Use this documentation for basic integration steps and to reference Billing Management Web Service calls.

The following basic steps are required to get started using the Apttus Billing Management Web Services API.

1. **Generate the Enterprise or Partner WSDL** – Integration with data stored in Salesforce requires you to first point your browser to the Salesforce Enterprise or Partner WSDL. This WSDL is generally provided by Apttus Professional Services. Refer to Salesforce Documentation for complete instructions on generating the Web Service WSDL.
2. **Generate the Apttus Web Services WSDL** – After you have connected to Apttus Web Service, go to your organization and download the WSDL for the appropriate Web Service.
3. **Import the WSDL Files Into Your Development Platform** – After you have generated the WSDL files, you can import them into your development platform. Apttus does not provide instructions for the import process. Refer to Salesforce documentation or documentation related to your development platform.
4. **Connect to Apttus** – Before you can begin using Billing Management Web Services, you must first authenticate to Apttus using the login() API.

- Generating the Apttus Web Services WSDL
  - To generate the Apttus Web Services WSDL
- Connecting to Apttus
  - To connect to Apttus Web Service using SoapUI

Generating the Apttus Web Services WSDL

Before you can import Apttus SOAP Web Service into your development or testing platform, you must generate and download the Apttus WSDL for the appropriate Web Service.

The example provided here uses SoapUI.

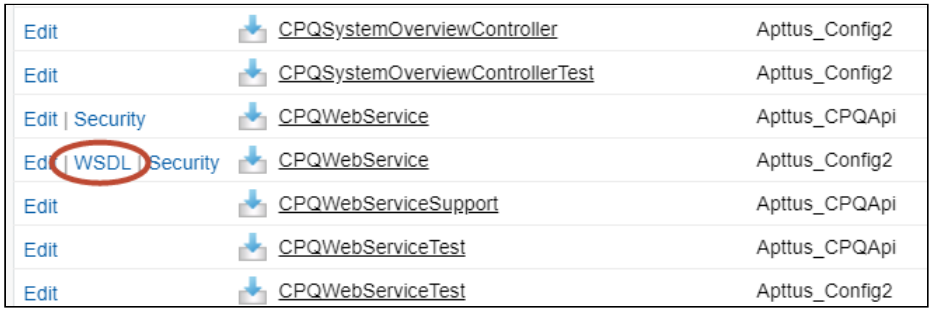
There a known bug in the WSDL Generator on Salesforce that does not include several field types, so it is recommended to update the WSDL file after you have generated it but *before* importing it into your development platform. You can find the details for any workaround tasks here:

- https://success.salesforce.com/issues_view?id=a1p3A000000eatxQAA&title=generated-wsdl-for-apex-webservices-is-malformed
- https://success.salesforce.com/issues_view?id=a1p300000008XKUAA2

When updating generated WSDL, make sure that the target namespace for any schema you add points to the correct Web Service (for example, schemas/class/Apttus_QPConfig/QPConfigWebService). If you are still having trouble, please ask Apttus Professional Services for a modified WSDL for the Web Services you are using.

To generate the Apttus Web Services WSDL

1. Log in to the Salesforce organization that contains you Apttus records and data (sandbox or production).
2. Go to **Setup > Develop > Apex Classes** (on Lightning, go to **Setup > Custom Code > Apex Classes**.
3. Find the Web Service you want to generate the WSDL for (for example, **BillingWebService**).
4. Click the **WSDL** link to generate the WSDL. The WSDL XML is generated and displayed in a new tab.



5. Right-click on the page and select **View Page Source**. Copy the XML content to any text editor.
6. Save the file with the extension **.wsdl**.
7. Open SoapUI (or wherever is required on your development platform).
8. Create a new SOAP project and import the Apttus Web Services WSDL. All methods under that Web Service are now available to call.

> Refer to the **Request/Response XML** section for any API in this reference to get the structure of the request and any prerequisite calls required for any API.

Connecting to Apttus

After you have downloaded the Enterprise or Partner WSDL, call the **login()** method to obtain a session ID from your org that you can use when calling Billing Management Web Services. After authenticating, you can use the same session ID until it either expires or your logout or login again.

The example provided here uses SoapUI, an API testing tool which can be downloaded for free here: https://www.soapui.org/.

Prerequisite: To authenticate with Apttus, please make sure to have your production or test org credentials on hand (username and password).

To connect to Apttus Web Service using SoapUI

1. Open SoapUI. Go to **File > New SOAP Project**.
2. Enter a name for the project.
3. Click **Browse**. Navigate to the saved Enterprise or Partner WSDL file that you downloaded and click **Open**.
4. Click **OK** to close the project window.
5. From the Navigation panel to the left, highlight the project folder and click to expand. Click to expand the **SoapBinding**. The list of methods that comprise the Enterprise or Partner services are displayed.
6. Scroll down and right click on **login**. Double-click on an existing **Request**. The request window opens in the SoapUI interface.

> If you are doing this for the first time, you need to right-click on the **login** method and select **New Request**.

7. Select and delete all content following the **<soapenv:Header>** tag and the **</soapenv:Header>** tag.
8. Enter the username for your org (must have appropriate privileges) between the **<urn:username>** and **</urn:username>** tags.
9. Enter the username for your org (must have appropriate privileges) between the **<urn:password>** and **</urn:password>** tags.

   The request should look like the following:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:urn="urn:partner.soap.sforce.com">
   <soapenv:Header>
   </soapenv:Header>
   <soapenv:Body>
      <urn:login>
         <urn:username>username@example.com</urn:username>
         <urn:password>password</urn:password>
      </urn:login>
   </soapenv:Body>
</soapenv:Envelope>
```

10. From the upper-left corner of the window, click the **Run** ( ) icon. The response is generated in the right-hand window.
11. Make note of the **serverURL** and the **sessionId** returned by the server. You will use the information to make Apttus Web Service calls.

Billing APIs

Apttus Billing Management APIs are categorized into:

Internal APIs

- Billing Services

External APIs

- REST Services
- WSDL Services

Billing Services

Apttus Billing APIs are based on Salesforce APIs and use the same standards and platforms.

You can invoke Apttus Billing APIs from the following command:

```
Apttus_Billing.BillingService.<Name of the Function>
where the name of the function is API Name and it parameters.
```

Here is a list of APIs along with their parameters:

- Creating Credit Memo Documents
- Creating Invoice Documents
- Creating Invoices
- Creating Invoices for Orders
- Previewing Pending Usage Inputs
- Processing Pending Usage Inputs
- Processing Rated Usage Inputs
- Updating Tax Calculations And Breakups on Credit Memos
- Updating Tax Calculations and Breakups on Invoice
- Forecasting Billing Schedules
- Creating Credit and Rebill
- Creating Billing Plans
- Forecasting Billing Schedules and Billing Summaries
- Creating Direct Credit Memos
- Creating Billing Plan with Product Configuration
- Forecasting Billing Schedules for Smart Cart
- Applying a Late Fee

- Cancelling a Late Fee
- Creating Direct Credit Memos Asynchronously
- Retrieving Session Key
- Use Case for Usage-Based Billing

Creating Credit Memo Documents

createCreditMemoDocuments API is used to generate credit memo documents.

API	Signature
createCreditMemoDocuments	*static void createCreditMemoDocuments(List creditMemoIds, String creditMemoTemplateName)*

It generates credit memo documents for the given list of credit memo IDs.

You can generate credit memo documents in the following formats:

- PDF
- DOCX
- DOC
- RTF

It accepts a list of Credit Memo IDs and a valid credit memo template name as input. If you do not specify a valid template name or pass **null** as a value, Billing Management system uses the default template provided at the Account or the Account Location.

If you call this API from a batch or a scheduled job, it can process only one CreditMemo ID. Otherwise, if you call this API from a non-batch or a non-scheduled job, it can process up to 10 CreditMemos.

Request			
Field	**Type**	**Required?**	**Description**
creditMemoIds	List <Id>	Yes	IDs of credit memo
creditMemoTemplateName	String	No	Name of the credit memo template

Code Sample

```
Account testAccount = new Account(Name = 'Test Account');
insert testAccount;
CreditMemo__c testCreditMemo = new CreditMemo__c(BillToAccountId__c = testAccount.Id,
 CreditAmount__c = 40.0,
 Status__c = CreditMemo.STATUS_DRAFT);
insert testCreditMemo;
Apttus__APTS_Template__c cmTemplate = new Apttus__APTS_Template__c(Name = 'Default',
 Apttus__IsActive__c = true,
 Apttus__Type__c = 'Credit Memo');
insert cmTemplate;

List<Id> cmIdList = new List<Id> {testCreditMemo.Id};
//create Credit Memo Documents
Apttus_Billing.BillingService.createCreditMemoDocuments(cmIdList, cmTemplate.Name);
```

Creating Invoice Documents

createInvoiceDocuments API creates invoice documents for a given list of Invoice. It accepts a list of invoice IDs as input and produces invoice documents for each corresponding invoice ID. It also accepts Invoice Template Name as a parameter if you want to create Invoice Documents using a specific template.

You can generate invoice documents in the following formats:

- PDF
- DOCX
- DOC
- RTF

The document is generated in the format set at the **Billing Preference** in *Invoice Output Format* field.

If you call this API from a batch or a scheduled job, it can process only one Invoice ID. Otherwise, if you call this API from a non-batch or a non-scheduled job, it can process up to 10 Invoices.

There are two ways of creating Invoice Documents:

- Using the default invoice template
- Using a specific invoice template

API	Signature
createInvoiceDocuments	*static void createInvoiceDocuments(List invoiceIds)*

This API uses the defualt Invoice Template specified on the Account or Account Location.

Request			
Field	**Type**	**Required?**	**Description**
invoiceIds	List <Id>	Yes	List of Invoice Ids

Code Sample

```
List<ID> invoiceIDs = new List<ID>();
//invoiceIDs.add('a4t1I000000H5hM');
Apttus_Billing.BillingService.createInvoiceDocuments(invoiceIDs);
```

API	Signature
createInvoiceDocuments	static void createInvoiceDocuments(List invoiceIds, String invoiceTemplateName)

This API is used to create invoice documents using a specific invoice template.

If you do not specify a valid template name or pass null as a value, Billing Management System uses the default template specified on the Account or Account Location.

Request			
Field	**Type**	**Required?**	**Description**
invoiceIds	List <Id>	Yes	List of Invoice Ids
invoiceTemplateName	String	Yes	Invoice Template Name

Code Sample

```
List<ID> invoiceIDs = new List<ID>();
//invoiceIDs.add('a4t1I000000H5hM');
//String invoiceTemplateName = 'myTestTemplate';
Apttus_Billing.BillingService.createInvoiceDocuments(invoiceIDs,
invoiceTemplateName);
```

Creating Invoices

createInvoices API is used to create Invoices automatically. It accepts a list of Account IDs and creates Invoices for each AccountID.

API	Signature
createInvoices	*static void createInvoices(Set billToAccountIds, Datetime targetDateTime, Date invoiceDate)*

This API is used to create invoices for given Account IDs. It accepts a Set of Account IDs, invoiceDate, and targetDateTime as input parameters. It creates invoices for all orders with billing schedules having status as pending billing and end date less than targetDateTime. All the new invoices are created with Invoice Date as the value mentioned in invoiceDate.

Request			
Field	**Type**	**Required?**	**Description**
billToAccountIds	Set <Id>	Yes	Set of bill to Account IDs
targetDateTime	Datetime	Yes	Process through Date
invoiceDate	Date	Yes	Invoice Creation Date

Code Sample

```
Set<ID> billToAccountIDs = new Set<ID>();
Date invoiceDate = Date.newInstance(2017, 1, 1);
Date myDate = Date.newInstance(2017, 1, 1);
Time myTime = Time.newInstance(3, 3, 3, 0);
DateTime targetDateTime = DateTime.newInstance(myDate, myTime);
Apttus_Billing.BillingService.createInvoices(billToAccountIDs, invoiceDate,
targetDateTime);
```

API	Signature
createInvoices	*static void createInvoices(Set billToAccountIds, Apttus_Billing.InvoiceCreationOptions options)*

This API is used to create invoices for given Account IDs. It accepts a set of Account IDs and a class containing invoice creation options. the class contains options that influence the creation of an Invoice such as Invoice Date, Invoice Through Date and Auto Approve.

Request			
Field	**Type**	**Required?**	**Description**
billToAccountIds	Set <Id>	Yes	Set of bill to Account IDs
options	Apttus_Billing.InvoiceCreationOptions	Yes	Class holding the Invoice Creation Options

DataObject - Billing.InvoiceCreationOptions		
Field	**Type**	**Description**
abort	Boolean	If the process is aborted.
autoApprove	Boolean	Mandatory Constructor. IfautoApprovefor Invoice is set to true.
autoApproveAmount	Decimal	Mandatory Constructor. The amount provided for auto approval.
autoApproveCreditMemo	Boolean	Mandatory Constructor. If the autoApproveCreditMemo is set to true

DataObject - Billing.InvoiceCreationOptions		
Field	**Type**	**Description**
autoApproveOperator	String	Mandatory Constructor. The selected autoApproveOperator such as Greater than, Less than etc.
invoiceDate	Date	Mandatory Constructor. The Invoice Date.
invoiceDateType	String	The Invoice Date Type such as month or year.
invoiceRun	Apttus_Billing __InvoiceRun__c	The Invoice Run Object API Name.
invoiceRunResult	Apttus_Billing __InvoiceRunResult__c	The Invoice Run Result Object API Name.
numberAccountsProcessed	Integer	The number of accounts processed as part of the Invoice Runs.
numberCreditMemos Genenerated	Integer	The number of Credit Memos generated.
numberInvoicesGenenerated	Integer	The number of Invoices generated.
numberOfAccountsThat GeneratedCreditMemos	Integer	Number of accounts for which Credit Memos are generated.
numberOfAccountsThat GeneratedInvoices	Integer	Number of accounts for which Invoices are generated.
numberOfAutoApprovedInvoices	Integer	Number of Invoices that are auto-approved.
numberOfCreditMemos AutoApproved	Integer	Number of Credit Memos that are auto-approved.

DataObject - Billing.InvoiceCreationOptions		
Field	**Type**	**Description**
numberOfSuppressedInvoices	Integer	Number of Invoices that are suppressed.
processThruDate	Date	Mandatory Constructor.The Process through date for invoice
splitInvoicesByOrder	Boolean	Mandatory Constructor. If Split Invoices By Order is set to true.
suppressInvoicesAmount	Decimal	Mandatory Constructor.The amount till which you want to suppress invoices.
suppressInvoicesOperator	String	Mandatory Constructor. .The selected Suppress Invoices operator such as Greater than, Less than etc.
taxCallbackStatus	String	The status of Tax Callback.
credtiMemoCreationOption	String	The credit memo creation option. For information on credit memo creation options, refer to Generating Credit Memos from Invoice Run.
autoSendEmailforInvoice	Boolean	Set it to true to auto-send an email once the invoice is generated.
autoSendEmailforCreditMemo	Boolean	Set it to true to auto-send an email once the credit memo is generated
InvoiceOverrideTemplate	String	Name of the invoice template to override the default invoice template
CreditMemoOverrideTemplate	String	Name of the credit memo template name to override the default credit memo template

⚠ Only the following values are currently supported for createInvoice API:

- **autoSendEmailforInvoice**: false
- **autoSendEmailforCreditMemo:** false
- **InvoiceOverrideTemplate:** NULL
- **CreditMemoOverrideTemplate:** NULL

Code Sample

```
Set<ID> billToAccountIDs = new Set<ID>();
Date invoiceDate = Date.newInstance(2017, 1, 1);
Date processThruDate = Date.newInstance(2017, 1, 1);
Boolean autoApprove = true;
Boolean splitInvoicesByOrder = true;
String autoApproveOperator = 'Greater than';
Decimal autoApproveAmount = 5.00 ;
String suppressInvoicesOperator = 'Less than';
Decimal suppressInvoicesAmount = 2.00;
Boolean autoApproveCreditMemo = true;
Apttus_Billing.InvoiceCreationOptions options = new
 Apttus_Billing.InvoiceCreationOptions(
invoiceDate,
processThruDate,
autoApprove,
splitInvoicesByOrder,
autoApproveOperator,
autoApproveAmount,
suppressInvoicesOperator,
suppressInvoicesAmount,
autoApproveCreditMemo);

Apttus_Billing.BillingService.createInvoices(billToAccountIDs, options);
```

Code Sample

Use this code sample to provide credit memo creation options while generating invoices through the createInvoices API.

```
Set<ID> billToAccountIDs = new Set<ID>();
Date invoiceDate = Date.newInstance(2018,1,1);
Date processThruDate = Date.newInstance(2018,9,9);
Boolean autoApprove = true;
Boolean splitInvoicesByOrder = false;
String autoApproveOperator = 'Greater than';
Decimal autoApproveAmount = 20.00 ;
String suppressInvoicesOperator = 'Less than';
Decimal suppressInvoicesAmount = 2.00;
Boolean autoApproveCreditMemo = true;
String credtiMemoCreationOption = 'Single Credit Memo for all Negative Schedules per
Invoice';
Boolean autoSendEmailForInvoice = false;
Boolean autoSendEmailForCreditMemo = false;
String InvoiceOverrideTemplate = NULL;
String CreditMemoOverrideTemplate = NULL;
billToAccountIDs.add('0017F00000tPMvL');
InvoiceCreationOptions
options = new InvoiceCreationOptions( invoiceDate,
processThruDate,
autoApprove,
splitInvoicesByOrder,
autoApproveOperator,
autoApproveAmount,
suppressInvoicesOperator,
suppressInvoicesAmount,
autoApproveCreditMemo,
credtiMemoCreationOption,
autoSendEmailForInvoice,
autoSendEmailForCreditMemo,
InvoiceOverrideTemplate,
CreditMemoOverrideTemplate);
BillingService.createInvoices(billToAccountIDs,options);
```

Creating Invoices for Orders

createInvoicesforOrder API is used to create invoices for given orders. It accepts a List of order IDs as input and produces invoices for each order ID.

API	Signature
createInvoicesForOrder	*static List createInvoicesForOrder(Id orderId, Datetime targetDateTime, Date invoiceDate)*

This API is used to create invoices for given order ID. It accepts the order ID, targetDateTime and invoiceDate as input parameters. It creates invoices for the billing schedules that have status as pending billing and end date less than targetDateTime. All the new invoices are created with Invoice Date as the value mentioned in invoiceDate.

Request			
Field	**Type**	**Required?**	**Description**
orderId	ID	Yes	Order Id
targetDateTime	Datetime	Yes	Invoice Process Through Date
invoiceDate	Date	Yes	Invoice Creation Date

Response		
Field	**Type**	**Description**
Apttus_Config2__Invoice_c	List of Invoice objects	Invoices created for each order ID

Code Sample

```
ID orderID = new ID();
Date invoiceDate = Date.newInstance(2017, 1, 1);
Date myDate = Date.newInstance(2017, 1, 1);
Time myTime = Time.newInstance(3, 3, 3, 0);
DateTime targetDateTime= DateTime.newInstance(myDate, myTime);
List <Invoice_c> invoices =
Apttus_Billing.BillingService.createInvoicesForOrder(orderID, targetDateTime,
invoiceDate);
```

API	Signature
createInvoicesForOrder	*static List createInvoicesForOrder(Id orderId, Apttus_Billing.InvoiceCreationOptions options)*

This API is used to create invoices for the given order ID. It accepts the order IDs and a class containing invoice creation options. the class contains options that influence the creation of an Invoice such as Invoice Date, Invoice Through Date and Auto Approve. It returns a list of invoices.

Request			
Field	**Type**	**Required?**	**Description**
orderId	ID	Yes	Order Id
options	Apttus_Billing.InvoiceCreationOptions	Yes	Class holding the Invoice Creation Options

Response		
Field	**Type**	**Description**
Apttus_Config2__Invoice_c	List of Invoice objects	Invoices created for each order ID

DataObject - Billing.InvoiceCreationOptions			
Field	**Type**	**Option Type**	**Description**
abort	Boolean	Output	If the process is aborted.
autoApprove	Boolean	Input	Mandatory Constructor. If autoApprove for Invoice is set to true.
autoApproveAmount	Decimal	Input	Mandatory Constructor. The amount provided for auto approval.
autoApproveCreditMemo	Boolean	Input	Mandatory Constructor. If autoApproveCreditMemo is set to true
autoApproveOperator	String	Input	Mandatory Constructor. The selected autoApproveOperator such as Greater than, Less than etc.
invoiceDate	Date	Input	Mandatory Constructor. The Invoice Date.
invoiceDateType	String	Input	The Invoice Date Type such as month or year.
invoiceRun	Apttus_Billing__InvoiceRun__c	Input	The Invoice Run Object API Name.
invoiceRunResult	Apttus_Billing__InvoiceRunResult__c	Input/Output	The Invoice Run Result Object API Name.
numberAccountsProcessed	Integer	Output	The number of accounts processed as part of the Invoice Runs.
numberCreditMemos Genenerated	Integer	Output	The number of Credit Memos generated.

DataObject - Billing.InvoiceCreationOptions			
Field	**Type**	**Option Type**	**Description**
numberInvoicesGenerated	Integer	Output	The number of Invoices generated.
numberOfAccountsThat GeneratedCreditMemos	Integer	Output	Number of accounts for which Credit Memos are generated.
numberOfAccountsThat GeneratedInvoices	Integer	Output	Number of accounts for which Invoices are generated.
numberOfAutoApprovedInvoices	Integer	Output	Number of Invoices that are auto-approved.
numberOfCreditMemos AutoApproved	Integer	Output	Number of Credit Memos that are auto-approved.
numberOfSuppressedInvoices	Integer	Output	Number of Invoices that are suppressed.
processThruDate	Date	Input	Mandatory Constructor. The Process through date for invoice
splitInvoicesByOrder	Boolean	Input	Mandatory Constructor. If Split Invoices By Order is set to true.
suppressInvoicesAmount	Decimal	Input	Mandatory Constructor. The amount till which you want to suppress invoices.
suppressInvoicesOperator	String	Input	Mandatory Constructor.The selected Suppress Invoices operator such as Greater than, Less than etc.

DataObject - Billing.InvoiceCreationOptions			
Field	Type	Option Type	Description
taxCallbackStatus	String	Output	The status of Tax Callback.

Code Sample

```
ID orderID = new ID();
Date invoiceDate = Date.newInstance(2017, 1, 1);
Date processThruDate = Date.newInstance(2017, 1, 1);
Boolean autoApprove = true;
Boolean splitInvoicesByOrder = true;
String autoApproveOperator = 'Greater than';
Decimal autoApproveAmount = 5.00 ;
String suppressInvoicesOperator = 'Less than';
Decimal suppressInvoicesAmount = 2.00;
Boolean autoApproveCreditMemo = true;

Apttus_Billing.InvoiceCreationOptions options = new
 Apttus_Billing.InvoiceCreationOptions(
invoiceDate,
processThruDate,
autoApprove,
splitInvoicesByOrder,
autoApproveOperator,
autoApproveAmount,
suppressInvoicesOperator,
suppressInvoicesAmount,
autoApproveCreditMemo);

Apttus_Billing.BillingService.createInvoicesForOrder(orderID, options);
```

Previewing Pending Usage Inputs

previewPendingUsageInputs API is used to preview the rating amount of usage inputs without altering the Billing Schedules, Usage Schedules, and Usage Inputs. You can rate a maximum of 2000 Usage Inputs at a time.

API	Signature
previewPendingUsageInputs	*static Map previewPendingUsageInputs(Set usageInputIds)*

The API accepts a set of Usage Input IDs as input. It validates the number of Usage Inputs to process. If the number of Usage Input Ids entered is less than 2000, it returns a map containing the rating amount for each Usage Input ID. If the number of Usage Input IDs given is more than 2000, Billing Management system throws an error.

Request			
Field	**Type**	**Required?**	**Description**
usageInputIds	Set <Id>	Yes	Set of Usage Input IDs

Response		
Field	**Type**	**Description**
mapRatedUsagInputIds	Map	Map containing Usage Inputs for each UsageInputId

Code Sample

```
Set<ID> usageInputIDs = new Set<ID>();
Map<ID, Apttus_Billing__UsageInput__c> mapRatedUsageInputs =
Apttus_Billing.BillingService.previewPendingUsageInputs(usageInputIDs);
for(Apttus_Billing__UsageInput__c usageInput : mapRatedUsageInputs.values()) {
                //Use usageInput.Apttus_Billing__RatedAmount__c
}
```

Processing Pending Usage Inputs

processPendingUsageInput API is used to process Usage Inputs. On completion of the batch job, the user receives a batch job status email.

You can process the usage inputs in two ways:

- Process all the loaded usage inputs
- Process an explicit set of usage inputs

API	Signature
processPendingUsageInput	*static void processPendingUsageInput()*

This API processes all the usage inputs with status as 'Loaded'.

Code Sample

```
Apttus_Billing.BillingService.processPendingUsageInput();
```

API	Signature
processPendingUsageInput	*static void processPendingUsageInput(Set usageInputIDs)*

This API processes usage inputs for all the usage inputs IDs with status as 'Loaded'. It accepts a Set of Usage Input IDs as input and checks the Usage Input Status for each Usage Input ID. It processes all the usage inputs with staus as 'Loaded'.

Request			
Field	**Type**	**Required?**	**Description**
usageInputIDs	Set <Id>	No	Set of usage input IDs

Code Sample

```
Set<Id> usageInputIds = new Set<ID>{usageInput1.Id, usageInput2.Id};
Apttus_Billing.BillingService.processPendingUsageInput(usageInputIds);
```

API	Signature
processPendingUsageInput	*static void processPendingUsageInput(Set usageInputIDs, Boolean processSynchronously)*

This API processes usage inputs for all the usage inputs IDs with status as 'Loaded'. It accepts a Set of Usage Input IDs as input and checks the Usage Input Status for each Usage Input ID. It processes all the usage inputs with staus as 'Loaded'.If processSynchronously is set to true, the usage inputs are processed synchronously without any batch job. You can process only a maximum of 1000 usage inputs. If processSynchronously is set to false, the API will schedule a batch job to process all the given usage inputs.

Request			
Field	**Type**	**Required?**	**Description**
usageInputIDs	Set <Id>	No	Set of usage input IDs
processSynchronously	Boolean	No	Indicates whether to process the Usage Inputs synchronously or asynchronously. The default value is false.

Code Sample

```
// To process Usage Inputs Synchronously
Set<ID> usageInputIDs = new Set<ID>(); //Max it can be 1000 IDs
Apttus_Billing.BillingService. processPendingUsageInput(usageInputIDs, true);
// To process Usage Inputs Asynchronously
Set<ID> usageInputIDs = new Set<ID>(); //Max it can be any number of IDs
Apttus_Billing.BillingService. processPendingUsageInput(usageInputIDs, false);
```

Processing Rated Usage Inputs

This API is used to unrate the processed usage input. It also reverts the Amount and Quantity from Billing Schedules, Usage Schedules, and Revenue Fee Schedules.

API	Signature
processRatedUsageInput	*static void processRatedUsageInput(Set usageInputIDs)*

It accepts a Set of Usage Input IDs as input. For all entered Usage Input IDs, it unrates all the usage Inputs and reverts the amount and quantity from the related billing schedules, usage scheules, and revenue fee schedules. On completion of the batch job, the user receives a batch job status email.

Request			
Field	**Type**	**Required?**	**Description**
usageInputIDs	Set <Id>	Yes	Set of usage input IDs

Code Sample

```
Set<Id> usageInputIds = new Set<ID>{usageInput1.Id, usageInput2.Id};
Apttus_Billing.BillingService.processRatedUsageInput(usageInputIds);
```

Updating Tax Calculations And Breakups on Credit Memos

updateCreditMemoTaxCalculationsAndBreakups API calculates Tax Amount and populates it on the Credit Memo.

API	Signature
updateCreditMemoTaxCalculationsAndBreakups	*static void updateCreditMemoTaxCalculationsAndBreakups(Set Creditmemolds, Boolean autoTransitionFromPendingApprovedToApproved)*

This API calculates Tax Amount and Tax Breakups for credit memos with status as 'Draft' or 'Pending Approved'. After tax computation, it updates the Tax Amount on each credit memo line item and creates or updates Credit Memo Line Item Tax Breakups. It then calculates the Total Tax Amount for the credit memo and creates or updates the Credit Memo Tax Breakups.

If autoTransitionFromPendingApprovedToApproved is set to true, it updates the status of Credit Memo from Pending Approved to Approved.

> ⚠ The billing administrator must register a Tax Callback class for updateCreditMemoTaxCalculationsAndBreakups API to work. For details on how to register a Tax Callback class, refer to Custom Settings for Tax.

Request			
Field	**Type**	**Required?**	**Description**
creditMemoIDs	Set <Id>	Yes	Set of Credit Memo Ids
autoTransitionFromPendingApprovedToApproved	Boolean	Yes	If set to true, the API updates the status of Credit Memo from Pending Approved to Approved. If set to false, the credit memo status is not updated.

Code Sample

The code sample below helps you calculate and fetch Tax Amount for credit memos.

```
Set<ID> creditMemoIDs = new Set<ID>();
//[{creditMemos[0].Id, creditMemos[1].Id};]
Apttus_Billing.BillingService.updateCreditMemoTaxCalculationsAndBreakups(creditMemoIDs, true);
```

Updating Tax Calculations and Breakups on Invoice

updateInvoiceMemoTaxCalculationsAndBreakups API calculates Tax Amount and populates it on the Invoice.

API	Signature
updateInvoiceTaxCalculationsAndBreakups	*static void updateInvoiceTaxCalculationsAndBreakups(Set InvoiceIds, Boolean autoTransitionFromPendingApprovedToApproved)*

This API calculates Tax Amount and Tax Breakups for invoices with status as 'Draft' or 'Pending Approved'. After tax computation, it updates the Tax Amount on each invoice line item and creates or updates Invoice Line Item Tax Breakups. It then calculates the Total Tax Amount for the invoice and creates or updates the Invoice Tax Breakups.

If autoTransitionFromPendingApprovedToApproved is set to true, it updates the status of the invoice from Pending Approved to Approved.

> ⚠ The billing administrator must register a Tax Callback class for updateInvoiceTaxCalculationsAndBreakups API to work. For details on how to register a Tax Callback class, refer to Custom Settings for Tax.

Request			
Field	**Type**	**Required?**	**Description**
invoiceIDs	List <Id>	Yes	List of Invoice Ids

Request			
Field	**Type**	**Required?**	**Description**
autoTransitionFromPendingApprovedToApproved	Boolean	Yes	If set to true, the API updates the status of the Invoice from Pending Approved to Approved. If set to false, the invoice status is not updated.

Code Sample

The code sample below helps you calculate and fetch Tax Amount for invoices.

```
Set<ID> invoiceIDs = new Set<ID>();
invoiceIDs.add('invoiceID')
Apttus_Billing.BillingService.updateInvoiceTaxCalculationsAndBreakups(invoiceIDs,true
);
```

Forecasting Billing Schedules

Forecast Billing Schedules API is used to generate and display forecast billing schedules for a Quote/Proposal. This API accepts the proposal ID as input parameter and returns the list of forecast billing schedules.

API	Signature
retrieveForecastedBillingSchedules	*static List* retrieveForecastedBillingSchedules(Id proposalId)

Forecast Billing Schedules API is used to generate and display forecast billing schedules for a Quote/Proposal. This API can be invoked by the implementation team on acceptance of the quote/proposal.

If the forecast billing schedules are already generated for the given proposal and no changes are made to the product configurations, the API displays the existing forecast billing schedules. If you perform any asset-based operations or change the product

configuration, the generated forecast billing schedules are deleted and new forecast billing schedules are generated to reflect the changes made to the product or the asset.

⚠ Forecast billing schedule functionality is not supported for:

- Evergreen products
- Quote/Proposal associated with a billing plan
- Informational line items of bundle and option products

Request			
Field	**Type**	**Required?**	**Description**
ProposalID	ID	Yes	The proposal ID

Response		
Field	**Type**	**Description**
forecastedbillingschedules	List	List of forecast billing schedules generated for the given proposal ID.

```
 If the Forecasted Billing Schedules have already been created and are "current" then
return the existing
 * Forecasted Billing Schedules, otherwise delete the existing "out of sync"
 Forecasted Billing Schedules and to
 * the following.
 *
 *  For each Line Item in the "active" Product Configuration create an in-memory
Order Line Item (and other
 * relevant in-memory objects) and call the BSM (Billing Schedule Manager) to
generate the resultant set of
 * _forecasted_ Billing Schedules.  When finished persist the aggregate list of
Forecasted Billing Schedules and
 * and return them
 *
 * @param ProposalId The Id of the Proposal (Quote) to persist and retrieve
Forecasted Billing Schedules for.
 *
 * @return The list of persisted Forecasted Billing Schedules, which will be sorted
by "Ready for Invoice Date".
 */
 global static List<ForecastedBillingSchedule__c>
retrieveForecastedBillingSchedules(ID ProposalId );
```

Creating Credit and Rebill

Credit and Rebill functionality allow you to credit an entire invoice and rebill it.

API	Signature
createCreditRebill	*static Map createCreditRebill(Id invoiceId, Boolean autoApproveCreditMemo)*

This API accepts the invoice ID and the option to Auto Approve the credit memo as input parameters.

Request			
Field	**Type**	**Required?**	**Description**
invoiceId	ID	Yes	ID of the invoice
autoApproveCreditMemo	Boolean	Yes	If set to true, credit memos are auto-approved

The API returns a Map<String, Value> as a response parameter.

Response		
Field	**Type**	**Description**
Result	Map	The result map contains the success string and a value

If the API executes successfully, the value contains the following keys:

- success - set to true
- creditMemoId - the ID of the created credit memo

If the API is not executed successfully, the value contains the following keys:

- success - set to false
- errorMessage - contains the error message
- stackTrace - contains the stackTrack of the API call

Code Sample

```
/**createCreditRebill This method creates a Credit Memo
@param  invoiceId
The id of the invoice which has to be credited and rebilled.
@param  autoApproveCreditMemo
Tells whether the credit memo should to be approved during creation
*
@return
Returns a map. If creation succeeds, returns success=true and creditMemoId
*
In case of error, success=false, errormessage and stackTrace
*/
global static Map<String, Object> Apttus_Billing.BillingService.createCreditRebill(Id
invoiceId,Boolean autoApproveCreditMemo);
```

Creating Billing Plans

createBillingPlan API is used to create a billing plan. It creates a billing plan and billing plan line items for the given proposal ID.

API	Signature
createBillingPlan	*static Apttus_Billing.BillingService.BillingPlanAPIResponse createBillingPlan(Id proposalId, Id billingPlanTemplateId, Set proposalLineItemIds)*

This API accepts Proposal ID, Billing Plan Template ID, and Proposal Line Item IDs as input parameters. It creates a billing plan and billing plan line items for the given proposal ID. Created billing plan is applied to the list of provided proposal IDs. If you do not provide any proposal line item IDs as input parameters, the created billing plan is applied to all the proposal line items. This API skips the proposal line items that are already linked to a billing plan.

For a billing plan template with plan type as Milestones, createBillingPlan API also creates milestones.

Request			
Field	**Type**	**Required?**	**Description**
proposalID	ID	Yes	ID of the proposal
billingPlanTemplateId	ID	Yes	IDof the Billing Plan Template
proposalLineItemIds	Set <Id>	No	Set of proposal Ids

Response		
Field	**Type**	**Description**
isSuccess	Boolean	Returns true if the API is executed correctly
errorMessage	String	Error Message if the API is not executed successfully.
billingPlanId	ID	Id of the created Billing Plan

Code Sample

```
Id proposalId = 'a2G1N000002RpRZUA0';
Id billingPlanTemplateId = 'a3D1N000000blZc';
Set<Id> proposalLineItemIds = new Set<Id>('a2E1N000001dHox', 'a2E1N000001eHox');
Apttus_Billing.BillingService.BillingPlanAPIResponse response;
// Call API without proposal line item ids parameter
response = Apttus_Billing.BillingService.createBillingPlan(proposalId,
billingPlanTemplateId);

// Call API with proposal line item ids parameter
response = Apttus_Billing.BillingService.createBillingPlan(proposalId,
billingPlanTemplateId, proposalLineItemIds);
```

createBillingPlan API is not supported for Evergreen Billing and Usage-based products.

Forecasting Billing Schedules and Billing Summaries

Forecast Billing Schedules and Summaries API is used to generate and display forecast billing schedules and forecast billing summaries for a Quote/Proposal. This API accepts the proposal ID and the option to extend the end date with renewal term as input parameters and returns the list of forecast billing schedules and forecast billing summaries.

API	Signature
retrieveForecastedBillingSchedulesAndSummaries	*static Apttus_Billing.ForecastedResults retrieveForecastedBillingSchedulesAndSummaries(Id proposalId, Boolean extendEndDateWithRenewalTerm)*

Forecast Billing Schedules API is used to generate and display forecast billing schedules and forecast billing summaries for a Quote/Proposal. This API can be invoked by the implementation team on acceptance of the quote/proposal.

If the forecast billing schedules or forecast billing summaries are already generated for the given proposal and no changes are made to the product configurations, the API displays the existing forecast billing schedules. If you perform any asset-based operations or change the product configuration, the generated forecast billing schedules and billing summaries are deleted and new forecast billing schedules and billing summaries are generated to reflect the changes made to the product or the asset.

Forecast billing schedule functionality is not supported for:

- Evergreen products
- Quote/Proposal associated with a billing plan
- Informational line items of bundle and option products

Request			
Field	**Type**	**Required?**	**Description**
proposalID	ID	Yes	The proposal ID
extendEndDateWithRenewal	Boolean	Yes	If extendEndDateWithRenewal is set to true, then for each proposal line item with Auto-Renew set to True and the Auto Renewal Type set to Fixed. the end date is extended based on the renewal term.

Response		
Field	**Type**	**Description**
forecastedSchedules	List	List of forecast billing schedules generated for the given proposal ID.
forecastedSummaries	List	List of forecast billing summaries generated for the given proposal ID.

```
/**
 * Class used to support forecasting of Billing Schedules and Invoices.
 */
global with sharing class ForecastedResults {

    global List<ForecastedBillingSchedule__c> forecastedSchedules { get; private
 set; }
    public List<ForecastedBillingSummary__c> forecastedSummaries { get; private
 set; }

    public ForecastedResults() {
        this.forecastedSchedules = new List<ForecastedBillingSchedule__c>();
        this.forecastedSummaries = new List<ForecastedBillingSummary__c>();
    }
}

/**
 * Create the list of Forecasted Billing Schedules by leveraging the BSM to
"forecast" each of
 * the Proposal Line Items belonging to the "active" Product Configuration of the
specified Proposal.
 * From the Forecasted Billing Schedules generate the list of Forecasted Billing
Summaries.
 *
 * If the Forecasted Billing Schedules/Summaries have already been created and are
"current",
 * return the existing forecasted Schedules and Summaries.  Otherwise delete the
existing
 * "out of sync" forecasted Schedules and Summaries and use the Product Configuration
of the
 * specified Proposal to re-generate (and persist) them.
 *
 * @param ProposalId The Id of the Proposal (Quote) to use when retrieving (and
possibly
 *   generating) the lists of forecasted Billing Schedules and Summaries.
 * @param extendEndDateWithRenewalTerm If the flag is true, then extend the End Date
 *   (based on the Renewal Term) for all Line Items that have Auto Renew set to True
and
 *   the Auto Renewal Type set to "Fixed".
 *
 * @return The list of persisted forecasted Billing Schedules and Billing Summaries.
The forecasted
```

```
 *   Billing Schedules will be sorted by "Ready for Invoice Date" and the forecasted
Billing Summaries
 *   will be sorted by Invoice Date and Summary Number.
 */
global static ForecastedResults retrieveForecastedBillingSchedulesAndSummaries(
    ID proposalId, Boolean extendEndDateWithRenewalTerm) {

    return new ForecastedResults();
}
```

Creating Direct Credit Memos

createDirectCreditMemos API is used to create credit memos for a list of invoices.

API	Signature
createDirectCreditMemos	*static* *List* createDirectCreditMemos(List creditMemoInputs)

directCreditMemoResult createDirectCreditMemos(directCreditMemoInputs)

This API is used to create credit memos for an invoice. It accepts a Set of Account IDs, invoiceDate, and targetDateTime as input parameters. For each DirectCreditMemoInput, a credit memo is created in the Draft Status. You can auto-approve a credit memo line item and auto-apply it to an invoice. If you have a Tax Callback registered, this API also calculates Tax and creates Tax breakups for a credit memo line item.

Request - DirectCreditMemoInput			
Field	**Type**	**Required?**	**Description**
invoiceId	ID	Yes	The Id of the approved Invoice the credit will be drawn from.
reasonCode	String	Yes	The reason for creating the Credit Memo. Must be null or a valid pick-list value for the Reason Code field declared of the Credit Memo object.

Request - DirectCreditMemoInput			
Field	**Type**	**Required?**	**Description**
isFullCredit	Boolean	No	If you set isFullCredit as true, the entire available credit amount of all the invoice line items is set as the credit amount. If you set the value as false, credit memo amount is calculated based on the values you provide in creditMemoLineItemInputs field. By default, it is set as false.
creditMemoLineItemInputs	List<DirectCreditMemoLineItemInput>	No	The list of inputs for each Credit Memo Line Item to create. This parameter is ignored if you set **isFullCredit** as *True.*
autoApprove	Boolean	No	If true the Credit Memo will be transitioned to Approved otherwise the Credit Memo will be created with a status of Draft. A value of null will be considered as false.
autoApplyCreditMemo	Boolean	No	This flag is only relevant when the Auto Approve option is true. If both options are true the newly created direct Credit Memo will be applied to the Invoice, the credit was drawn from. A value of null will be considered as false.
TemplateId	ID	No	The Id of the Credit Memo template to use when creating the Credit Memo attachment. This parameter is optional and can be null.
calculateTax	Boolean	Yes	If this flag is true Tax will be calculated for a non tax-exempt Asset. If false, no tax will be calculated.

⚠ Ensure that you provide value for either isFullCredit or creditMemoLineItemInputs parameters. If you leave both the fields as null, the API returns an error.

Request - CreditMemoLineItemInput			
Field	**Type**	**Required?**	**Description**
invoiceLineItemId	ID	Yes	The Id of the affiliate Invoice Line Item. It must be a child of the specified Invoice.
creditAmount	Decimal	Yes	The amount of credit to draw from the corresponding Invoice Line Item. The amount must be a positive number and cannot exceed the available credit of the corresponding Invoice Line Item.

Response		
Field	**Type**	**Description**
DirectCreditMemoResult	List	A result parameter is returned for each request parameter.

Create Direct Credit Memo API returns an error if:

- The specified invoice does not exist.
- The status of your specified invoice is not Approved.
- Your specified reason code is invalid. Reason code can be null or one of the picklist values declared in the Reason Code field of Credit Memo object.

  ⓘ Create Direct Credit Memo API does not support the following reason codes:
  - Wallet Application
  - Credit & Rebill

- Credit Amount is negative or 0.
- Template ID does not exist.
- Template Type is other than Credit Memo.
- Specified Invoice Line Item does not exist.
- Specified Invoice Line Item does not belong to the specified invoice.
- Invoice Line Item is already referenced in more than one Credit Memo Line Item.
- Credit Amount is greater than the Available Credit.

Code Sample 1

```
// Set Credit Memo input parameters
List<DirectCreditMemoInput> creditMemoInputs = new List<DirectCreditMemoInput>();
List<DirectCreditMemoInput.DirectCreditMemoLineItemInput> creditMemoLineItemInputs;
Set<Id> invoiceIds = new Set<Id>

{'a2V1N000002QbuC', 'a2V1N000002QbuM'}
;
List<Invoice__c> invoices = SELECT Id, Status__c, (Select Id, Amount__c From
InvoiceLineItems__r) FROM Invoice__c Where Id IN:invoiceIds;
Id templateId = 'a1b1N000003QCS';
Decimal creditAmount = 100.00;
String reasonCode = null;

for(Invoice__c invoice : invoices){
creditMemoLineItemInputs = new
 List<DirectCreditMemoInput.DirectCreditMemoLineItemInput>();
for(InvoiceLineItem_c invoiceLineItem : invoice.InvoiceLineItems_r)

{ creditMemoLineItemInputs.add(new
 DirectCreditMemoInput.DirectCreditMemoLineItemInput(invoiceLineItem.Id,
creditAmount)); }
creditMemoInputs.add(new DirectCreditMemoInput(invoice.Id,
reasonCode,
true,
true,
templateId,
true,
creditMemoLineItemInputs));
}
List<DirectCreditMemoResult> actualCreditMemoResults =
Apttus_Billing.BillingService.createDirectCreditMemos(creditMemoInputs);
System.debug('actualCreditMemoResults========================'
+actualCreditMemoResults);
```

Code Sample 2 - To provide Full Credit

```
Apttus_Billing.DirectCreditMemoInput dcm = new
 Apttus_Billing.DirectCreditMemoInput();
dcm.autoApplyCreditMemo = true;
dcm.autoApprove = true;
dcm.calculateTax = true;
dcm.creditMemoLineItemInputs = null;
dcm.invoiceId = 'a5F5x000001hqtJ';
dcm.isFullCredit = true;
dcm.reasonCode = 'Refund';
dcm.templateId = 'a03f200000lccG2';
List<Apttus_Billing.DirectCreditMemoInput> creditMemoInputs = new
 List<Apttus_Billing.DirectCreditMemoInput>();
creditMemoInputs.add(dcm);
List<Apttus_Billing.DirectCreditMemoResult> actualCreditMemoResults =

Apttus_Billing.BillingService.createDirectCreditMemos(creditMemoInputs);
```

Creating Billing Plan with Product Configuration

createBillingPlanWithProductConfiguration API is used to create a billing plan for the product configurations. This API is used to creates a billing plan without the quote flow, directly from the cart page. It creates a billing plan with billing plan line items. For plan type as Milestone, the API also creates milestones.

API	Signature
createBillingPlanWithProductConfiguration	*static Apttus_Billing.BillingService.BillingPlanAPIResponse createBillingPlanWithProductConfiguration(Id productConfigurationId, Id billingPlanTemplateId, Set* lineItemIds)

This API accepts Product Configuration ID, Billing Plan Template ID, and a Set of Line Item IDs as input parameters. It creates a billing plan and billing plan line items for the given proposal ID. Created billing plan is applied to the list of provided Line Item IDs. If you do not provide any line item IDs as input parameters, the created billing plan is applied to all the line items.

For a billing plan template with plan type as Milestones, createBillingPlanWithProductConfiguration API also creates milestones.

Request			
Field	**Type**	**Required?**	**Description**
productConfiguraionID	ID	Yes	ID of the Product Configuration
billingPlanTemplateId	ID	Yes	ID of the Billing Plan Template
lineItemIds	Set <Id>	No	Set of Line Item Ids

Response: BillingPlanAPIResponse		
Field	**Type**	**Description**
isSuccess	Boolean	Returns true if the API is executed correctly
errorMessage	String	Error Message if the API is not executed successfully.
billingPlanId	ID	Id of the created Billing Plan

Code Sample

```
Id productconfigurationId = 'a2G1N000002RpRZUA0';
Id billingPlanTemplateId = 'a3D1N000000blZc';
Set<Id> LineItemIds = new Set<Id>('a2E1N000001dHox', 'a2E1N000001eHox');

response =
Apttus_Billing.BillingService.createBillingPlanWithProductConfiguration(productconfig
uratlId, billingPlanTemplateId);

// Call API with line item ids parameter
response =
Apttus_Billing.BillingService.createBillingPlanWithProductConfiguration(productconfig
uratlId, billingPlanTemplateId, LineItemIds);
```

⚠ createBillingPlanWithProductConfiguration API is not supported for Evergreen Billing and Usage-based products.

Forecasting Billing Schedules for Smart Cart

`forecastBilling` API is used to generate forecast billing schedules for Smart Cart. A smart cart is a cart with a large number of product line items. This API run in the asynchronous mode. This API accepts the proposal ID and the option to extend the end date with renewal term as input parameters.

API	Signature
forecastBilling	*static Id forecastBilling(Id proposalId, Boolean extendEndDateWithRenewalTerm)*

Forecast Billing Schedules API is used to generate forecast billing schedules, forecast billing summaries, forecast options, and forecast line item options for a large number of product line items. This API can be invoked by the implementation team after the cart is finalized.

`forecastBilling` API submits an asynchronous batch job to delete all the existing forecast billing schedules and returns the ID of the submitted batch job as a response parameter. After deleting the existing forecast data, `forecastBilling` API submits a batch job to generate forecast billing data for all the line items associated with the given proposal ID.

⚠ Forecast billing schedule functionality is not supported for:
- Quote/Proposal associated with a billing plan
- Informational line items of bundle and option products

Request			
Field	**Type**	**Required?**	**Description**
proposalID	ID	Yes	The proposal ID

Request			
Field	**Type**	**Required?**	**Description**
extendEndDateWithRenewal	Boolean	Yes	If extendEndDateWithRenewal is set to true, then for each proposal line item with Auto-Renew set to True and the Auto-Renewal Type set to Fixed, the end date is extended based on the renewal term. If you pass a null value, the API considers it as a false value.

Response		
Field	**Type**	**Description**
apexJobId	ID	Id of the batch job submitted to delete the forecast billing schedules.

```
/**
* For a given proposal, delete exising forecast billing data, if any, and regenerate
the forecast data.
* This API supports the forecasting for samrt cart.
*
* @param proposalId The proposal Id for which the forecasting needs to be done.
* @param extendEndDateWithRenewalTerm States whether to generate forecasting with
current term or extend it with renewal term.
* A 'null' will default to 'false' value
*
* @return The ID of the new batch job (AsyncApexJob).
*
* @note The API functionality involves two batch jobs, one for deleting the existing
forecast data,
* and another chained batch is generating the new forecast data.
* Hence, the Id returned by the API is of the first deletion batch job.
*/
global static Id forecastBilling(Id proposalId, Boolean extendEndDateWithRenewalTerm)
{
return DeleteForecastedBillingBatchJob.deleteAndRegenerateForecastData(proposalId,
extendEndDateWithRenewalTerm);
}
```

Applying a Late Fee

With `applyLateFeesToInvoice` API, you can charge your customers a Late Fee, if they do not complete their payment by the Due Date.

API	Signature
applyLateFeesToInvoices	*static List applyLateFeesToInvoices(List inputs)*

`applyLateFeesToInvoice` API accepts Invoice ID and Late Fee ID as input parameters. This API calculates the late fee amount and adds the calculated amount to the Total Due Amount of the specified invoice Id. A Destinated Related A/R Transaction record is created for the late fee applied on the invoice. `applyLateFeesToInvoice` API returns ApplyInvoiceLateFeeResult object as response parameters.

Request: ApplyInvoiceLateFeeAmount			
Field	**Type**	**Required?**	**Description**
invoiceId	ID	Yes	The invoice Id to apply the late fee.
lateFeeId	ID	Yes	Id of the late fee to calculate penalty amount for the specified invoice

Response: ApplyInvoiceLateFeeResult		
Field	**Type**	**Description**
isSuccess	Boolean	Value is set as true if the Late Fee is applied successfully. Value is set as false if the API encounters an error.
errorMessage	String	If the API execution false, errorMessage string contains a detailed error message.
lateFeeAmount	Decimal	Value is set as the calculated late fee amount.
relatedA/RTransactionId	ID	Value is set as the ID of the created Related A/R Transaction record

```
Apttus_Billing__LateFee__c lateFee = new Apttus_Billing__LateFee__c(
        Name = 'Test Late Fee',
        Apttus_Billing__LateFeeValue__c = 100,
        Apttus_Billing__LateFeeType__c = 'Amount'
    );
    insert lateFee;

List<Apttus_Billing.CustomClass.ApplyInvoiceLateFeeInput> lateFeeInputs = new
 List<Apttus_Billing.CustomClass.ApplyInvoiceLateFeeInput>();
Apttus_Billing.CustomClass.ApplyInvoiceLateFeeInput lateFeeInput1 = new
 Apttus_Billing.CustomClass.ApplyInvoiceLateFeeInput(invoiceId, lateFee.Id);
lateFeeInputs.add(lateFeeInput1);
List<Apttus_Billing.CustomClass.ApplyInvoiceLateFeeResult> invoiceLateFeeResultList =
Apttus_Billing.BillingService.applyLateFeesToInvoices(lateFeeInputs);
```

Cancelling a Late Fee

With `cancelInvoiceLateFees` API, you can cancel a late fee incorrectly applied to an invoice.

API	Signature
cancelInvoiceLateFees	*static List cancelInvoiceLateFees(List inputs)*

`cancelInvoiceLateFees` API accepts Related A/R Transaction ID and Description as input parameters. This API cancels the late fee and deducts the late fee amount from the Total Due Amount of the Invoice. A Destinated Related A/R Transaction record is created for the canceled late fee. `cancelInvoiceLateFees` API returns cancelInvoiceLateFeeResult object as a response parameter.

Request: cancelLateFeeInputs			
Field	**Type**	**Required?**	**Description**
relatedLateFeeTransactionId	ID	Yes	ID of the Related A/R Transaction record
description	String	Yes	Reason to cancel the late fee.

Response: cancelInvoiceLateFeeResult		
Field	**Type**	**Description**
isSuccess	Boolean	Value is set as true if the Late Fee is canceled successfully. Value is set as false if the API encounters an error.
errorMessage	String	If the API execution false, errorMessage string contains a detailed error message.
invoiceId	ID	Value is set as the ID of the invoice for which the late fee was canceled successfully.
cancellationTransactionId	ID	Value is set as the ID of the Related A/R Transaction created record the cancellation of late fee

```
List<CustomClass.CancelInvoiceLateFeeInput> cancelLateFeeInputs = new
 List<CustomClass.CancelInvoiceLateFeeInput>();

CustomClass.CancelInvoiceLateFeeInput cancelInput1 = new
 CustomClass.CancelInvoiceLateFeeInput('a5p2v000000dpZ3', 'test description');

cancelLateFeeInputs.add(cancelInput1);

List<CustomClass.CancelInvoiceLateFeeResult> cancelInvoiceLateFeeResult =
BillingService.cancelInvoiceLateFees(cancelLateFeeInputs);
```

Creating Direct Credit Memos Asynchronously

You can use createDirectCreditMemosAsync API to create credit memos for an invoice with large number of invoice line items. This API is an asynchronous API and it submits batch jobs to create direct credit memos. You can use this API to create credit memos for a large number of invoices or invoices with a large number of invoice line items.

API	Signature
createDirectCreditMemosAsync	*static* *List* *createDirectCreditMemosAsync(List creditMemoInputs)*

This API is used to create credit memos for an invoice. It accepts a Set of Account IDs, invoiceDate, and targetDateTime as input parameters. For each DirectCreditMemoInput, a credit memo is created in the Draft Status. You can auto-approve a credit memo line item and auto-apply it to an invoice. If you have a Tax Callback registered, this API also calculates Tax and creates Tax breakups for a credit memo line item.

Request - DirectCreditMemoInput			
Field	**Type**	**Required?**	**Description**
invoiceId	ID	Yes	The Id of the approved Invoice the credit will be drawn from.
reasonCode	String	Yes	The reason for creating the Credit Memo. Must be null or a valid pick-list value for the Reason Code field declared of the Credit Memo object.
isFullCredit	Boolean	No	If you set isFullCredit as true, the entire available credit amount of all the invoice line items is set as the credit amount. If you set the value as false, credit memo amount is calculated based on the values you provide in creditMemoLineItemInputs field. By default, it is set as false.

Request - DirectCreditMemoInput			
Field	**Type**	**Required?**	**Description**
creditMemoLineItemInputs	List<DirectCreditMemoLineItemInput>	No	The list of inputs for each Credit Memo Line Item to create. This parameter is ignored if you set **isFullCredit** as *True.*
autoApprove	Boolean	No	If true the Credit Memo will be transitioned to Approved otherwise the Credit Memo will be created with a status of Draft. A value of null will be considered as false.
autoApplyCreditMemo	Boolean	No	This flag is only relevant when the Auto Approve option is true. If both options are true the newly created direct Credit Memo will be applied to the Invoice, the credit was drawn from. A value of null will be considered as false.
TemplateId	ID	No	The Id of the Credit Memo template to use when creating the Credit Memo attachment. This parameter is optional and can be null.
calculateTax	Boolean	Yes	If this flag is true Tax will be calculated for a non tax-exempt Asset. If it is set to false, no tax will be calculated.

⚠ Ensure that you provide value for either isFullCredit or creditMemoLineItemInputs parameters. If you leave both the fields as null, the API returns an error.

Request - CreditMemoLineItemInput			
Field	**Type**	**Required?**	**Description**
invoiceLineItemId	ID	Yes	The Id of the affiliate Invoice Line Item. It must be a child of the specified Invoice.
creditAmount	Decimal	Yes	The amount of credit to draw from the corresponding Invoice Line Item. The amount must be a positive number and cannot exceed the available credit of the corresponding Invoice Line Item.

Response		
Field	**Type**	**Description**
DirectCreditMemoResult	List	A result parameter is returned for each request parameter.

Create Direct Credit Memo API returns an error if:

- The specified invoice does not exist.
- The status of your specified invoice is not Approved.
- Your specified reason code is invalid. Reason code can be null or one of the picklist values declared in the Reason Code field of Credit Memo object.

  > Create Direct Credit Memo API does not support the following reason codes:
  > - Wallet Application
  > - Credit & Rebill

- Credit Amount is negative or 0.
- Template ID does not exist.
- Template Type is other than Credit Memo.
- Specified Invoice Line Item does not exist.
- Specified Invoice Line Item does not belong to the specified invoice.
- Invoice Line Item is already referenced in more than one Credit Memo Line Item.
- Credit Amount is greater than the Available Credit.

Code Sample

```
Apttus_Billing.DirectCreditMemoInput dcm = new
 Apttus_Billing.DirectCreditMemoInput();
dcm.autoApplyCreditMemo = true;
dcm.autoApprove = true;
dcm.calculateTax = true;
dcm.creditMemoLineItemInputs = null;
dcm.invoiceId = 'a5F5x000001hqtJ';
dcm.isFullCredit = true;
dcm.reasonCode = 'Refund';
dcm.templateId = 'a03f200000lccG2';
List<Apttus_Billing.DirectCreditMemoInput> creditMemoInputs = new
 List<Apttus_Billing.DirectCreditMemoInput>();
creditMemoInputs.add(dcm);
List<Apttus_Billing.DirectCreditMemoResult> actualCreditMemoResults =

Apttus_Billing.BillingService.createDirectCreditMemosAsync(creditMemoInputs);
```

Retrieving Session Key

docGenAPIUserKey API is used to retrieve the session key. The session key is generated for the user credentials stored in the Merge Server API Credentials. Billing administrators can use the session key to call the Merge Service API.

API	Signature
DocGenApiUserKey	*static String DocGenApiUserKey()*

This API returns the session key for Doc Gen. This API returns the session key only if the API credentials are configured. You can configure the API credentials from the Billing Console Page by clicking **Merge Service API Credentials** link under **Setup.**

Response		
Field	**Type**	**Description**
sessionKey	String	Returns the session key for Doc Gen

Code Sample

```
String sessionKey = docGenAPIUserKey();
```

Use Case for 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.

API Flow for Usage-Based Billing

1. Create an Order using createOrder API.
2. Create usage inputs.
3. Before you rate the usage inputs, preview them using the previewPendingUsageInputs API.
4. Rate the usage inputs using processPendingUsageInput API. For incorrectly rate usage inputs, you can use the processRatedUsageInput API to unrate them.
5. Invoice your customers using createInvoicesForOrder API.
6. Generate the invoice document using Creating Invoice Documents API and email it to your customers.

WSDL Services

Apttus Billing Management offers the following WSDL Services:

- Applying Credit Memos to Invoices
- Applying Payments to Invoices
- Adding a A/R Transaction - Deprecated
- Adding Multiple A/R Transactions - Deprecated

Prerequisites for invoking WSDL Services:

1. You need to Generate Entripse WSDL and Apex WSDL for ARTransactionService. For information on generating enterprise WSDL, refer to Generating WSDL.
2. Add both the Generated WSDLs as a reference in your .Net Project.

Applying Credit Memos to Invoices

This API indirectly applies a Credit Memo to an Invoice by creating Related A/R Transaction records to reduce the tracked balance of the Invoice and the Credit Memo (double entry accounting).

You can use this API for a single invoice or multiple invoices. For each Related A/R Transaction Input passed in, a Related A/R Transaction Result will be returned. However, the order of the results may not be the same as the order of the inputs.

For more details on A/R Transactions, refer to Related A/R transactions in Billing Management User Guide.

API	Signature
applyCreditMemosToInvoices	*webService static* *List applyCreditMemosToInvoices(List inputs)*

Request

Field	Type	Required?	Description
inputs	List <RelatedARTxnInput2>	Yes	List of RelatedARTaxInput2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput2

Field	Type	Description
Description	String	Description for the transaction.
DestinationObjId	ID	Id of the Destination Object.
ExternalSystemStatus	String	Status of the External System.
IntegrationDate	Datetime	Date of the system Integration.
ReasonCode	String	Reason for applying the A/R transaction.
SourceObjId	ID	Id of the Source Object.
transactionAmount	Decimal	The transaction amount.
transactionDate	Datetime	The transaction date.
transactionISOCurrency	String	The transaction Currency.
transactionNumber	String	The transaction number.
transactionSubType	String	Sub-type of transaction (picklist value).

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput2		
transactionType	String	Type of transaction.

Response		
Field	**Type**	**Description**
results	List <RelatedARTxnResult2>	List of RelatedARTaxResult2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult2		
Field	**Type**	**Description**
destinationARTransactionId	ID	A/R transaction Id for destination object.
destinationObjId	ID	The id of Destination object.
errorString	String	The Error message. Null value implies success and non-empty string value implies failure.
sourceObjId	ID	The id of Source object.
Status	String	The Status of a transaction will be success or failure.
transactionNumber	String	The transaction number.

Code Sample

```
using System;
using sforce = ConsoleApplication1.Sforce1;
using ARService = ConsoleApplication1.ARTransactionService;
using System.Net;

namespace ConsoleApplication1
{
    class Program
    {

        static void Main(string[] args)
        {

            String sessionId;
            using (sforce.SoapClient client = new sforce.SoapClient())
            {
                ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 |
SecurityProtocolType.Tls11;

                //call login service
                sforce.LoginResult result = client.login(null, "[ORG_UserName]",
"[ORG_Password]");
                //extract authentication token
                 sessionId = result.sessionId;
                //serviceUrl = result.serverUrl;
                Console.WriteLine("Token generated: " + sessionId);

            }


            //call Related AR Transaction SOAP API
            ARService.SessionHeader header = new ARService.SessionHeader();
            header.sessionId = sessionId;

            using (ARService.RelatedARTransactionServicePortTypeClient soapClient =
new ARService.RelatedARTransactionServicePortTypeClient())
            {
                ARService.RelatedARTxnInput2[] inputs = new
 ARService.RelatedARTxnInput2[1];
                ARService.RelatedARTxnInput2 input2 = new
 ARService.RelatedARTxnInput2();
                input2.destinationObjId = "[InvoiceID]";
                input2.sourceObjId = "[CreditMemoID]";
```

```
                input2.transactionAmount = 100.00M;
                input2.description = "test description";
                input2.integrationDate = DateTime.Today;
                input2.transactionDate = DateTime.Today;
                input2.transactionNumber = "AR12342";
                input2.transactionType = "Credit Memo";

                inputs[0] = input2;
                ARService.RelatedARTxnResult2[] arResults = new
 ARService.RelatedARTxnResult2[1];
                soapClient.applyCreditMemosToInvoices(header, null, null, null,
inputs, out arResults);
                Console.WriteLine("ARResult is: " + arResults[0].destinationARTransac
tionId);
                Console.ReadLine();
            }
        }
    }
}
```

Integration Details

Use the following information in your integrations with Apttus Intelligent Workflow Approvals API. Refer to Integrating Apttus with External Systems for information on how to get started.

API Prerequisites

None.

Response/Request XML

Example Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:rel="http://soap.sforce.com/schemas/class/Apttus_Billing/
RelatedARTransactionService">
   <soapenv:Header>
      <rel:SessionHeader>
         <rel:sessionId>00Df2000000vacp!
ARIAQE7yD6Xo8Be5p8.rrdqyO5K93M7qwAvc1FIav6nuWMt10X2jRQJ.aGYxUTHvWAB8NZRR7ngYsyamyEthU
Y3HbpuvoooN</rel:sessionId>
      </rel:SessionHeader>
   </soapenv:Header>
   <soapenv:Body>
      <rel:applyCreditMemosToInvoices>
         <!--Zero or more repetitions:-->
         <rel:inputs>
            <!--Optional:-->
            <rel:description></rel:description>
            <!--Optional:-->
            <rel:destinationObjId>a5F5x000001pVOn</rel:destinationObjId>
            <!--Optional:-->
            <rel:externalSystemStatus></rel:externalSystemStatus>
            <!--Optional:-->
            <rel:reasonCode>Write Off From Invoice</rel:reasonCode>
            <!--Optional:-->
            <rel:sourceObjId>a525x000002AqB2</rel:sourceObjId>
            <!--Optional:-->
            <rel:transactionAmount>10</rel:transactionAmount>
            <!--Optional:-->
            <rel:transactionDate>2020-07-01T00:00:00Z</rel:transactionDate>
            <!--Optional:-->
            <rel:transactionISOCurrency>USD</rel:transactionISOCurrency>
            <!--Optional:-->
            <rel:transactionNumber></rel:transactionNumber>
            <!--Optional:-->
            <rel:transactionSubType></rel:transactionSubType>
            <!--Optional:-->
            <rel:transactionType>Credit Memo</rel:transactionType>
         </rel:inputs>
      </rel:applyCreditMemosToInvoices>
```

```
   </soapenv:Body>
</soapenv:Envelope>
```

Example Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns="http://soap.sforce.com/schemas/class/Apttus_Billing/
RelatedARTransactionService" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <applyCreditMemosToInvoicesResponse>
         <result>
            <destinationARTransactionId>a5J5x0000012BRKEA2</
destinationARTransactionId>
            <destinationObjId>a5F5x000001pVOnEAM</destinationObjId>
            <errorString xsi:nil="true"/>
            <sourceObjId>a525x000002AqB2AAK</sourceObjId>
            <status>Success</status>
            <transactionNumber/>
         </result>
      </applyCreditMemosToInvoicesResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Applying Payments to Invoices

This API indirectly applies a Payment to an Invoice by creating Related A/R Transaction records to reduce the tracked balance of the Invoice and the Payment (double entry accounting).

If no Payment exists for the specified Transaction Number then a Payment will be created with the Payment Amount set to the Transaction Amount.

The API can be for a single Invoice or multiple Invoices. For each Related A/R Transaction Input passed in, a Related A/R Transaction Result will be returned. However, the order of the results may not be the same as the order of the inputs.

For more details on A/R Transactions, refer to Related A/R transactions in Billing Management User Guide.

API	Signature
applyPaymentsToInvoices	*webService static* *List applyPaymentsToInvoices(List inputs)*

Request		
Field	**Type**	**Description**
inputs	List <RelatedARTxnInput2>	List of RelatedARTaxInput2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput2		
Field	**Type**	**Description**
Description	String	Description for the transaction.
destinationObjId	ID	Id of the Destination Object.
externalSystemStatus	String	Status of the External System.
integrationDate	Datetime	Date of the system Integration.
reasonCode	String	Reason for applying the A/R transaction.
sourceObjId	ID	Id of the Source Object.
transactionAmount	Decimal	The transaction amount.
transactionDate	Datetime	The transaction date.
transactionISOCurrency	String	The transaction Currency.
transactionNumber	String	The transaction number.
transactionSubType	String	Sub-type of transaction (picklist value).

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput2		
transactionType	String	Type of transaction.

Response		
Field	**Type**	**Description**
results	List <RelatedARTxnResult2>	List of RelatedARTaxResult2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult2		
Field	**Type**	**Description**
destinationARTransactionId	ID	A/R transaction Id for destination object.
destinationObjId	ID	The id of Destination object.
errorString	String	The Error message.Null value implies success and non-empty string value implies failure.
sourceObjId	ID	The id of Source object.
Status	String	The Status of a transaction will be success or failure.
transactionNumber	String	The transaction number.

Code Sample

```
using System;
using sforce = ConsoleApplication1.Sforce1;
using ARService = ConsoleApplication1.ARTransactionService;
using System.Net;

namespace ConsoleApplication1
{
    class Program
    {

        static void Main(string[] args)
        {

            String sessionId;
            using (sforce.SoapClient client = new sforce.SoapClient())
            {
                ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 |
SecurityProtocolType.Tls11;

                //call login service
                sforce.LoginResult result = client.login(null, "[ORG_UserName]",
"[ORG_Password]");
                //extract authentication token
                 sessionId = result.sessionId;
                //serviceUrl = result.serverUrl;
                Console.WriteLine("Token generated: " + sessionId);

            }


            //call Related AR Transaction SOAP API
            ARService.SessionHeader header = new ARService.SessionHeader();
            header.sessionId = sessionId;

            using (ARService.RelatedARTransactionServicePortTypeClient soapClient =
new ARService.RelatedARTransactionServicePortTypeClient())
            {
                ARService.RelatedARTxnInput2[] inputs = new
 ARService.RelatedARTxnInput2[1];
                ARService.RelatedARTxnInput2 input2 = new
 ARService.RelatedARTxnInput2();
                input2.destinationObjId = "[InvoiceID]";
                input2.transactionAmount = 100.00M;
```

```
                input2.description = "test description";
                input2.integrationDate = DateTime.Today;
                input2.transactionDate = DateTime.Today;
                input2.transactionNumber = "AR12342";
                input2.transactionType = "Payment";

                inputs[0] = input2;
                ARService.RelatedARTxnResult2[] arResults = new
 ARService.RelatedARTxnResult2[1];
                soapClient.applyPaymentsToInvoices(header, null, null, null, inputs,
out arResults);
                Console.WriteLine("ARResult is: " + arResults[0].destinationARTransac
tionId);
                Console.ReadLine();
            }
        }
    }
}
```

Integration Details

Use the following information in your integrations with Apttus Intelligent Workflow Approvals API. Refer to Integrating Apttus with External Systems for information on how to get started.

API Prerequisites

None.

Response/Request XML

Example Request

```
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rel="http://soap.sforce.com/schemas/class/Apttus_Billing/
RelatedARTransactionService">
   <soapenv:Header>
      <rel:SessionHeader>
         <rel:sessionId>00Df2000000vacp!
ARIAQE7yD6Xo8Be5p8.rrdqyO5K93M7qwAvc1FIav6nuWMt10X2jRQJ.aGYxUTHvWAB8NZRR7ngYsyamyEthU
Y3HbpuvoooN</rel:sessionId>
      </rel:SessionHeader>
   </soapenv:Header>
   <soapenv:Body>
      <rel:applyPaymentsToInvoices>
         <!--Zero or more repetitions:-->
         <rel:inputs>
            <!--Optional:-->
            <rel:description>Payment</rel:description>
            <!--Optional:-->
            <rel:destinationObjId>a5F5x000001pVSt</rel:destinationObjId>
            <!--Optional:-->
            <rel:integrationDate>2020-07-31T00:00:00Z</rel:integrationDate>
            <!--Optional:-->
            <rel:transactionAmount>50</rel:transactionAmount>
            <!--Optional:-->
            <rel:transactionDate>2020-07-31T00:00:00Z</rel:transactionDate>
            <!--Optional:-->
            <rel:transactionISOCurrency>USD</rel:transactionISOCurrency>
            <!--Optional:-->
            <rel:transactionNumber>P_123</rel:transactionNumber>
            <!--Optional:-->
            <rel:transactionType>Payment</rel:transactionType>
         </rel:inputs>
      </rel:applyPaymentsToInvoices>
   </soapenv:Body>
</soapenv:Envelope>
```

Example Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns="ht
tp://soap.sforce.com/schemas/class/Apttus_Billing/RelatedARTransactionService"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <applyPaymentsToInvoicesResponse>
         <result>
            <destinationARTransactionId>a5J5x0000012BWVEA2</
destinationARTransactionId>
            <destinationObjId>a5F5x000001pVStEAM</destinationObjId>
            <errorString xsi:nil="true"/>
            <sourceObjId>a5H5x000001peerEAA</sourceObjId>
            <status>Success</status>
            <transactionNumber>P_123</transactionNumber>
         </result>
      </applyPaymentsToInvoicesResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Adding a A/R Transaction - Deprecated

addRelatedARTransaction(RelatedARTxnInput input)

This API applies Credit Memo to Invoices as part of the A/R transaction.

Request		
Field	**Type**	**Description**
input	RelatedARTxnInput	List of RelatedARTaxInput2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput		
Field	**Type**	**Description**
collectionStatus	String	The Collection Status.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput		
description	String	Description for A/R Transaction.
externalSystemStatus	String	The Status of external system.
impactARAmount	Decimal	The A/R impact amount.
integrationDate	Datetime	Date of Integration.
invoiceId	ID	Invoice Id.
reasonCode	String	Reason code for creating A/R transaction.
relatedTransactionId	String	Related transacion id.
transactionAmount	Decimal	Transaction amount.
transactionDate	Datetime	Transaction date.
transactionEffectiveDate	Datetime	The effective date of transaction.
transactionNumber	String	Transaction number.
transactionType	String	The type of transaction.

Response		
Field	**Type**	**Description**
result	RelatedARTxnResult	List of RelatedARTaxResult2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult		
Field	**Type**	**Description**
errorString	String	Error string showing the error message.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult		
invoiceId	ID	Invoice Id.
relatedARTransactionId	ID	The related A/R transaction Id.
Status	String	Status of the transaction.
transactionNumber	String	Transaction number.

⚠ This API is deprecated. Calling this API will result in an error.

Adding Multiple A/R Transactions - Deprecated

This API invokes RelatedAR Transactions.

addRelatedARTransactions(List<RelatedARTxnInput> inputs)

This API is used to accept A/R Transactions in bulk.

Request			
Field	**Type**	**Required?**	**Description**
inputs	RelatedARTxnInput	Yes	List of RelatedARTaxInput2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput		
Field	**Type**	**Description**
collectionStatus	String	The Collection Status.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnInput		
description	String	Description for A/R Transaction.
externalSystemStatus	String	The Status of external system.
impactARAmount	Decimal	The A/R impact amount.
integrationDate	Datetime	Date of Integration.
invoiceId	ID	Invoice Id.
reasonCode	String	Reason code for creating A/R transaction.
relatedTransactionId	String	Related transacion id.
transactionAmount	Decimal	Transaction amount.
transactionDate	Datetime	Transaction date.
transactionEffectiveDate	Datetime	The effective date of transaction.
transactionNumber	String	Transaction number.
transactionType	String	The type of transaction.

Response		
Field	**Type**	**Description**
results	List <RelatedARTxnResult>	List of RelatedARTaxResult2 records

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult		
Field	**Type**	**Description**
errorString	String	Error string showing the error message.

Apttus_Billing.RelatedARTransactionService.RelatedARTxnResult		
invoiceId	ID	Invoice Id.
relatedARTransactionId	ID	The related A/R transaction Id.
Status	String	Status of the transaction.
transactionNumber	String	Transaction number.

⚠ This API is deprecated. Calling this API will result in an error.

REST Services

Apttus Billing Management offers the following REST Services:

- Creating Invoices for Orders
- Creating Invoices
- Processing Pending Usage Inputs

Billing Management REST APIs

Apttus Copyright Disclaimer

quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Apttus is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

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

DOC ID: BMSFSUM20APIG20200731