



# Order Management on Salesforce Spring 2020 SOAP API Guide



# Table of Contents

- About this Guide..... 3
- What's New ..... 5
- About Order Management ..... 8
  - Key Terminology ..... 10
- Getting Started ..... 14
  - API Standards and Development Platforms..... 14
    - Standards ..... 14
    - Development Platforms..... 14
  - Field Types..... 15
- API Reference ..... 16
  - Order Web Service ..... 16
    - Creating an Order ..... 16
    - Creating a Cart for an Order ..... 18
    - Synchronizing a Cart to an Order ..... 22
    - Accepting an Order ..... 23
    - In-Flight Order Changes and Cancellation ..... 25
    - Cloning an Order..... 42
    - Fetching Price Agreements ..... 44
    - Fetching Price Agreement Items..... 49
    - Creating a Partial Order from a Quote or Agreement..... 53
    - Creating a Direct Order..... 61
    - Executing Order Workflow ..... 66
  - Customer Purchase Order (CPO) Web Service ..... 67
    - Accepting Customer Purchase Orders..... 67
    - Amending Customer Purchase Orders..... 69
    - Reverting Amended Customer Purchase Orders..... 71
    - Cancelling Customer Purchase Orders ..... 72
    - Reverting Cancelled Customer Purchase Orders ..... 73
    - Creating Orders from PO..... 75
- Apttus Copyright Disclaimer ..... 80

## About this Guide

The *Order Management on Salesforce User Guide* provides a collection of SOAP (Simple Object Access Protocol) service APIs to allow users to extend the feature functionality of Order Management on Salesforce. Use the SOAP APIs in this reference guide as a tool for designing solutions to complex business needs. All APIs in the reference are defined using the SOAP, WSDL, and WS-I standards.

Topic	Description
What's Covered	This guide defines and provides examples of Order Web Service APIs that can be used to create and manage orders, including partial orders from price agreements and customer purchase orders, as well as in-flight order changes and cancellation.
Primary Audience	<ul style="list-style-type: none"> <li>• Order Management Administrators</li> <li>• Apttus Professional Services</li> </ul>
IT Environment	Refer to the latest <i>Order Management on Salesforce Release Notes</i> for information on System Requirements and Supported Platforms.
Updates	For a comprehensive list of updates to this guide for each release, see the <a href="#">What's New</a> topic.
Other Resources	<ul style="list-style-type: none"> <li>• <a href="#">Order Management on Salesforce User Guide</a>: Refer to this guide for installing and setting up Order Management in your organization.</li> <li>• <a href="#">Order Management on Salesforce Administrator Guide</a>: Refer to this guide for installing and setting up Order Management in your organization.</li> <li>• <a href="#">Order Management on Salesforce Release Notes</a>: Refer to this guide for the new feature, enhancements, resolved, and known issues.</li> <li>• <a href="#">CPQ on Salesforce Administrator Guide</a>: Refer to this guide for setting up products, price lists, and constraint rules.</li> <li>• <a href="#">Salesforce SOAP API Developer Guide</a></li> </ul>

This guide describes the following tasks:

- Implementing Order Management Web Service
- Implementing Customer Purchase Order Web Service

Before using Order Management APIs, you must be familiar with the following:

- Basic Salesforce administration knowledge
- Apttus CPQ and Apttus CLM administration
- Salesforce and Apttus terms and definitions

- Apttus Order Management basics
- Experience with SOAP/WSDL API protocols

## What's New

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

Release	Topic	Description
Spring 2020 (2001)	<a href="#">About This Guide</a>	Updated topic.
	<a href="#">About Order Management</a>	Updated topic.
	<a href="#">Getting Started</a>	Renamed topic from "Overview." Updated topic.
	<a href="#">API Supported Packages</a>	Deleted topic.
	<a href="#">API Reference</a>	Updated topic.
	<a href="#">Creating an Order</a>	Updated topic.
	<a href="#">Creating a Cart for an Order</a>	Updated topic.
	<a href="#">Synchronizing a Cart to an Order</a>	Updated topic.
	<a href="#">Accepting an Order</a>	Updated topic.
	<a href="#">In-Flight Order Changes and Cancellation</a>	New contextual parent topic for Amend Order, Undo Amend Order, and Cancel Order APIs.
	<a href="#">Amending an Order</a>	Updated topic.
	<a href="#">Reverting an Amendment to an Order</a>	New topic. Content taken from "Amending an Order."
	<a href="#">Cancelling an Order</a>	Updated topic.
	<a href="#">Reverting a Cancelled Order</a>	New topic. Content taken from "Cancelling an Order."
	<a href="#">Cloning an Order</a>	Updated topic.
<a href="#">Fetching Price Agreements</a>	Updated topic.	

Release	Topic	Description
	Fetching Price Agreement Items	Updated topic.
	Creating a Partial Order from a Quote or Agreement	Renamed topic. Updated topic.
	Creating a Direct Order	Updated topic.
	Executing Order Workflow	Updated topic.
	Customer Purchase Order (CPO) Web Service	Updated topic.
	Accepting Customer Purchase Orders	Updated topic.
	Amending Customer Purchase Orders	Updated topic.
	Reverting Amended Customer Purchase Orders	Updated topic.
	Cancelling Customer Purchase Orders	Updated topic.
	Reverting Cancelled Customer Purchase Orders	Updated topic.
	Creating Orders from PO	Updated topic.
Winter 2019 (1903)	About Order Management	Updated topic.
	Overview	Updated topic.
	Document Setup	Deleted topic. Will be added back when scenarios are documented.

Release	Topic	Description
	API Supported Packages	Updated topic. Updated table to reflect the latest package versions.
Summer 2019 (1902)	Executing Order Workflow	New topic. New feature for this release.
Spring 2019 (1901)	No updates	No new APIs are introduced in this release. The guide is updated to reflect the product name change.
Winter 2018 (1803)	No updates	No new APIs are introduced in this release. The guide is updated to reflect the product name change.

## About Order Management

An order is a document that serves as a confirmation of a purchase created for a customer before delivering goods or services. Apttus Order Management allows customer sales and customer support representatives manage the life of the generated order before it creates/updates an asset. Order Lifecycle Management offers a common, streamlined process for managing orders regardless of the channel used for the creation (direct sales, partner sales, telesales, digital commerce, and Electronic Data Interchange (EDI) ) through the lifecycle of the order through fulfillment.

The three main stages of the lifecycle are:

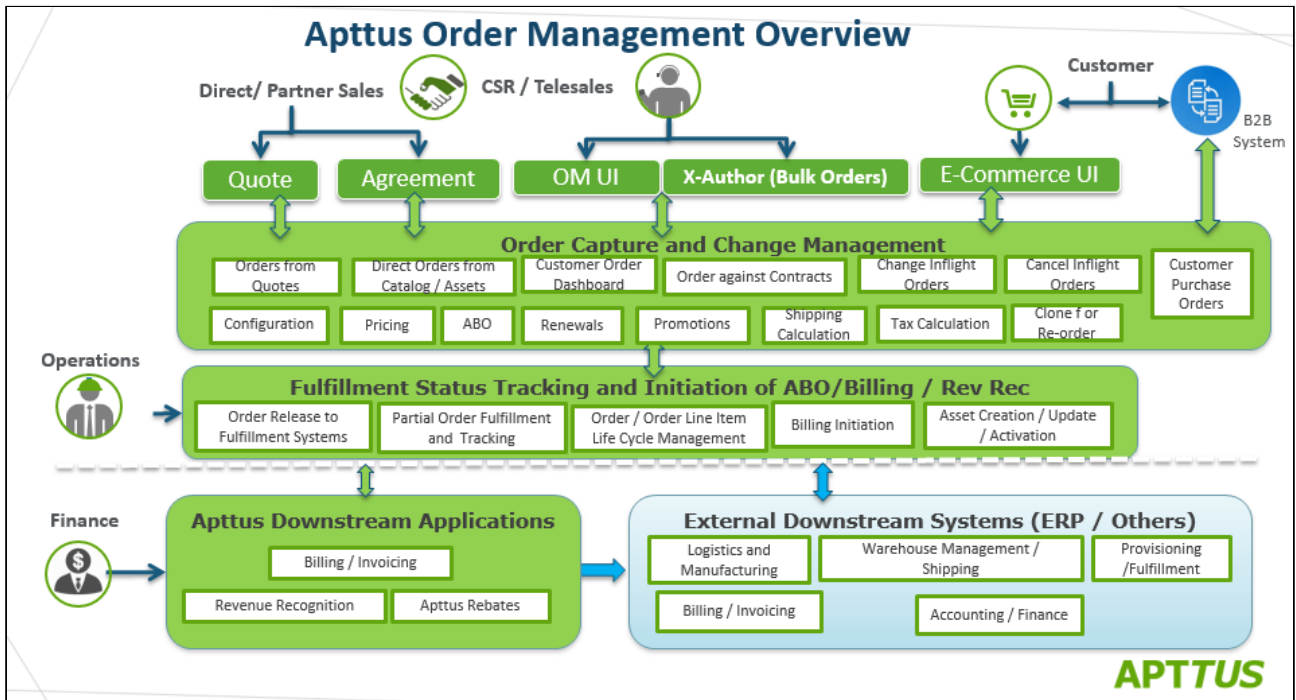
- Order Capture: Order Capture: Capturing direct orders through the Order Management interface, creating customer purchase orders from a quote, and creating digital commerce orders, including orders submitted through Max Apttus AI.
- Order Change: Change management allows users to make changes to configuration, pricing, and shipping, add promotions, change or cancel in-flight orders, and reorder before the order has created or updated an asset.
- Order Fulfillment Tracking: Order release to multiple fulfillment systems and locations. Order fulfillment can be tracked and managed at the order and order line item level.

Apttus Order Management allows users to integrate with external systems, including ERP, logistics and manufacturing, accounting and finance, and warehouse management.

Advanced features within Order Management include design of custom validation and enrichment rules for purchase orders, as well as automation of in-flight order changes.

The following diagram shows the interaction with different entities at various stages in the omni-channel sales process.





Apttus Order Management allows a user to perform the following tasks:

- Managing orders in the Quote-to-Cash Process
- Managing order lifecycle and status
- Creating order management workflows
- Capturing orders
- Creating partial orders from customer price agreement (Quote/Contract)
- Managing in-flight order changes and cancellation
- Creating distributed order fulfillment
- Activating an order
- Billing for an order
- Creating customer purchase orders

Apttus currently supports order creation through direct and partner sales and digital commerce interfaces. From the Order Management interface, customer support representatives can create orders on behalf of customers.

## Key Terminology

It is important to understand how terms are used when working with Apttus Order Management.

Term	Description
ABO	Asset-based ordering (ABO) functionality enables the customers to manage their existing subscriptions or install base using actions such as change, renew, swap, and terminate.
Administrators	Individual responsible for installing, configuring, and maintaining Order Management software, including creating direct orders, customer purchase orders, and managing partial order fulfillment and automation.
Assets	Assets define a purchased product or service. An asset is associated with an account. After being processed and fulfilled, the line items associated with new quotes, agreements, or orders result in the creation of new assets, which can then be viewed or managed from the customer's account.
Amend Order	An action a user can take from the order record page. You can use Amend Order to create a new version of the Order classified as an amendment with changes to the field values of the order.
Attributes	Features of a product, such as color, size, weight, and more.
Bundled Products	A combination of standalone products that offer added value to the customer while increasing overall sales.
Cart	A product and pricing view for the user to review all configuration and pricing information at a glance.
Catalog	A view that allows hierarchical categorization of products for users to search through and add to their configuration.
Clone	To replicate a field, record, template, etc.

Term	Description
Clone Order	An action a user can take from the order record page which creates a copy of the order record.
Contract Price List	A price list that helps the customer keep track of specific price agreements as applicable for that account.
Customer Purchase Order	A customer document that represents the initial offer of negotiated types, quantities, and prices for products or services. Customer Purchase Orders can be created by sales users or administrators as sales orders for long-term contracts.
In-Flight Order	The flow status of an order that is in the process of fulfillment.
Order	A document that serves as a confirmation of a purchase created for a customer before delivering goods or services. The Order object is used in Order Management to capture and track orders created from quotes, agreements, or other sources, such as customer purchase orders.
Order Fulfillment Line Items	A line item on the order record that comprises order fulfillment information for the record. Order fulfillment line items are displayed in the Order Fulfillment Line Items related list.
Order Line Item	A line item on the order record corresponding to a product, bundle, or service that is part of the order. Order line items are displayed in the Order Line Items related list.
Order Line Item Status	A column on the Order Line Items related list. This status refers to the current status of the corresponding line item. The possible status values are: PartiallyFulfilled, InFulfilment, Pending, Draft, Accepted, In-Amendment, BeingAmended, Superseded, Activated, Pending Cancellation, BeingCancelled, Cancelled, and Fulfilled.
Order Number	The unique ID of an order.
Order Source	A field on the order record that specifies the source from which an order is created. For example: Account, Quote, or an Agreement.
Order Status	A field on the order record that denotes the status of an order that you have created or updated. The possible status values are: PartiallyFulfilled, InFulfilment, Pending, Draft, Accepted, In-Amendment, BeingAmended, Superseded, Activated, PendingCancellation, BeingCancelled, Cancelled, and Fulfilled.

Term	Description
Partial Order	An order placed against a price agreement or customer purchase order for a partial amount of the committed quantities captured for an order.
Price Agreement	An agreement associated with a generated contract price list that represents the negotiated pricing and quantity commitments for one or more configurations of products and services.
Price Lists	Containers of items that are grouped in a price list. A price list controls which products are visible to the end user. A price list contains several price list items; each linked to a product. A product can be set up with one or more price list items.
Price Matrix (Matrices)	These are an advanced pricing concept used to define tiered pricing paradigms, or complex pricing structures with multiple criteria. Common examples are pricing tiers for a product based on user count or particular customer or transactional dimensions.
Price Rule	Represents a single rule in a price ruleset.
Price Rule Sets	Price Rulesets are a mechanism to allow particular families, categories or groupings of products to have either line item pricing adjustments applied or summary pricing adjustments applied. Typical examples of these are volume discounting rules or promotional pricing rules.
Product	A product or service that can be sold on its own as a standalone item, an option of other products, or as part of a bundled product.
Product Attribute Group	Represents a product attribute group that contains attributes shared by products.
Product Attribute Value	Represents the attribute values for a product class. For example, color has attribute values such as red, green, blue, orange and so on.
Product Group	A logical grouping of one or more product records. This construct allows you to create combinations of products with similar characteristics/qualities for use in a Rule.

<b>Term</b>	<b>Description</b>
Quote or Proposal	A structured definition of a prospective sale that contains product configurations, pricing, and customer opportunity information.
Standalone Products	Refers to a device or software that is self-contained, one that does not require any other devices or software to function.

# Getting Started

Execution of Apttus Order Management requires a working knowledge of API standards and development platform, as well as the subset of supported data and field types in Salesforce. Refer to the *Order Management on Salesforce Release Notes* for the supported Order Management package for the latest release.

Refer to the following topics:



## 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	<a href="http://www.w3.org/TR/2000/NOTE-SOAP-20000508">http://www.w3.org/TR/2000/NOTE-SOAP-20000508</a>
Web Service Description Language (WSDL) 1.1	<a href="http://www.w3.org/TR/2001/NOTE-wsdl-20010315">http://www.w3.org/TR/2001/NOTE-wsdl-20010315</a>
WS-I Basic Profile 1.1	<a href="http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html">http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html</a>

### Development Platforms

Apttus SOAP APIs work with standard SOAP development environments. For a list of compatible development platforms, see [Salesforce Developer Force API](#) details.

**Note**

A later version of this content will contain sample code.

## Field Types

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

The following table lists the APIs that Apttus provides. For a comprehensive list of all field types supported by Salesforce, see [Salesforce Data Types](#).

Type	Description
Boolean	The <b>Boolean</b> field has a <code>true</code> (or 1) or <code>false</code> (or 0) value.
Data object	The <b>Data Object</b> field is an ID type and is represented by <code>CPQ.nno0</code> in this document.
Date	The <b>Date</b> field contains date values only and does 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 <b>DateTime</b> field.
Decimal	The <b>Decimal</b> field provides an exact numeric value and you can arbitrarily size the precision and scale of the value.
ID	The <b>ID</b> 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 <code>priceListId</code> . The ID field acts like foreign keys and their values can be changed using an <code>update()</code> call.
Integer	The <b>Integer</b> field contains whole numbers only. There are no digits after the decimal.
List	The <b>List</b> 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 <b>String</b> field contains text and may have different length restrictions based on the data you store in the specific field. For instance, <code>City</code> may be limited to 50 characters, while <code>AddressLine1</code> is limited to 255 characters.

## API Reference

The following set of Order Management APIs are divided into two main web service categories:

- [Order Web Service](#): Use Order Web Service APIs to create and manage orders and execute order workflow automation rules.
- [Customer Purchase Order \(CPO\) Web Service](#): Use the CPO Web Service APIs to manage the customer purchase order lifecycle and create sales orders from purchase orders.

## Order Web Service

Use the Order Web Service APIs to manage the order lifecycle, create direct orders, and create partial orders from Price Agreements.

Use the Order Web Service APIs to complete the following tasks:

- [Creating an Order](#)
- [Creating a Cart for an Order](#)
- [Synchronizing a Cart to an Order](#)
- [Accepting an Order](#)
- [In-Flight Order Changes and Cancellation](#)
- [Cloning an Order](#)
- [Fetching Price Agreements](#)
- [Fetching Price Agreement Items](#)
- [Creating a Partial Order from a Quote or Agreement](#)
- [Creating a Direct Order](#)
- [Executing Order Workflow](#)

## Creating an Order

`createOrder()`

Creates an order for a specific Account and Price List.

## API Details



Parameters		
Name	Type	Description
request	CPQStruct.CreateOrderRequestDO	The request data object.

Request Data Object - CPQStruct.CreateOrderRequestDO			
Field	Type	Required?	Description
AccountId	ID	Yes	The Id of the Account to be associated with the order.
PriceListId	ID	Yes	The Id of the PriceList to be associated with the order. Only the products associated with the Price List appear on the cart.
PricingDate	Datetime	No	Date and time associated with the order and represents the date when the order will be priced.

## API Response

Response Data Object - CPQStruct.CreateOrderResponseDO		
Field	Type	Description
orderId	ID	The Id of the newly created order object

## Code Sample

The following code sample enables you to create an order for a valid account with an Account Id and a Price List with a valid Price List Id. Using the sample, you can search for a valid account using an account number. If an account exists with the account number entered, you can create an order using the `createorder()` API. You can invoke this API in use cases when you want to create an order based on the account and price list.

```

1  public void createOrder()
2  {
3      // Create the request
4      Apttus_Config2.CPQStruct.CreateOrderRequestDO request = new
      Apttus_Config2.CPQStruct.CreateOrderRequestDO();
5
6      // Add request parameters
7      request.AccountId = accountId;
8      request.PriceListId = priceListId;
9      request.PricingDate = Datetime.now();
10
11     // Create a new order
12     Apttus_Config2.CPQStruct.CreateOrderResponseDO result =
      Apttus_Config2.OrderWebService.createOrder(request);
13
14     // Assign orderId to local variable
15     orderId = result.Orders0.Id;
16 }

```

## Creating a Cart for an Order

### createCart()

This creates a cart for the order referenced by OrderId.

## API Details

Parameters		
Name	Type	Description
request	CPQStruct.CreateCartRequestDO	The request data object.

Request Data Object - CPQStruct.CreateCartRequestDO			
Field	Type	Required?	Description
OrderId	ID	Yes	The Id of the order to be associated with the cart.
Properties	List<Config2.Property>	No	The list of properties applicable to the cart.

Data Object - Config2.Property			
Field	Type	Required?	Description
Name	String	Yes	Specify the features applicable to the cart. Applicable values are: <ul style="list-style-type: none"> <li>• useAdvancedApproval: Enables Advanced Approval for a cart.</li> <li>• useDealOptimizer: Enables Deal Optimizer for a cart.</li> </ul>
Value	String	No	The applicable values are true or false. Specifying the value as true enables the feature for a cart.

## API Response

Response Data Object - CPQStruct.CreateCartResponseDO		
Field	Type	Description
CartId	ID	The Id of the newly created cart object.

## Code Sample

The following code sample enables you to create a cart for a valid order with an Order Id. Using the sample, you can search for a valid order using an order number. If an order exists

with the order number entered, you can create a cart using `createCart()`. You can invoke this API in use cases when you want to show a cart page based on the order.

```

1  public void createCart()
2  {
3      if(String.isNotBlank(orderId))
4      {
5          // Create config properties
6          List<Apttus_Config2.Property> configProps = new
List<Apttus_Config2.Property>();
7
8          Apttus_Config2.Property prop = new Apttus_Config2.Property();
9
10         prop.Name = 'useAdvancedApproval';
11         prop.Value = 'false';
12         configProps.add(prop);
13
14         prop = new Apttus_Config2.Property();
15         prop.Name = 'useDealOptimizer';
16         prop.Value = 'false';
17         configProps.add(prop);
18
19         // Create the request
20         Apttus_Config2.CPQStruct.CreateCartRequestDO request = new
Apttus_Config2.CPQStruct.CreateCartRequestDO();
21
22         // Add request parameters
23         request.OrderId = orderId;
24         request.Properties.addAll(configProps);
25
26         // Create a new cart for Order
27         Apttus_Config2.CPQStruct.CreateCartResponseDO result =
Apttus_Config2.OrderWebService.createCart(request);
28
29         // Get the cart Id
30         cartId = result.CartId;
31     } else {
32         ApexPages.addMessage(new
ApexPages.Message(ApexPages.severity.info, 'Invalid or Blank Order
Id. '));
33     }
34 }

```

# Synchronizing a Cart to an Order

## SynchronizeCart()

Sync shopping cart line items with order line items. Before you synchronize your cart add products using `addMultiProducts()`, add bundles using `addBundle()`. The API will generally be invoked when the user clicks the Place Order action from the cart.

## API Details

Parameters		
Name	Type	Description
request	CPQStruct.SynchronizeCartRequestDO	The request data object.

Request Data Object - CPQStruct.SynchronizeCartRequestDO			
Name	Type	Required?	Description
CartId	ID	Yes	The Id of the cart (Product Configuration) you want to synchronize.

## API Response

Response Data Object - CPQStruct.SynchronizeCartResponseDO		
Name	Type	Description
IsSuccess	Boolean	Indicates whether synchronizing the cart items was successful.

## Code Sample

The following code sample enables you to synchronize cart items with an order. When the customer has finalized the cart, you can synchronize the products in the cart with the order used to generate the cart. Invoke this API after the user has finalized the cart.

```

1  public void synchronizeCart()
2  {
3      if(String.isNotBlank(cartId))
4      {
5          // Create the request
6          Apttus_Config2.CPQStruct.SynchronizeCartRequestDO request2 = new
Apttus_Config2.CPQStruct.SynchronizeCartRequestDO();
7
8          // Add request parameters
9          request2.CartId = cartId;
10
11         // Synchronize cart
12         Apttus_Config2.CPQStruct.SynchronizeCartResponseDO result2 =
Apttus_Config2.OrderWebService.synchronizeCart(request2);
13
14     } else {
15         ApexPages.addMessage(new
ApexPages.Message(ApexPages.severity.info, 'Invalid or blank cart Id.));
16     }
17 }

```

## Accepting an Order

### AcceptOrder()

Finalize a draft order on customer confirmation. Accepting an order performs the following functions:

- IF Auto Activate Order = False
  - Order and Order Line Item status is set to "Pending".
  - Asset Line Items are created in "Pending" status.
  - If "Ready for Billing Date" is set, billing schedule generation is initiated.
- IF Auto Activate Order = True
  - Order and Order Line Item status is set to "Activated".
  - Asset Line Items are created in "Activated" status.
  - If "Ready for Billing Date" is set, the billing schedule generation is initiated.

## API Details

Parameters		
Name	Type	Description
Request	Apttus_Config2.CPQStruct.AcceptOrderRequestDO	The request data object.

Request Data Object - Apttus_Config2.CPQStruct.AcceptOrderRequestDO			
Field	Type	Required?	Description
OrderId	ID	Yes	The Id of the Order to be finalized.

## API Response

Response Data Object - CPQStruct.AcceptOrderResponseDO		
Field	Type	Description
IsSuccess	boolean	Specifies whether the order is finalized successfully.



Response Data Object - CPQStruct.AcceptOrderResponseDO		
Field	Type	Description
AssetLineItems	List<Apttus_Config2__AssetLineItem__c>	List of Asset Line Items

## Code Sample

The following code sample enables you to finalize an order for a valid order with an Order Id.

```

1  public void acceptOrder() {
2
3      Apttus_Config2.CPQStruct.AcceptOrderRequestDO request = new
4      Apttus_Config2.CPQStruct.AcceptOrderRequestDO();
5
6      request.OrderId = orderId;
7
8      Apttus_Config2.CPQStruct.AcceptOrderResponseDO response =
9      Apttus_Config2.OrderWebService.acceptOrder(request);
10
11     ApexPages.addMessage(new
12     ApexPages.Message(ApexPages.severity.info, 'acceptOrder :' +
13     response.IsSuccess));
14
15     ApexPages.addMessage(new
16     ApexPages.Message(ApexPages.severity.info, 'List of Assets :' +
17     response.AssetItems));
18
19 }

```

## In-Flight Order Changes and Cancellation

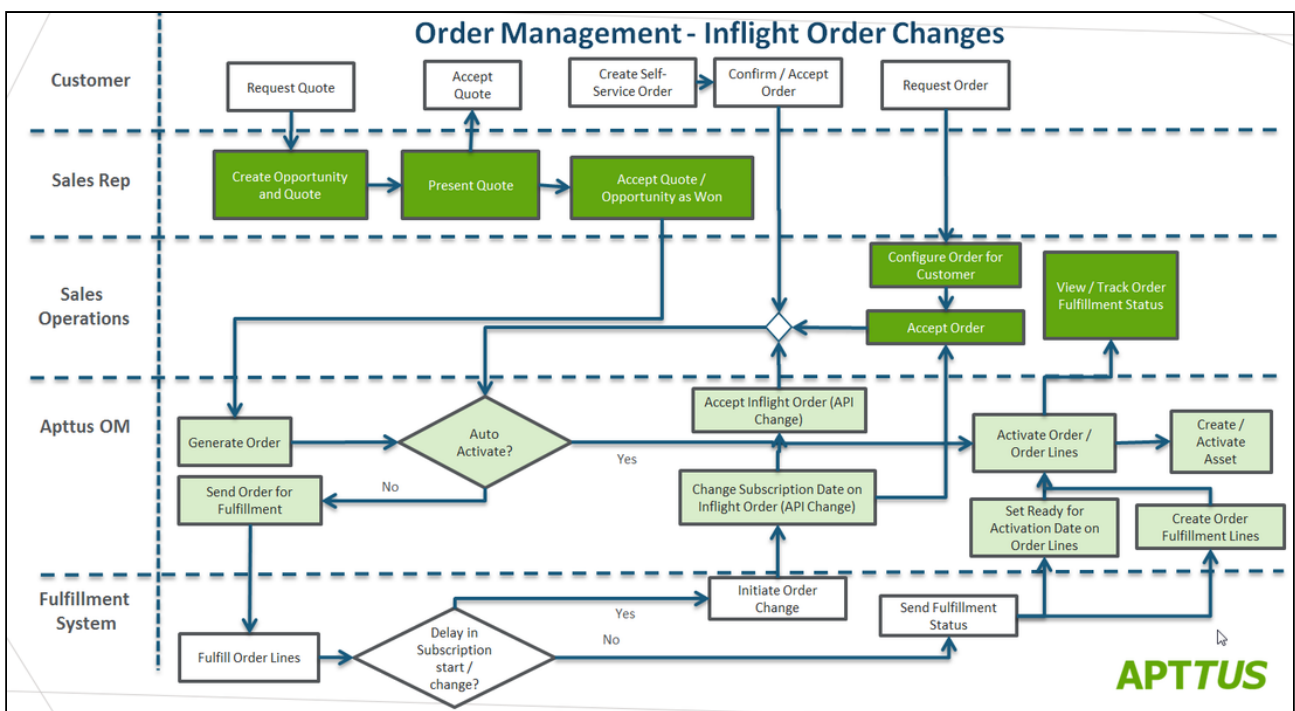
In-Flight Order Changes and Cancellation allows you to change or cancel orders after order confirmation and before fulfillment. Changes to orders come in the form of versioned amendments, to existing order line items that include modification of quantity, subscription dates, discounts, and other pricing parameters. You can achieve the following using the APIs:

- Add a new order line item

- Change an existing order line item
- Revert the changes if required
- Cancel an order and its order line items
- Revert cancellation of an order

Using the AmendOrder() API, you can make subscription date changes on order line items of an "In Flight" order (an order with the status Pending, In Fulfillment, or Partially Fulfilled). For example, you may want to amend an order in a scenario when there is a delay in provisioning or fulfillment of the subscription after the order has been confirmed by the customer and before it is fulfilled.

The following diagram visually represents the workflow for In-Flight Order Changes



Refer to the *Order Management on Salesforce Administrator Guide* for complete steps to configure In-Flight Order Changes and Cancellation. At a minimum, the "Enable In-Flight Changes and Cancellation" custom setting must be enabled for any of the following APIs to function. You can also create workflow automation rules for in-flight order changes and cancellation using the `execWorkflow()` API.

## In-Flight Changes and Cancellation APIs

Invoke the following APIs to make In-Flight Order Changes to orders (amendments) or to cancel an order.



## Order and Order Line Item Behavior on Amendment

Action	Behavior
In-flight Order changes allowed	<ul style="list-style-type: none"> <li>• Change to Start Date and/or End Date</li> <li>• Change the term as needed based on new start date/end date</li> <li>• Recalculate Net Price as needed</li> <li>• Add a new Product (order line) for the existing order: For instance, after placing an order for salesforce licenses, you can add more products for the SFDC Admin Training.</li> <li>• Cancel an order line: Cancel a particular order line from an existing order and keep rest of the order lines unchanged.</li> <li>• Change an existing order line for the existing order: Change the quantity or start/end date for a particular order line on an existing order. Price and Term are recalculated after changing quantity and dates (start/End) respectively.</li> </ul>
Order Versioning	<ul style="list-style-type: none"> <li>• New order and order line version created to make the in-flight change</li> <li>• The previous version superseded once the new change is confirmed</li> </ul>
Delayed Asset Creation	<ul style="list-style-type: none"> <li>• Option added to "Create Assets on Order Activation" in Order System Properties. This option needs to be enabled for In-flight change capability.</li> <li>• With this option, assets are created and activated only when the order/order line is activated</li> <li>• For bundles / multi-charge types, assets created/ updated when last related option or charge type is fulfilled</li> </ul>

Action	Behavior
Billing Changes	<ul style="list-style-type: none"> <li>• Billing can only be initiated after order/order line activation</li> <li>• "Initiate Billing on Order Activation" is not relevant for In-flight Order changes and is always treated as if the property is enabled.</li> </ul>
Revenue Recognition	<ul style="list-style-type: none"> <li>• No impact on Revenue Recognition as scheduled, after activation</li> </ul>
<p>DRAFT or IN AMENDMENT Order</p> <p>Setting Ready for Activation Date or (Ready for Billing Date) on Order Line Item on orders in Draft / In Amendment Order</p>	<p>When the orders are in "Draft" or "In Amendment" status, users can set "Ready for Activation Date" or "Ready for Billing Date" on the order or order line item level. This will not initiate order/order line item activation or billing while order stays in "Draft" or "In Amendment" status.</p> <p>If "Ready for Activation Date" is set on the order or order line item then this will become a trigger to activate order or order line item respectively when the order is accepted.</p>

Action	Behavior
<p>ACCEPT ORDER / QUOTE</p> <p>Order line item activation on Accept Order or Quote</p>	<p>When the order or quote is accepted, the system uses the following behavior on the order line items irrespective of delayed asset creation flag.</p> <p>READY FOR ACTIVATION DATE ON THE ORDER HEADER</p> <p>Mark all the line items as Fulfilled</p> <p>Initiate logic for order line item activation, which will activate all the line items as well as order header. It will also create and activate assets.</p> <p>INDEPENDENT LINE ITEM:</p> <p>If ready for activation date is set on the order line item and status of the order line is in "In Amendment" or "Draft", then system will mark the line item as "Fulfilled" and initiate "order line item activation logic</p> <p>BUNDLES OR MULTI-CHARGE TYPES WHEN Create Asset on Order Activation = TRUE</p> <p>If ready for activation date is set and status of the order line is in "In Amendment" or "Draft", then system will mark the line item as "Fulfilled"</p> <p>Initiate activation logic, if the current fulfilled item is the last item in the bundle or multi-charge items</p> <p>BUNDLES OR MULTI-CHARGE TYPES WHEN Create Asset on Order Activation = FALSE</p> <p>If ready for activation date is set and status of the order line is in "In Amendment" or "Draft", the system will mark the line item as "Fulfilled"</p> <p>Initiate activation even if there are other items in the bundle or multi-charge types that are still NOT in "Fulfilled" status</p>

Action	Behavior
<p>PENDING / IN FULFILLMENT / PARTIALLY FULFILLED ORDERS</p> <ul style="list-style-type: none"> <li>Activating Order Line Items</li> </ul>	<p>READY FOR ACTIVATION DATE ON THE ORDER HEADER</p> <ul style="list-style-type: none"> <li>Mark all the line items as Fulfilled</li> <li>Initiate logic for order line item activation</li> </ul> <p>INDEPENDENT LINE ITEM:</p> <ul style="list-style-type: none"> <li>If ready for activation date is set on the order line item and status of the order line is in "Pending" or "In Fulfillment", then the system will mark the line item as "Fulfilled" and initiate "order line item activation logic"</li> <li>If Fulfilled Quantity &gt;= Delta Quantity and status of the order line is in "Pending" or "In Fulfillment" or "Partially Fulfilled", then mark the line item as "Fulfilled" and initiate "order line item activation logic"</li> </ul> <p>BUNDLES OR MULTI-CHARGE TYPES WHEN Create Asset on Order Activation = TRUE</p> <ul style="list-style-type: none"> <li>If ready for activation date is set and status of the order line is in "Pending" or "In Fulfillment", then mark the line item as "Fulfilled"</li> <li>If Fulfilled Quantity &gt;= Delta Quantity and status of the order line is in "Pending" or "In Fulfillment" or "Partially Fulfilled", then mark the line item as "Fulfilled"</li> <li>Initiate activation logic, if the current fulfilled item is the last item in the bundle or multi-charge items</li> </ul> <p>BUNDLES OR MULTI-CHARGE TYPES WHEN Create Asset on Order Activation = FALSE</p> <ul style="list-style-type: none"> <li>If ready for activation date is set and status of the order line is in "Pending" or "In Fulfillment", then mark the line item as "Fulfilled"</li> <li>If Fulfilled Quantity &gt;= Delta Quantity and status of the order line is in "Pending" or "In Fulfillment" or "Partially Fulfilled", then mark the line item as "Fulfilled"</li> </ul>

Action	Behavior
	<ul style="list-style-type: none"> <li>Initiate activation logic, even if there are other items in the bundle or multi-charge types that are still NOT in "Fulfilled" status.</li> </ul>

## Use Cases for Changing In-Flight orders

This section highlights the use cases when you would make changes to an In-Flight order, For example, delay in the provisioning of Subscription from a confirmed order, impacts the subscription effective dates, duration, and Price. Accordingly on provisioning, change In-Flight order, order line item, asset, and other related data.

<p><b>New Subscription</b></p> <p>Change in subscription dates without changing Term/Duration or Quantity</p>	<p>New subscription start is delayed by a month due to provisioning delay.</p> <p>Ordered subscription: 01-Jan-2017 to 31-Dec-2017</p> <p>Provisioned subscription: 01-Feb-2017 to 31-Jan-2018</p>
<p><b>Amend Subscription</b></p> <p>Change amendment start date without changing the End Date and Quantity</p>	<p>Provisioning of an add-on order on an existing subscription is delayed by a month.</p> <p>Ordered subscription: 01-Aug-2017 to 31-Jan-2018</p> <p>Provisioned subscription: 01-Sep-2017 to 31-Jan-2018</p>
<p><b>Terminate Subscription</b></p> <p>Change termination date of subscription</p>	<p>Termination of an existing subscription is delayed by a month due to provisioning delay</p> <p>Ordered subscription Termination: End Date: 31-Oct-2017</p> <p>Provisioned subscription Termination: End Date: 30-Nov-2017</p>

<p><b>Swap Subscription</b></p> <p>Change start date of upgrade and thus change termination date of original subscription</p>	<p>Start date of the upgrade of an existing subscription is delayed by a month due to provisioning delay. Accordingly, termination of existing subscription also need to be delayed</p> <p>Ordered subscription Upgrade: Gold Warranty: End Date: 31-July-2017 Platinum Warranty: 01-Aug-2017 to 31-Jan-2018</p> <p>Provisioned subscription Upgrade: Gold Warranty: End Date: 31-Aug-2017 Platinum Warranty: 01-Sep-2017 to 31-Jan-2018</p>
<p><b>Add a new order line</b></p> <p>Addition of n number of products to the existing order</p>	<p>Addition of Dimond Warranty to the existing order</p> <p>Add new order line to the existing order by adding Dimond warranty</p>
<p><b>Cancel an existing Order Line</b></p>	<p>Canceling Gold Warranty for the existing order</p> <p>Gold Warranty Status is <i>Cancelled</i>.</p>
<p><b>Change an existing order line</b></p>	<p>Provisioning of an add-on order on an existing subscription is shifted by a month and quantity needs to be increased from 1 to 2</p> <p>Ordered subscription: 01-Aug-2017 to 31-Jan-2018</p> <p>Provisioned subscription: 01-Sep-2017 to 31- Feb -2018 (Quantity: 2)</p>



<p><b>Calculate the end date by providing a start date, selling term and selling frequency</b></p>	<p>Provide your input for start date, selling term, selling frequency to the order line item and ensure that the end date on order line item must be null while passing the API.</p> <p>Code Sample:</p> <pre>for(Apttus_Config2_OrderLineItem_c oli: orderLineList) { oli.Apttus_Config2__EndDate__c = null; oli.Apttus_Config2__StartDate__c = oli.Apttus_Config2__StartDate__c.addMonths(4); oli.Apttus_Config2__SellingTerm__c = 24; oli.Apttus_Config2__SellingFrequency__c = 'Monthly'; oli.Apttus_Config2__Quantity__c = oli.Apttus_Config2__Quantity__c; }</pre>
<p><b>Calculate the selling term by providing a start date, end date, and selling frequency</b></p>	<p>Provide your input for start date, end date, selling frequency to the order line item and ensure that the selling term on order line item must be "0" while passing the API.</p> <p>Code Sample:</p> <pre>for(Apttus_Config2_OrderLineItem_c oli: orderLineList) { oli.Apttus_Config2__SellingTerm__c = 0; oli.Apttus_Config2__EndDate__c = oli.Apttus_Config2__EndDate__c.addMonths(4); oli.Apttus_Config2__StartDate__c = oli.Apttus_Config2__StartDate__c; oli.Apttus_Config2__SellingFrequency__c = 'Monthly'; oli.Apttus_Config2__Quantity__c = oli.Apttus_Config2__Quantity__c; }</pre>

## Amending an Order

### AmendOrder()

Make In-Flight order changes. Use this API to make the following changes to Order Line Items (OLI):

- Change the start date or end date of subscription on one or more order line items that are in "Pending", "In Fulfillment", or "Partially Fulfilled" status.
- Make subscription date changes on the standalone item, bundles, and multiple charge line items.

- Cancel order line items in "Pending," "In Fulfillment," or "Partially Fulfilled" status. This changes their status to "Pending Cancellation."
- Override the "Base Price" field.
- Change the Quantity.
- Change the Selling Term, Selling Frequency, and/or Selling Unit-of-Measure (Selling Uom).
- Modify Contract Numbers
- Change the Adjustment Type/Adjustment Amount.

The following changes to an order occur after it has been amended:

- The change in subscription start and/or end date recalculates the price for the line item or bundle after the changes are applied.
- The changes to the order line items are also reflected in the corresponding assets.
- Assets are only created on activation of the order line item.
- Creates a new version of the order to make and track changes and has reference to the previous version. The previous version of the order and order line items are "Superseded", once the work-in-progress (In Amendment) changes are confirmed on "Accept" of the "In Amendment" order.
- Cancelled order line items are assigned the "Pending Cancellation" status. The order must be accepted to confirm cancellation.



- Order line items that are added from catalog (line status = "New") or created through ABO actions (line status In "Amended", "Cancelled", "Renewed", "Upgraded") can be amended.
- Any order configured directly or generated from quote or eCommerce can be amended.

For more information on amending an order, refer to the topic "Managing In-Flight Changes and Cancellation" in the *Order Management on Salesforce User Guide*.

## API Details

Parameters		
Name	Type	Description
request	Apttus_Config2.CPQStruct.AmendOrderRequestDO	The request data object.

Request Data Object - Apttus_Config2.CPQStruct.AmendOrderRequestDO			
Field	Type	Required?	Description
OrderId	ID	Yes	<p>Reference to the order. Order Id can be any one of the following:</p> <ul style="list-style-type: none"> <li>Order in "Pending, In Fulfillment, Partially Fulfilled" status that requires amendment.</li> <li>Order in "In Amendment" status for which additional line item changes are required.</li> </ul>
OrderItems	List	No	<p>List of order line items and changes to make:</p> <ul style="list-style-type: none"> <li>Order line item id: Reference to the order line item</li> <li>Start Date: New Start date for the order line item. Not needed for the order line item if the termination date for existing subscription needs to be changed.</li> <li>End Date: New end date for the order line item. If the termination date needs to be changed on existing subscription then just specify the new end date.</li> <li>Contract Numbers: If the termination date needs to be changed on existing subscription then set this value to null or same as End Date.</li> <li>Base Price: New base price for the order line item.</li> <li>Quantity: New quantity for the order line item (see sample below).</li> <li>Selling Term / Selling Frequency: The new Selling Term or Selling Frequency for an order line item.</li> <li>Selling Unit-of-Measure: The new Selling Uom for an order line item.</li> <li>Adjustment Type / Adjustment Amount: The new price Adjustment Type and/or Adjustment amount for the order line item.</li> </ul>

## API Response

Response Data Object - CPQStruct.AmendOrderResponseDO		
Field	Type	Description
AmendedOrderItems	List	List of amended order items
AmendOrderSO	Standard Data Object	The fields of the AmendOrder Object.

## Code Sample

The following code sample enables you to amend an order for a valid order with an Order Id.

```

1  Id orderid = 'a2n4C0000007NFpQAM';
2  List<Apttus_Config2__OrderLineItem__c> lineitemlist =
3
4  [ Select Id, Name, Apttus_Config2__StartDate__c,Apttus_Config2__EndDate__c
5  from Apttus_Config2__OrderLineItem__c
6  where
7  Id IN ('a2k4C000000D7yt', 'a2k4C000000D7yu', 'a2k4C000000D7yv', 'a2k4C00000
8  0D7yw')];
9
10         lineitemlist[0].Apttus_Config2__StartDate__c=date.newInstance
11         e(2017, 10, 01);
12         lineitemlist[0].Apttus_Config2__EndDate__c=date.newInstance(2
13         018, 8, 31);
14         /*****For line item, where cancellation date needs to be
15         changed, provide the end date and set Contract Numbers field to
16         null*****/
17         lineitemlist[1].Apttus_Config2__EndDate__c=date.newInstance
18         (2018, 9, 30);
19         lineitemlist[1].Apttus_Config2__ContractNumbers__c=null;
20
21         /****to change the quantity of line item ****/
22         lineitemlist[2].Apttus_Config2__Quantity__c=3;
23
24         /***Canceling the line item by changing the status of line
25         item to Pending cancellation ****/
26         lineitemlist[3].Apttus_Config2__Status__c='Pending
27         Cancellation';
28
29         Apttus_Config2.CPQStruct.AmendOrderRequestDO request = new
30         Apttus_Config2.CPQStruct.AmendOrderRequestDO();
31         request.OrderId = orderid; request.OrderItems =
32         lineitemlist;
33         Apttus_Config2.CPQStruct.AmendOrderResponseDO
34         Amend_response = Apttus_Config2.OrderWebService.amendOrder (request);
35         system.debug('OrderID_Of_New_Version
36         -'+Amend_response.AmendOrderSO.id); /**This ID should be passed While
37         Processing Undo Amend request **/
38
39         /***** Accepting new order *****/
40         Apttus_Config2.CPQStruct.AcceptOrderRequestDO request1 =
41         new Apttus_Config2.CPQStruct.AcceptOrderRequestDO();

```

```


30         /*Apttus_Config2__Order__c newOrder = [ Select Id, Name,
Apttus_Config2__ActivatedDate__c from Apttus_Config2__Order__c
31         where Apttus_Config2__PreviousVersion__c = :orderid];
32
33         request1.OrderId = newOrder.Id; /** you can either use id
for above query **/
34         OR */
35         request1.OrderId=Amend_response.AmendOrderS0.id // you can
use the id of new orderversion from above response
36
37
38         /* ***** updating ready for activation date on the
order or order line items for auto activation of the amended lines *****/
39
40         //newOrder.Apttus_Config2__ActivatedDate__c = system.now();
41
42         List<Apttus_Config2__OrderLineItem__c> lineitemlist1 =
43
44         [Select Id, Name, Apttus_Config2__ActivatedDate__c from
Apttus_Config2__OrderLineItem__c where Apttus_Config2__OrderId__c = :
45         .Id and Apttus_Config2__Status__c = 'In Amendment'];
46         lineitemlist1[0].Apttus_Config2__ActivatedDate__c =
system.now();
47         lineitemlist1[1].Apttus_Config2__ActivatedDate__c =
system.now();
48
49         update lineitemlist1;
50
51         Apttus_Config2.CPQStruct.AcceptOrderResponseD0 response =
Apttus_Config2.OrderWebService.acceptOrder(request1);

```

## Reverting an Amendment to an Order

### UndoAmendOrder()

Enables you to revert an amendment to an order. For example, you can roll back the changed order date to the previous date. UndoAmendOrder is composite API.

 For more information on how an order and its line items are affected when you revert an amendment to an order, refer to the *Order Management on Salesforce User Guide*.

## API Details

Parameters			
Name	Type	Description	
request	Apttus_Config2.CPQStruct.UndoAmendOrderResponseDO	The request data object.	
Request Data Object - Apttus_Config2.CPQStruct.UndoAmendOrderRequestDO			
Field	Type	Required?	Description
orderId	ID	Yes	Unique identifier of the amended order.

## API Response

Response Data Object - CPQStruct.UndoAmendOrderResponseDO		
Field	Type	Description
PreviousOrderSO	Standard Data Object	The fields of the Previous Order Object.

## Code Sample

Use the following code sample to revert an amendment to an order.

```

1  Apttus_Config2.CPQStruct.UndoAmendOrderRequestDO Undo_Amend_request = new
    Apttus_Config2.CPQStruct.UndoAmendOrderRequestDO();
2  Undo_Amend_request.orderid = 'a2n4C0000007NGJ'; /** This is the Id of the
    order whose status is 'In Amendment' **/
3  Apttus_Config2.CPQStruct.UndoAmendOrderResponseDO UndoAmend_response =
    Apttus_Config2.OrderWebService.undoAmendOrder(Undo_Amend_request);
4  system.debug('Order_OLD_Version - ' +
    UndoAmend_response.PreviousOrderSO.id); /** returns the OrderId of the
    previous version **/

```

## Cancelling an Order

### CancelOrder()

Cancels an order. The status of the order and its line items changes to *Pending Cancellation* and a new version is created. The status of the previous order is set to *Being Cancelled*. Please note the following:

- If you decide to revert (undo) the cancelled order, the new order version is deleted and the previous order version shows *Pending* status.
- If you decide to cancel the order, you must accept the cancellation of the order. After accepting cancellation, a new version is created showing with *Cancelled* status for the order as well as its line items. In this case, the previous version shows *Superseded* status for the order and its line items.

**i** For more information on how an order and its line items are affected when you cancel an order, refer to the *Order Management on Salesforce User Guide*.

## API Details

Parameters		
Name	Type	Description
request	Apttus_Config2.CPQStruct.CancelOrderRequestDO	The request data object.

Request Data Object - Apttus_Config2.CPQStruct.CancelOrderRequestDO			
Field	Type	Required?	Description
orderId	ID	Yes	Unique identifier of the order to be cancelled.



## API Response

Response Data Object - CPQStruct.CancelOrderResponseDO		
Field	Type	Description
cancelOrderSO	Standard Data Object	The fields of the Cancel Order Object.

## Code Sample


The following code sample allows you to cancel an order.

```
ID Orderid = 'a2n4C0000007NFpQAM';
Apttus_Config2.CPQStruct.CancelOrderRequestDO CancelRequest = new
    Apttus_Config2.CPQStruct.CancelOrderRequestDO();
CancelRequest.orderid=Orderid;
Apttus_Config2.CPQStruct.CancelOrderResponseDO Cancel_Response =
    Apttus_Config2.OrderWebService.cancelOrder(CancelRequest);
system.debug('OrderID_New_Version - '+Cancel_Response.CancelOrderSO.id); /**This ID
should be passed While Processing Undo Cancel request **/
```

## Reverting a Cancelled Order

### UndoCancelOrder()

Revert the cancellation of an order.

 For more information on how an order and its line items are affected when you revert a cancelled order, refer to the *Order Management on Salesforce User Guide*.

## API Details

Parameters		
Name	Type	Description
Request	Apttus_Config2.CPQStruct.UndoCancelOrderRequestDO	The request data object.

Request Data Object - Apttus_Config2.CPQStruct.UndoCancelOrderRequestDO			
Field	Type	Required?	Description
orderId	ID	Yes	Unique identifier of the cancelled order.

## API Response

Response Data Object - CPQStruct.UndoCancelOrderResponseDO		
Field	Type	Description
PreviousOrderSO	Standard Data Object	The fields of the Previous Order Object.

## Code Sample

The following API lets you undo or revert the cancelled order.

```
Apttus_Config2.CPQStruct.UndoCancelOrderRequestDO UndoCancelrequest = new
    Apttus_Config2.CPQStruct.UndoCancelOrderRequestDO();
UndoCancelrequest.orderid=Cancel_Response.CancelOrderSO.id;
Apttus_Config2.CPQStruct.UndoCancelOrderResponseDO UndoCancel_Response=
Apttus_Config2.OrderWebService.undoCancelOrder (UndoCancelrequest);
system.debug('OrderID_OLD_Version - '+UndoCancel_Response.PreviousOrderSO.id);/**
returns the OrderId of the previous version **/
```

# Cloning an Order

## CloneOrder()

Perform a deep clone of an order. Creates a new copy of the order along with its related objects rather than creating a new version. Deep clone can only be performed for orders where all order line items have the status "New." Any order with ABO (Asset-Based Ordering) line items cannot be cloned in this way.

The following applies when the CloneOrder() API is invoked:

- Cloning an existing order sets the status of the order header and order line items to "Draft".
- Cloning an existing order resets the following fields: Asset Line Item, Ready for Activation Date, and Ready for Billing Date.

- Clones the corresponding cart and cart line items. Cart line items can then be configured and change similar to new draft order.

## API Details

Parameters		
Name	Type	Description
request	Apttus_Config2.CPQStruct.CloneOrderRequestDO	The request data object.

Request Data Object - Apttus_Config2.CPQStruct.CloneOrderRequestDO			
Field	Type	Required?	Description
OrderId	ID	Yes	The order Id to clone. This is the order number that remains constant from version to version and is referred by the customer.

## API Response

Response Data Object - CPQStruct.CloneOrderResponseDO		
Field	Type	Description
OrderItems	List	List of cloned order items.
CloneOrderSO	Object	The fields of the Clone Order object.

## Code Sample

The following code sample enables you to clone an order for a valid order with an Order Id.

```

1  /***** Clone an order *****/
2  Id orderid = 'a2n3C000000CfAC';
3  Apttus_Config2.CPQStruct.CloneOrderRequestDO request = new
    Apttus_Config2.CPQStruct.CloneOrderRequestDO();
4  request.OrderId = orderid;
5
6
7  Apttus_Config2.CPQStruct.CloneOrderResponseDO response
    =Apttus_Config2.OrderWebService.cloneOrder (request);
    
```

## Fetching Price Agreements

### GetPriceAgreements()

Fetches price agreements based on proposals or agreements.

### API Details

Parameters		
Name	Type	Description
request	CPQApiStruct.GetPriceAgreementsRequestDO	The request data object.

Request Data Object - CPQApiStruct.GetPriceAgreementsRequestDO			
Field	Type	Required?	Description
PriceAgreementType	String	Yes	The type of price agreement. Valid Values: Proposal, Agreement.
Criteria	CPQApiStruct.FilterDo	No	The criteria details like API agreement name and proposal. This is based on FilterDo object.
Fields	List<String>	No	The list of fields to be displayed in the response data object based on the price agreement type.

Request Data Object - CPQApiStruct.GetPriceAgreementsRequestDO			
Field	Type	Required?	Description
AccountIds	List<Id>	No	The account IDs passed to fetch the price agreements. This field narrows down the price agreements based on the filter of account IDs.

### CPQApiStruct.FilterDO

This is an API object that feeds to criteria variable of below request objects:

- CPQApiStruct.GetPriceAgreementsRequestDO
- GetPriceAgreements CPQApiStruct.GetPriceAgreementItemsRequestDO

Request Data Object - CPQApiStruct.FilterDO			
Field	Type	Required?	Description
SObjectName	String	Y	API name of the subject valid values are: Apttus_Proposal__Proposal__c, Apttus__APTS_Agreement__c
Predicates	List<CPQApiStruct.PredicateDO>	Y	List of conditions in where clause
ConditionExpr	String	N	To define the complex logic among multiple conditions.  Example for 2 conditions (1 OR 2)

### CPQApiStruct.PredicateDO

This API defines a filter based on which criteria specifications are set. List of this needs to feed Predicates variable of FilterDO object type variable.

Parameters		
Name	Type	Description
request	CPQApiStruct.PredicateDO	The request data object.

Request Data Object - CPQApiStruct.PredicateDO			
Field	Type	Required?	Description
RowNum	Integer	Y	The row numbers.
FieldName	String	Y	The API name of the field.
CompOper	String	Y	The type of query comparator. Valid values: equal to, not equal to, less than, less than or equal to, greater than, greater than or equal to, in, not in, like, not like, includes, excludes, starts with, ends with, contains, does not contain.
FieldValue	String	Y	The value based on which data should be fetched.

## API Response

Response Data Object - CPQApiStruct.GetPriceAgreementsResponseDO		
Field	Type	Description
PriceAgreements	List<sObject>	The list of price agreements fetched based on the specified input fields.

## Code Sample

The following sample enables you to create an order for a valid account with an Account Id and a Price List with a valid Price List Id. Using the sample below, you can search for a valid account using an account number. If an account exists with the account number entered, you can create an order using the `createorder()` API. You can invoke this API in use cases when you want to create an order page based on the account and price list.

```

1  Apttus_Config2.CPQApiStruct.GetPriceAgreementsRequestDO request = new
    Apttus_Config2.CPQApiStruct.GetPriceAgreementsRequestDO();
2
3  // possible values Proposal,Agreement
4  request.PriceAgreementType = 'Proposal';
5
6  Apttus_Config2.CPQApiStruct.FilterDO criteriaSpec = new
    Apttus_Config2.CPQApiStruct.FilterDO();
7
8  // Proposal,Agreement subjectApi name
9  criteriaSpec.SObjectName = 'Apttus_Proposal__Proposal__c';
10
11 // setup predicates
12 Apttus_Config2.CPQApiStruct.PredicateDO predicate = new
    Apttus_Config2.CPQApiStruct.PredicateDO();
13 predicate.RowNum = 1;
14 predicate.FieldName = 'Name';
15 predicate.CompOper = '=';
16 predicate.FieldValue = 'Q-11122';
17 criteriaSpec.Predicates.add(predicate);
18 request.Criteria = criteriaSpec;
19
20 List<String> fieldNames = new List<String>{'Id','Name'};
21 request.Fields.addAll(fieldNames);
22
23
24 List<Id> accIds = new List<Id>();
25 accIds.add('0015000000Fn6m0');
26
27 request.AccountIds.addAll(accIds);
28
29 //select Id, Name From Apttus_Proposal__Proposal__c Where
    Apttus_Proposal__Approval_Stage__c =:Accepted
30
31 Apttus_CPQApi.PriceAgreementApiCallback apiCall = new
    Apttus_CPQApi.PriceAgreementApiCallback();
32 Apttus_Config2.CPQApiStruct.GetPriceAgreementsResponseDO result =
    apiCall.getPriceAgreements(request);
33 List<SObject> PriceAgreements = new List<SObject>();
34
35 if (!result.PriceAgreements.isEmpty())
36 PriceAgreements = result.PriceAgreements;

```



```
37 system.debug(PriceAgreements);
```

## Fetching Price Agreement Items

### GetPriceAgreementItems()

Fetches price agreement items for a specified price agreement type.

## API Details

Parameters			
Name	Type	Description	
request	CPQApiStruct.GetPriceAgreementItemsRequestDO	The request data object.	
Request Data Object - CPQApiStruct.GetPriceAgreementItemsRequestDO			
Field	Type	Required?	Description
PriceAgreement Type	String	Yes	The type of price agreement. Valid Values: Proposal, Agreement.
Criteria	CPQApiStruct.FilterDo	No	The criteria details like API agreement name and proposal. This is based on FilterDo object.
fieldNames	List<String>	No	The list of fields to be displayed in the response data object based on the price agreement type.
priceAgreement Numbers	List<String>	No	The list of price agreements numbers.

### CPQApiStruct.FilterDO

This is an API object that feeds to criteria variable of below request objects:

- CPQApiStruct.GetPriceAgreementsRequestDO
- GetPriceAgreements CPQApiStruct.GetPriceAgreementItemsRequestDO

Request Data Object - CPQApiStruct.FilterDO			
Field	Type	Required?	Description
SObjectNa me	String	Y	API name of the subject valid values are: Apttus_Proposal__Proposal__c, Apttus__APTS_Agreement__c
Predicates	List<CPQApiStruct.P redicateDO>	Y	List of conditions in where clause
ConditionE xpr	String	N	To define the complex logic among multiple conditions.  Example for 2 conditions (1 OR 2)

### CPQApiStruct.PredicateDO

This API defines a filter based on which criteria specifications are set. List of this needs to feed Predicates variable of FilterDO object type variable.

Parameters		
Name	Type	Description
request	CPQApiStruct. PredicateDO	The request data object.

Request Data Object - CPQApiStruct.PredicateDO			
Field	Type	Required?	Description
RowNu m	Integer	Y	The row numbers.
FieldNa me	String	Y	The API name of the field.

Request Data Object - CPQApiStruct.PredicateDO			
Field	Type	Required?	Description
CompOper	String	Y	The type of query comparator. Valid values: equal to, not equal to, less than, less than or equal to, greater than, greater than or equal to, in, not in, like, not like, includes, excludes, starts with, ends with, contains, does not contain.
FieldValue	String	Y	The value based on which data should be fetched.

## API Response

Response Data Object - CPQApiStruct.GetPriceAgreementItemsResponseDO		
Field	Type	Description
PriceAgreementItems	List<sObject>	The list of price agreement items fetched based on the specified input fields.

## Code Sample

```

1  Apttus_Config2.CPQApiStruct.GetPriceAgreementItemsRequestDO request = new
    Apttus_Config2.CPQApiStruct.GetPriceAgreementItemsRequestDO ();
2  // possible values Proposal,Agreement
3  request.PriceAgreementType = 'Proposal';
4
5  Apttus_Config2.CPQApiStruct.FilterDO criteriaSpec = new
    Apttus_Config2.CPQApiStruct.FilterDO();
6  // subject name
7
8  criteriaSpec.SObjectName = 'Apttus_Proposal__Proposal_Line_Item__c';
9  // setup predicates
10 Apttus_Config2.CPQApiStruct.PredicateDO predicate = new
    Apttus_Config2.CPQApiStruct.PredicateDO();
11 predicate.RowNum = 1;
12 predicate.FieldName =
    'Apttus_Proposal__Proposal__r.Apttus_Proposal__Approval_Stage__c';
13 predicate.CompOper = 'equal to';
14 predicate.FieldValue = 'Accepted';
15
16 Apttus_Config2.CPQApiStruct.PredicateDO predicate2 = new
    Apttus_Config2.CPQApiStruct.PredicateDO();
17 predicate2.RowNum = 1;
18 predicate2.FieldName =
    'Apttus_Proposal__Proposal__r.Apttus_Proposal__ExpectedStartDate__c';
19 predicate2.CompOper = '=';
20 predicate2.FieldValue = ':Apttus_QPConfig__StartDate__c';
21
22 criteriaSpec.Predicates.add(predicate2);
23
24 criteriaSpec.ConditionExpr = '1 AND 2';
25
26 List<String> fieldNames = new List<String>{'Id','Name'};
27 request.Fields.addAll(fieldNames);
28
29 List<String> priceAgreementNumbers = new List<String>();
30 priceAgreementNumbers.add('Q-00002175');
31
32 // contract numbers
33 request.PriceAgreementNumbers.addAll(priceAgreementNumbers);

```

```


34
35
36
37 Apttus_CPQApi.PriceAgreementApiCallback apiCall = new
    Apttus_CPQApi.PriceAgreementApiCallback();
38 Apttus_Config2.CPQApiStruct.GetPriceAgreementItemsResponseDO result =
    apiCall.getPriceAgreementItems(request);
39 system.debug(result.PriceAgreementItems);

```

## Creating a Partial Order from a Quote or Agreement

### CreateOrderRequest()

Creates a partial order from a quote or agreement.

 For detailed information on partial orders, refer to the *Order Management on Salesforce User Guide*.

## API Details

Parameters			
Name	Type	Description	
request	CPQApi.CPQ.CreateOrderRequestDO	The request data object.	
Request Data Object - CPQApi.CPQ.CreateOrderRequestDO			
Field	Type	Required?	Description
CustomFields	List<String>	No	List of custom fields on orderSO
OrderInput	Apttus_Config2__Order__c	Yes	OrderSO
OrderItems	List<Apttus_CPQApi.CPQ.SelectedProductDO>	No	List of order items

Request Data Object - CPQApi.CPQ.CreateOrderRequestDO			
Field	Type	Required?	Description
OrderItemSource	String	Yes	Source of order items
Properties	List<Apttus_Config2.Property>	No	List of properties for cart created for order
Request Data Object - CPQApi.CPQ.SelectedProductDO			
Field	Type	Required?	Description
AttributeValues	List<Apttus_Config2__ProductAttributeValue__c>	No	Product attribute value for the product
Comments	String	No	Comments
CopyBundleConfigurationFromSource	Boolean	No	For bundles, to copy bundle configuration
CustomData	Apttus_Config2__LineItem__c	No	Custom data line item
CustomFields	List<String>	No	List of custom fields
EndDate	Date	No	End date
ProductId	Id	No	Product Id
Quantity	Decimal	No	Quantity
RelatedLineItems	List<Apttus_Config2__RelatedLineItem__c>	No	List of related line items
SellingTerm	Decimal	No	Selling term
SourceFields	List<String>	No	List of source custom fields to be copied

Request Data Object - CPQApi.CPQ.SelectedProductDO			
Field	Type	Required?	Description
SourceId	Id	No	Id of a source item
StartDate	Date	No	Start Date

## Code Sample

The following sample uses an explicit product definition for order items while creating partial order from quote/agreement.

```

1  Apttus_CPQApi.CPQ.CreateOrderRequestDO request
2  = new Apttus_CPQApi.CPQ.CreateOrderRequestDO();
3
4  // order item source
5
6  request.OrderItemSource = Apttus_CPQApi.CPQ.ITEMSOURCE_QUOTE_LINEITEM;
7
8
9  // create order input
10
11 Apttus_Config2__Order__c inputSO = new Apttus_Config2__Order__c();
12
13
14 inputSO.Apttus_Config2__SoldToAccountId__c = '0014C00000F9Zz3';
15
16 inputSO.Apttus_Config2__PriceListId__c = 'a174C0000001MQ1';
17
18 inputSO.Apttus_Config2__PricingDate__c = Datetime.now();
19
20 inputSO.Apttus_Config2__OrderDate__c = Datetime.now();
21
22 inputSO.Apttus_Config2__OrderStartDate__c = Date.today();
23
24 inputSO.Apttus_Config2__OrderEndDate__c = Date.today().addMonths(12);
25
26 // add custom fields to orderSO (optional)
27
28
29 inputSO.put('Apttus_QPConfig__ProposalId__c', 'a0Y4C0000021Cm5');
30
31
32 request.OrderInput = inputSO;
33
34 request.CustomFields.add('Apttus_QPConfig__ProposalId__c');
35
36 // add order items (optional)
37
38
39 // selected item
40
41
42 Apttus_CPQApi.CPQ.SelectedProductDO productDO
43 = new Apttus_CPQApi.CPQ.SelectedProductDO();

```



```
44
45
46 productDO.Comments = 'Test Create Order API';
47
48
49 productDO.CopyBundleConfigurationFromSource = false;
50
51 productDO.StartDate = Date.today();
52
53 productDO.EndDate = Date.today().addMonths(12);
54
55 productDO.SellingTerm = 12;
56
57 productDO.Quantity = 1;
58
59 productDO.ProductId = '01t4C000001pB1J';
60
61 // add attributes to the selected product (optional)
62
63
64 Apttus_Config2__ProductAttributeValue__c
65 attVal = new Apttus_Config2__ProductAttributeValue__c();
66
67
68 attVal.APTS_TestAttColor__c = 'Black';
69
70
71 productDO.AttributeValues.add(attVal);
72
73 // add related line items to selected product (optional)
74
75
76 Apttus_Config2__RelatedLineItem__c
77 lineItem = new Apttus_Config2__RelatedLineItem__c();
78
79
80 lineItem.Apttus_Config2__RelationType__c = 'Wallet';
81
82
83 productDO.RelatedLineItems.add(lineItem);
84
85
86 // source id and the explicit product definition are mutually exclusive
87
88
```

```
89 // productD0.SourceId = 'a0W4C000001MRRF';
90
91
92 request.OrderItems.add(productD0);
93
94 // order properties (optional)
95
96
97 // request.Properties.add(new
98 Apttus_Config2.Property('useAdvancedApproval' 'false'));
99
100
101 // request.Properties.add(new Apttus_Config2.Property('useDealOptimizer'
102 'false'));
103
104
105 // request.Properties.add(new Apttus_Config2.Property('retId',
106 'a0Y4C0000021Cm5'));
107
108
109 // request.Properties.add(new Apttus_Config2.Property('cntrNbr_1',
110 'Q-00004442'));
111
112
113 // create order api call
114
115
116 Apttus_CPQApi.CPQ.CreateOrderResponseD0 result
117 = Apttus_CPQApi.CPQWebService.createOrder(request);
```

### Code Sample

This sample uses an explicit product definition for order items while creating partial order from quote/agreement.

```
1 Apttus_CPQApi.CPQ.CreateOrderRequestDO request
2 = new Apttus_CPQApi.CPQ.CreateOrderRequestDO();
3
4 // order item source
5
6 request.OrderItemSource = Apttus_CPQApi.CPQ.ITEMSOURCE_QUOTE_LINEITEM;
7
8 // create order input
9
10 Apttus_Config2__Order__c inputSO = new Apttus_Config2__Order__c();
11
12 inputSO.Apttus_Config2__SoldToAccountId__c = '0014C00000F9Zz3';
13
14 inputSO.Apttus_Config2__PriceListId__c = 'a174C0000001MQ1';
15
16 inputSO.Apttus_Config2__PricingDate__c = Datetime.now();
17
18 inputSO.Apttus_Config2__OrderDate__c = Datetime.now();
19
20 inputSO.Apttus_Config2__OrderStartDate__c = Date.today();
21
22 inputSO.Apttus_Config2__OrderEndDate__c = Date.today().addMonths(12);
23
24
25 // add custom fields to orderSO (optional)
26
27 inputSO.put('Apttus_QPConfig__ProposalId__c', 'a0Y4C0000021Cm5');
28
29 request.OrderInput = inputSO;
30
31 request.CustomFields.add('Apttus_QPConfig__ProposalId__c');
32
33 // add order items (optional)
34
35
36 // selected item
37
38
39 Apttus_CPQApi.CPQ.SelectedProductDO productDO
40 = new Apttus_CPQApi.CPQ.SelectedProductDO();
41
42 productDO.Comments = 'Test Create Order API';
43
44 productDO.CopyBundleConfigurationFromSource = false;
```

```
45
46 productDO.StartDate = Date.today();
47
48 productDO.EndDate = Date.today().addMonths(12);
49
50 productDO.SellingTerm = 12;
51
52 productDO.Quantity = 1;
53
54 //productDO.ProductId = '01t4C000001pB1J';
55
56
57 // add attributes to the selected product (optional)
58
59
60 Apttus_Config2__ProductAttributeValue__c
61 attVal = new Apttus_Config2__ProductAttributeValue__c();
62
63 attVal.APTS_TestAttColor__c = 'Black';
64
65 productDO.AttributeValues.add(attVal);
66
67
68 // add related line items to selected product (optional)
69
70
71 Apttus_Config2__RelatedLineItem__c
72 lineItem = new Apttus_Config2__RelatedLineItem__c();
73
74
75 lineItem.Apttus_Config2__RelationType__c = 'Wallet';
76
77
78 productDO.RelatedLineItems.add(lineItem);
79
80
81 // source id and the explicit product definition are mutually exclusive
82
83
84 productDO.SourceId = 'a0W4C000001MRRF';
85
86
87 request.OrderItems.add(productDO);
```

```
88
89
90 // add order properties to use contract pricing (optional)
91
92
93 // order properties (optional)
94
95
96 // request.Properties.add(new
97 Apttus_Config2.Property('useAdvancedApproval' 'false'));
98
99
100 // request.Properties.add(new Apttus_Config2.Property('useDealOptimizer'
101 'false'));
102
103
104 // request.Properties.add(new Apttus_Config2.Property('retId',
105 'a0Y4C0000021Cm5'));
106
107
108 request.Properties.add(new Apttus_Config2.Property('cntrNbr_1',
109 'Q-00004442'));
110
111 // Q-00004442 is the contract number in this example (Contract Number
112 field on PriceList object)
113
114
115 // create order api call
116
117
118 Apttus_CPQApi.CPQ.CreateOrderResponseDO result
119 = Apttus_CPQApi.CPQWebService.createOrder(request);
```

## Creating a Direct Order

### CreateOrderRequest

Create a direct order without source Id.

## API Details

Parameters			
Name	Type	Description	
request	CPQApi.CPQ.CreateOrderRequestDO	The request data object.	
Request Data Object - CPQApi.CPQ.CreateOrderRequestDO			
Field	Type	Required?	Description
CustomFields	List<String>	No	List of custom fields on orderSO
OrderInput	Apttus_Config2__Order__c	Yes	OrderSO
OrderItems	List<Apttus_CPQApi.CPQ.SelectedProductDO>	No	List of order items
OrderItemSource	String	Yes	Source of order items
Properties	List<Apttus_Config2.Property>	No	List of properties for cart created for order
Request Data Object - CPQApi.CPQ.SelectedProductDO			
Field	Type	Required?	Description
AttributeValues	List<Apttus_Config2__ProductAttributeValue__c>	No	Product attribute value for the product
Comments	String	No	Comments
CopyBundleConfigurationFromSource	Boolean	No	For bundles, to copy bundle configuration
CustomData	Apttus_Config2__LineItem__c	No	Custom data line item

Request Data Object - CPQApi.CPQ.SelectedProductDO			
Field	Type	Required?	Description
CustomFields	List<String>	No	List of custom fields
EndDate	Date	No	End date
ProductId	Id	No	Product Id
Quantity	Decimal	No	Quantity
RelatedLineItems	List<Apttus_Config2__RelatedLineItem__c>	No	List of related line items
SellingTerm	Decimal	No	Selling term
SourceFields	List<String>	No	List of source custom fields to be copied
SourceId	Id	No	Id of a source item
StartDate	Date	No	Start Date

## Code Sample

```
1  Apttus_CPQApi.CPQ.CreateOrderRequestDO
2  request = new Apttus_CPQApi.CPQ.CreateOrderRequestDO();
3
4
5  // order item source
6
7  request.OrderItemSource = Apttus_CPQApi.CPQ.ITEMSOURCE_QUOTE_LINEITEM;
8
9
10 // create order input
11
12 Apttus_Config2_Orderc
13 inputSO = new Apttus_Config2Order_c();
14
15
16 inputSO.Apttus_Config2_SoldToAccountId_c
17 = '0014C00000F9Zz3';
18
19 inputSO.Apttus_Config2_PricelistId_c
20 = 'a174C0000001MQ1';
21
22 inputSO.Apttus_Config2_PricingDate_c
23 = DateTime.now();
24
25 inputSO.Apttus_Config2_OrderDate_c
26 = DateTime.now();
27
28 inputSO.Apttus_Config2_OrderStartDate_c
29 = Date.today();
30
31 inputSO.Apttus_Config2_OrderEndDate_c
32 = Date.today().addMonths(12);
33
34
35 inputSO.put('Apttus_QPConfig_ProposalId_c', 'a0Y4C0000021Cm5');
36
37 request.OrderInput = inputSO;
38
39 request.CustomFields.add('Apttus_QPConfig_ProposalId_c');
```



```
40
41
42 // add order items (optional)
43
44 // selected item
45
46 Apttus_CPQApi.CPQ.SelectedProductDO productDO = new
47 Apttus_CPQApi.CPQ.SelectedProductDO();
48
49 productDO.Comments = 'Test Create Order API';
50
51 productDO.CopyBundleConfigurationFromSource = false;
52
53 productDO.StartDate = Date.today();
54
55 productDO.EndDate = Date.today().addMonths(12);
56
57 productDO.SellingTerm = 12;
58
59 productDO.Quantity = 1;
60
61 productDO.ProductId = '01t4C000001pB1J';
62
63
64 Apttus_Config2_ProductAttributeValuec
65 attVal = new Apttus_Config2ProductAttributeValue_c();
66
67 attVal.RY_Color__c = 'Black';
68
69
70 productDO.AttributeValues.add(attVal);
71
72 // productDO.SourceId = 'a0W4C000001MRRF';
73
74 request.OrderItems.add(productDO);
75
76
77 // create order
78
79 Apttus_CPQApi.CPQ.CreateOrderResponseDO result =
80 Apttus_CPQApi.CPQWebService.createOrder(request);
```

## Executing Order Workflow

### `execOrderWorkflow()`

Executes an Order Workflow Ruleset (Composite API).

## API Details

Parameters			
Name	Type	Description	
request	Apttus_Config2.CPQStruct.ExecOrderWorkflowRequestDO	The request data object.	
Request Data Object - Apttus_Config2.CPQStruct.ExecOrderWorkflowRequestDO			
Field	Type	Required?	Description
OrderId	ID	Yes	Reference to the order. Order iD can be any one of the following: <ul style="list-style-type: none"> <li>Order in "Pending, In Fulfillment, Partially Fulfilled" status that requires amendment</li> <li>Order in "In Amendment" status for which additional line item changes are required</li> </ul>

## API Response

Response Data Object - Apttus_Config2.CPQStruct.ExecOrderWorkflowResponseDO		
Field	Type	Description
IsSuccess	Boolean	Workflow is a success true/false
JobId	ID	Id of asynchronous apex thread. The status of this job can be tracked as a Salesforce Apex Job

## Code Sample

The following sample enables you to execute an order workflow.

```

1  // id of an order record where status is Pending or In Fulfillment or
    Partially Fulfilled
2  Id orderId = 'a2n2f0000008Swz';
3
4  // request
5
6  Apttus_Config2.CPQStruct.ExecOrderWorkflowRequestDO request = new
7  Apttus_Config2.CPQStruct.ExecOrderWorkflowRequestDO();
8  request.OrderId = orderId;
9
10 // response
11
12 response =
13 Apttus_Config2.OrderWebService.execOrderWorkflow(request);
14 system.debug('Success = '+response.IsSuccess);
15 system.debug('Async Job Id = '+response.JobId);


```

## Customer Purchase Order (CPO) Web Service

Use the CPO Web Service APIs to manage customer purchase orders.

Use CPO Web Service APIs to complete the following tasks:

- [Accepting Customer Purchase Orders](#)
- [Amending Customer Purchase Orders](#)
- [Reverting Amended Customer Purchase Orders](#)
- [Cancelling Customer Purchase Orders](#)
- [Reverting Cancelled Customer Purchase Orders](#)
- [Creating Orders from PO](#)

 While capturing a Customer Purchase Order for Purchase Order and Purchase Order Items, refer to standard Salesforce SOQL APIs.

## Accepting Customer Purchase Orders

### AcceptCustomerPO()

Accepts a Customer Purchase Order in **Draft** status. Changes the status of the Customer Purchase Order and Purchase Order Items to **Accepted**.

## API Details

Parameters		
Name	Type	Description
request	Apttus_Config2.CPQStruct.AcceptCustomerPORequestDO	The request data object.

Request Data Object - Apttus_Config2.CPQStruct.AcceptCustomerPORequestDO			
Field	Type	Re qui red ?	Description
Customer POId	ID	Yes	The Id of the Order to be finalized.

## API Response

Response Data Object - CPQStruct.AcceptCustomerPOResponseDO		
Field	Type	Description
CustomerPurchaseOrderSO	Standard Object	The response data object.
Response Data Object - CustomerPurchaseOrderSO		
Field	Type	Description
IsSuccess	boolean	Specifies whether the order is finalized successfully.

### Code Sample

The following enables you to accept an order for a valid customer purchase order with an Order ID.

```

1 // accept the order
2 CPQStruct.AcceptCustomerPORequestDO request = new
   CPQStruct.AcceptCustomerPORequestDO();
3 request.CustomerPOId = customerPO.Id;
4
5 CPQStruct.AcceptCustomerPOResponseDO result =
   CustomerPOWebService.acceptOrder(request);
6 system.debug(result.CustomerPurchaseOrderSO)

```

## Amending Customer Purchase Orders

### AmendCustomerPO()

Modify or amend a Customer Purchase Order without changing the value of the order line items (Composite API).

## API Details

Parameters			
Name	Type	Description	
request	Apttus_Config2.CPQStruct.AmendCustomerPORequestDO	The request data object.	
Request Data Object - Apttus_Config2.CPQStruct.AmendCustomerPORequestDO			
Field	Type	Required?	Description
CustomerPOId	ID	Yes	Reference to the order. Order Id can be any one of the following: <ul style="list-style-type: none"> <li>Order in "Accept, Partially Processed" status that requires amendment</li> <li>Order in "In Amendment" status for which additional line item changes are required</li> </ul>
CustomerPOItems	List<CustomerPOItem__C>	No	CPO Items to be amended

## API Response

Response Data Object - CPQStruct.AmendCustomerPOResponseDO		
Field	Type	Description
AmendCustomerPO	Standard Data Object	The fields of the AmendOrder Object
AmendedCustomerPOItems	List	List of amended order items

## Code Sample

The following sample enables you to amend an order for a valid order with an Order Id.

```

1  // create the amend request CPQStruct.AmendCustomerPORequestDO request =
   new CPQStruct.AmendCustomerPORequestDO();
2  request.customerPOId = customerPO.ID;
3  request.CustomerPOItems = CustomerPOItems;
4  CPQStruct.AmendCustomerPOResponseDO result =
   CustomerPOWebService.amendCustomerPO(request);
5  system.debug(result.AmendCustomerPOS0);
6  system.debug(result.AmendedCustomerPOItems);

```

## Reverting Amended Customer Purchase Orders

### UndoAmendCustomerPO()

Undo or revert an amended Customer Purchase Order (Composite API). For example, You can roll back the changed order date to the previous date using this API.

## API Details

Parameters		
Name	Type	Description
request	Apttus_Config2.CPQStruct.UndoAmendCustomerPORequestDO	The request data object.

Request Data Object - Apttus_Config2.CPQStruct.UndoAmendCustomerPORequestDO			
Field	Type	Required?	Description
CustomerP Old	ID	Yes	Unique identifier of the amended order.

## API Response

Response Data Object - CPQStruct.UndoAmendCustomerPOResponseDO		
Field	Type	Description
PreviousCustomerPurchaseOrderSO	Standard Data Object	The fields of the Previous Order Object.

## Code Sample

The following sample allows you to roll back an amendment to an order.

```

1 //undoamend customerPurchaseOrder
2 CPQStruct.UndoAmendCustomerPORequestDO request = new
   CPQStruct.UndoAmendCustomerPORequestDO();
3 request.CustomerPOId = CustomerPOId;
4
5 CPQStruct.UndoAmendCustomerPOResponseDO result =
   CustomerPOWebService.undoAmendCustomerPO(request);
6 system.debug(result.PreviousCustomerPurchaseOrderSO)
    
```

## Cancelling Customer Purchase Orders

### CancelCustomerPO()

Cancels a Customer Purchase Order

## API Details

Parameters		
Name	Type	Description
request	Apttus_Config2.CPQStruct.CancelCustomerPORequestDO	The request data object.



Request Data Object - Apttus_Config2.CPQStruct.CancelCustomerPORequestDO			
Field	Type	Required?	Description
CustomerP Old	ID	Yes	Unique identifier of the latest version of the customer purchase order to be cancelled.

## API Response

Response Data Object - CPQStruct.CancelCustomerPOResponseDO		
Field	Type	Description
CancelCustomerPOSO	Standard Data Object	The fields of the Cancel Order Object.

## Code Sample

The following sample allows you to cancel a customer purchase order.

```

1  // Create the Cancel Request Data Object
2  CPQStruct.CancelCustomerPORequestDO request = new
    CPQStruct.CancelCustomerPORequestDO();
3  request.CustomerPOId = customerPO.Id;
4
5  CPQStruct.CancelCustomerPOResponseDO result = new
    CPQStruct.CancelCustomerPOResponseDO();
6  result = CustomerPOWebservice.cancelCustomerPO(request);
7  // returns next version of customerpurchaseOrder
8  system.debug(result.CancelCustomerPOSO);

```

## Reverting Cancelled Customer Purchase Orders

### UndoCancelCustomerPO()

Revert the cancellation of a customer purchase order.

## API Details

Parameters		
Name	Type	Description
request	Apttus_Config2.CPQStruct.UndoCancelCustomerPORequestDO	The request data object.

Request Data Object - Apttus_Config2.UndoCancelCustomerPORequestDO			
Field	Type	Required?	Description
CustomerPOId	ID	Yes	Unique identifier of the latest version of cancelled customer purchase order.

## API Response

Response Data Object - CPQStruct.UndoCancelOrderResponseDO		
Field	Type	Description
PreviousCustomerPurchaseOrderSO	Standard Data Object	The fields of the previous customer purchase order object.

## Code Sample

The following sample allows you to roll back cancellation of a customer purchase order.

```
//undocancel customerPurchaseOrder
CPQStruct.UndoCancelCustomerPORequestDO request = new
    CPQStruct.UndoCancelCustomerPORequestDO();
request.CustomerPOId = CustomerPOId;

CPQStruct.UndoCancelCustomerPOResponseDO result =
    CustomerPOWebService.undoCancelCustomerPO(request);
system.debug(result.PreviousCustomerPurchaseOrderSO);
```

## Creating Orders from PO

### CreateOrderForPO()

Creates a sales order from a Customer Purchase Order: single order, multiple order using split criteria, or a partial order.

**i** You must enable Split Group Properties field on PO item in order to consume this functionality. Refer to the *Order Management on Salesforce Administrator Guide* for steps.

## API Details

Parameter 1: Create Order for Customer Purchase Order Data Object		
Name	Type	Description
request	Apttus_CPQApi.CPQ.CreateOrderForPORequestDO	The request data object.
Request Data Object - CreateOrderForPORequestDO		
Field	Type	Description
orderInputSO	Standard Object	To create the order inputs.
PurchaseOrderId	ID	To create a single order.
SplitGroupFields	List<String>	To group by orders

Request Standard Object - orderInputSO			
Field	Type	Required?	Description
SoldToAccountId	ID	Yes	The id of the Account to which the order is placed.
PriceListId	ID	Yes	The id of the PriceList to be associated with the order.
PricingDate	Datetime	No	Date and time associated with the order and represents the date when the order will be priced
OrderDate	Datetime	No	Date and time when the order is created.
OrderStartDate	Datetime	No	Date and time when the order fulfillment starts.
OrderEndDate	Datetime	No	Date and time when the order fulfillment ends.
useAdvancedApproval	Boolean	No	Whether to use advanced approval. The default value is False.
useDealOptimizer	Boolean	No	Whether to use the Deal Optimizer. The default value is False.

Parameter 2: Create Selected Purchase Order Item Data Object		
Name	Type	Description
request	Apttus_CPQApi.CPQ.SelectedPOItemDO poltemDO	The request data object.

Request Data Object - poltemDO (Required for partial order only)			
Field	Type	Required?	Description
POItemId	ID	Yes	Customer PO item number.
ProductId	ID	Yes	Customer Product item number.

Request Data Object - poltemDO (Required for partial order only)			
Field	Type	Required?	Description
Quantity	Integer	Yes	Quantity of the order.
StartDate	Datetime	No	Date and time when the PO fulfillment starts.
EndDate	Datetime	No	Date and time when the PO fulfillment ends.
Selling Term	Integer	No	Duration of the customer PO.
Comments	Text	No	Any notes
Sourceld	ID	Yes	Source ID of the order. For example, Quote, Contract, or Agreement ID
Comments	Text	No	Comments about the agreement
ContractNumbers	Text	No	Contract number from the agreement

## API Response

Response Data Object - Apttus_CPQApi.CPQ.CreateOrderForPOResponseDO		
Name	Type	Description
OrderId	ID	Order number which is created for customer PO.

## Code Sample

```

1 String ACCOUNT_NAME = 'RY_Account_002';
2 String PRICELIST_NAME = 'RY_PriceList';
3 Integer SELLING_TERM = 12;
4 Id CUSTOMER_PO_ID = 'a440R00000002Tif';
5 String SPLIT_ORDER_FIELD = 'RY_GroupingTest__c';
6 Id PO_ITEM_ID = 'a430R00000002tC';
7 Id SOURCE_ITEM_ID = 'a0W0R0000000JT9F';
8
9 Account myAccount = [SELECT Id FROM Account WHERE Name =: ACCOUNT_NAME];

```

```

10  Apttus_Config2__PriceList__c myPriceList = [SELECT Id FROM
    Apttus_Config2__PriceList__c WHERE Name =: PRICELIST_NAME];
11  Apttus_Config2__CustomerPOItem__c poItem = [SELECT Id, RY_GroupingTest__c,
    Apttus_Config2__ProductId__c, Apttus_Config2__Quantity__c,
    Apttus_Config2__StartDate__c, Apttus_Config2__EndDate__c FROM
    Apttus_Config2__CustomerPOItem__c WHERE Apttus_Config2__PurchaseOrderId__c
    =: CUSTOMER_PO_ID AND Id =: PO_ITEM_ID LIMIT 1];
12
13
14  /*
15
16  Create Order for Customer Purchase Order Data Object */
17  Apttus_CPQApi.CPQ.CreateOrderForPORequestDO request = new
    Apttus_CPQApi.CPQ.CreateOrderForPORequestDO();
18
19  // Create the Order Input
20  Apttus_Config2__Order__c orderInputSO = new Apttus_Config2__Order__c();
21
22  orderInputSO.Apttus_Config2__SoldToAccountId__c = myAccount.Id;
23  orderInputSO.Apttus_Config2__PriceListId__c = myPriceList.Id;
24  orderInputSO.Apttus_Config2__PricingDate__c = DateTime.now();
25  orderInputSO.Apttus_Config2__OrderDate__c = DateTime.now();
26  orderInputSO.Apttus_Config2__OrderStartDate__c = Date.today();
27  orderInputSO.Apttus_Config2__OrderEndDate__c =
    Date.today().addMonths(SELLING_TERM);
28  request.OrderInput = orderInputSO;
29
30
31
32  // Add Order Properties (Optional)
33  request.Properties.add(new Apttus_Config2.Property('useAdvancedApproval',
    'false'));
34  request.Properties.add(new Apttus_Config2.Property('useDealOptimizer',
    'false'));
35
36
37
38  // Assign the Customer Purchase Order (For single order)
39  request.PurchaseOrderId = CUSTOMER_PO_ID;
40
41
42
43  // Add Order Items (Optional)

```

```

44
45 // Selecting Items
46 // Create Selected Purchase Order Item Data Object
47
48 Apttus_CPQApi.CPQ.SelectedPOItemDO poItemDO = new
    Apttus_CPQApi.CPQ.SelectedPOItemDO(); (For partial order)
49
50 //WebService Apttus_Config2__LineItem__c CustomData
51 //WebService List CustomFields
52
53 poItemDO.POItemId = poItem.Id;
54 poItemDO.ProductId = poItem.Apttus_Config2__ProductId__c;
55 poItemDO.Quantity = poItem.Apttus_Config2__Quantity__c;
56 poItemDO.StartDate = poItem.Apttus_Config2__StartDate__c;
57 poItemDO.EndDate = poItem.Apttus_Config2__EndDate__c;
58 poItemDO.SellingTerm =
    Apttus_Config2.CPQWebService.computeTerm(poItemDO.StartDate,poItemDO.EndDate, 'Monthly');
59 poItemDO.Comments = 'Purchase order item test';
60
61
62
63 // Set Additional Parameters From Source Line Item
64 poItemDO.SourceId = SOURCE_ITEM_ID;
65 poItemDO.SourceFields.add('Apttus_QPConfig__Comments__c');
66 poItemDO.SourceFields.add('Apttus_QPConfig__ContractNumbers__c');
67 request.OrderItems.add(poItemDO);
68
69
70
71 // Split Order Criteria
72 //request.SplitGroupFields.add(SPLIT_ORDER_FIELD); (For group by orders)
73
74
75
76 // Create the Order(s)
77 Apttus_CPQApi.CPQ.CreateOrderForPOResponseDO result =
    Apttus_CPQApi.CPQWebService.createOrderForPO(request);
78 system.debug('Order Id = ' + result.OrderId);

```

# Apttus Copyright Disclaimer

Copyright © 2021 Apttus Corporation (“Apttus”) and/or its affiliates. All rights reserved.

No part of this document, or any information linked to or referenced herein, may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written consent of Apttus. All information contained herein is subject to change without notice and is not warranted to be error free.

This document may describe certain features and functionality of software that Apttus makes available for use under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not, in any form, or by any means, use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part of the software. Reverse engineering, disassembly, decompilation of, or the creation of derivative work(s) from, the software is strictly prohibited. Additionally, this document may contain descriptions of software modules that are optional and for which you may not have purchased a license. As a result, your specific software solution and/or implementation may differ from those described in this document.

U.S. GOVERNMENT END USERS: Apttus software, including any operating system(s), integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Neither the software nor the documentation were developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Apttus and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Apttus and X-Author are registered trademarks of Apttus and/or its affiliates.

The documentation and/or software may provide links to Web sites and access to content, products, and services from third parties. Apttus is not responsible for the availability of, or any content provided by third parties. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the



relationship is directly between you and the third party. Apttus is not responsible for: (a) the 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: OMSFSPR20API20200402