**Choose the Right Term**

Subscription products are special because they’re sold as a service that lasts for days, months, or years. Since time is involved, you must consider what unit of time you want to use when quoting subscription products. For example, do you typically quote in terms of months, like a video streaming service? Or do you sell in units of days, like you might when renting industrial equipment? Before we can set up any subscription products, Salesforce CPQ must know if you plan to use months or days to describe subscription terms. This choice is made in the CPQ package settings by setting the subscription term unit that applies to every product. Since the units are set org-wide, it’s not possible for some products to use months as a unit while others use days.

Choosing the right subscription term unit is important because it affects how CPQ calculates prices. It’s a good idea to decide which subscription term unit is best for your business when implementing CPQ, and then never change it. Thankfully most businesses know how they want to quote subscription products, so it’s an easy decision. While you can change the subscription term unit after implementation, we don’t recommend it.

It’s also worth noting that choosing months for the subscription term unit still allows your reps to sell subscriptions for less than one month. CPQ calculates the price for just the portion of the term, not the whole term. This is known as prorated pricing.

Your CPQ-enabled developer org is already configured to use month as the subscription term unit, and it should stay that way! All of our examples and exercises assume a monthly term unit.

## Default Subscription Term

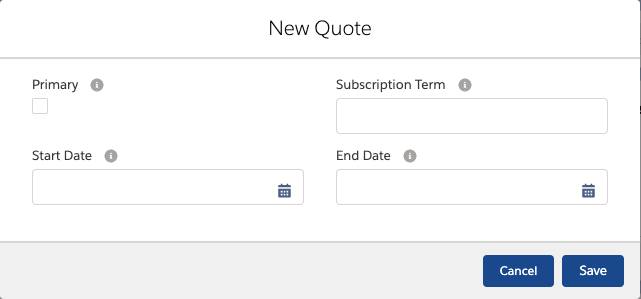
CPQ uses the Default Subscription Term field to calculate subscription pricing. It’s found on the quote line object. You can think of this field as the length of a “normal” subscription for exactly the price book price of the product.

If the product record has a blank value for the Subscription Term field, the Default Subscription Term field will also be blank. However, CPQ treats blank values in these fields as 12 for practical purposes.

Once the quote line is created and the default subscription term value is copied over, all pricing calculations use that quote line value. The Subscription Term field from the product record is no longer in use. This is because the product record might change in the future, and we don't want pricing to mysteriously change while a quote is in progress.

## Many Ways to Define Time

Another important factor in subscription pricing is the length of the subscription. When you created your first quote, you learned that you can set a quote's subscription term to 24, 28, or whatever duration your customer needs. CPQ used the subscription term of 24 to calculate the $60 prorated price of the Password Manager License. However, that’s not the only way to tell CPQ the length of a subscription. You might remember seeing that you can also set a start date and end date for the quote.

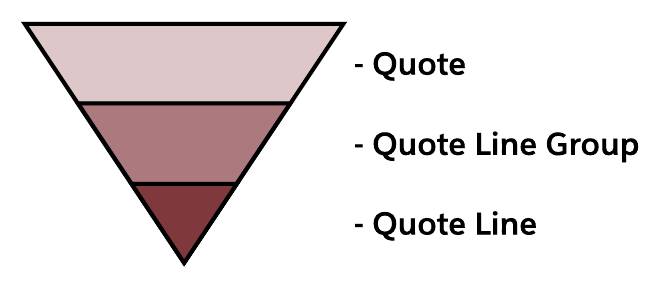


If you’d left the Subscription Term field blank, and instead entered Jan 1, 2020 for the start date, and Dec 31, 2021 for the end date, you would still see the same prices since those dates also describe a duration of 2 years. Sales reps can use whichever method they prefer; the results will be the same.

Although it's common to set subscription terms or start/end dates on the quote, you can also define them on every quote line.

This lets your sales reps tell CPQ the lengths of all subscriptions in a group by just setting one or two fields.

In total, there are nine different fields used to define subscription length. So how does CPQ know which ones to use if many of them are filled in? To understand which have priority, it helps to think of the quote as having three levels: the quote level (the most generic), the quote line group level (a little more specific), and the quote line level (the most specific).



CPQ determines the actual length of a subscription quote line in three stages.

* First, CPQ looks for the most specific start date and the most specific end date and uses those. It’s OK if the start date is from one level (such as quote) and the end date from another (such as line). If either the start date or end date is unknown, it goes to the next stage.
* Next, CPQ looks for the most specific subscription term. If it can’t find a subscription term anywhere, it goes to the last stage.
* Finally, if all else fails, CPQ uses the quote line’s default subscription term (or inputs 12 if the field is blank).

Remember, a given subscription quote line can only have one length. While it may take some detective work to uncover what contributes to its length, you can count on CPQ to always use the above process.

## Prorate Multiplier and Subscription Prorate Precision

So far we've discussed two things that affect subscription pricing of a quote line: the default subscription term and the length of the subscription. CPQ combines these two factors to create what is known as the prorate multiplier, which is the actual number CPQ uses to calculate a prorated price. For example, when you added the Password Manager License to your 24-month quote, CPQ used a prorate multiplier of 2, multiplying $30 by 2 to get $60. The question is, where did the 2 come from?

In this case, the prorate multiplier was calculated by dividing the subscription term (24) by the default subscription term (12). So, 24/12 = 2. When you changed the subscription term to 28, that gave you a new prorate multiplier: 28/12 = 2.3333. That was then used to calculate a new prorated price: $30 x 2.3333 = $70.

The prorate multiplier is easy to calculate if using subscription terms for your subscription lengths. It gets a little more complicated when using start and end dates. That’s because now the subscription length has to account for how many days are in a month, if a leap year is involved, and if the subscription includes a partial month. To handle different business needs, CPQ supports five different methods of calculating the prorate multiplier when using start and end dates, seen in the table below. In this table, DST = default subscription term. The third column shows an example calculation that includes a leap year (2020).

| **Proration Method** | **Math** | **January 10, 2020 to April 15, 2021 with 12-month DST** | **Prorate Multiplier** |
| --- | --- | --- | --- |
| Month | Months (rounded up) / DST | 16/12 | 1.3333 |
| Monthly + Daily | (Months + [Remaining days / {365/12}]) / DST | (15+[6/{365/12}])/12 In this example the first month spans from January 10 to February 9. The last month spans from March 10 to April 9. | 1.2664 |
| Calendar Monthly + Daily | ([Partial days in start month / days in start month] + Whole months + [Partial days in end month / days in end month]) / DST | ([22/31]+14+[15/30])/12 | 1.2675 |
| Day | Days / Number of days necessary to complete DST from start date | 462/366 | 1.2623 |
| Day With Calendar Month Weighted | Years + (Remaining days / X) where X = 366 if remaining days includes February 29, otherwise X = 365 | 1 + (96/365) In this example the first year spans from January 10 to January 9. | 1.2630 |

The **Day method** is unusual for a few reasons.

* If your org’s subscription term unit is Day, you must choose the Day method—all other methods are locked out.
* The package level setting named “Ignore Leap Year Days” appears when the Day method is selected. Checking it tells CPQ to completely remove February 29 from all calculations. That would make the example in the table 461/365 instead of 462/366.
* Two subscriptions with the same length can differ in price if your subscription term unit is Month. Imagine you have a product with a default subscription term of 1. Selling a subscription from March 1 to May 15 gives us a prorate multiplier of 2.4516 (76 days/31 days). We divide the total days by 31 because March has 31 days, so it would take 31 days to complete the default subscription term. But the 76 days spanning from April 1 to June 15 have a prorate multiplier of 76/30, or 2.533. Here we divide by 30 because April only has 30 days. If the subscription started in February, we would have to divide by 28 (or 29 if a leap year).

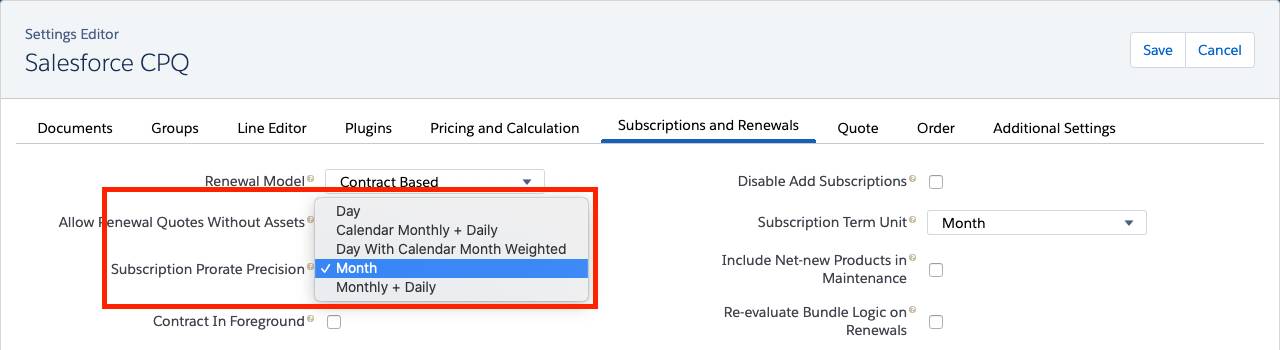
The **Day With Calendar Month Weighted method** also has the option to Ignore Leap Year Days. Selecting it makes the X in the above table 365 for all cases.

Note

#### Note

The **Calendar Monthly + Daily method**, and the **Day With Calendar Month Weighted method** only calculate correctly if your org is set up to use the CPQ Advanced Calculator. To use the Advanced Calculator navigate to Setup > Installed Packages > Configure (for Salesforce CPQ). On the Pricing and Calculation tab, uncheck Use Legacy Calculator, then click the Authorize New Calculation Service link. When prompted, allow the Advanced Calculator to access your org data.

The proration method is set in the CPQ package level setting named Subscription Prorate Precision.



It’s best to keep the proration method the same for the life of your org. If you do change it, be aware that old quotes are NOT retroactively recalculated. Quotes that are recalculated by the reps or by some automated process will use the new proration method.

Wow, proration involves a lot of math! Good thing CPQ runs those calculations for you. You’ve learned how subscriptions are priced. Next you look at how to break a 3-year subscription into three 1-year pieces so each piece can be discounted separately.

## Not Your Typical One-Dimensional Deals

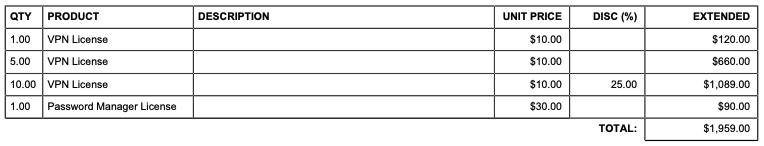
While multidimensional quoting (MDQ) sounds straight out of science fiction, it’s really just breaking a long subscription into smaller pieces, in order to treat each piece differently. For example, you might have a 3-year subscription and you want to discount only the first year. Or maybe you want to change the subscription quantity of just the last year. Sales reps like this flexibility because customers are often willing to commit to longer subscriptions if they can start small and increase their service over time.

Imagine you have a $10 monthly subscription product, VPN License, that you want to sell in yearly segments so sales reps can increase the quantity in the second year. As an admin, start by navigating to the product record.

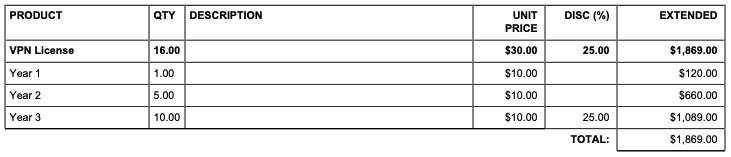
## Proposal Options

Multidimensional quoting allows sales reps to pack in more information about a subscription over its duration. It’s important that your customer can see those MDQ details when they receive a proposal.

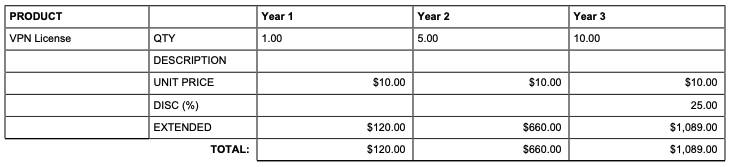
CPQ gives you three out-of-the-box ways to present MDQ subscriptions to your customers. The first treats the multiple quote lines like any other, so they appear as rows in the line items table.



Second, you can tell CPQ to put all MDQ subscriptions into their own specially made table, with a row for each segment, along with a summary row. This is a great option for MDQ subscriptions with lots of segments, like if you sell quarterly segments over a number of years.



Third, you can have CPQ create a special table for each MDQ subscription. It creates a column for each segment, much like how it appears in the Quote Line Editor. Keep in mind that a page is only so wide, so this option works best with fewer segments.



One last thing to note: Any product that is used as the top level of a bundle cannot be enabled for MDQ. That said, MDQ products can be included within bundles as options.

Setting up and selling MDQ subscriptions is easy, but there’s even more you can do to take MDQ to the next level. You learn how in the next unit.

To add fields into the MDQ header, update the Segmented Line Editor Summary field set.  Just be aware that these fields can’t be edited in the Quote Line Editor—they are for informational purposes only.

## Accounting for Partial Segments

In the MDQ examples so far, we’ve had perfectly sized subscription terms so that each segment was a full year. In the real world, you sometimes need to sell subscriptions that have odd lengths like 18 months. So, how does CPQ break 18 months into two segments?

By default, CPQ puts full segments first and the partial segment at the end.

Sometimes customers don’t want the partial segment to appear at the end. Maybe they’re trying to buy a subscription that coincides with existing service that renews July 1, not January 1. In that example, the partial segment would work best in the front.

Thankfully CPQ gives sales reps a way to control where breaks in subscription segments fall. The field named First Segment Term End Date is a mouthful, but it’s how you can shift breakpoints for MDQ subscriptions. This quote-level field tells CPQ where the first segment should end, which affects all subsequent segments.

## Custom Segments

You may remember seeing a field on the Price Dimension record named Type when you set up the VPN License product to support MDQ. You left it as “Year” then, which forced CPQ to create year-long segments. But what if you want to give your sales reps the option of setting their own segment lengths, such as 6 months? You can do that by setting the Type field to Custom.

By default, CPQ does not allow gaps or overlaps in custom segments. For example, if you try to start the second segment a month after the first ends, you receive an error message.

If you ever want your sales reps to have the option to include subscription gaps and overlaps, you must navigate to Setup > Installed Packages > Configure (for Salesforce CPQ) > Pricing and Calculation tab, and change the Allow Non-Consecutive Custom Segments setting to true.

Lastly, regardless of this package setting, be aware that sales reps can edit the start and end dates of custom segments if the fields are included in the layout.

Currently, editing the dates here does NOT produce a validation message if gaps or overlaps exist, even if you don’t allow nonconsecutive custom segments. If this is a concern, either remove the date fields from the layout, or create your own validation rule to check for gaps and overlaps.

## One-time Price Dimensions

While MDQ is mainly used to split a subscription into pieces, it has one more trick to offer. Imagine that anytime you sell a VPN License, there’s a one-time $20 setup fee for the customer regardless of subscription length, and this fee can’t be discounted. You can do that and more using MDQ one-time price dimensions.

## Term Discount Level

Using term-based discount schedules is easy, but there’s one use case that requires a little more consideration. Imagine you have a quote with a 3-year term, and it has a bunch of subscriptions lines. Most subscriptions run the whole 3 years, but your customer only wants 1 year of Password Manager License. So, you change its quote line subscription term to 12.

So here’s the question: Does the Password Manager License qualify for the “25% for 2 years” term discount schedule? The quote is 3 years, but the quote line is only 1...

Before answering the question, let’s review what CPQ does when the quote and line-level subscription terms differ. In a previous unit, you learned that CPQ determines the length of a subscription quote line by first looking for the most specific start and end dates. If CPQ can’t find either of those dates, it looks for the most specific subscription term.

By default, CPQ does the same thing when deciding if a quote line qualifies for a term discount schedule. So in the Password Manager License example, it would NOT get the discount because the 12 months at the line level is not enough.

But wait—you can change what happens! On every product there is a field named Term Discount Level. Basically, this field tells CPQ which level (quote, quote line group, or quote line) is “most specific,” and to ignore lower levels (unless they’re the only source of data). This is a little hard to imagine at first, so let’s look at a few examples.

|  | **Example 1** | **Example 2** | **Example 3** | **Example 4** |
| --- | --- | --- | --- | --- |
| Quote Start/End Dates |  |  | Jan 1, 2020 to Dec 31, 2020 |  |
| Quote Sub Term | 36 |  |  | 36 |
| Group Start/End Dates |  |  |  | Jan 1, 2020 to Dec 31, 2020 |
| Group Sub Term |  |  | 36 |  |
| Line Start/End Dates |  |  | Jan 1, 2020 to Dec 31, 2022 | Jan 1, 2020 to Dec 31, 2020 |
| Line Sub Term | 12 | 12 |  |  |
| **Term Discount Level (TDL)** | **Group** | **Group** | **Group** | **Quote** |
| Recognized Term Length | 36 | 12 | 12 | 36 |
| 25% for 2 years applied? | Yes | No | No | Yes |
| Explanation | The group level is most specific, but it has no data. CPQ ignores data “below” the TDL if there’s data “above” it. In this case it uses the 36 months at the quote level. | The group level is most specific, but it has no data. CPQ uses data “below” the TDL only when there’s nothing else to use. In this case, the 12 months at the line level. | Although group level is most specific, CPQ always prioritizes the most specific dates "above" the TDL. CPQ ignores the dates “below” the TDL. | The quote level is most specific and data exists there. CPQ ignores all data “below” the TDL, even though they’re dates. |

One important thing to note about term discount level is that CPQ looks up the TDL value from the product record every time a recalculation of the quote occurs. Be careful when changing the TDL on a product because the new value is applied every time you recalculate a quote that contains that product. Avoid changing the TDL unless absolutely necessary.

Finally, you might notice that Term Discount Level defaults to a blank value. That’s fine, when blank, it behaves as though you chose “Line,” which is the most specific you can get anyhow.