COMMUNICATIONS

MEDIA & ENTERTAINMENT

ENERGY & UTILITIES

INSURANCE

HEALTH

GOVERNMENT

DIGITAL INTERACTION PLATFORM

CME EPC

Table of Contents

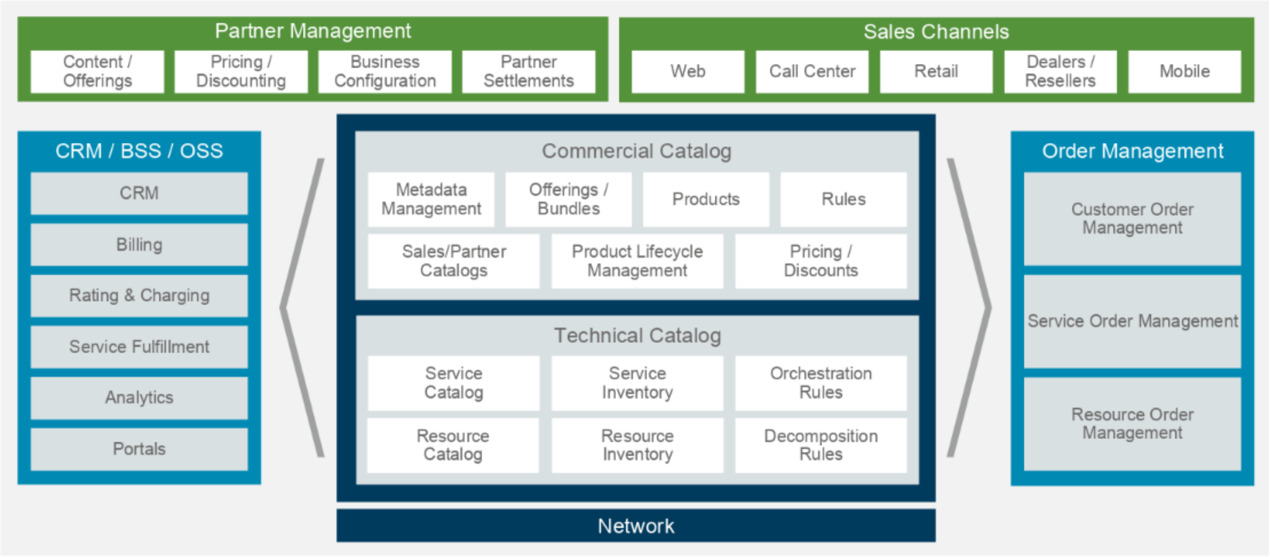
Vlocity Enterprise Product Catalog (EPC)

Vlocity ‘s Enterprise Product Catalog (EPC) enables you to introduce and effectively manage a portfolio of products that are relevant to customers, released at the most opportune time, and at a low cost. EPC helps you manage the end-to-end processes and product data relevant to the product development, product launch, and in-life product management activities of your organization.

Vlocity EPC provides the core components necessary to centralize, configure, integrate, and maintain the product and service portfolio across the enterprise. Vlocity EPC, a centralized catalog, contains all of the commercial and technical elements you use to define reusable product and service building blocks and resulting offers

The following diagram shows where EPC fits into the enterprise. The catalog consists of a commercial and a technical catalog.

Product Catalog in Unified BSS/CRM/OSS Environment



Vlocity Enterprise Product Catalog uses the IDX Workbench build tool to synchronize the product management process and corresponding data into a single, automated, error-resilient stream through reuse, shared resources and common tools:

- Through an enterprise scope, enables the definition of a single product catalog on one place, with multiple subscribers and systems that consume the product catalog.
- Enables the creation of products once, making them available for reuse across the enterprise by different systems.
- Gives you a unified UI with common tools, enabling your teams to collaboratively and rapidly create, manage, and deliver product offers to the market.

- Once implemented, eliminates data silos and includes both commercial and technical entities in a single model.
- Allows you to import products from other systems. For example, if you want to move from a development to a production environment, you can use Vlocity DataPacks and document the import process.
- EPC can be leveraged by different types of users, such as product administrators, product managers, system administrators, and developers.

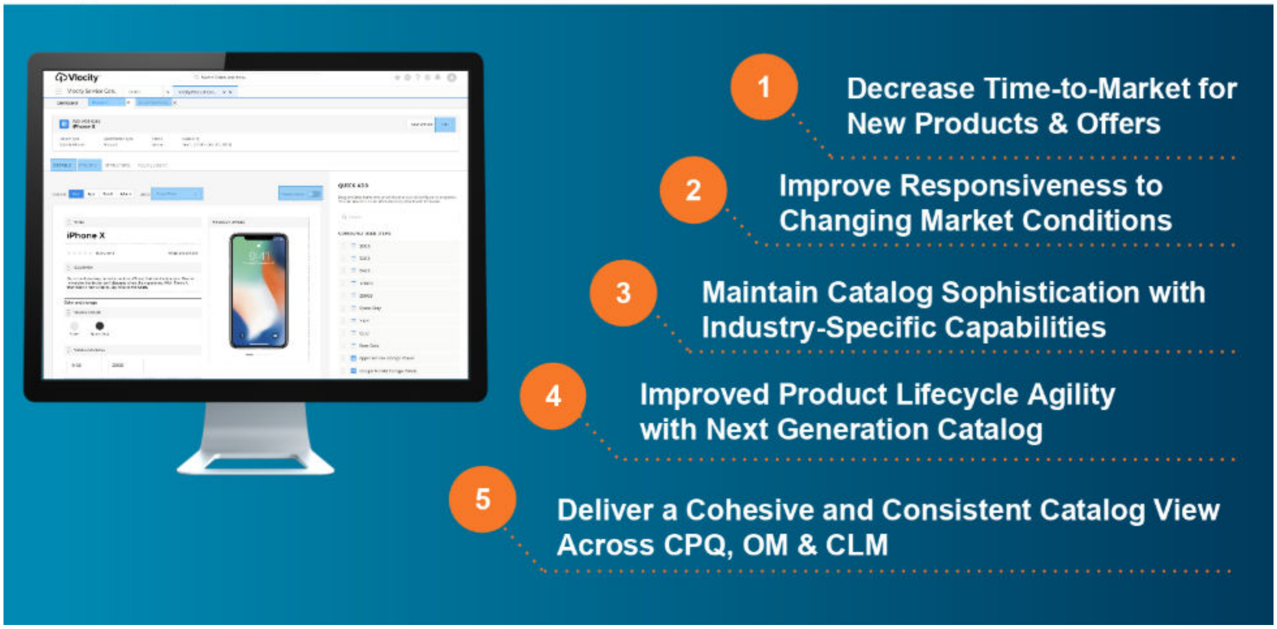
Vlocity EPC gives your business an industry-leading, catalog-driven platform that revolutionizes the CPQ and order management space by:

- Using a metadata-driven approach to accommodate the incarnations of a product as it travels from twinkle in the eyes of customer to physical implementation customer’s hands.
- Integrating with your end-to-end ecosystem using a product-service-resource (PSR) model to drive sales, configure-price-quote (Vlocity CPQ) and Vlocity Order-Management capabilities.
- Using an evolutionary approach to transform both front-office and back-office systems into a shared-catalog-driven architecture.

Business Benefits

Communications, media, and energy companies gain significant benefits through the adoption of an enterprise-wide, centralized, product-management capability. An enterprise product-management solution yields more highly competitive products with stronger profit margins and greater customer satisfaction. Vlocity Enterprise Product Catalog delivers these benefits, improving the bottom line through faster time-to-market and lower cost-to-market, enabling innovation and improving quality across product lifecycle management.

Vlocity Enterprise Catalog Benefits



1. Decrease Time-to-Market for New Products & Offers
2. Improve Responsiveness to Changing Market Conditions
3. Maintain Catalog Sophistication with Industry-Specific Capabilities
4. Improved Product Lifecycle Agility with Next Generation Catalog
5. Deliver a Cohesive and Consistent Catalog View Across CPQ, OM & CLM

Decrease Time-to-Order for New Products and Offers: Deliver a unified catalog across CPQ, CLM and OM for all products & services to reduce sales cycle and order orchestration times.

Accelerate Responsiveness to Changing Market Conditions: Centralize product catalog management within an agile, cloud platform to improve responsiveness and flexibility for changing business needs.

Improve Product Lifecycle Agility and Efficiency: Leverage next-generation catalog capabilities. View customer assets across all product or service stacks. Automate notifications to appropriate teams for order fallouts or manual intervention requirements.

Deliver a Cohesive and Consistent Catalog: Eliminate reliance on legacy systems and deliver a unified catalog for centralized lifecycle product management.

Time-to-Market

Faster time-to-market means:

- Reduced design, build, test and launch time for products through component reuse.
- A reduced volume of data required to assemble and establish new offerings in the architecture.
- A first-mover advantage.
- Simplified and accelerated new product and service introduction through improved clarity around what to offer, which is critical in responding to competitors.

Cost-to-Market

Lower cost-to-market means:

- Reduced operational and delivery costs through fewer disparate data points to manage.
- Reduced complexity of business processes.
- Structured components that maximize compatibility with long-term reference architecture.

Innovation

Enabling innovation provides:

- Increased ability to support complex propositions due to simpler set-up from any channel.
- Rapid monetization of data moving among people and systems through simplified definition of a unified product.
- Improved customer satisfaction through better understanding of the product purchased.
- Ability to compete with and support web-application and cloud-based solutions.

Solution Features

Vlocity Enterprise Product Catalog product provides the core components necessary to centralize, configure, integrate, and maintain the product and service portfolio across the enterprise. Vlocity Enterprise Product Catalog brings together, into a single catalog, all of the commercial and technical elements that define the reusable product and service building blocks and resulting offers.

The Product Console user interface provides a single point-of-entry and single point-of-access to the product and service data, through which users can perform all the activities necessary to manage the full catalog of product offerings and underlying services, including the creation, modification, reuse, launch, maintenance, and retirement of products and services.

Key features of Vlocity Product Console include:

- Collaboration environment for both business and IT users.
- Cross-function, role-based user definitions and security.
- Centralized Data Management for the product and service catalog:
  - Componentized data model for offers, products, services, devices, pricing, target system rules, and processes.
  - Simple and complex product and service structures.
  - Definition of commercial and technical aspects of product and service components.
  - Sharing of product components across different product and market categories.
  - Cloning and modification of product structures or individual entities to enable rapid product modeling.
  - Business and technical modeling of product offers and underlying component specifications.
  - Highly configurable capability for defining rating attributes for charges and discounts, to support attribute-based, tiered, threshold, and flat rating and discounting, variable across product lines.
  - Channel-, market-, and customer-specific price lists to manage the reuse of products across target markets through different offers.
  - SID-compliant and SID-based XML available.

What's New in EPC

This section provides a summary of the new features and enhancements for Vlocity's Enterprise Product Catalog. Check back frequently as this page is updated on a regular basis, in between releases.

Fall '20

Overview of New and Enhanced Features

The Vlocity Enterprise Product Catalog (EPC) includes the following feature enhancements for Fall '20.

Vlocity Product Designer

The next generation Vlocity Product Designer provides a unified user experience and the agility to quickly launch products and services. In this release, Vlocity introduces several enhancements to the Vlocity Product Designer.

- **Bundling Products** — For improved usability, the Structure view of a product uses icons so that you can more easily differentiate products and their relationships without having to open each to view the details. The Structure view also includes a legend of the icons and their meanings.
  - Specification types (product, offers, service, and resource)
  - Simple and complex product bundles
  - Relationship types between products:
    - Parent-child relationship
    - Associated product specification
- **Attribute Metadata** — You can now set select attributes to be excluded from searches of the basket cache for the same bundled product to maximize the basket cache hits in Vlocity Digital Commerce solution. Use this for attributes that are specific to a customer. If enabled, the attribute is not part of the response cache when the Digital Commerce ConfigureOffer API is called. For example, you can set the IMEI or serial number attribute to be excluded from the cache. By default, this property is set to False. Use this property for attributes whose values are unique for each account. See Define Attributes in the Product Designer.
- **Reports (Beta)** — To help with audits and product reviews, you can export a report of products with their properties and attributes from the product search page in Vlocity Product Designer. You can also configure the columns, or fields, used for the products and attributes.

EPC Projects

In Vlocity Product Designer, you can use Projects to track all changes made to any object in your product catalog. Changes tracked when using Vlocity Product Designer but not with the Vlocity Product Console. In this release, the Projects feature has these enhancements:

- You can select multiple changed items in a project and move them to another project or remove them from the current project.
- You can also track the action (such as Add, Change, and Delete) taken on the referenced project item.

- The options for the **Lifecycle Status** field has changed.
  - The **Deployed** status is replaced with **Released**. You can use IDX Workbench to export Projects in **Released** status in Fall '20 and **Deployed** status in Spring '20.
  - The **Testing** status is replaced with **In-Test**.
  - The **Rejected** and **Approved** states are deprecated.
  - The **Canceled** status is new.

See Track Product Catalog Changes with Projects.

Versioning (Beta)

Products define a company. The products evolve over time and require changes in their configuration. Versioning enables a company to control its products across the different stages of their maturity and marketability.

With Vlocity Product Designer, you can create different versions of the same offer and specifications and manage their lifecycles from inception to retirement. This versioning provides increased agility to introduce changes without disrupting the ongoing transactions.

IMPORTANT

As a beta feature, the Version Management feature of the Vlocity Product Designeris a preview and isn’t part of the “Services” under agreement with Vlocity. This feature is for evaluation purposes only in a sandbox environment, not for production use. This feature is offered as is and isn’t supported, and Vlocity has no liability for any harm or damage arising out of or in connection with it. Turning on Versioning causes irreversible changes that render an org inoperable if Versioning is thereafter turned off. Use this feature at your sole discretion, and make your purchase decisions only on the basis of generally available products and features. Vlocity doesn’t guarantee general availability of this feature within any particular time frame or at all, and we can discontinue it at any time. For access to beta documentation, contact the Vlocity Customer Success team. For information on enabling this feature, contact Vlocity.

Any use of this beta feature is subject to the Beta Services terms.

NOTE

Versioning is supported for commercial offers and product specifications only. It is not supported for technical products.

For this release, Versioning is enhanced with the following features.

- Version history and comparison — To see changes with products or offerings, you can compare versions for additions, modifications, and deletions. For products and offers, you can see version changes for the details (attributes and fields), pricing, and structure. Pricing includes changes to overrides and adjustments.
- Versioning and EPC entities — When you create a new version of an object, commercial product, or picklist, the related entities for each are either cloned or reused across the versions.
- CPQ Support — With the Versioning feature enabled, CPQ supports browsing and searching products as well as order capture and MACD operations on assetized versions.
- New CMT jobs — New Upgrade EPC Schema jobs are provided to update the existing product catalog data to align with the schema changes. This process should be run regardless of the use of the Versioning feature.

  You must run both jobs in the order displayed. These jobs are safe to run more than once. If you create entries in Vlocity Product Console, you must run these jobs again before updating to Vlocity Product Designer.

  - Populate New Fields job — Populates the **GlobalGroupKey** and **VersionLabel** fields in Product2, Object Type, and Picklist objects. It does not modify the existing field values if they're already populated.
  - Populate Product Hierarchy Group Key Path job — the ProductHierarchyGroupKeyPath field in Promotion Items, Override Definition, Vlocity Attachment, Opportunity Items, Quote Items, Order Items, and Assets. Run this job after the Populate New Fields job.

  See Upgrade EPC Schema.

Realization of Product Offering based on Specification (Beta)

Communications Service Providers define products decoupled from commercial view (Product Offering and Product Specification) and technical view (Customer Facing Service and Resource Facing Service). TM Forum suggests this decoupling to efficiently support the needs of product marketing and delivery groups.

A product specification represents a tangible or intangible product as perceived by the business use. It also specifies what product marketing can sell at a functional level. A product offering describes tangible and intangible goods (such as a product specification) made available for a certain price to the market in the form of product catalogs.

For this release, the Offer Specification Realization feature has the following enhancements.

- An offering (even a simple, non-bundled one) can realize a product specification that is a bundle. From the Structure tab of a simple or bundled offering, you can see the parent product specification and its child specifications. You can change the group and local cardinality of the parent or child specifications. You can also override the attributes of the parent and immediate child product specifications from an offering.

IMPORTANT

As a beta feature, the Realization of Product Offering based on Specification feature of the Vlocity Product Designer is a preview and isn't part of the "Services" under agreement with Vlocity. This feature is for evaluation purposes only in a sandbox environment, not for production use. This feature is offered as is and isn't supported, and Vlocity has no liability for any harm or damage arising out of or in connection with it. Turning on Versioning causes irreversible changes that render an org inoperable if Versioning is thereafter turned off. Use this feature at your sole discretion, and make your purchase decisions only on the basis of generally available products and features. Vlocity doesn't guarantee general availability of this feature within any particular time frame or at all, and we can discontinue it at any time. For access to beta documentation, contact the Vlocity Customer Success team. For information on enabling this feature, contact Vlocity.

Any use of this beta feature is subject to the Beta Services terms.

Spring '20

Overview of New and Enhanced Features

Vlocity Product Designer

Vlocity Enterprise Product Catalog delivers a unified, cloud-based solution to manage product, service, and resource catalogs and life cycles in a central repository. It provides the core components necessary to centralize, configure, integrate, and maintain the product and service portfolio across the enterprise. You can use Vlocity Enterprise Product Catalog to manage many complex configurations of commercial offers, specifications, pricing, and rules.

In this release, Vlocity introduces the next generation Product Catalog Vlocity Product Designer that provides a unified user experience and the agility to quickly launch products and services. The Vlocity Product Designer provides intuitive user interface, quick actions, drag-n-drop capabilities, enhanced searching, and improved performance to increase the productivity of the users. This allows the business to react more effectively to changing market dynamics.

You can use Vlocity Product Designer to manage the configuration of products, product characteristics, bundling, and pricing. See Overview of Product Creation in Product Designer.

The Vlocity Installation Assistant contains instructions for installing the Vlocity Product Designer. See Vlocity Installation Assistant.

The Vlocity Product Designer can co-exist with the existing Vlocity Product Console. You can view and manage products created using the Vlocity Product Console in the Vlocity Product Designer with few exceptions, as long as the product attributes are assigned to the General Properties layout. See Manage Existing Object Types in Product Designer.

Winter '20

Overview of New and Enhanced Features

Versioning - Beta

With Vlocity Unified Product Catalog, you can create and configure a Product, Vlocity Picklist, or Object Type and log and track any changes so you can introduce those changes without making them visible to sales channels or other catalog functions. You can then release a specific draft version of the Product, Picklist, or Object Type with effective dates and a custom auto-generated version number to sales channels or other catalog functions.

Once the Product, Picklist, or Object Type is deployed, any further changes are not allowed. Any changes require you to create a new version of the product, object type, or picklist.

IMPORTANT

The Version Management feature of the Vlocity Unified Product Catalog is a Beta version. Therefore, you cannot use the Beta version for development or production because there is no committed support. You can deploy the Beta version in a sandbox environment to try it and provide feedback to Vlocity to make further improvements. Turning on Versioning causes irreversible changes that render an org inoperable if Versioning is thereafter turned off.

Projects

In Vlocity Product Catalog, you can use Projects to track all changes made to any object in your product catalog. When you set a project as the default, all changes are saved to that project automatically. You can review the changed object from the change history. You can also manage changes manually, such as moving a change from one project to another.

Each project has an owner, and only that owner can change the project—its default status, lifecycle status, history, and any other properties. A user can own multiple projects.

See Track Product Catalog Changes with Projects. To configure, see Enable and Set Up Projects for Vlocity Product Designer.

Realization of Product Offering based on Specification - Beta

Communications Service Providers define products decoupled from commercial view (Product Offering and Product Specification) and technical view (Customer Facing Service and Resource Facing Service). TM Forum suggests this decoupling to efficiently support the needs of product marketing and delivery groups.

A product specification represents a tangible or intangible product as perceived by the business use. It also specifies what product marketing can sell at a functional level. A product offering describes tangible and intangible goods (such as a product specification) made available for a certain price to the market in the form of product catalogs.

The Vlocity Product Console provides the following capabilities:

- Configure logical entities of a product, such as product offerings, product specifications, service specification, and resource specification.
- Configure simple or bundled product offerings and atomic or composite specifications.
- Realize product offerings based on product specifications with characteristics, inheritance, and overrides.
- Ensure backward compatibility with existing implementations.

IMPORTANT
The Offer Specification Realization feature of the Vlocity Unified Product Catalog is a Beta version. Therefore, you cannot use the Beta version for development or production because there is no committed support. You can deploy the Beta version in a sandbox environment to try it and provide feedback to Vlocity to make further improvements.

Unified Product Catalog - Beta

The Unified Product Catalog has the following features:

- Usability Pricing -- You can define price and cost, create promotions and offers, and apply adjustments, and overrides.
- Price Adjustments and Overrides -- You can adjust or override a product’s price without changing the price itself. You can manually adjust or override a price in the cart. You can add products to the bundle and then change their prices when they appear in that context. You can also apply a promotion to a product in the cart.
- String Translations -- You can configure the Unified Product Catalog to translate the names and descriptions of catalog data to multiple languages for rendering in CPQ operations or eCommerce applications.

IMPORTANT
The Vlocity Unified Product Catalog is a Beta version. Therefore, you cannot use the Beta version for development or production because there is no committed support. You can deploy the Beta version in a sandbox environment to try it and provide feedback to Vlocity to make further improvements. For access to beta documentation, contact the Vlocity Customer Success team.

Fall '19

Overview of New and Enhanced Features

This section provides a summary of new EPC features and enhancements in the CME Fall '19 release.

Product Console (Beta Version)

Configuring Translation

You can configure the Unified Product Console to display object and field labels in languages other than English. You can translate the names and descriptions of catalog data to multiple languages for rendering in CPQ operations or eCommerce applications. This feature is supported by the Multilingual Product Catalog Feature. You manage the Commercial and Technical catalog in a single base language, such as English, Spanish, French. For every deployment or upgrade, follow the post-install steps for translation and upload the translated JSON file.

Creating an Attribute Category

You can define attribute categories to group product attributes and influence a consumer's purchase decision. Attribute categories appear as headings at runtime in Vlocity Cart’s Filter and Configuration windows. Attribute categories are required to create product attributes. Once you create attributes categories, you can add new attributes.

Applying Attributes to Objects

You can associate an attribute with an object. When you create an attribute, you can specify the product objects to which the attribute applies. You create the attribute first. The attribute then appears in the Attributes list, from where you can assign it to an object.

Summer'19

Overview of New and Enhanced Features

Product Console (Beta Version)

- The Load More button no longer appears on the Product Hierarchy tab when displaying direct child items. All the direct child items of a product are shown at once. This fix has been chosen because there are no customer use cases where a node in the tree has more than 100 direct children.
- The capability to add promotion or offer based adjustments or overrides is now located under the Structure tab. This allows customers to define contextual adjustments or overrides for bundled products as well.

  You can select any node in the structure and modify an existing promotion, add a new promotion, or create an offer-level adjustment or override.
- When deleting price list entries, there is now a prompt where you need to provide confirmation.
- When the price list is changed, the currency updates with new currency values. If no currency values are present in the price list, the currency field is blank and a price list entry is not created. The customer must input the currencies in the price list to allow the price list entry creation. No error message appears.

- In the Pricing tab, all tables only show active records by default.
- New Product Console (Beta): Each table in the Pricing tab has an arrow button which, when clicked, either expands or collapses the corresponding table. On load, only tables with records appear expanded. All other tables are collapsed.
- The new order for the tables in the Pricing tab is:

  1. Price Table
  2. Promotion and Offers Table
  3. Cost Table
- For the root product, the following cardinalities are displayed:
  - Minimum, maximum, and default.
  - Group Cardinalities (min, max) if the root product has children

  You can override local and group cardinalities.
- You can now set the column layout and tab order in a section and indicate the column in a section element.
- In the Structure tab, you can configure the relationship when adding a product to a bundle. The relationship consists of Relationship Type, such as Decomposition, and setting the quantity.
- New Product Console (Beta): In the Structure tab, you can now re-order nodes in the structure tree using drag and drop, and the sequence updates accordingly.

Change of Plans

- As part of Change of Plans, transforming multiplay offers, you can:
  - Merge multiple existing individual offers that belong to different product families into a new single multi-play bundle.
  - Split a single existing multi-play bundle into individual offers that belong to different product families.
  - Perform a combination of Merge and Split.
- As part of Change of Plans, transforming multiplay offers, you can use a comparison feature that recursively finds all the related items and dependencies so you are not missing anything in your comparison and so you get an accurate price.
- A one-time charge of the intermediate matching parent products with action=Add and subAction=Reassign is set to 0 after a Replace/Transform. For each matched leaf, the parent product is also matched. Only exact parent products are matched. A new item relationship record is created with a reassignment type and the sub action on the source and target line item is set to Reassign.
- With the introduction of the xli relationship object for the change of plan functionality, the existing move implementation has been updated to create a consistent set of records with the new Replace APIs for use with Vlocity Order Management. You now run a Create Relationship Records Upgrade job from the CMT Admin console to allow customers to update their existing order which creates the required relationship records for Movement.
- As part of Change of Plans, an API returns line item attributes and values to display on the Comparison UI. The compare API will consider the includeAttributes parameter with possible values of true or false. The default for this parameter is false if not passed. The parameter can be defined as part of Vlocity Cards.

- As part of Change of Plans, transforming multiplay offers, the ReplaceOffer API and the Submit Order API support additional custom fields to copy during replacements of offers. You must define a Field Settings record for each field to copy during the Replace process.

For more information on Change of Plans, Transform Multiplay Offers, see Transforming Multiplay Offers.

Vlocity Support for TM Forum Frameworx

Telecommunication service providers/operators and service providers in other verticals are facing challenges when producing and deploying new products or upgrading existing products. The time-to-market is very slow because of the complexity and volume involved in the enterprise catalogue.

The Vlocity Enterprise Product Catalog (EPC) solution is designed to help Communications Service Providers (CSPs) to deliver innovative, profitable products at a lower cost and in less time. Other industries like utilities, water, retail etc can implement the same solution as their key enterprise catalogue management application integrated with their core billing application.

The Enterprise Product Catalog (EPC) solution helps CSPs create and maintain their product catalogues in a user-friendly and cost effective way. The solution ensures the accuracy and quality of the product catalogues and significantly reduces time to market. Vlocity Enterprise Product Catalog empowers communication service providers (CSPs) to place products at the heart of their business. Featuring an easy-to-use collaboration environment for business and IT users alike, Vlocity Product Enterprise Catalog provides CSPs total control over their product data and processes.

Vlocity Enterprise Product Catalog takes previously inconsistent practices and products and consolidates them into a single, unified catalog with up-front, role-based access and back-end system interoperability. The result is a multidimensional, cross-product and channel catalog that sits at the heart of the CSP’s order-to-install process, ultimately improving time to market, reducing the cost to market and improving innovation and quality.

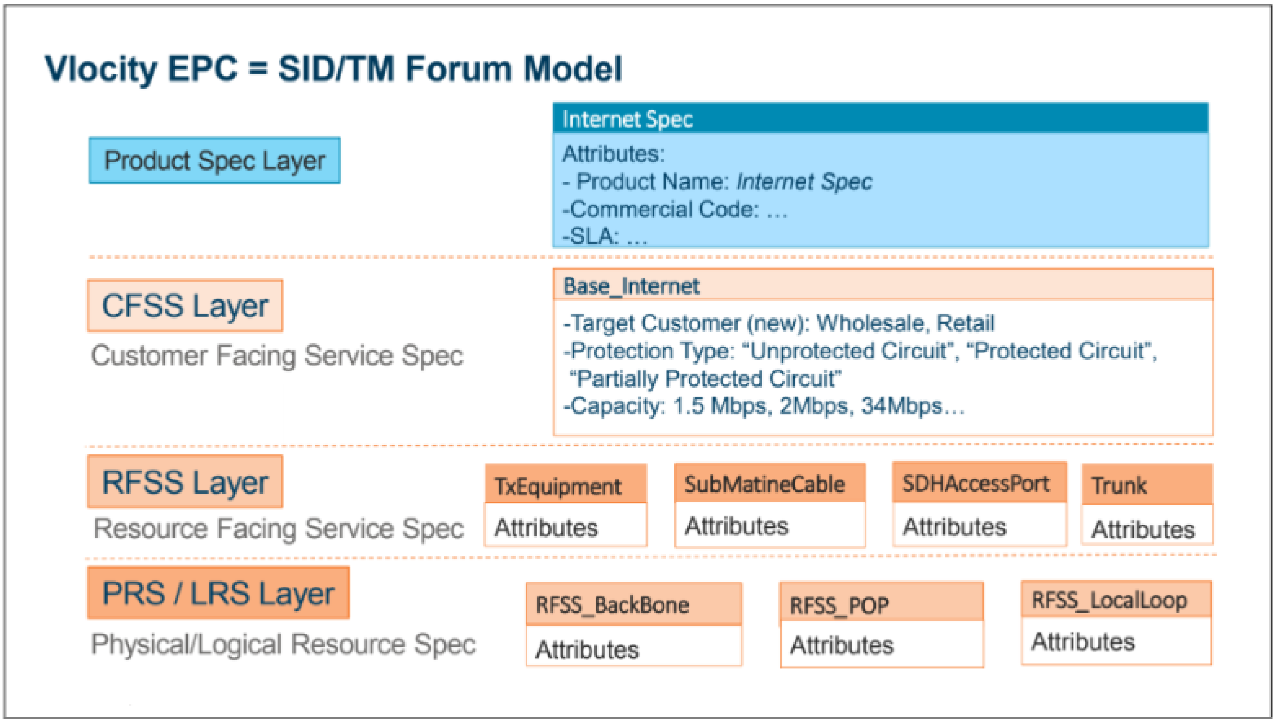
Vlocity Enterprise Product Catalog allows flexible hierarchical structure to suite any vertical, and has a powerful rules engine to dynamically differentiate product properties such as price, presentation etc., and to enforce overall product selection rules. Sophisticated cross product or cross vertical bundling can be achieved using the hierarchical structure and the rules engine.

Vlocity EPC is powered by a shared product catalog. Vlocity’s shared product catalog uses a metadata-driven approach to accommodate all of the incarnations of a product, as it travels from a twinkle in the customer’s eyes to the physical implementation in the customer’s hands. Vlocity EPC is TMForum SID compliant. It supports all levels of product definitions, including offers, product specifications, services, and resource specifications.

- Offers are designed by Marketing, sold to particular channels, customer segments with specific pricing.
- The Product Spec is created by Product Management and describes all of the product components.
- Using Vlocity Order Management, the Product Spec is translated into the various technical and manufacturing spec products that are tuned for each of the downstream systems that use them.

Vlocity EPC and the SID/TM Forum Model

Vlocity EPC is compliant with the SID/TM Forum model (frameworx), and supports various levels of product definitions, including offers, product specifications, services, and resources. Marketing teams can design product offers for sale to particular channels and customer segments with specific pricing.



Vlocity EPC = SID/TM Forum Model

Product Spec Layer

Internet Spec

Attributes:
- Product Name: *Internet Spec*
-Commercial Code: …
-SLA: …

CFSS Layer

Customer Facing Service Spec

Base_Internet

-Target Customer (new): Wholesale, Retail
-Protection Type: "Unprotected Circuit", "Protected Circuit", "Partially Protected Circuit"
-Capacity: 1.5 Mbps, 2Mbps, 34Mbps…

RFSS Layer

Resource Facing Service Spec

TxEquipment
Attributes

SubMatineCable
Attributes

SDHAccessPort
Attributes

Trunk
Attributes

PRS / LRS Layer

Physical/Logical Resource Spec

RFSS_BackBone
Attributes

RFSS_POP
Attributes

RFSS_LocalLoop
Attributes

Product Spec Layer

The blue, Product Spec Layer is the commercial view of what the business sells. Product management teams define the product specification, which describes all product components: Product name, code, attributes, other commercial characteristics, service-level agreement (SLA), download speed and physical aspects, such as size and color. The product specification can have a business hierarchy with child products.

Vlocity Order Management translates the product specification into technical and manufacturing spec products tuned for the downstream systems that use them. The commercial view can be translated to the service and the underlying technical resources.

Customer Facing Service Spec Layer

The next level is the Customer Facing Service Spec (CFSS) Layer, which includes technical products and technical services. Multiple products can use the CFSS Layer, for example, a broadband internet service requires a port and an authentication service to deliver the product.

Resource Facing Service Spec Layer

The third layer is the Resource Facing Service Spec Layer, which supports the CFSS Layer. For example, this layer may include equipment, cable attributes, access port, trunk and other attributes, typically to support the network.

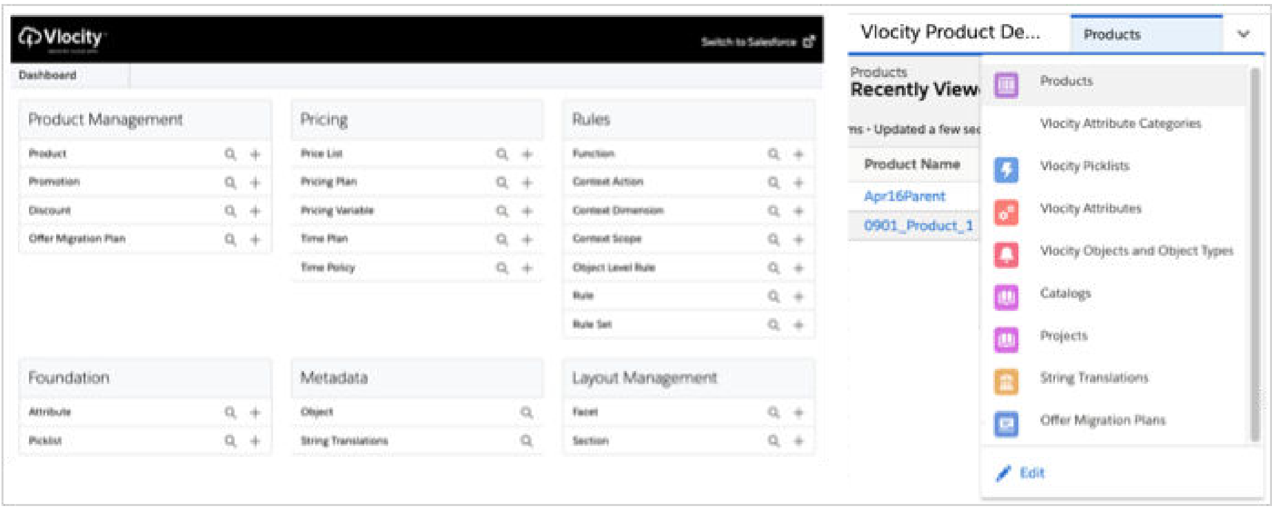
The last layer is the Physical/Logical Resource Spec Layer, for physical resources such as phone number, IP address, POP, local loop connection ID, physical cable – aspects that need to be allocated to the customer. These concepts are important, especially from a data-modelling perspective. For more information, check the TMForum website.

Comparison of the Product Catalog Applications

In Spring '20, Vlocity introduced the Vlocity Product Designer for Lightning. Vlocity also supports its legacy versions of the Product Catalog administration applications.

- Vlocity Product Designer uses Lightning Web Components (LWC) and is available in Spring '20 and later. You have access to the product catalog entities from a menu list.
  Vlocity Product Designer has enhanced searching and improved performance. It also aligns with the move to LWC design in Vlocity and Salesforce for a common user experience.Vlocity Product Designer improves the usability for managing attributes, offers, product specifications, object types, and bundles. For example, you can drag and drop products to create product structures or files to add attachments. The Vlocity Product Designer
- Vlocity Product Console uses Angular, and you have access to the product catalog entities from a dashboard.
- Product Console for Salesforce Classic uses the Salesforce Classic interface. This is the first generation of Vlocity's Product Console.

The following image shows Vlocity Product Console with the dashboard on the left and the Vlocity Product Designer on the right.



The Vlocity Product Designer supports creating and managing products:

- Object types
- Attribute assignment overrides
- Price list entries
- Cost list entries
- Product attachments

- Attribute configuration rules
- Fulfillment rule

The Vlocity Product Designer supports creating and managing product structures:

- Bundle composition
- Attribute overrides
- Cardinality overrides
- Pricing overrides and adjustments

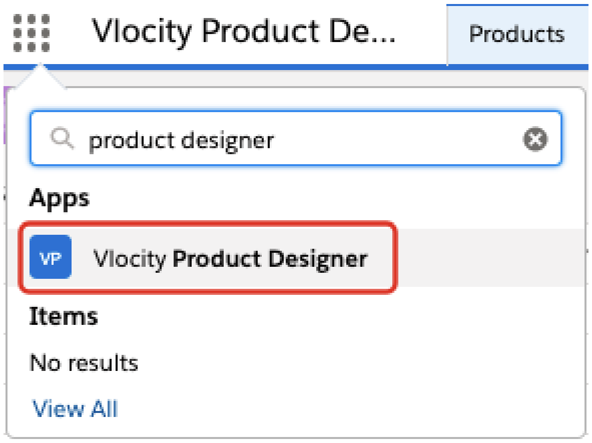
The Vlocity Product Designer supports managing these product catalog entities:

- Attributes
- Picklists
- Catalog
- Migration plans
- String translations
- Object types

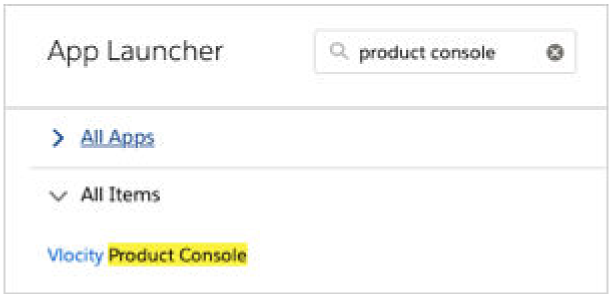
For the administration of other product catalog entities, use the Vlocity Product Console.

Access to the Applications

To use Vlocity Product Designer, search for **`product designer`**, and select it in the list of apps.



To use Vlocity Product Console , search for **`product console`**, and select it in the All Items list.



App Launcher

product console

See Also

- The Vlocity Product Designer
- The Vlocity Product Console Interface
- Manage Existing Object Types in Product Designer

Overview of Product Creation in Product Designer

With Vlocity EPC, you can create an offering to sell to customers, such as an iPhone 11. Before creating an offering, you must create other entities, such as object types, attributes, and product specifications. You can then create bundled offerings and set pricing.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Create Products in Vlocity Product Console.

> **NOTE**
>
> You must use Vlocity Product Console to create pricing elements. See Pricing Definition.
>
> You must also use Vlocity Product Console to update an older product that has layout facets other than General Properties. See Manage Existing Object Types in Product Designer.

You must create *attributes* to capture specifics about the product that needs to be tracked. Some attributes are options that might affect the product's price, such as storage capacity, while others might customize an offering for a customer, such as the iPhone color. For attributes with specific options, like storage and color, you need to create *picklists* so that a customer can select the option from a list.

Example 1. iPhone Offering Example

Before creating an iPhone 11 offering, you need to create more generic objects that have common options and traits. For example, you can create a smartphone object with common attributes, such as model number, bandwidth, operating system, storage, and dimensions. When you create an iPhone or a Samsung phone based on the smartphone object, they inherit those attributes. You don't have to create them for each phone maker and model. You can set the attributes for each phone offering. For example, for the iPhone 11, you might have a picklist for the colors specific to that phone and another picklist of colors specific to the Samsung S10.

In Vlocity Product Console, create the pricing elements — price lists, pricing plans, time plans and policies, promotions, and discounts.

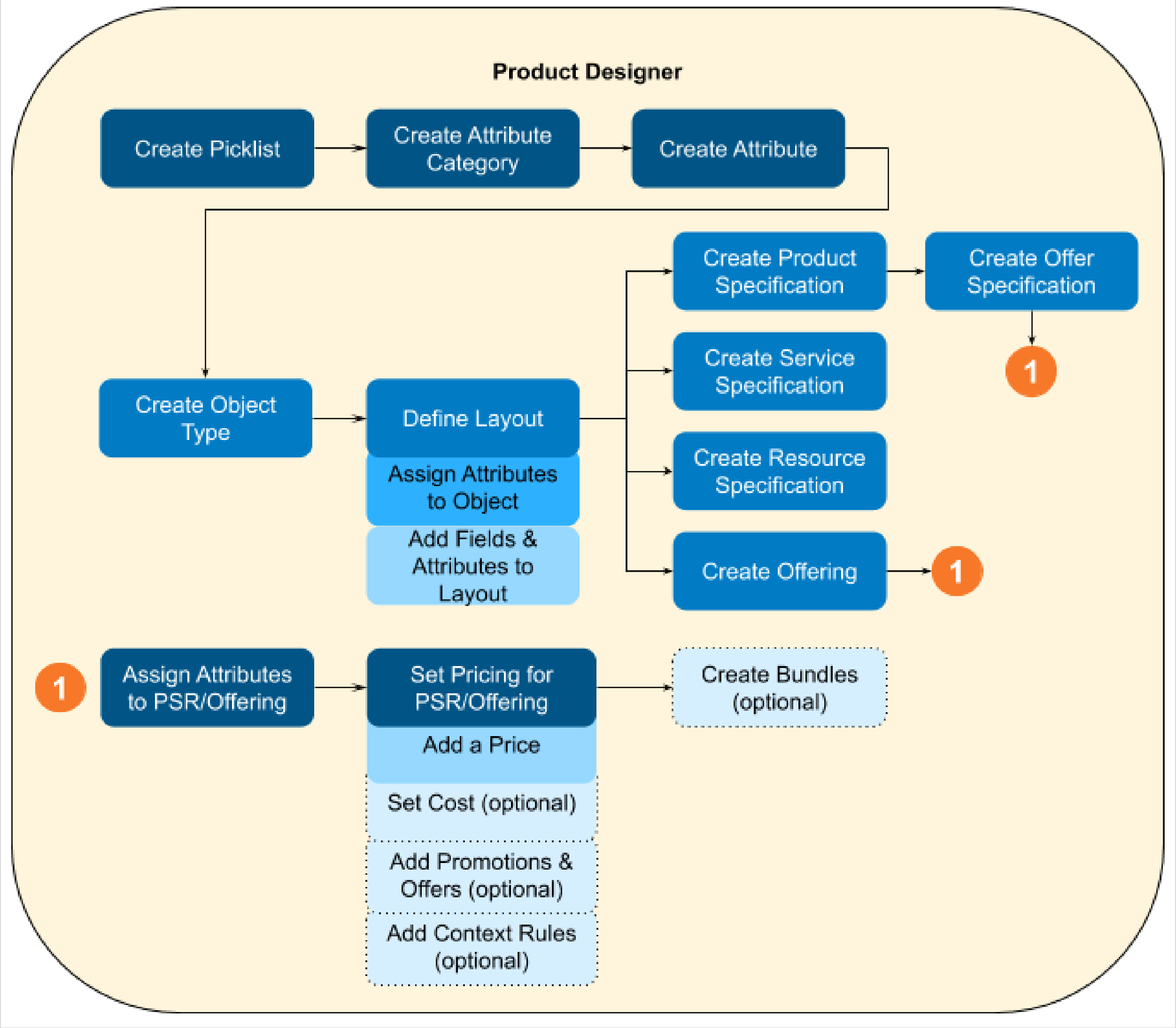
In Vlocity Product Designer, create product bundles, or a set of products that are sold together.

For example, you could create a new account bundle of the iPhone 11, Apple Watch Series 5, and a mobile data plan.

Then, in Vlocity Product Designer, you can set the pricing, promotions, and discounts for standalone and bundled offerings.

For instance, by itself, an iPhone 11 is priced at $699. In a new account bundle, the iPhone 11 has a $50 discount.

Figure 1. Product Creation Workflow


Product Designer
Create Picklist
Create Attribute Category
Create Attribute
Create Product Specification
Create Offer Specification
1
Create Object Type
Define Layout
Assign Attributes to Object
Add Fields & Attributes to Layout
Create Service Specification
Create Resource Specification
Create Offering
1
1
Assign Attributes to PSR/Offering
Set Pricing for PSR/Offering
Add a Price
Set Cost (optional)
Add Promotions & Offers (optional)
Add Context Rules (optional)
Create Bundles (optional)

Before You Begin
• You must have at least one price book. You can create one generic price book and have multiple price lists that point to the same price book. See Create a Price Book.

NOTE
Complete the following tasks in the specified order.

1. In the Vlocity Product Designer, create the attributes, object types, and specifications needed to define an offering.
   a. Create picklists.
   b. Create attribute categories.
   c. Create attributes.
   d. Create object types and subtypes as needed.
   e. Define the layout for the object type.
   f. Create product specifications.
   g. Create offer specifications.
2. Vlocity Product Designer, pricing is auto-created or re-used while defining Pricelist entries. But you can define pricing elements in the Vlocity Product Console, define the pricing elements.
   a. Create price lists, pricing plans, pricing variables, time plans, and time policies.
   b. Define promotions and discounts.
   c. Associate products as promotion or discount products.
3. In Vlocity Product Designer, create product bundles.
4. Set pricing for offers.
   a. Add a price.
   b. Add promotions and offers.
   c. Add a cost.

What's Next

Create Picklists and Picklist Items in the Product Designer

The Vlocity Product Designer

Vlocity Product Designer is the next generation Product Catalog administration application. It provides a unified user experience with improved efficiency and agility to launch products quickly to market. It gives both business and IT users access to all the product management capabilities they need.

For a comparison of the Vlocity Product Designer and Vlocity Product Console, see Comparison of the Product Catalog Applications.

The Vlocity Product Designer has several benefits:

- Defines pricing.
- Improves usability for product definition, pricing, and structure.
- Increases productivity for product marketing and admin users.
- Provides a cohesive view of shared catalog elements across Configure Price Quote (CPQ), Contract Lifecycle Management, and Order Management.

- Provides omnichannel capabilities directly within the catalog.
- Decreases time to market for new products.

Vlocity Product Designer provides seamless integration between the shared product catalog and the ability to configure, price, and quote the products in the catalog.

Product managers use Vlocity Product Designer to:

- Define and maintain product entities.
- Manage product structure, rules, and offers.
- Build foundational constructs, including attributes and picklists.
- Build metadata constructs, such as product object types.

Catalog administrators use Vlocity Product Designer for the following tasks:

- Work with predefined objects, for example, to create reusable object types that define properties, such as attributes and fields.
- Define common layouts for product entities, such as offers or product specifications, that are instantiated from the object types.
- Perform initial setup tasks for Vlocity EPC implementation.

The Vlocity Product Designer can co-exist with the existing Vlocity Product Console. You can view and manage products created using the Vlocity Product Console in the Vlocity Product Designer with few exceptions, as long as the product attributes are assigned to the General Properties layout. For more information, see Manage Existing Object Types in Product Designer.

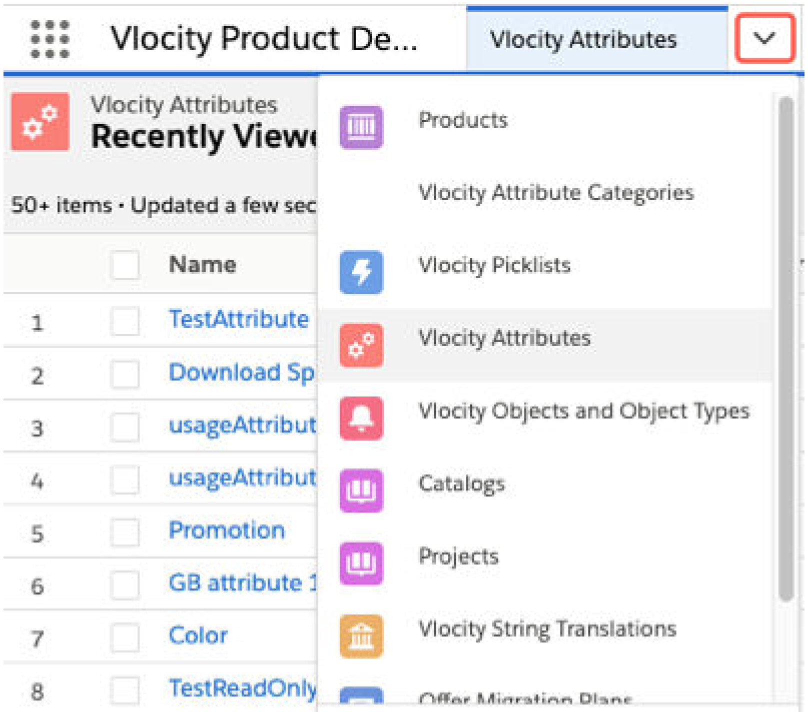
Open the Vlocity Product Designer

1. Make sure you have completed the Configure Vlocity Product Designer.
2. Click the App Launcher and enter **`Vlocity Product Designer`** in the Quick Find box.
3. Click the Vlocity Product Designer app.

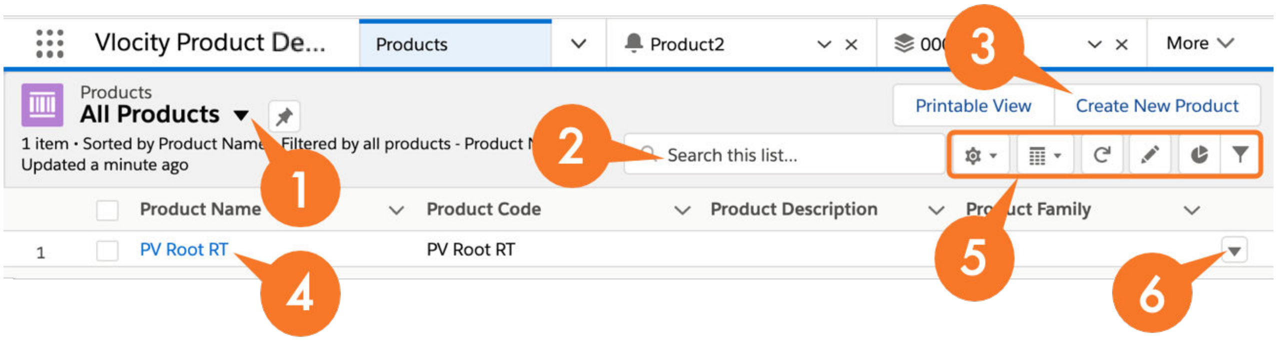
Navigate the Vlocity Product Designer

When you open Vlocity Product Designer, the first tab is the title, and you can't interact with it.

Use the tab after the title to manage the different product catalog entities. The ⌄ navigation menu lists the product entities that you can manage.



When you select an entity, an entity manager opens in that tab, such as **Vlocity Objects and Object Types**. The following image and list describe the common features of the new interface of the Vlocity Product Designer.

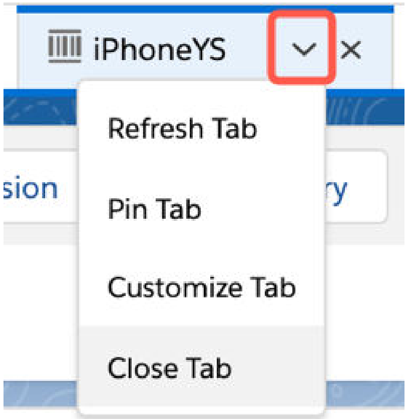


1. Change views, such **Recently Viewed** or **All**. You can pin a view as the default.
   - For the **Products** manager, Vlocity recommends that you make **All Products** the default view. After you select **All Products**, click the pin icon to make it the default view.
   - For the **Vlocity Objects and Objects Types** manager, Vlocity recommends that you do not use the New button or click objects to edit from the list view. Instead, click Product2 Object to create, edit, and delete objects.
2. Search for a specific entity.

3. Create a new entity.
4. From the list of entities, click the name to edit the entity's properties.
5. Use the controls to update and modify the entity list.
6. Select an action for an entity, such as delete or edit.

When you create or edit an entity, it opens in a new tab.

Each tab has an action menu that you can use to refresh the tab's content, pin the tab (so that it opens when you open the Vlocity Product Console), or rename the tab.



Manage Existing Object Types in Product Designer

If an existing product created in Vlocity Product Console uses facets other than General Properties, you must manually modify the associated object type layout definition to make the product compatible with Vlocity Product Designer.

The Vlocity Product Designer uses only the General Properties facet to manage the presentation of Product2 fields and attributes. You can add multiple sections to this facet to organize the orientation of fields and attributes in a single or two-column layout.

How you make the updates depends on whether a product inherits its attribute and field layout from an associated object type:

- Products with Attribute Assignments Using Object Types
- Products with Attribute Assignments Without Using Object Types

There are no tools to convert the object type layout. In Vlocity Product Console, you must create sections in the General Properties facet and move fields and attributes from other facets into them. You cannot use Vlocity Product Designer for this task because it does not show facets other than General Properties.

Moving attributes and fields to another facet does not invalidate or change any attribute overrides. This task changes only the presentation of attributes and fields.

Products with Attribute Assignments Inherited from Object Types Only

To manage existing products in Vlocity Product Designer that were created in Vlocity Product Console, you must modify the layout for the associated object type to use one facet, General Properties.

In this use, attributes are not assigned directly to the product but are assigned to an object type that is associated with a product.

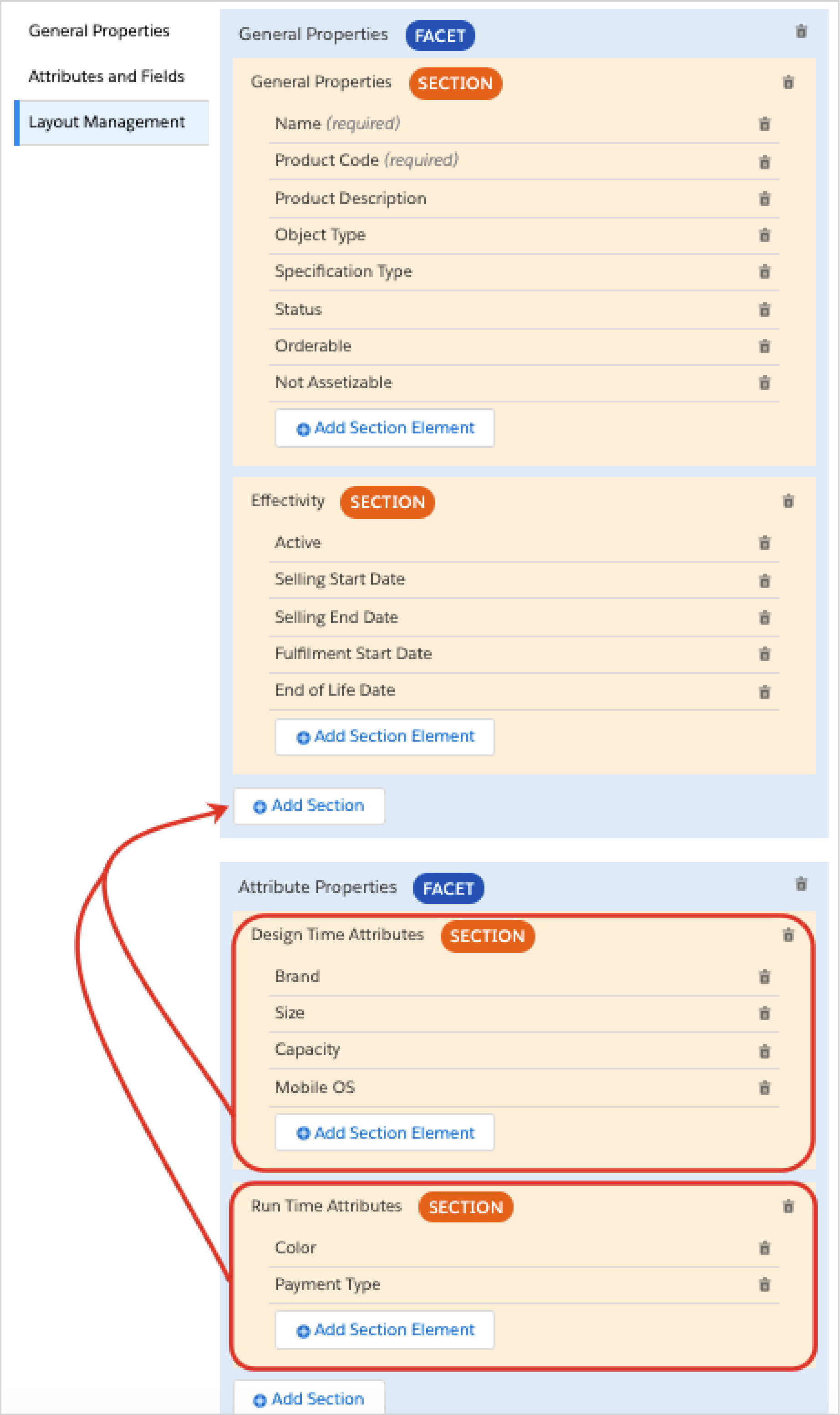
The Vlocity Product Designer shows only the fields and attributes that are configured in the General Properties facet. All other facets are redesigned using web components in the Vlocity Product Designer These other facets can be left for backward compatibility with Vlocity Product Console. To display fields and attributes in the other facets, you must create corresponding new sections in the General Properties facet.

Example 2. Smart Devices Object Type

Monique's team wants to use Vlocity Product Designer to manage their products. The Smart Devices object type has an Attribute Properties facet with the Design Time Attributes and Run Time Attributes sections. Monique must recreate these two sections in the General Properties facet.

To update layout for an object type:

1. In the App Launcher, type **`product console`** in the quick find box, and open the Vlocity Product Console.
2. Open an object type.
   1. In the Vlocity Product Console, in the Metadata section, click the search icon next to Object.
   2. In the Search Object page, search for Product2.
   3. Click the pen icon next to the Product2 Object.
   4. In the left menu, click **Object Types**.
   5. Click an object type.
      For example, Monique opens the Smart Devices object type.
3. In the object editor, click the **Layout Management** facet.
4. Review the object's facets other than the General Properties facet.
   For example, the Smart Devices object has an Attribute Properties facet with the Design Time Attributes and Run Time Attributes sections. Monique must recreate these sections in the General Properties facet.



General Properties

Attributes and Fields

Layout Management

General Properties FACET

General Properties SECTION

- Name *(required)*
- Product Code *(required)*
- Product Description
- Object Type
- Specification Type
- Status
- Orderable
- Not Assetizable

Add Section Element

Effectivity SECTION

- Active
- Selling Start Date
- Selling End Date
- Fulfilment Start Date
- End of Life Date

Add Section Element

Add Section

Attribute Properties FACET

Design Time Attributes SECTION

- Brand
- Size
- Capacity
- Mobile OS

Add Section Element

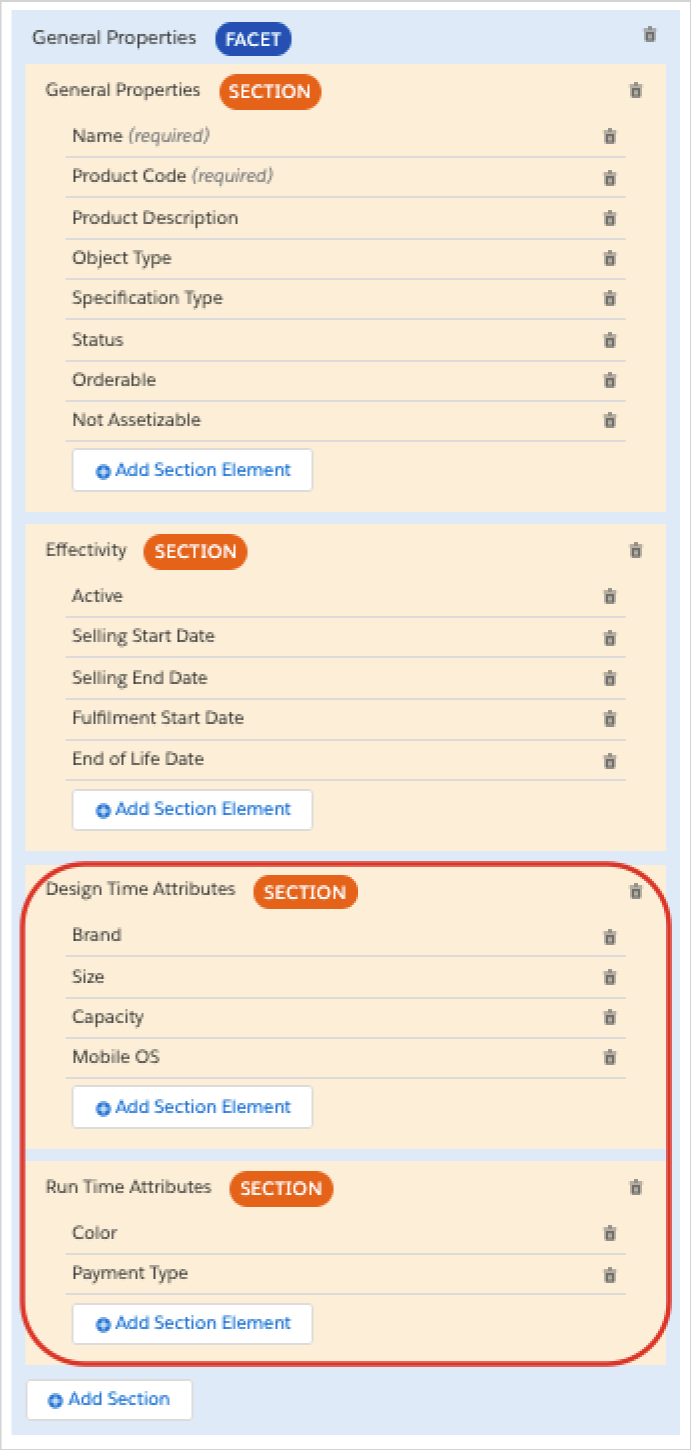
Run Time Attributes SECTION

- Color
- Payment Type

Add Section Element

Add Section

5. In the General Properties facet, create a section for each section that is not in the General Properties facet.
    1. Click **Add Section**.
    2. From the **Section** list, select a name for the section.
       For example, Monique selects the Design Time Attributes section.
    3. In the **Sequence** field, enter a numeric value for the order in which the section appears in the facet.
    4. From the **View Type** picklist, select **Field Layout** to add desired fields and attributes.
    5. Click **Save**.
6. In the new section in the General Properties facet, add the attributes that are in the original section.
   For example, in the new Design Time Attributes section, Monique adds the Brand, Size, Capacity, and Mobile OS attributes.



General Properties FACET

General Properties SECTION

- Name *(required)*
- Product Code *(required)*
- Product Description
- Object Type
- Specification Type
- Status
- Orderable
- Not Assetizable

Effectivity SECTION

- Active
- Selling Start Date
- Selling End Date
- Fulfilment Start Date
- End of Life Date

Design Time Attributes SECTION

- Brand
- Size
- Capacity
- Mobile OS

Run Time Attributes SECTION

- Color
- Payment Type

1. In a recreated section, click **Add Section Element**.
2. From the **Field** picklist, select the field name.

    3. In the **Display Label** field, enter the label for the field as you want it to appear in the UI.
    4. In the **Sequence** field, enter a numeric value for the order in which the field appears in the section.
    5. Click **Save**.
7. After you confirm that the new sections and attributes appear in the Vlocity Product Designer, you can remove the facets with the sections that you recreated.

Results

All Product2 attributes, and fields are in the General Properties facet.

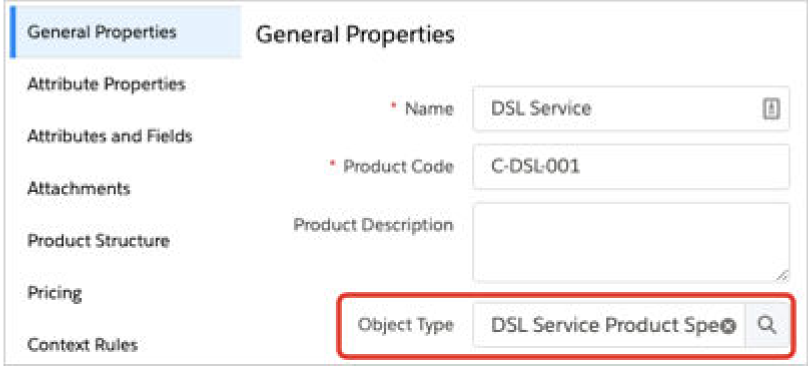
Products with Direct Attribute Assignments

Vlocity Product Designer currently doesn't support associating attributes directly to a product. As a best practice, Vlocity recommends associating product attributes through an object type.

In Vlocity Product Console, products can have attributes that are not inherited from object types. Attributes can be assigned directly to a product, whether or not the product is associated with an object type. Vlocity Product Designer does not allow attributes assigned directly to products, so attributes must be assigned to an object type that is associated with a product.

To create a new object type for a product with attribute assignments:

1. Create a new object type for the product specification or offering.
   See Create Object Types and Layouts in the Product Designer.
2. If the product created in Vlocity Product Console has attributes assigned directly to the product, assign these attributes to the new object type.
3. Recreate the product's layout in the new object type.
   You can use only the General Properties facet. TheVlocity Product Designer shows only this facet.
    1. In the Vlocity Product Designer, create a section for each section that is not in the General Properties facet for the product.
    2. In the new section in the General Properties facet, add the attributes that were assigned directly to the product.
4. In Vlocity Product Console, note and delete overrides in promotions for the attributes assigned directly to the product.
5. Note and delete attribute overrides in products that are not associated with an object type.
6. In the **Object Type** field for the product, select the new object type to associate the product with.



General Properties

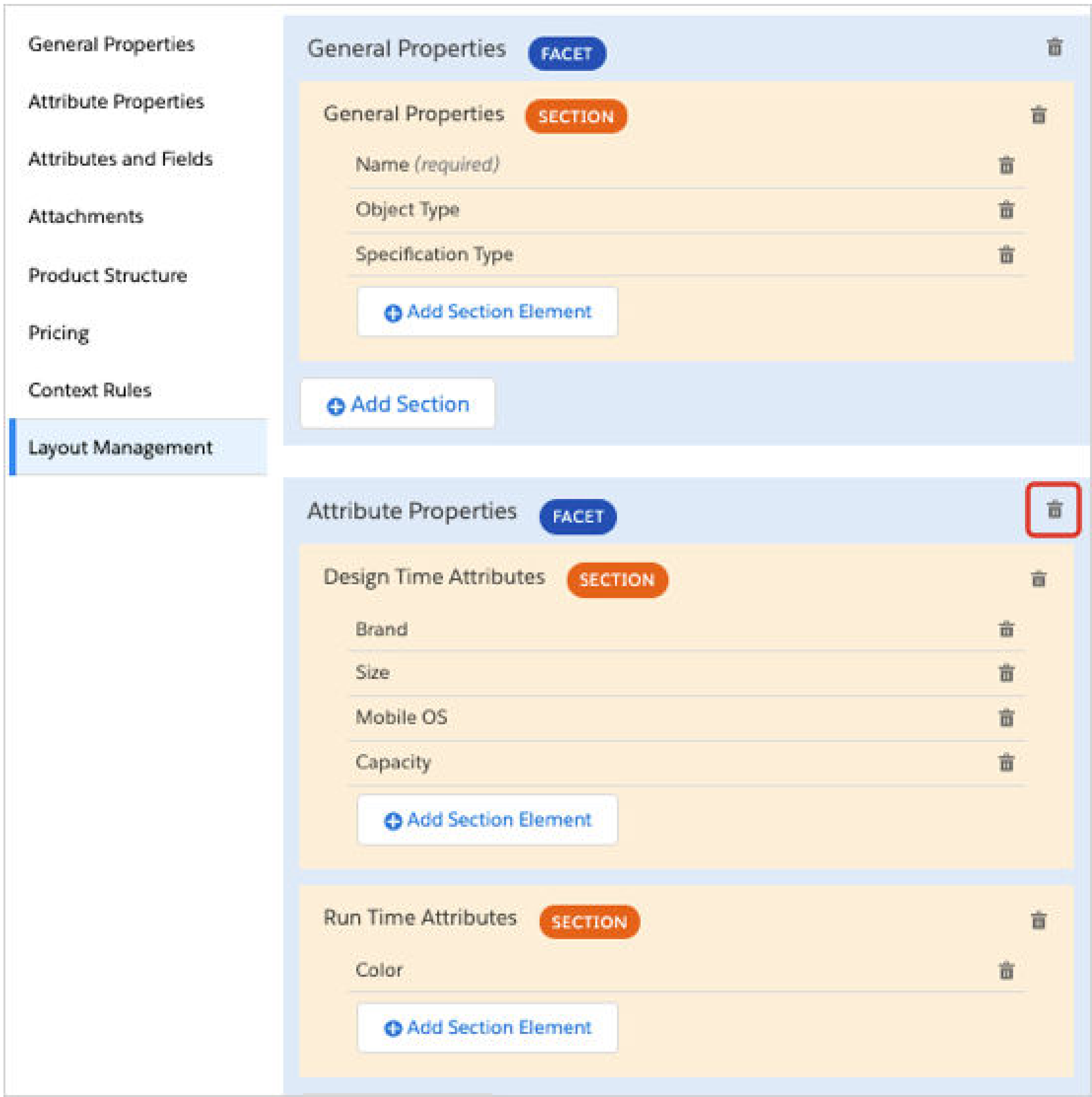
* Name: DSL Service

* Product Code: C-DSL-001

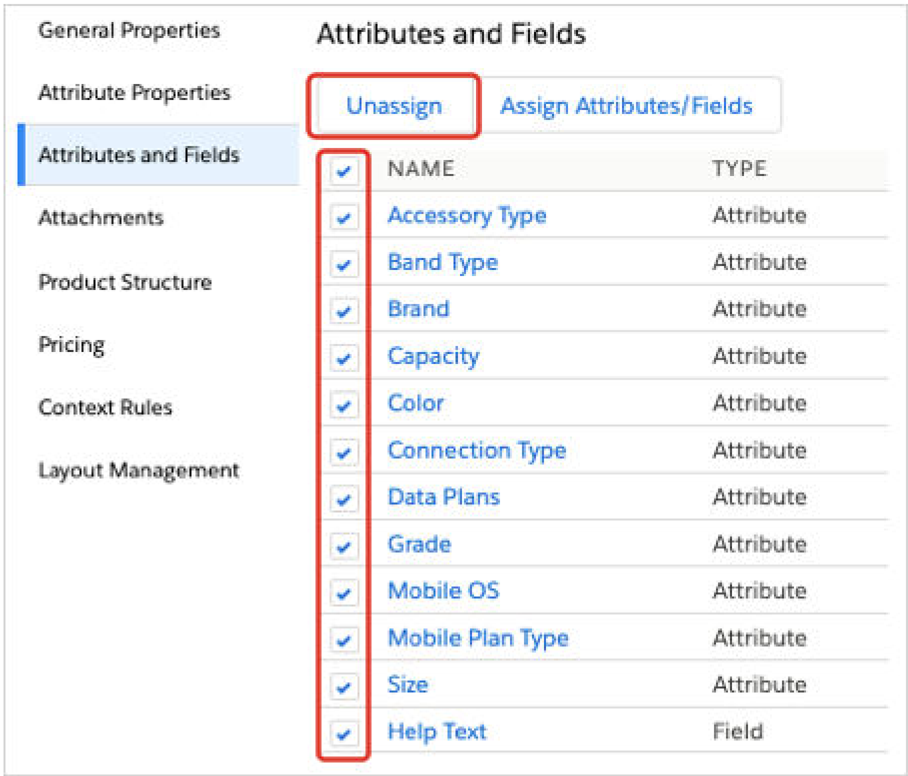
Product Description:

Object Type: DSL Service Product Spe

7. After moving the attributes to an object type, edit the product, and delete the facets, sections, or elements that you created and directly associated with the product.



8. Unassign the attributes and fields that are directly assigned to the product.



☑	NAME	TYPE
☑	Accessory Type	Attribute
☑	Band Type	Attribute
☑	Brand	Attribute
☑	Capacity	Attribute
☑	Color	Attribute
☑	Connection Type	Attribute
☑	Data Plans	Attribute
☑	Grade	Attribute
☑	Mobile OS	Attribute
☑	Mobile Plan Type	Attribute
☑	Size	Attribute
☑	Help Text	Field

9. In Vlocity CMT Administration, run EPCProductAttribJSONBatchJob.
   See Running the EPCProductAttribJSONBatchJob.
10. In the promotion, add the override for the newer attribute (which is assigned to the new object type).

Results

All Product2 attributes, and fields are moved from the product to a new object type, which is associated with the product.

Product Lifecycles

Using Vlocity Shared Catalog, you can configure products at any level in the bundle, add new products, and importantly, maintain the product entities through their entire product lifecycle. Vlocity Shared Catalog supports the commercial product lifecycle by allowing you to define current, future, past and retired products.

You can consolidate products and services from different lines of business in commercial packages or bundles. Customers ordering such packages can take an advantage from special pricing applied to the bundles of products and from new functional capabilities created by combining different types of products or services.

- Current Product is a product that is currently available for sale.

- Future Product is a product that will be sold in the future. This product can be configured and ordered, but will be fulfilled once its selling period and fulfillment begins. In order to ensure that future products are not assetized until they are fulfilled, Future Date Ordering functionality must be implemented.
- Past Product is a product that is no longer for sale but still can be fulfilled for orders that have already been placed and changed for orders that have already been assetized.
- Retired Product is a product that is no longer supported, cannot be ordered or fulfilled, or changed for orders that have been assetized.

Vlocity Shared Catalog defines these stages of the product lifecycle using Product Selling Period dates. For more information, see the following topics.

This topic pertains to Vlocity Communications, Media, and Energy Fall '18 and later.

Product Selling Period Dates

Product Selling Period Dates allow you to set the selling period, the fulfillment start date, and a date to retire commercial products. The selling period is the time period between the selling start date and the selling end date of a product, and it controls when the product can be ordered or sold.

The fulfillment start date supports pre-orders for commercial products that are can be fulfilled at a later date. An end of life date is used to retire commercial products when the business decides to discontinue selling or supporting permanently.

Starting in Fall '18, you can create products in the shared catalog with the following dates:

- Selling Start Date(**vlocity_cmt__SellingStartDate__c**): This defines when the commercial product can be ordered and sold.
- Selling End Date (**vlocity_cmt__SellingEndDate__c**): This defines when the commercial product can be stopped from ordering. Changes to existing assets are supported after the end of selling and before it is retired.
- Fulfillment Start Date (**vlocity_cmt__FulfilmentStartDate__c**): This defines when the commercial product is ready to be fulfilled. By default, this is same as the selling start date but can be a future date.
- End Of Life Date (**vlocity_cmt__EndOfLifeDate__c**): The EOL date of a product is the date on which the commercial product is no longer supported.

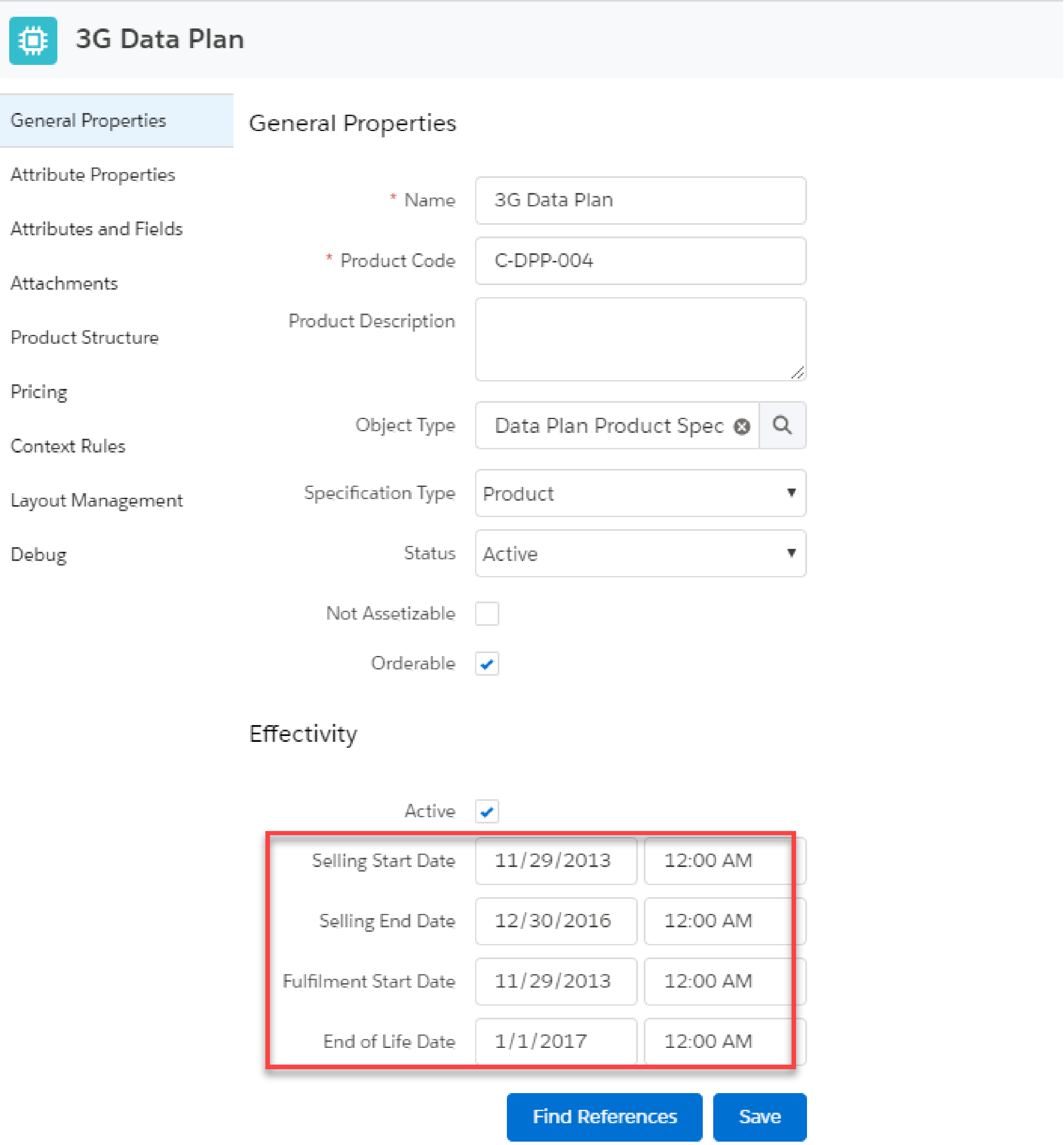
NOTE

These product selling period fields have a data type of Date/Time and are intended for commercial products only. Technical products and promotions do not use these fields and should continue to use the existing effectivity date fields. Assets do not use these fields.

This topic pertains to Vlocity Communications, Media, and Energy Fall '18 and later.

Product Selling Period Definition

You set the product selling period fields in the product dialog or in the **General Properties** facet for products in **Vlocity Product Console**.



3G Data Plan

General Properties

* Name	3G Data Plan
* Product Code	C-DPP-004
Product Description	
Object Type	Data Plan Product Spec
Specification Type	Product
Status	Active
Not Assetizable	□
Orderable	☑

Effectivity

Active	☑	
Selling Start Date	11/29/2013	12:00 AM
Selling End Date	12/30/2016	12:00 AM
Fulfilment Start Date	11/29/2013	12:00 AM
End of Life Date	1/1/2017	12:00 AM

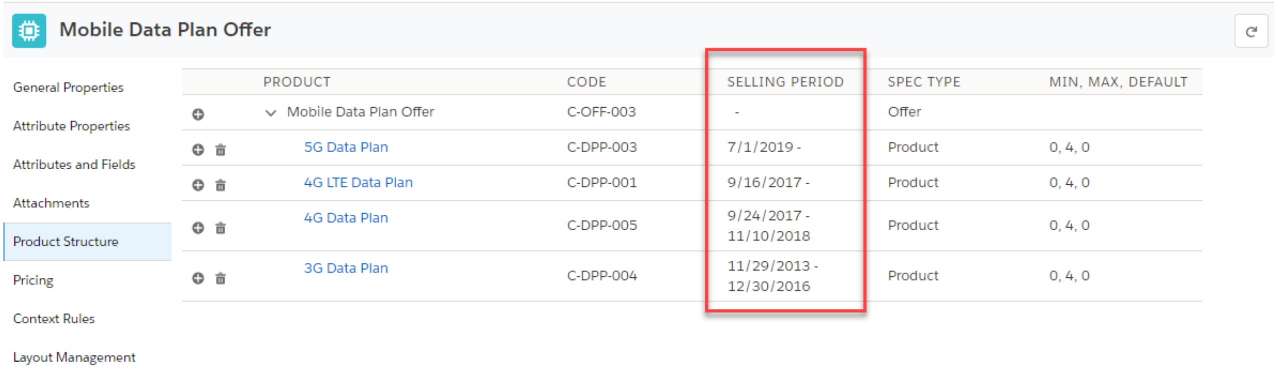
You can specify a date and a time for each field. If no time is specified, it will default to 12:00 AM. You also can choose to have no date or time, which implies that the product will be available for all of time, or you can set some of the dates but not all, as long as the following is true:

SellingStartDate <= FulfillmentStartDate <= SellingEndDate <= EndOfLifeDate

Vlocity Product Console validates all dates and times entered using this rule and will not allow you to save products until the dates are set correctly.

Product selling period data is stored in the platform cache. After you make changes to product selling period fields, you must update the product hierarchy data in the platform cache by running Product Hierarchy Maintenance, Clear Managed Platform Cache and Refresh Pricebook in this sequence.

When a product is used in bundles, it is the responsibility of the product administrator to ensure that the selling periods are consistent. To assist administrators, Vlocity Product Console has been enhanced to display the **Selling Period** dates in the **Product Structure** facet.



PRODUCT	CODE	SELLING PERIOD	SPEC TYPE	MIN, MAX, DEFAULT
Mobile Data Plan Offer	C-OFF-003	-	Offer	
5G Data Plan	C-DPP-003	7/1/2019 -	Product	0, 4, 0
4G LTE Data Plan	C-DPP-001	9/16/2017 -	Product	0, 4, 0
4G Data Plan	C-DPP-005	9/24/2017 - 11/10/2018	Product	0, 4, 0
3G Data Plan	C-DPP-004	11/29/2013 - 12/30/2016	Product	0, 4, 0

In addition, product administrators can use the **Find References** button in **Vlocity Product Console** to find references to a given product in active promotions, product bundles, assets, quote line items, order line items and opportunity line items. This button assists administrators in determining the effect of changing product selling period and end of life dates. For more information, see Finding Product References in the Shared Catalog.

This topic pertains to Vlocity Communications, Media, and Energy Fall '18 and later.

Create Picklists and Picklist Items in the Product Designer

A picklist is an ordered list of values that customers can choose from when buying a product. If an attribute includes predefined values, a picklist represents all available values for that attribute. You can reuse picklists across attributes assigned at the object type level or the product level.

This information is for the Vlocity Product Designer and later. For the Vlocity Product Console, see Create a Picklist.

You must define picklists before creating attributes.

You can create picklists for these data types: text, number, Boolean (True/False), date or datetime.

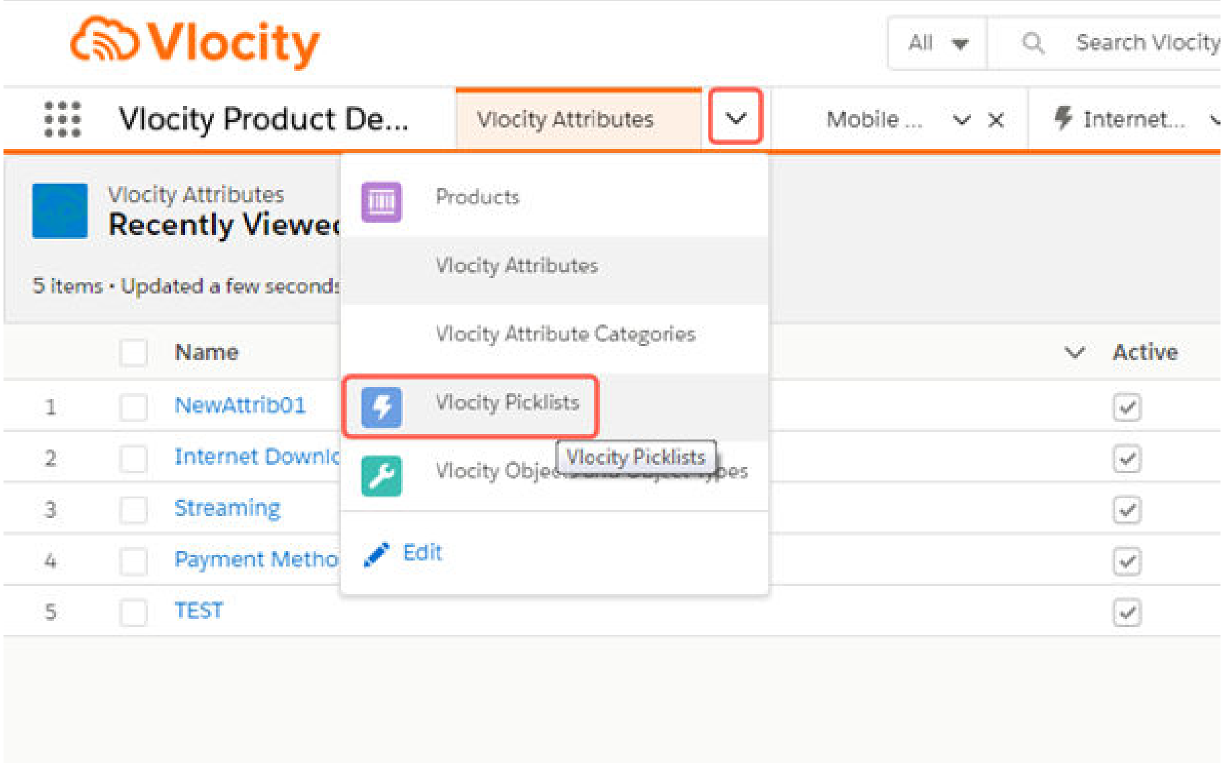
Example 3. Example: iPhone picklists

The SmartPhone object has color and storage attributes, but the iPhone 11 has specific options for each. Monique is a product modeler and creates two picklists with the following list items for the iPhone 11. She makes the first option the default for each.

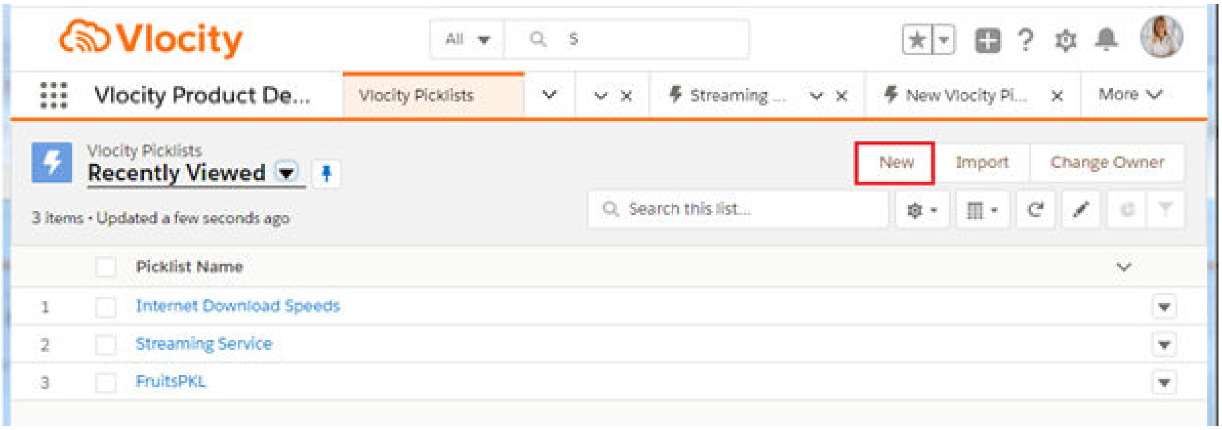
- iPhone 11 Colors — black, white, red, purple, green, and yellow
- iPhone 11 Storage — 64 GB, 128 GB, and 256 GB

To create a picklist:

1. From the Vlocity Product Designer, click the ⌄ button, and select **Vlocity Picklists**.



2. Click **New**.



Vlocity Picklists
Recently Viewed

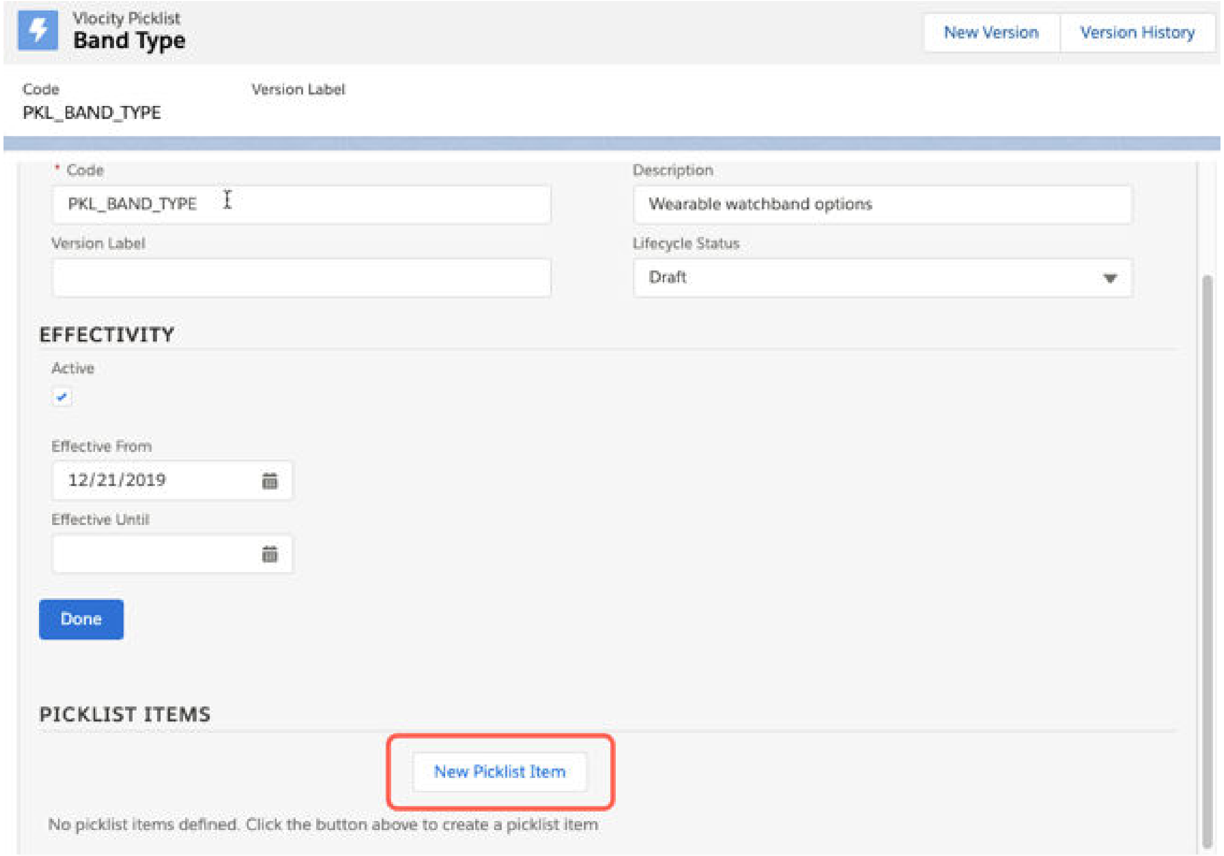
3 items · Updated a few seconds ago

	Picklist Name
1	Internet Download Speeds
2	Streaming Service
3	FruitsPKL

3. In the New Vlocity Picklist page, enter the following information.

Field	Description
Picklist Name	Enter a name. The name is used at design-time, not runtime. Customers do not see this name.
Code	Enter a unique code that is shown only during design-time. Vlocity recommends PKL_<alphanumeric-value> for the naming convention for attributes. Example: PKL_BANDWIDTH.
Version Label	This field is reserved for the Versioning functionality (in Beta). Enter a value to indicate a new version, such as v1.0.
Data Type	Enter the type of data for picklist items. **NOTE** Picklist attribute values are stored as text, even if the picklist you select has another Data Type, such as Number. If you use a Picklist attribute in a Set Value or Set Default Value rule, verify that the rule expresses the attribute's data type as text for the Value to Set and in the rule's Expression box.
Description	Enter a description of the picklist, such as its purpose or associated offering.
Lifecycle Status	This field is reserved for the Versioning functionality (in Beta). Set and track the lifecycle status of the picklist. You can only set this field if Specification Type is set to Product or Offer. **CAUTION** Don't use the Released value in Fall '20 because it can have undesirable effects.
Active	Indicate that the picklist is currently in use.
Effective From Effective Until	Enter a date or date range when the item is effective.

4. Click **Done**.
The page for the new picklist appears with a section for Picklist Items.



Vlocity Picklist
Band Type

Code
PKL_BAND_TYPE

Version Label

* Code: PKL_BAND_TYPE

Description: Wearable watchband options

Version Label:

Lifecycle Status: Draft

EFFECTIVITY

Active ☑

Effective From: 12/21/2019

Effective Until:

PICKLIST ITEMS

No picklist items defined. Click the button above to create a picklist item

5. Under the Picklist Items heading, click **New Picklist Item**.
6. Enter the following information.

Field	Description
Value	Enter the value of a picklist item. This value must be unique within a picklist. For example, for a Color picklist, you cannot have one item labeled **`Silver`** with a value of **`Silver`** and another item labeled **`Default`** with a value of **`Silver`**. The Default item must either have a different, unique value or be removed. The data type of this field is set by the picklist data type. The text value is important if the picklist item's data type is other than text. For example, if the data type is numeric, and the value is 1, the text value is one (or 1 as text).
Code	Enter the unique code (unique within this picklist) for the picklist item.
Label	Enter the item name as it appears on the picklist.
Active	Indicates that the picklist item is in use. Picklists show only items that are set to Active. In Vlocity Cart, for example, if a line item has an attribute with a picklist, only the active picklist items are shown. Inactive picklist items are not. Also, batch jobs that use attribute assignments might be affected, such as EPCProductAttribJSONBatchJob which updates the JSONAttribute field on Product2 based on its attribute assignments.

Field	Description
Effective From Effective Until	Enter a date or date range when the item is effective.

7. Click **Done**.
8. Repeat steps 5-7 until you've added all your picklist items.
9. To modify an existing picklist item, click the **Edit** icon, change fields, and click **Done**.
10. To change the sequence in which the picklist items display at runtime, drag and drop the picklist items, and click **Save Changes**.

Results

The new picklist with its picklist items is saved and available to use.

What's Next

Create Attribute Categories in the Product Designer

See Also

- Overview of Product Creation in Product Designer

Attributes and Overrides

Products may or may not have attributes. A product that does not have attributes comes only in one form, and customers do not make any decisions about the features of the product, such as a headset or a computer cord that come in one color.

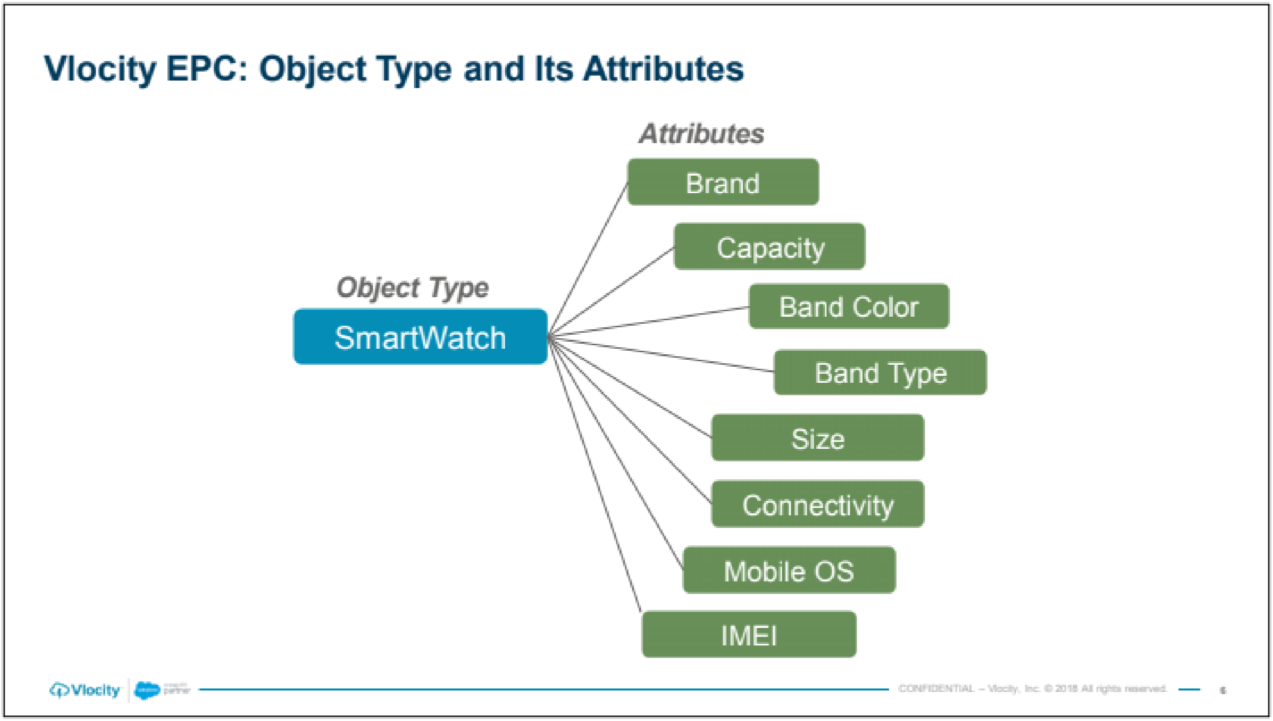
If a customer can choose from a selection of product features, you must define these features as attributes before creating an object type. Also, if an attribute includes multiple values that a customer can choose from, you must define picklists before defining the attribute.

Product attributes provide a powerful and flexible way to extend product entities. Product attributes are stored in an aggregate JSON blob on the Product object. They are used extensively throughout Vlocity CPQ and Vlocity Order Management.

Uses of Attributes

In Vlocity CPQ, you can use product attributes to filter products at runtime, displaying the best-fit products to customers. You can use them to configure product specifications, allowing users to specify product colors, sizes, and other options. You can also use attributes to change product pricing using attribute-based pricing rules.

Finally, in Vlocity Order Management, you can use attributes to map commercial products, or the products that customers see, with technical products that communicate with backend systems to fulfill the products that are purchased.



Vlocity EPC: Object Type and Its Attributes

Object Type

SmartWatch

Attributes

Brand

Capacity

Band Color

Band Type

Size

Connectivity

Mobile OS

IMEI

CONFIDENTIAL – Vlocity, Inc. © 2018 All rights reserved. 6

In the above diagram, the object is a smartwatch with various attributes, shown on the right. The provisioning process requires the definition of the attributes, such as size and data capacity to cater to customer preference.

When a customer orders an offering based on the smartwatch object, such as an Apple Watch Series 5, you can think of it as a collection of statements that describe what they are ordering.

Table 1. SmartWatch product attributes

Attribute	Value
Brand	Apple
Band type	sport band
Band color	gray
Capacity	8 GB
Product width	38mm
Operating system	watchOS 4
Feature	GPS
International Mobile Equipment Identity (IMEI) number	assigned during provisioning

You do not want to redefine all of these statements each time a customer orders a product. Instead, you want to reuse attributes and the definition of a product. Some of these attributes are set during design-time, some at runtime, and others during order decomposition (provisioning).

With Vlocity EPC, you create attributes once and assign them to an object type for the product. For example, Google and Apple smartwatches could both have the object type SmartWatch, each with a different set of attributes. Having this structure in place helps the CSR support customers by giving them the correct product details and place consistent orders.

Overrides

Although you cannot delete any inherited attributes or fields, you can override them at the product level. In the SmartWatch example, you could override the color picklist inherited from the SmartWatch object type to include colors that are specific to the Series 3 Apple Watch.

You can also add additional attributes directly to the offering level.

Encrypted and Not Assetizable Attributes

In Vlocity Product Designer, you can flag attributes for technical products as not assetizable and encrypted.

Not Assetizable

The **Not Assetizable** attribute metadata flag is useful for transitory information that should not persist or does not need to persist beyond the completion of order processing where assetization would usually occur. This metadata flag is stored in Vlocity Order Management, not in the Salesforce platform.

An example of transitory information is the initial password that a customer uses to log in to the provider's website for the first time. During fulfillment, the initial password is generated within an orchestration order item, then saved on the technical fulfillment request line, which is then passed to the next orchestration item that emails the password to the customer. The password does not need to persist at the end of order management. Other information that does not need to persist can be the result of back-end services connected to order items that do not become customer assets, such as a service needed to fulfill a customer’s order. As an example, there might be a check to see whether a required modem is in stock and on the delivery vehicle. This information is not needed after that check has been made.

Encrypted

The **Encrypted** attribute metadata flag is used to protect personally identifiable information (PII) and other data such as the customer name, phone number, social security number, and so on. These pieces of information can be captured as attributes and should be sent downstream only for legitimate purposes. You can use the encrypted flag to indicate the PII or other information must be stored securely.

Setting Encrypted and Not Assetizable

Not Assetizable and **Encrypted** are attribute metadata flags. You set these flags when an attribute is assigned to an object type in Unified Product Catalog. These flags can only be set when the **Run-time Configurable** flag is also set. Once the attribute is marked as **Run-time Configurable**, you can then set **Not Assetizable** or **Encrypted** or both.

If none are selected, attribute values will be stored in the technical inventory after decomposition. If both are selected, the attribute will be encrypted during processing and not stored with the inventory items. However, if you set the **Encrypted** flag to True, and **Not Assetizable** to False, the attribute is stored in the technical inventory, but the attribute will be encrypted. Technical assetization stores the output of decomposition into the technical tables in Vlocity Order Management, not in the Salesforce platform.

Once assigned to the object type, every product that is created in that object type will inherit these metadata flags, along with the other attribute and metadata. **Not Assetizable**, **Encrypted** and **Run-time Configurable** are then non-overridable at the product level. This prevents any product manager or EPC administrator from removing these flags during product creation.

If you set these flags on existing attributes, it will not affect attributes already stored in existing technical inventory.

Not Assetizable and **Encrypted** metadata flags are intended to trigger functions in downstream order management systems, such as Vlocity Order Management. There is no encryption to these attributes when used on commercial products in Vlocity CPQ or in Vlocity Order Management, and the information will remain stored and accessible on the originating order. Consequently, these attribute flags are intended for use in technical products only.

Using Encrypted and Not Assetizable

Once these attribute metadata flags have been set on an object type and inherited by a product, you must create decomposition rules to map the attributes to the downstream fulfillment requests, which are the basis for the technical inventory. See Partial Assetization of Product Attributes.

Attributes vs. Fields

When defining product properties, you can use attributes or fields. The choice depends on how you plan to implement property data during order capture.

- *Attributes*: A list of all attributes and their associated Attribute Codes assigned to the Product2 Object.
  If a data element, such as a SKU number, is intended to be common to all products across your product catalog, then you should create a new field on your product object. Typically, adding new fields requires Salesforce system admin privileges.
- *Fields*: A list of all fields and their associated Field API name for the Product2 Object.
  If the data element is intended to be specific to a product, or class of products, then you should create an attribute, such as download speed. At runtime, attributes are visible to customers in Vlocity Cart and can be used to filter the catalog and provide configuration capabilities. Adding attributes requires access to the Product Console.

The fundamental differences between fields and attributes are:

	Fields	**Attributes**
Where the value is created?	Created in Salesforce Setup on a Salesforce object. Field example: Orderable.	In Vlocity attribute/category interface. Attribute example: Color.
Where the value is stored (including metadata, such as default value, picklist, etc.)?	Each field value is stored in a separate Salesforce field. Related metadata is stored separately in Salesforce. For example, if you create 15 fields, there will be 15 field records in Salesforce and each record will store its value.	Each attribute value and its' metadata (including picklists, default value, etc.) is stored in a JSON blob (i.e. in a FIELD in JSON). For example, if you create 15 attributes, all 15 attributes and all their metadata will be stored in one JSON blob which is stored in one Salesforce field.
Who can create?	Requires Salesforce System Administrator privileges.	A Product Manager with sufficient privileges.

	Fields	Attributes
Is searchable at runtime?	Each field is stored in a separate Salesforce database field and is easily searchable.	Since all attribute data is stored in one JSON blob, expect performance implications when using search.
When to use at design-time to define Product2 properties	Use for commonly used properties that are searchable and runtime (order time) configurable.	Use for properties that are sparely used.
When to use at runtime to define asset properties	Do not use.	Use attributes only. For example, IMEI number, VIN #, etc.

Attribute and Picklist Overrides for the Object Type and at the Product Level

An attribute is a reusable key/value pair. An attribute can be associated with a picklist that has possible values for the attribute. You can set the default value of an attribute. When you associate the attribute with an object type, child object types and products inherit the attribute but can override it.

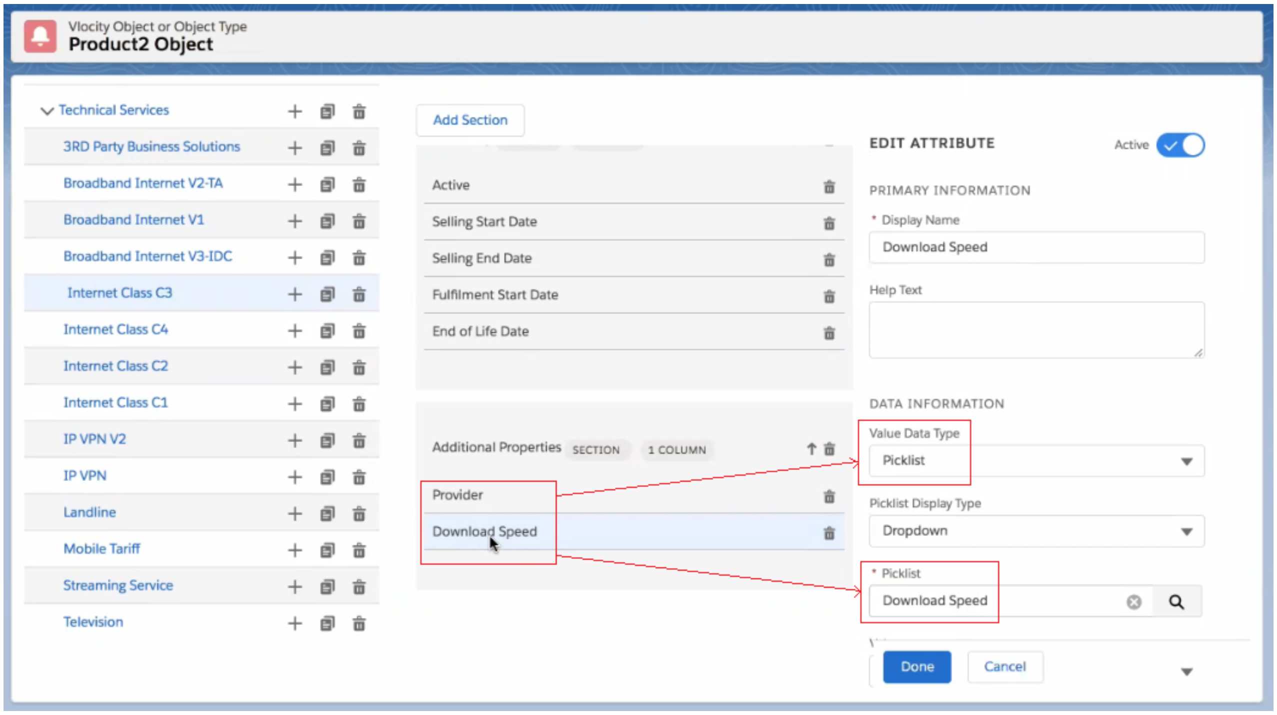
In summary, you are copying the attribute, which inherits the properties of the original attribute, then modifying those properties.

At the object type level, you can also change or replace the Picklist Display Type and configure the attribute behaviors.

At the product level, you can only change the attribute value for a product that was created from the object type. For an attribute that uses a picklist, you can select a different value, but you cannot change or replace the picklist itself.

Example 4. Example: Download Speed

As an example, you may have an object type to which you have added an attribute for Download Speed. The attribute is associated with a picklist named Download Speed.



Vlocity Object or Object Type
Product2 Object

- Technical Services
  - 3RD Party Business Solutions
  - Broadband Internet V2-TA
  - Broadband Internet V1
  - Broadband Internet V3-IDC
  - Internet Class C3
  - Internet Class C4
  - Internet Class C2
  - Internet Class C1
  - IP VPN V2
  - IP VPN
  - Landline
  - Mobile Tariff
  - Streaming Service
  - Television

EDIT ATTRIBUTE Active

PRIMARY INFORMATION

* Display Name: Download Speed

Help Text:

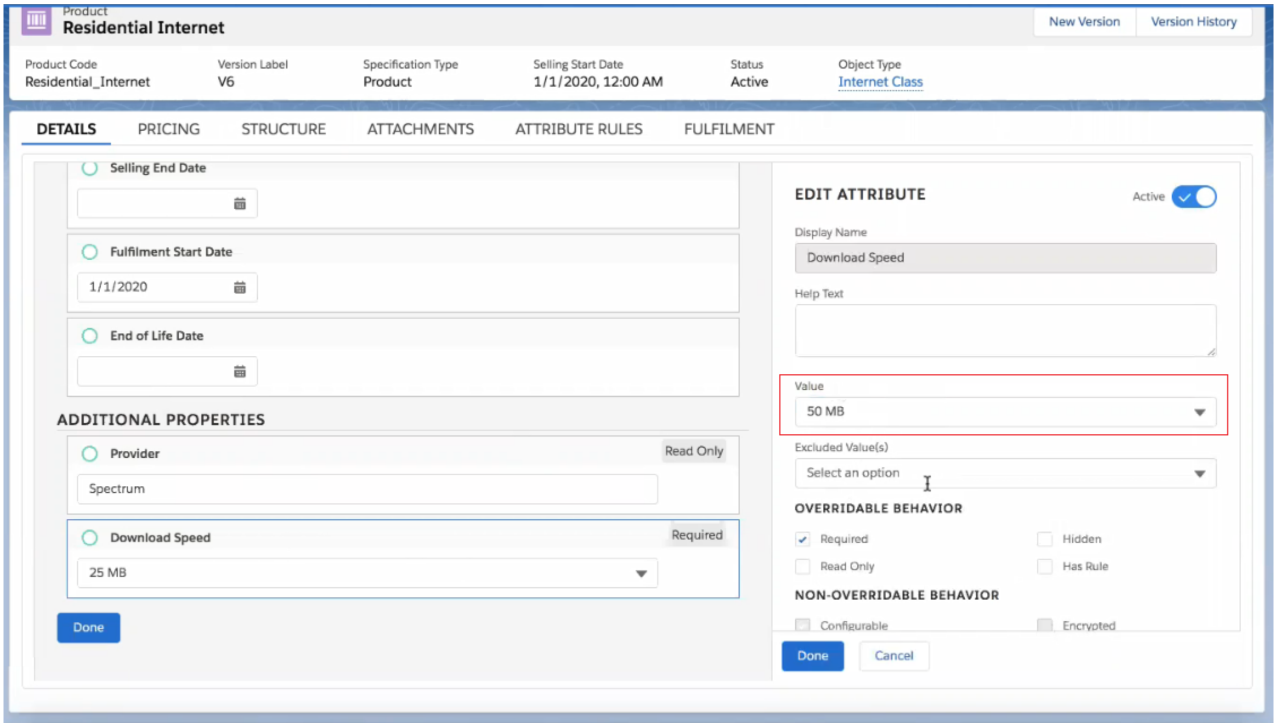
DATA INFORMATION

Value Data Type: Picklist

Picklist Display Type: Dropdown

* Picklist: Download Speed

The default value for Download Speed is initially 25 MB, but you can change it, for example, to 50 MB.



Product
Residential Internet

Product Code	Version Label	Specification Type	Selling Start Date	Status	Object Type
Residential_Internet	V6	Product	1/1/2020, 12:00 AM	Active	Internet Class

Selling End Date:

Fulfilment Start Date: 1/1/2020

End of Life Date:

ADDITIONAL PROPERTIES

Provider — Read Only: Spectrum

Download Speed — Required: 25 MB

EDIT ATTRIBUTE Active

Display Name: Download Speed

Help Text:

Value: 50 MB

Excluded Value(s): Select an option

OVERRIDABLE BEHAVIOR

☑ Required ☐ Hidden
☐ Read Only ☐ Has Rule

NON-OVERRIDABLE BEHAVIOR

☐ Configurable ☐ Encrypted

Create Attribute Categories in the Product Designer

Use attribute categories to group product attributes. They also appear as headings at runtime in Vlocity Cart's Filter and Configuration windows.

This information is for the Vlocity Product Designer. For Vlocity Product Console, see Create New Product Attribute Categories.

Attribute categories are required to create product attributes. You define the type of attribute at the Attribute Category level.

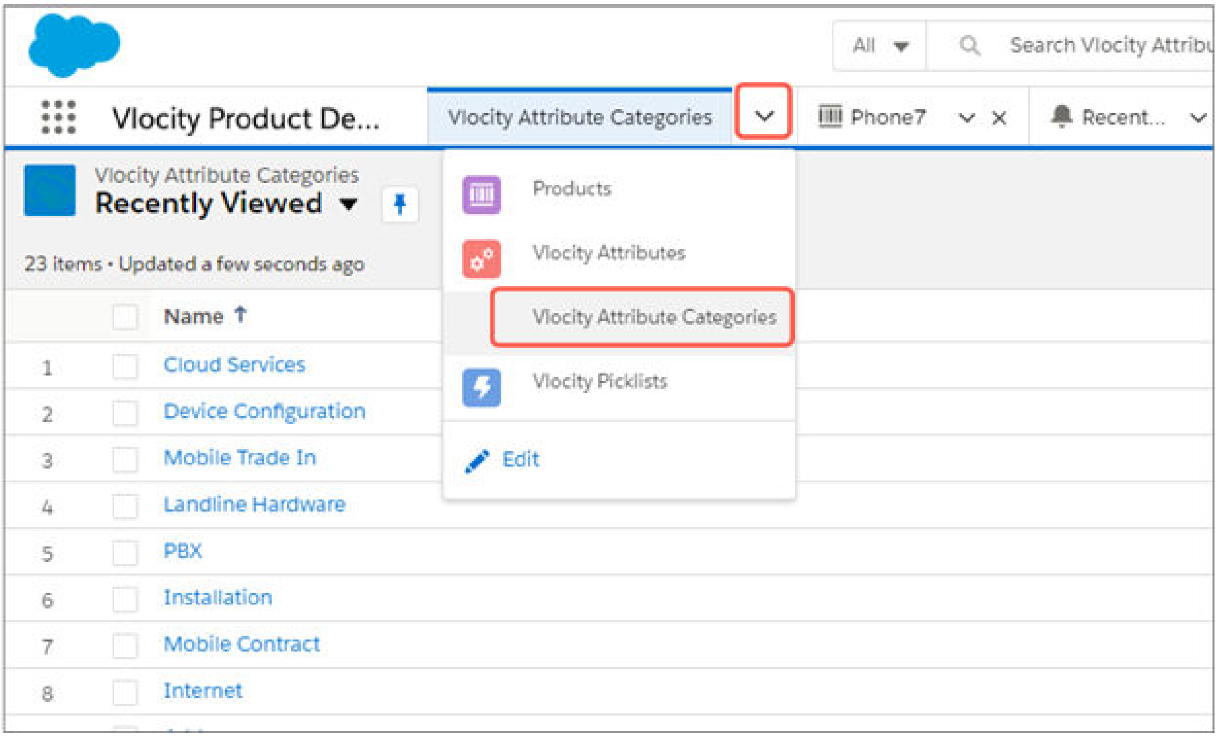
Example 5. Example: Smartphones and Wearables Attribute Categories

As the product modeler, Monique has attributes that are common to both smartphones and wearables, so she creates the SmartDevices attribute category. But smartwatches have options that phones don't have, so she creates a Wearables category.

These categories are general to types of products rather than specific to a particular brand or model. She associates the categories with the Base Product object in her product catalog so that they're inherited by all child objects, including the Smartphone and Smart Wearable objects.

To create an attribute category:

1. From the Vlocity Product Designer app, click the ⌄ button, and select **Vlocity Attribute Categories**. The Vlocity Attribute Categories page opens with a list of attribute categories.

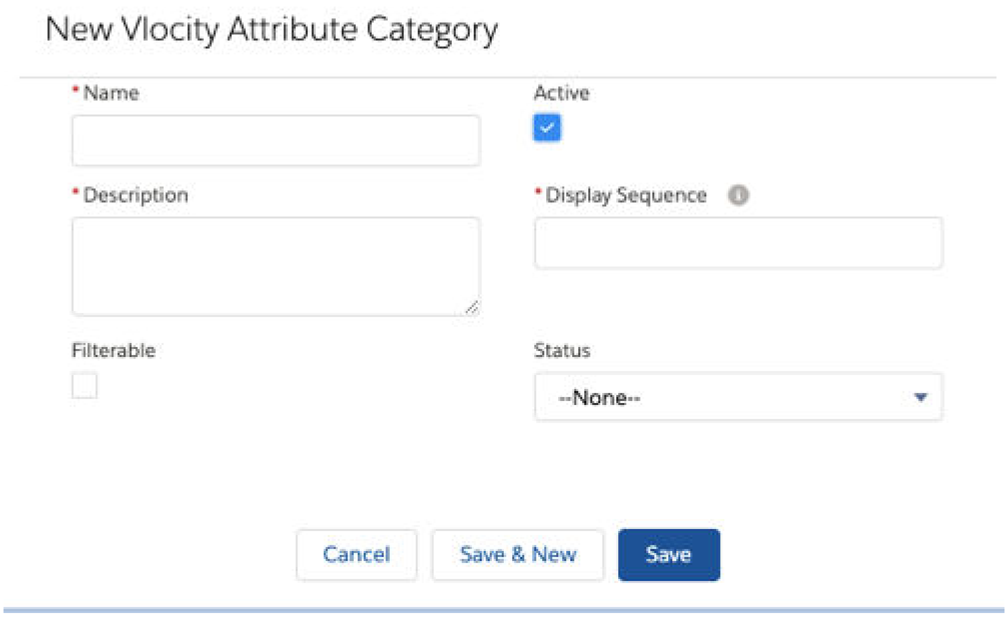


Vlocity Attribute Categories
Recently Viewed

23 items • Updated a few seconds ago

	Name ↑
1	Cloud Services
2	Device Configuration
3	Mobile Trade In
4	Landline Hardware
5	PBX
6	Installation
7	Mobile Contract
8	Internet

2. Click **New**.
   The New Vlocity Attribute Category dialog box opens.



3. In the New Vlocity Attribute Category dialog box, enter the following information.

Name	Description
Name	Enter a name for the attribute category. This name is shown at runtime in Vlocity Cart's filter and configuration windows.
Description	This description is used only during design-time.
Applicable Types	Select Product2.
Applicable Sub Type	Select one of the following: • Product Attribute — For Product records. Only Attribute categories set to Product Attribute are available in Entity Filter Conditions. • Profile Attribute — For Profile records. Profile attributes are used with Vlocity Intelligence. • Inventory Item — For attributes to set up housing filters.
Display Sequence	Enter a unique number to indicate the sequence in which the category is displayed on the record detail page. It orders the attribute category in Vlocity Cart's filter and configuration windows. Use a sequence of 10, 20, 30 so that you can add to the sequence at a later time.
Active	Indicate that the category is in use.
Filterable	If enabled, use the attribute category to filter products.
Status	Set and track the lifecycle status of the attribute category.

4.
5. Click **Save**.

Results

The new attribute category appears in the Vlocity Attribute Categories page. You can add attributes to this category.

What's Next

Define Attributes in the Product Designer

See Also

- Overview of Product Creation in Product Designer

Define Attributes in the Product Designer

A product attribute is a characteristic that defines a particular product and affects a consumer's purchase decision. Attributes can include product characteristics such as make, color, bandwidth, storage, and other options.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Create Attributes.

You can use attributes in several ways:

- Configure product specifications, allowing users to specify product colors, sizes, and other options.
- Change product pricing using attribute-based pricing rules.
- In Vlocity Order Management, map commercial products (the products that customers see) to technical products that communicate with backend systems to fulfill the products that are purchased.

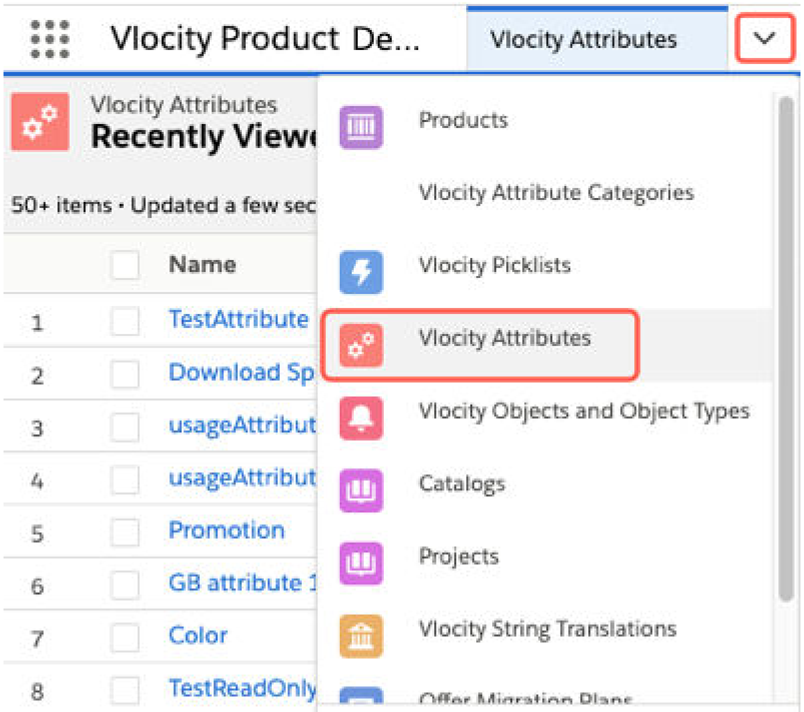
Example 6. Example: Smart Devices Attributes

In modeling the smart devices, Monique creates common attributes for the Smartphones and Wearables attribute categories. She associates the attributes with the Base Product object in her product catalog so that they’re inherited by all child objects, including the Smartphone and Smart Wearable objects.

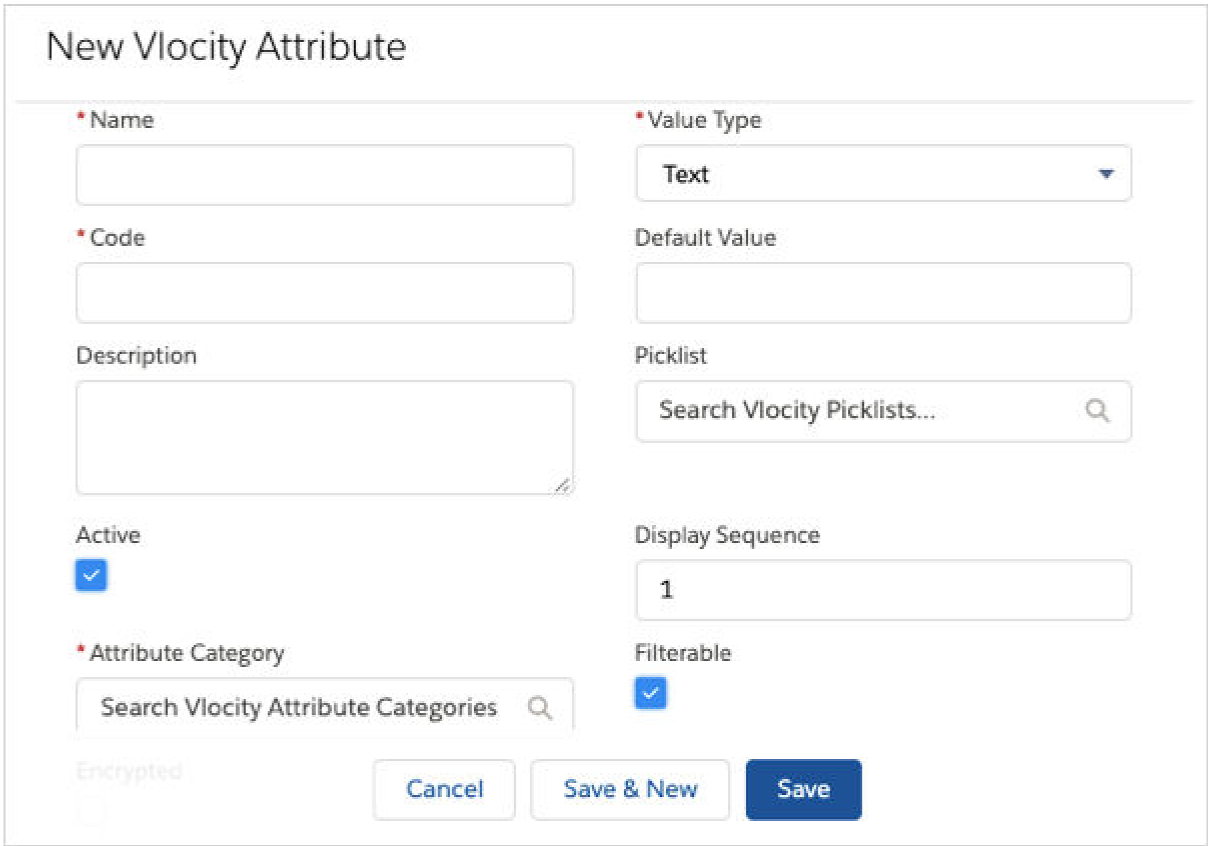
- Smart Devices attribute category — Color, OS, Brand, Storage, Connectivity, Size
- Wearables attribute category — Band Type

Before You Begin
• Create an attribute category that the new attribute is assigned to. See Create Attribute Categories in the Product Designer. • If needed, create any picklists that the new attribute uses. See Create Picklists and Picklist Items in the Product Designer.

1. From the Vlocity Product Designer, click the ⌄ button, and select **Vlocity Attributes**.
   The Vlocity Attributes page opens with a list of attributes.



2. Click **New**.
   The New Vlocity Attribute Category dialog box opens.



New Vlocity Attribute

* Name

* Value Type: Text

* Code

Default Value

Description

Picklist: Search Vlocity Picklists...

Active: ☑

Display Sequence: 1

* Attribute Category: Search Vlocity Attribute Categories

Filterable: ☑

Encrypted: □

3. Enter the following information.

Field	Description
Name	Names are used to identify attributes in searches.
Value Type	Select the data type for the attribute, such as None, Checkbox, Number, Percent, or Text.
Code	A unique code for the attribute that is shown only during design-time. Vlocity recommends ATT_<alphanumeric-value> for the naming convention for attributes. Example: ATT_BANDWIDTH.
Default Value	Enter a value for value types such as Text or Number. If the attribute uses a picklist, this value isn't used.
Description	Descriptions help users identify the purpose of the attribute. This description is used only during design-time.
Picklist	Select the name of the picklist to use with this attribute (if the values are to be selected from a picklist).
Active	Indicate that the attribute is in use.
Display Sequence	Enter a unique number to indicate the sequence in which the attribute is displayed in the attribute category and on the detail page when configuring in Vlocity Cart. Use a sequence of 10, 20, 30 so that you can add to the sequence at a later time.

Field	Description
Attribute Category	Attributes are organized into predefined categories. Select or create a category. • Click inside the field and search Vlocity Attribute Categories to select one. • From the list, click **New Vlocity Attribute Category** to add a category.
Filterable	If enabled, use the attribute to filter products in the cart.
Encrypted	For use with technical products, enable **Encrypted** on an attribute to ensure privacy, such as Personally Identifiable Information (PII). See Encrypted and Not Assetizable Attributes.
Exclude From Basket Cache	If enabled, the attribute is excluded from searches of the basket cache for the same bundled product. The attribute isn't part of the response cache when the Digital Commerce ConfigureOffer API is called. For example, you can set the IMEI or serial number attribute to be excluded from the cache. The benefit is that you can find more matches when you searching the basket cache, while excluded attributes don't have the same values. Use this property for attributes whose values are unique for each account. The CPQ engine doesn't calculate the price and rules. The assumption is that if the attribute isn't cached, the customer has no price and rules on that attribute. Attributes with the setting enabled aren't available for rules that check the attributes in the cache. For example, attribute-based configuration and qualification rules wouldn't be able to use an attribute that is excluded from the cache because it wouldn't be present. Also, in Vlocity Rules with Product Relationships, Modify Attribute relationships shouldn't be defined that are based on attributes excluded from the cache. **NOTE** The CPQ Cart APIs don't process this setting. Only the Digital Commerce APIs process it.

4. Click **Save**.
5. Under **Applicable Objects**, review the object types assigned to the attribute.
   The default object type is Product (Product2 Object). All objects based on Product2 Object have the new attribute. You might also select one of your base object types.
6. If you change the assigned objects, click **Save**.

Results

The new attribute is saved, and you can add it to object types and can configure for product specifications and offers.

What's Next

Create Object Types and Layouts in the Product Designer

See Also

- Overview of Product Creation in Product Designer
- Export a Product Report in Product Designer

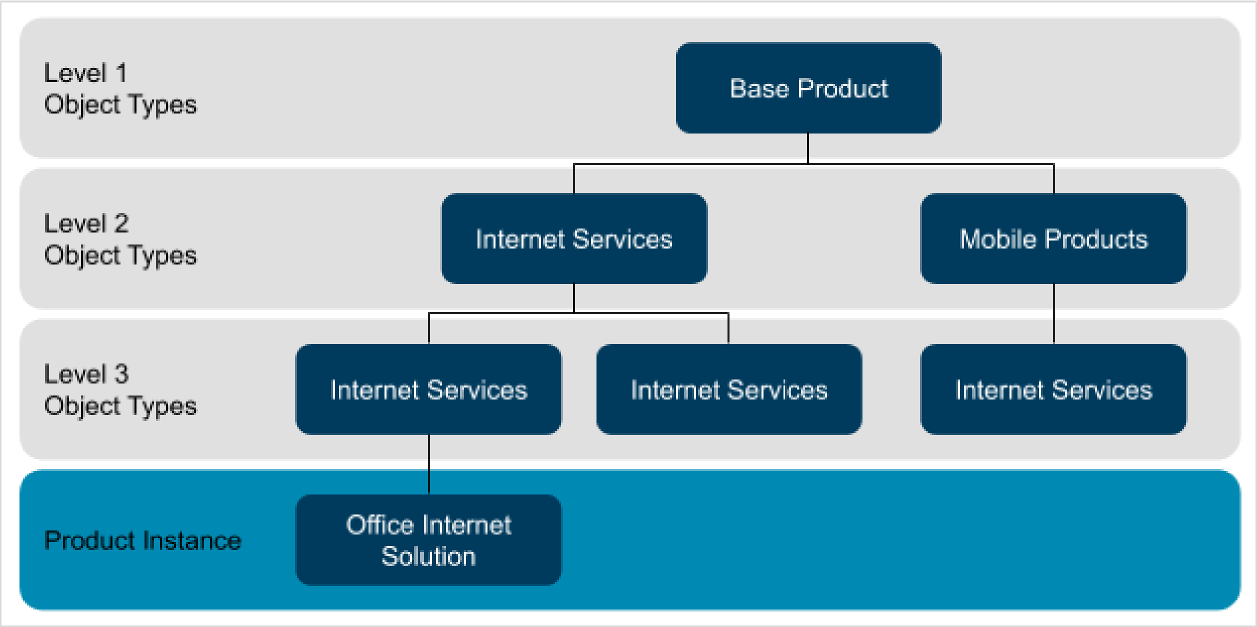
Object Types and Guidelines

An object type is a reusable entity that defines properties, such as fields and attributes, and layouts for all product instances. Object types are like Salesforce record-types but with enhanced capabilities. You can use object types to group products with similar characteristics and ensure consistent behavior and application of rules.

Object Types Inheritance

Object types support hierarchy inheritance, which allows you to create objects such as a mobile device with attributes like brand, capacity, and color that are common among mobile devices. From the object type, you can create a child object type, such as an iPhone, and assign specific attributes, such as color. Any product linked to the iPhone object type will automatically inherit the iPhone attributes from the mobile device parent object type as well as any iPhone-specific attributes of the iPhone child object type.

Vlocity's object types are designed as an "Is A" inheritance architecture, meaning you can create relationships between abstractions, where one object type is a subtype of another object type. The first level of abstraction is known as an object type, and the subsequent levels are known as subtypes. When you apply this relationship to a product, you create a relationship consisting of a product instance.



NOTE

You must add attributes or fields to the layout. Otherwise, they are not displayed when you create new products from the object type.

When defining object types and their inheritance:

- Vlocity recommends creating a *base object* during the initial setup. This is your product catalog's prime object from which all the other objects descend, and should include all the fields (but not attributes) in the product catalog.
- You can then create child object types for all the product classes and all the products in your catalog, and they inherit the layout of the base product and all the associated fields and attributes.
- You create the layout for the base product manually.
- An object subtype inherits the layout via deep copy, but any subsequent changes to the parent object type's layout are not replicated down the hierarchy after initial creation. This allows you to change the layout, allowing you to add or delete fields for each object type without altering the layout with the addition of any fields on the parent object type.
- Inherited attributes and fields cannot be deleted but must be unassigned from the parent object type. Attributes are specific to products and classes of products and cannot be removed.
- New attributes and fields assigned to an object type are inherited dynamically by its subtypes. If you assign new attributes to an object type, all object subtypes and products linked to the object type hierarchy are dynamically assigned those attributes.

Guidelines for Creating Object Types

- Each object type can have only one layout.
- Each layout has sections and elements.
- The General Properties facet can have different sections to organize your elements.
- You can add metadata.
- Attribute Category is not hard-coded, so you can define it any way you want.

See Also

Create Object Types and Layouts in the Product Designer

Create Object Types and Layouts in the Product Designer

An object type that you create serves as a template that stores the configuration of attributes, fields, and layouts. Object subtypes (or child object types) or associated products inherit the configuration of the base object's layout, fields, and attributes.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Create an Object Type.

You can modify this configuration for each object type. But you must modify it before you create object subtypes so that they inherit the newly modified configuration.

With an object type's layout, you control the layout and organization of its attributes and fields. A layout has one or more sections, and each section has attributes or fields.

TIP

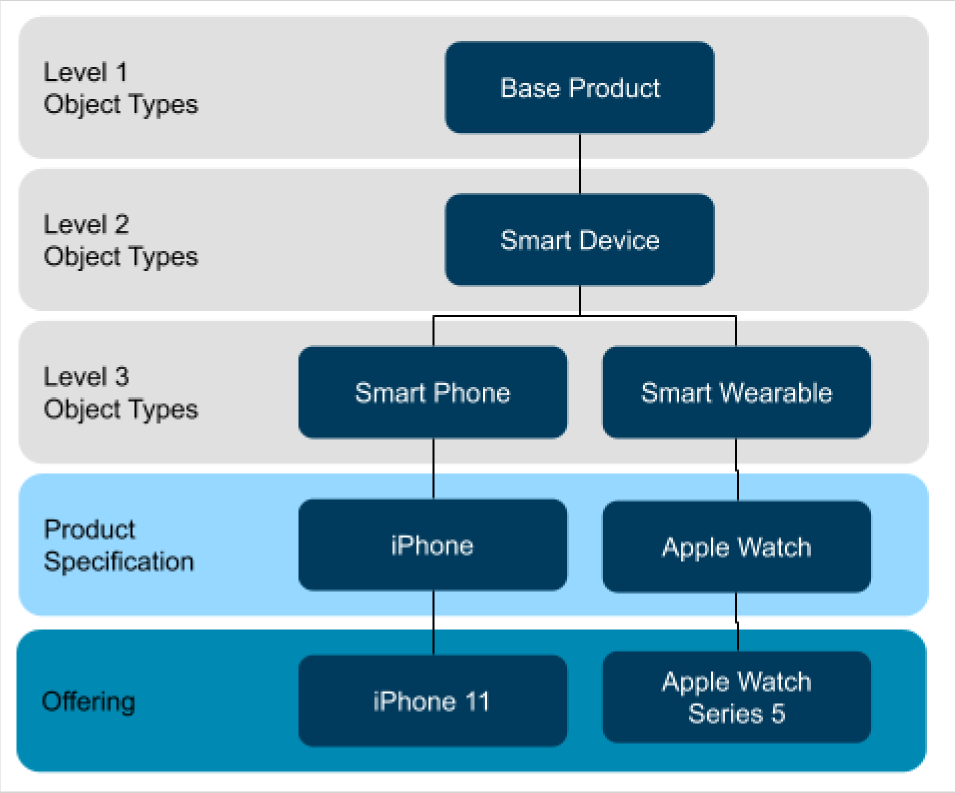
Object types are internal classifications of products. Object types differ from specifications in two key ways:

- Object types are used in EPC but not by order capture.
- Product specifications have cardinalities while object types don’t.

Example 7. Example: Smart Device Object and Child Objects

To model smart phones and wearables, Monique creates a Base Product object type for all object types. She creates a child object type named Smart Device, which can be the parent for a variety of object types. In this case, Monique creates two children of the Smart Devices object: Smart Phone and Smart Wearable.

The following diagram shows the objects that Monique creates.



NOTE

Don’t create child object types until you define attributes, fields, and layouts for the base object type. This order ensures the children inherit the basic settings, which you can later modify.

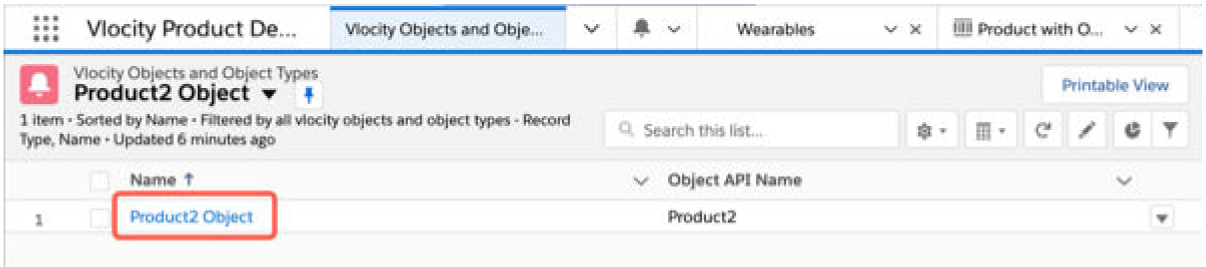
Create an Object Type

Before You Begin
• Review the guidelines for creating object types. • Create the attributes that the new object type uses. See Define Attributes in the Product Designer.

1. From the Vlocity Product Designer app, click the ⌄ button, and select **Vlocity Objects and Object Types**.
   The Vlocity Objects and Object Types page opens.

   TIP

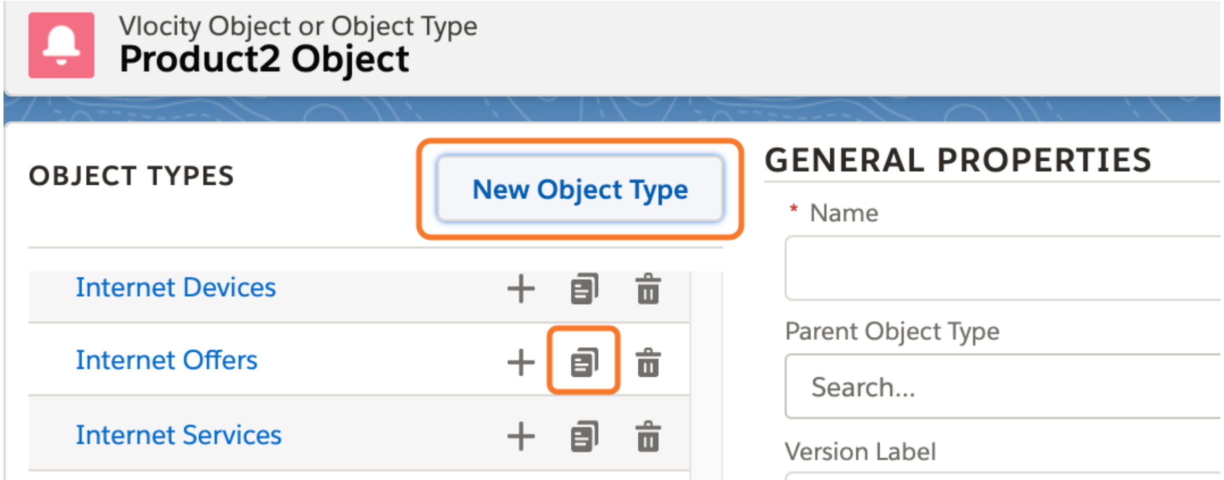
   Vlocity recommends that you make **Product2 Object** the default view. After you select **Product2 Object**, click the pin icon to make it the default view.


Vlocity Product De...
Vlocity Objects and Obje...
Wearables
Product with O...
Vlocity Objects and Object Types
Product2 Object
Printable View
1 item · Sorted by Name · Filtered by all vlocity objects and object types · Record Type, Name · Updated 6 minutes ago
Search this list...
Name
Object API Name
1
Product2 Object
Product2

2. Click Product2 Object.
   The page for Product2 Object opens.
   - The left side of the page shows the hierarchy of all the object types that are linked to the Product2 Object.
   - You can select an object type and click the Details tab to view the general properties and effectivity dates for that object. From the Details tab, you can change settings for an object type, unless other objects link to it.
3. Click **New Object Type**.

TIP

You can also create a duplicate of an existing object and modify its settings. Click the **Clone** icon for a specific object type.



Vlocity Object or Object Type

Product2 Object

4. In the object type form, enter the following information.

Field	Description
Name	Enter the name of the object type.
Parent Object Type	Specify a parent object type when the new object type must inherit common properties. For example, the Smart Phone object has the Color and Storage attributes. The Samsung Phone object must share those attributes. To inherit those attributes, its parent object type is Smart Phone. If you're creating an object subtype, select the parent object type. 1. Click the **Search** button. 2. In the Lookup Search dialog box, enter the name of the parent object, and click **Search**. 3. From the search results, click **Select** for a parent object.
Version Label	This field is reserved for the Versioning functionality (in Beta). Enter a value to indicate a new version, such as v2.0.
Lifecycle Status	This field is reserved for the Versioning functionality (in Beta). Set and track the lifecycle status of the specification. You can only set this field if Specification Type is set to Product or Offer. **CAUTION** Don't use the Released value in Fall '20 because it can have undesirable effects.
Active	Indicate that the object type is in use.

Field	Description
Effective From Effective Until	Enter a date or date range when the object type is available to use.

5. Click **Done**.
   The new object type is saved, and the Layout Management tab is available.

Define a Layout with Fields and Attributes

Use this procedure for new object types (not for the Product2 Object). An object type's layout has sections, which contain attributes and fields that you can sequence. You can modify inherited sections and create ones for child objects.

For each section, you can define the number of columns (one or two) and the movement direction when the tab key is used (up-down or left-right).

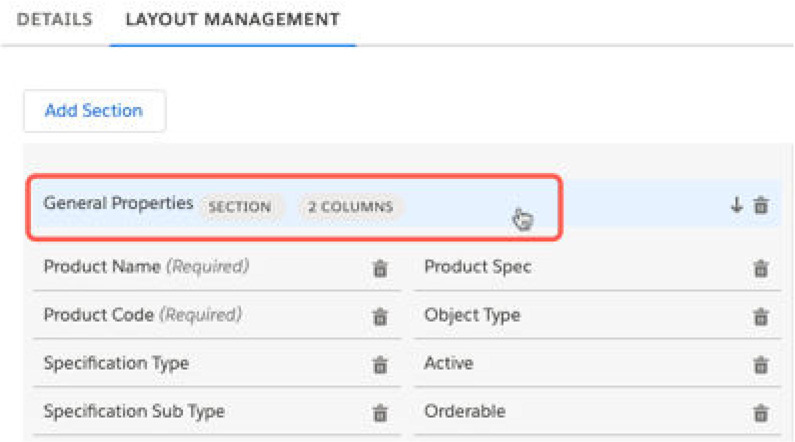
Changes in the Layout Management tab are automatically saved.

Before You Begin
• If needed, assign attributes that you created to the Product2 object so that you can add them to the layout in the following steps. See Define Attributes in the Product Designer.

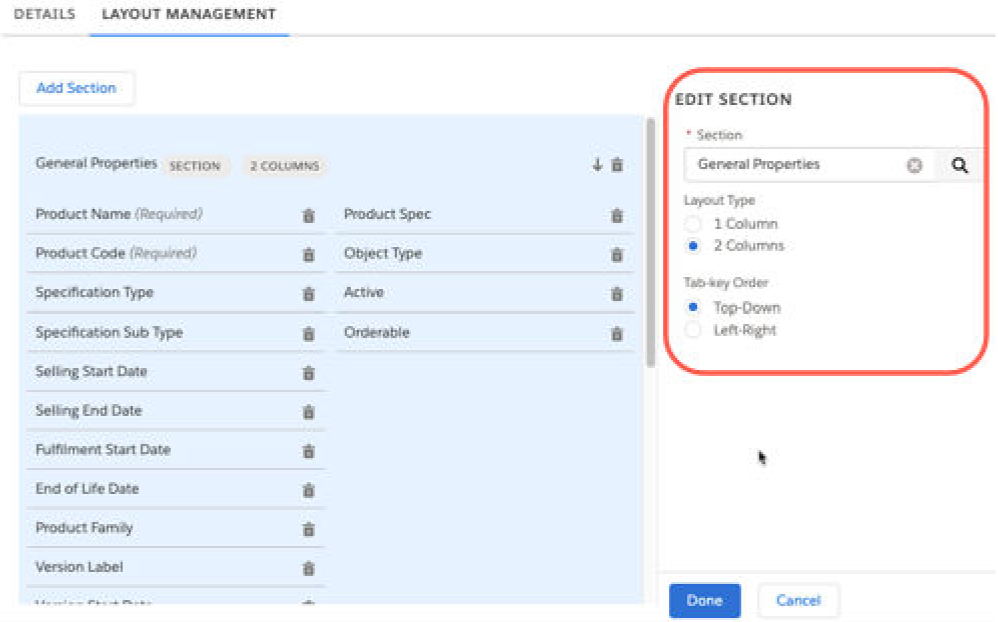
1. Click the Layout Management tab.
   The layout form opens with a panel on the right that has two tabs.

Tab	Description
Attributes	A list of all attributes and their associated Attribute Codes assigned to Product2 Object. If a data element, such as a SKU number, is intended to be common to all products across your product catalog, create a field on your product object instead. Typically, adding new fields requires Salesforce system admin privileges.
Fields	A list of all fields and their associated Field API name for Product2 Object. If the data element is intended to be specific to a product, or class of products, create an attribute, such as download speed, instead. At runtime, attributes are visible to customers in Vlocity Cart and can be used to filter the catalog and provide configuration capabilities. Adding attributes requires access to the Product Console.

2. Drag and drop fields or attributes onto the General Properties section in the appropriate locations.
   The attribute or field that you drop is added to the layout. Child objects also have the sections with the added attributes or fields.
3. To change a section, click the section heading, such as General Properties.



4. From the Edit Section panel, you can modify the section and click **Done**:
   - Search for a section name.
   - Set the Layout Type to use one or two columns in the Details tab of a product specification or offering.
   - Set the tab key order. When the user uses the **Tab** key in the Details form, the tab order is either left to right or top to bottom.



EDIT SECTION

* Section: General Properties

Layout Type
○ 1 Column
● 2 Columns

Tab-key Order
● Top-Down
○ Left-Right

5. To edit a field or attribute, click in the section.
6. From the Edit Attribute panel, modify the attribute characteristics, and click **Done**.
   The available characteristics depend on the type of attribute, such as the following characteristic examples.
   - Required
   - Help text

  The text displays at runtime in the Vlocity Cart if users hover over the information icon (lowercase letter “i”).
- Data type and value
- For attributes with picklists, set **Excluded Value(s)** so that products of this object type don’t show the excluded values.
- For attributes with picklists, ensure that the **Value Data Type** is set to Picklist and the appropriate picklist is linked.
- Behaviors (which define the runtime behaviors in CPQ)

7. To add more sections, click **Add Section**, and repeat steps 2–6.
   - You can clone the Object layout or the Parent Object Type layout if a parent object type is present.
   - To resequence elements in the layout, drag them into the desired locations.
   - Use the delete icons to delete sections and elements.
   - When you define a new product, the New Product page appears at runtime and contains the layout you defined for the product object type at design time. The layout also shows in the Details tab for the product.

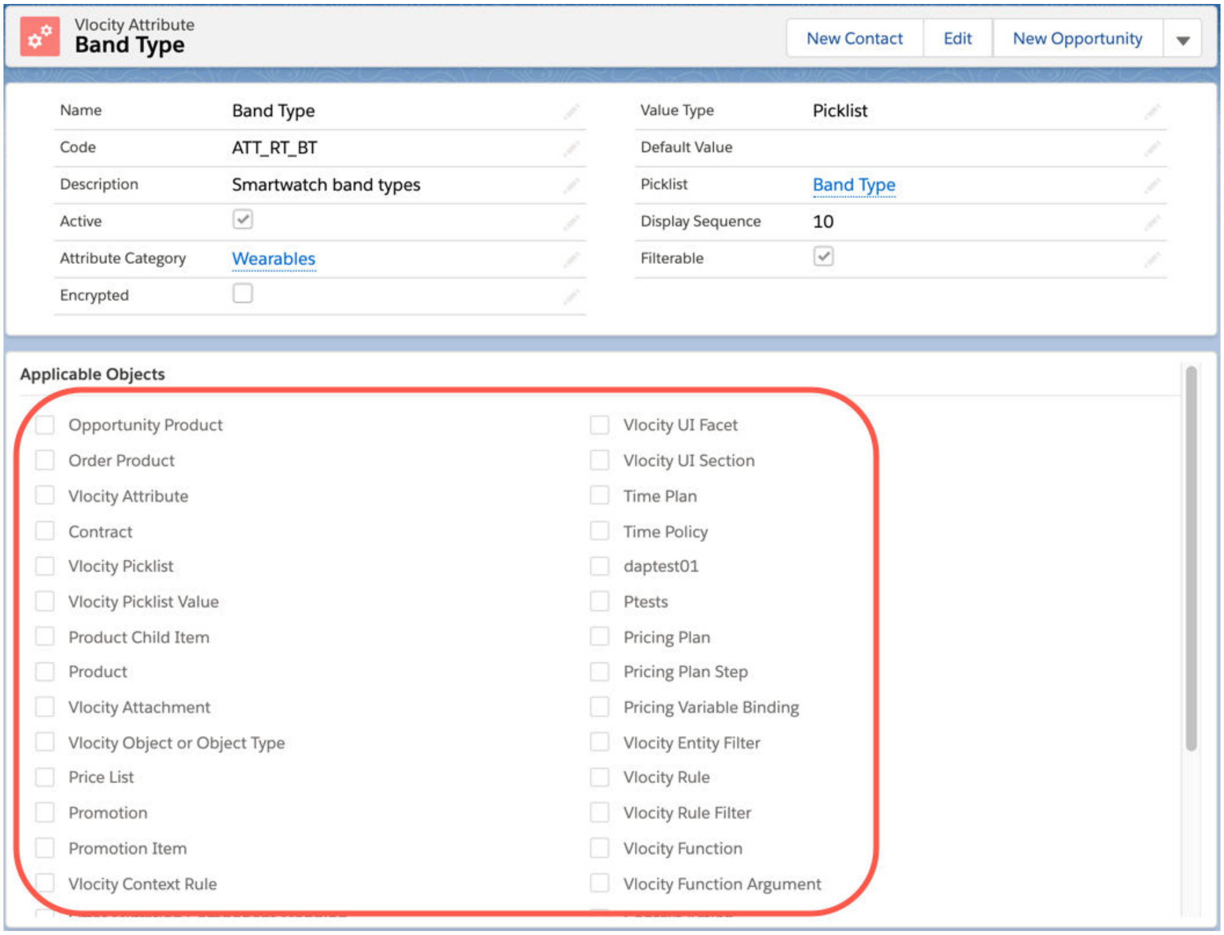
Applying Attributes to an Object

When you create an attribute, you can specify the objects to which the attribute applies.

Use this procedure if you created an object after you defined an attribute or if you didn't assign objects when you created the attribute.

Before You Begin
• Create attributes

1. From the Vlocity Product Designer app, click the ⌄ button, and select **Vlocity Attributes**.
2. Click the name of an attribute.
   The page for that attribute opens.



Vlocity Attribute
Band Type

Name	Band Type	Value Type	Picklist
Code	ATT_RT_BT	Default Value	
Description	Smartwatch band types	Picklist	Band Type
Active	☑	Display Sequence	10
Attribute Category	Wearables	Filterable	☑
Encrypted	☐		

Applicable Objects

- ☐ Opportunity Product
- ☐ Order Product
- ☐ Vlocity Attribute
- ☐ Contract
- ☐ Vlocity Picklist
- ☐ Vlocity Picklist Value
- ☐ Product Child Item
- ☐ Product
- ☐ Vlocity Attachment
- ☐ Vlocity Object or Object Type
- ☐ Price List
- ☐ Promotion
- ☐ Promotion Item
- ☐ Vlocity Context Rule
- ☐ Vlocity UI Facet
- ☐ Vlocity UI Section
- ☐ Time Plan
- ☐ Time Policy
- ☐ daptest01
- ☐ Ptests
- ☐ Pricing Plan
- ☐ Pricing Plan Step
- ☐ Pricing Variable Binding
- ☐ Vlocity Entity Filter
- ☐ Vlocity Rule
- ☐ Vlocity Rule Filter
- ☐ Vlocity Function
- ☐ Vlocity Function Argument

3. In the Applicable Objects section, select the objects to which the attribute applies.
4. Click **Save**.
5. When you add an attribute to an existing object type that is associated with products, run the EPCProductAttribJSONBatchJob job.
   See Running the EPCProductAttribJSONBatchJob.
6. If an associated product has at least one attribute with an override, run the EPCFixCompiledAttributeOverrideBatchJob.
   See Running the EPCFixCompiledAttributeOverrideBatchJob.

What's Next

Create Product Specifications in the Product Designer

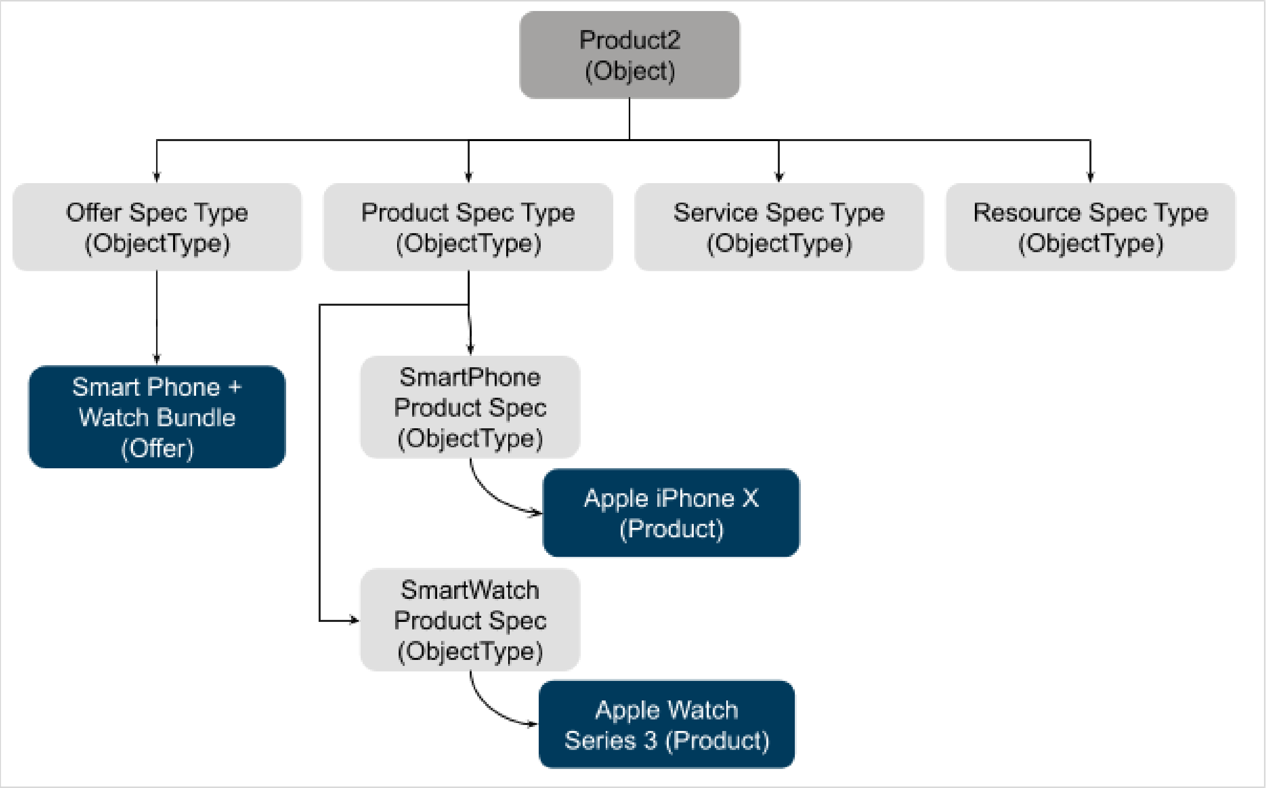
See Also

- Overview of Product Creation in Product Designer
- Fix Duplicate and Empty Attributes

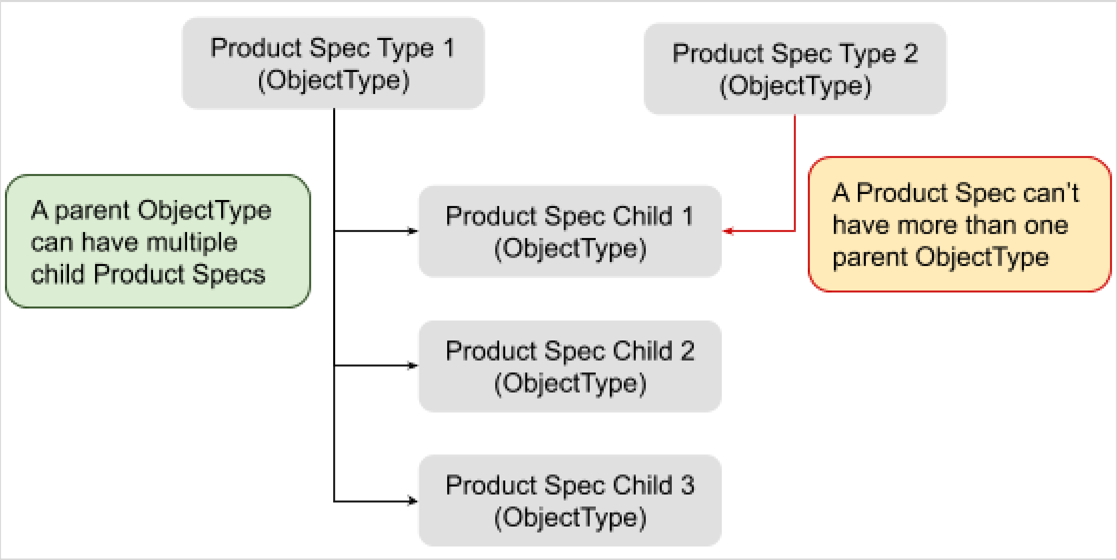
Product Specifications and Characteristics for Product Designer

A product specification is a fundamental entity for which you define parameters (attributes or fields), such as bundling, pricing rules, fulfillment rules, and compatibility rules.

A product specification is an actual product that you plan to sell. Before you can define a product specification, you must have a corresponding object type. A product specification inherits all the attributes of the object type. For example, the iPhone specification has all attributes defined for the smartphone object type (such as brand, finish, and capacity).


Product2 (Object)
Offer Spec Type (ObjectType)
Product Spec Type (ObjectType)
Service Spec Type (ObjectType)
Resource Spec Type (ObjectType)
Smart Phone + Watch Bundle (Offer)
SmartPhone Product Spec (ObjectType)
Apple iPhone X (Product)
SmartWatch Product Spec (ObjectType)
Apple Watch Series 3 (Product)

You can create many product specs from the same object type, but each product specification can only be associated with one object type.


Product Spec Type 1 (ObjectType)
Product Spec Type 2 (ObjectType)
A parent ObjectType can have multiple child Product Specs
Product Spec Child 1 (ObjectType)
A Product Spec can't have more than one parent ObjectType
Product Spec Child 2 (ObjectType)
Product Spec Child 3 (ObjectType)

Types of Specifications

The following specification types are available:

- Product Specification — Specifies the details of a product. An example of a product specification is a iPhone with attributes such as phone model, color, storage, size, and operating system.
- Offer Specification — Specifies a product that is available to clients for purchase and has the pricing information.
- Service Specification — Specifies the details of a Customer Facing Service (CFS). An example of a service specification is an Internet service for which you can define parameters, such as upload or download speed, broadband type (ADSL or Fiber), and so on.
- Resource Specification — Specifies the resources needed to implement the Customer Facing Service (CFS). For example, an IP port address, or a router.

You can use the Product Console to create a new Object Template for defining Product Specifications, Customer facing Services, Resource Facing Specifications and Offer Specifications with attributes, fields and layouts configurations.

See Also

Create Product Specifications in the Product Designer

Create Product Specifications in the Product Designer

A Product specification is a record in the product object. You create product specifications from your object types. Product specifications inherit all fields, attributes, and layouts defined for the associated object type.

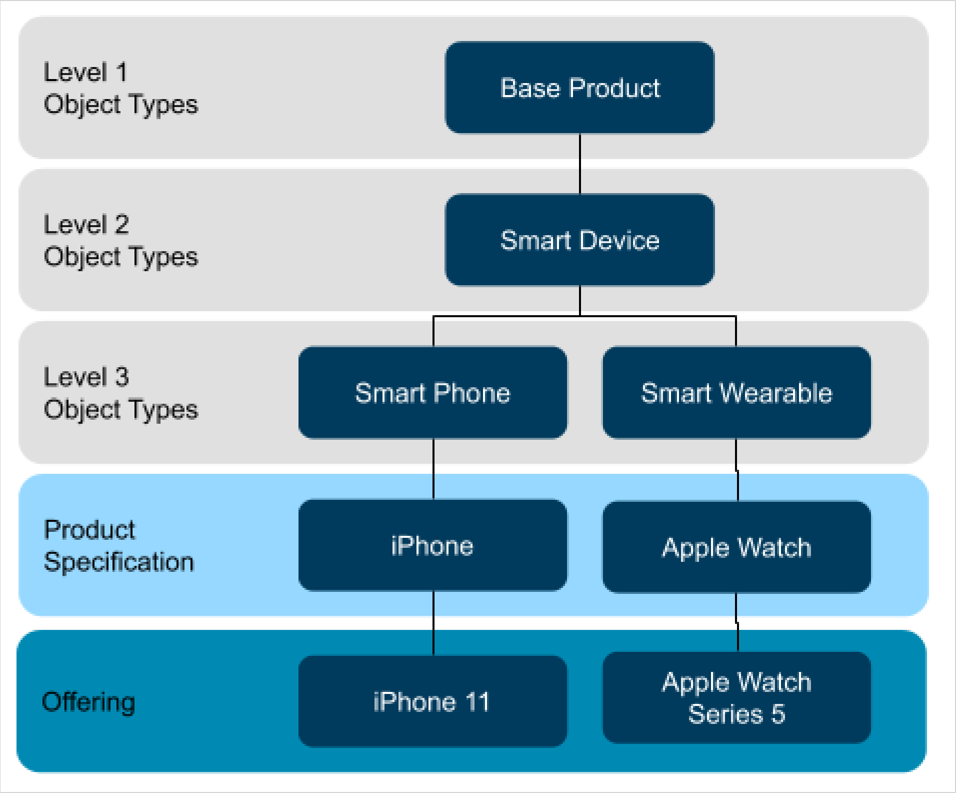
This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Create a Product Specification.

An object type has a one-to-many relationship with a product specification. In other words, you can create many product specifications from the same object type. For example, for the Smart Phone object type, you can create iPhone and Samsung Galaxy product specifications.

After you create a product specification, you specify values for the attributes and fields defined for the object type.

Example 8. Example: iPhone Specification

After Monique creates the Smart Phone and Smart Wearable objects, she creates a product specification for each: an iPhone specification and an Apple Watch specification.



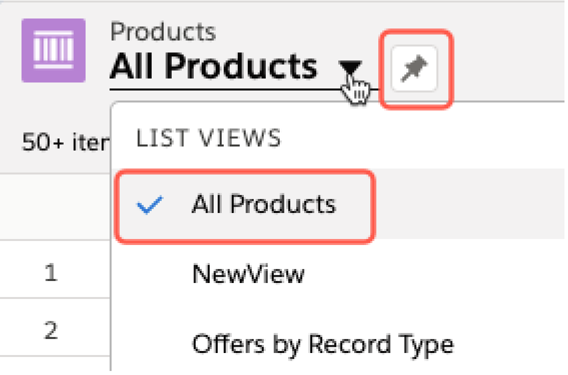
The Smart Device object has a layout with a section with the model, size, storage, color, operating system, and processor attributes. The iPhone product specification inherits its layout with the same fields and attributes.

Before You Begin
• Create the required object types. See Create Object Types and Layouts in the Product Designer. • Review Product Specifications and Characteristics for Product Designer.

1. From the Vlocity Product Designer, click the ⌄ button, and select **Products**.
2. If the view isn't **All Products**, click the menu, and select **All Products** from the list.

TIP

Vlocity recommends that you make **All Products** the default view. After you select **All Products**, click the pin icon to make it the default view.

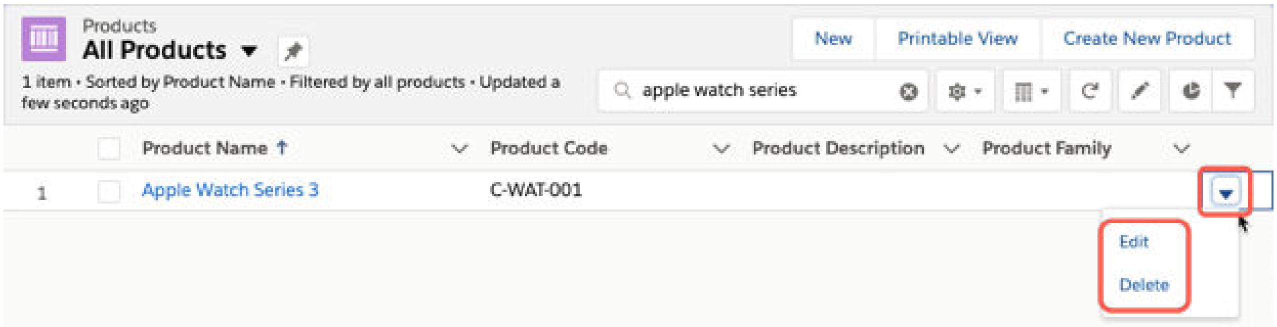


3. Click **Create New Product**.
4. In the New Product: Class page, enter the following information.

Field	Description
Product Name	Enter a name for the product. Example: iPhone
Product Code	Enter a unique code for the product. At runtime in the Vlocity Cart, the Product Details window shows this code. Example: C-DPP-001
Product Description	Enter a description that is shown at runtime in the Product Details window of Vlocity Cart. It's also searchable in the Search dialog.
Product Family	Select a group of products that this specification belongs to. Leave this field blank if this specification doesn't fit into a product family or if you don't use product families. If the Product Family list doesn't show all active values, click **Refresh Tab** from the tab menu for a product.
Specification Type	Select from a specification type or offering.
Product Spec	This field is reserved for the Offer Specification Realization functionality (in Beta). Select a product specification for an offering. You can only select this option if Specification Type is set to Offer.
Object Type	Enter the name of the object type and select from the list. Example: Smart Phone

Field	Description
Specification Sub Type	This field is reserved for the Offer Specification Realization functionality (in Beta). Select one of the following types or leave it blank for backward compatibility. If blank, the product is treated as a bundle and can have child specifications. • Simple — The product doesn't have child specifications and can't be associated with another product specification. It can be a child of a bundle product specification and can refer to an object type. In the Structure view, it can't be set as a **Virtual Item**. • Bundle — The product can have child specifications and refer to an object type. In the Structure view, it can be set as a **Virtual Item** when the specification is a child of a parent specification. But it can't be virtual when the specification is a root product in a bundle.
Help Text	Enter the text that displays at runtime in the Vlocity Cart if users hover over the information icon (lowercase letter "i").
Status	This field is a customer-defined.
Orderable	For product specifications, this field is typically not checked. An offering for the product specification is what is orderable.
Type	This field is customer-defined. Use it to classify products.
Version Label	This field is reserved for the Versioning functionality (in Beta). Enter a value to indicate a new version, such as v2.0. You can only set this field if Specification Type is set to Product or Offer.
Version Start and End Date	This field is reserved for the Versioning functionality (in Beta). Enter a date or date range when the new version is available. The dates for different versions of a specification can't overlap. You can only set this field if Specification Type is set to Product or Offer.
Lifecycle Status	This field is reserved for the Versioning functionality (in Beta). Set and track the lifecycle status of the specification. You can only set this field if Specification Type is set to Product or Offer. ⚠ **CAUTION** Don't use the Released value in Fall '20 because it can have undesirable effects.
Active	Indicate that the specification is in use. Deactivating a product after it's Active and available to sales channels can have multiple impacts, such as product hierarchies, rules, and existing orders. For example, a deactivated product is removed from Vlocity Cart, but the Vlocity Cart price isn't changed.
Selling Start and EndDate	Enter a date or date range when a commercial product can be ordered and sold.
Fulfillment Start Date	Enter a date when a commercial product is ready to be fulfilled. By default, this date is the same as the selling date, but it can be a future date to support pre-orders.
End of Life Date	Enter a date when a commercial product is no longer available to be fulfilled.
Sub Type	This field is customer-defined field. Use it to classify products.

5. Click **Done**.



Results

The new product specification is saved.

To modify or remove an existing product specification, click the arrow on the product row and select **Edit** or **Delete**. You can also delete a product if it's opened in a tab.

What's Next

Create Offerings in the Product Designer

See Also

- Overview of Product Creation in Product Designer

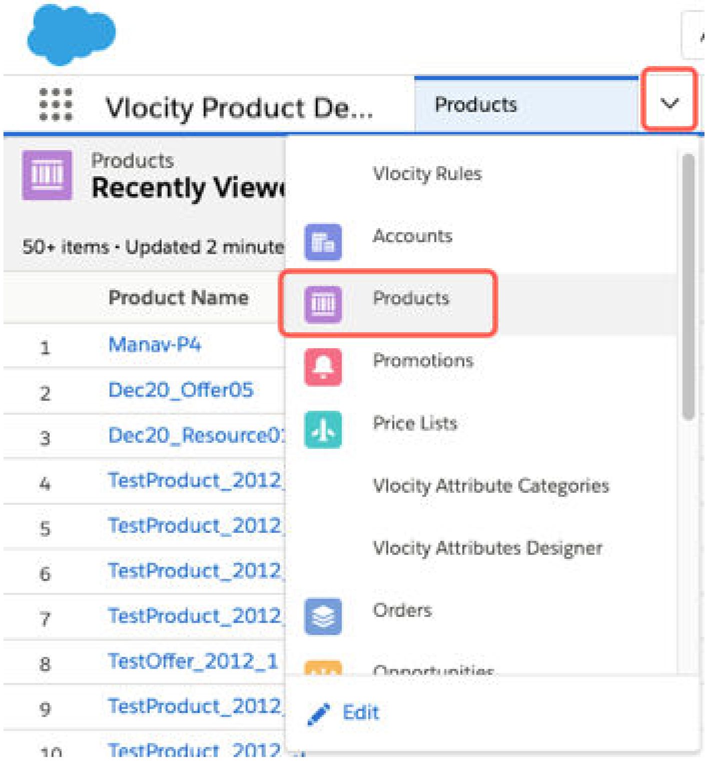
Create Offerings in the Product Designer

After you define product specifications, you can associate them with prices, promotions, and context rules to create offers or commercial products that customers can purchase.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Create a Product Specification.

To create a offering:

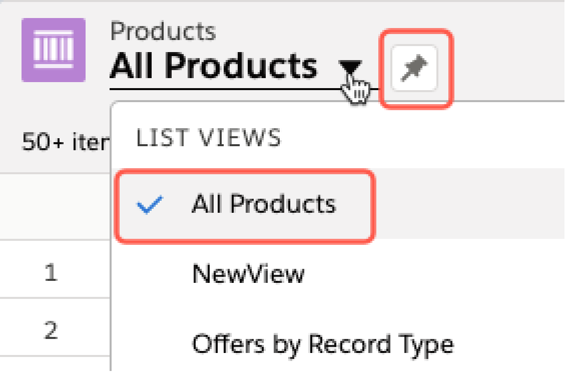
1. From the Vlocity Product Designer app, click the ⌄ button, and select **Products**.



2. If the view isn't **All Products**, click the menu, and select **All Products** from the list.

TIP

Vlocity recommends that you make **All Products** the default view. After you select **All Products**, click the pin icon to make it the default view.



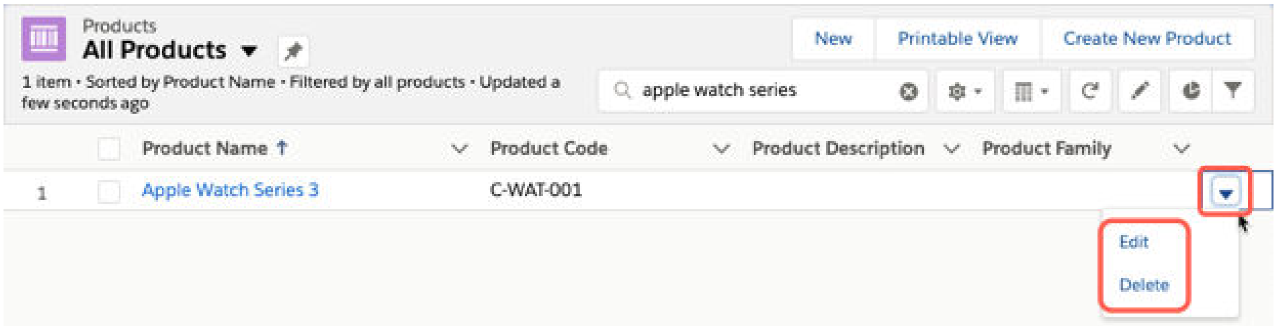
3. Click **Create New Product**.
4. In the New Product: Product page, enter the following information.

Field	Description
Product Name	Enter a name for the product. Product names appear to customers in the product catalog. Example: iPhone 6S Plus
Product Code	A unique code for the product. Example: VLO-MOB-0003
Product Description	Enter a description that is shown at runtime in the Product Details window of Vlocity Cart. It's also searchable in the Search dialog.
Product Family	Select a group of products that this specification belongs to. Leave this field blank if this offer doesn't fit into a product family or if you don't use product families. If the Product Family list doesn’t show all active values, click **Refresh Tab** from the tab menu for a product.
Specification Type	Select **Offer**.
Product Spec	This field is reserved for the Offer Specification Realization functionality (in Beta). Select a product specification for an offering. You can only select this option if Specification Type is set to Offer.
Object Type	Select the object type you have previously created. Example: Smart Phone type
Specification Sub Type	This field is reserved for the Offer Specification Realization functionality (in Beta). Select one of the following types or leave it blank for backward compatibility. If blank, the product is treated as a bundle and can have child specifications. • Simple — The product doesn’t have child specifications and can’t be associated with another product specification. It can be a child of a bundle product specification and can refer to an object type. In the Structure view, it can’t be set as a **Virtual Item**. • Bundle — The product can have child specifications and refer to an object type. In the Structure view, it can be set as a **Virtual Item** when the specification is a child of a parent specification. But it can’t be virtual when the specification is a root product in a bundle.
Help Text	Enter the text that displays at runtime in the Vlocity Cart if users hover over the information icon (lowercase letter “i”).
Status	This field is a customer-defined.
Orderable	Enable the offering to be ordered. This setting is typically enabled for offerings and not the product specifications that the offerings are based on.
Type	This field is customer-defined. Use it to classify products.
Version Label	This field is reserved for the Versioning functionality (in Beta). Enter a value to indicate a new version, such as v2.0. You can only set this field if Specification Type is set to Product or Offer.
Version Start and End Date	This field is reserved for the Versioning functionality (in Beta). Enter a date or date range when the new version is available. The dates for different versions of a specification can’t overlap. You can only set this field if Specification Type is set to Product or Offer.

Field	Description
Lifecycle Status	This field is reserved for the Versioning functionality (in Beta). Set and track the lifecycle status of the specification. You can only set this field if Specification Type is set to Product or Offer. ⚠ **CAUTION** Don’t use the Released value in Fall '20 because it can have undesirable effects.
Active	Indicate that the offering is active and in use. Deactivating a product after it's Active and available to sales channels can have multiple impacts, such as product hierarchies, rules, and existing orders. For example, a deactivated product is removed from Vlocity Cart, but the Vlocity Cart price isn’t changed.
Selling Start and End Date	Enter a date or date range when a commercial product can be ordered and sold.
Fulfillment Start Date	Enter a date when a commercial product is ready to be fulfilled. By default, this date is the same as the selling date, but it can be a future date to support pre-orders.
End of Life Date	Enter a date when a commercial product is no longer available to be fulfilled.

5. Click **Done**.

To modify or remove an existing offering, click the arrow on the product row and select **Edit** or **Delete**.



1 item • Sorted by Product Name • Filtered by all products • Updated a few seconds ago

apple watch series

	Product Name	Product Code	Product Description	Product Family
1	Apple Watch Series 3	C-WAT-001		

Results

The offering is saved, and you can add attachments, set attribute rules, or add context rules.

What's Next

Create Product Bundles in the Product Designer

See Also

- Overview of Product Creation in Product Designer
- Export a Product Report in Product Designer
- Fix Duplicate and Empty Attributes

Configure Attribute Rules for Offerings in the Product Designer

You can display attributes and attribute values that correspond to specific offering (or product specification) configurations. Product compatibility rules support attribute configuration. Configuration evaluations are server-side and actions are client-side. Changing an attribute triggers attribute configurations.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Configure Attribute Rules.

You must explicitly define all required behavior. After an action is applied, only another configuration can change it. For example, you can define a product configuration where changing the color attribute affects the memory attribute choices.

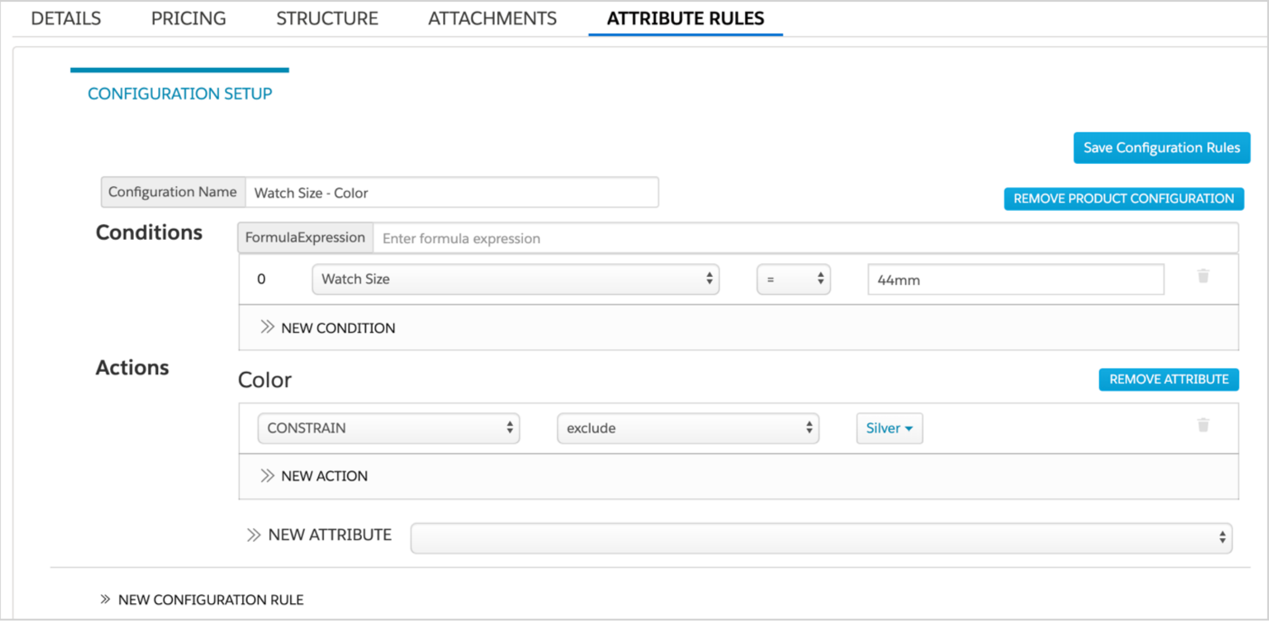
Each configuration requires two sections—the CONDITIONS section and the ACTIONS section.

- Use the CONDITIONS section to create the conditions that trigger the configuration.
- Use the ACTIONS section to define the actions.

Example 9. Example:

For the Apple Watch Series 5, Monique needs to remove silver as a color option if the watch is 44mm. For the configuration rule, she defines a condition based on the Watch Size attribute that has a value of 44mm. If that condition is true, she creates an action for the Color attribute that excludes the silver option.

She creates the following attribute rule for the configuration.



Configuration Name: Watch Size - Color

Conditions

FormulaExpression: Enter formula expression

0 | Watch Size | = | 44mm

Actions

Color

CONSTRAIN | exclude | Silver

The configuration setup is similar to setting up rules, which include, from left to right:

- index number (for each condition)
- attribute
- operator
- value

In the example above, 0 is the index number, watch size is the attribute, the equal sign (=) is the operator, and 44mm is the value.

Before You Begin

- Create attributes.
- Add required attributes to the layout of the object that the offering is based on.
- Create an offering.

1. From the Vlocity Product Designer, click the ⌄ button, and select **Products**.
2. From the Products list, click an offering or product specification.
   The product opens in a new tab.
3. In the Attribute Rules tab, enter a name for the rule in the **Configuration Name** field.
4. In the Conditions section, click New Condition and define a condition.
   a. Select an attribute.
   b. Select an operator.
   c. Enter a value.
5. To add more conditions, repeat step 4.
   All of the defined conditions in a configuration rule must be true so that the action occurs.
6. If an action should occur when any one of several conditions is met, enter the appropriate expression in the Formula Expression box.
   By default, all conditions must be true for the action to occur unless you specify a formula expression. You can specify that only one of the conditions must be true.
   A formula expression defines how multiple conditions are joined. Use the AND or OR operator with the index numbers of the conditions. For example, if you've created two conditions, the action can occur if only one condition is true with the following formula:
   `{index 0} OR {index 1}`.
   You can use bracket { } to group conditions. For example, the following example has three conditions. But either the first and second conditions must be true or the third condition must be true.
   `{{index 0} AND {index 1}} OR {index 2}`
7. From the New Attribute list, select an attribute to modify.
8. Select an action.
   - **Constrain** includes or excludes specific picklist choices. It is only applicable for attributes in picklists.
   - **Assign** defines a value to an attribute. For example, the Color attribute is assigned a value of White.
   - **Disable** makes an attribute that is visible but cannot be modified.
   - **Require** indicates that an attribute must be defined, such as the size of a smart watch.
   - **Hide** any attribute in the user interface. The attribute definition is in the JSON but is hidden from the user.
9. To add more actions for the selected attribute, click **New Action** and repeat step 8.
10. To modify other attributes for the same conditions, click **New Attribute** and repeat steps 7 through 9.

11. To create another rule with new conditions and actions, click **New Configuration Rule**, and repeat steps 3 through 10.
12. When you are finished, click **Save Configuration Rules**.

What's Next

Create Product Bundles in the Product Designer

See Also

- Add Attachments in the Product Designer
- Overview of Product Creation in Product Designer

Add Attachments in the Product Designer

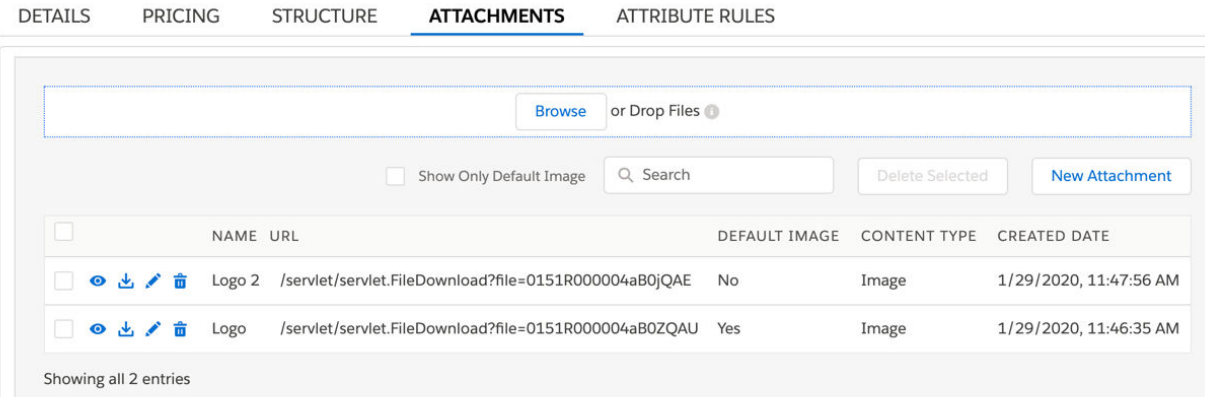
You can upload files for an offering or specification, including images and documents. On the Attachments tab, you can indicate the product image to show across different channels. Previews of images and videos are shown in the Details tab for a product.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Add an Image to a Product.

In the Attachments tab, you can also use the icons to preview, download, edit, or delete attachments. When previewing attachments such as images, you can use the arrows to scroll through the list of attachments.

To define an attachment for the product:

1. From the Vlocity Product Designer, click the ⌄ button, and select **Products**.
2. Click on a product specification or offering to edit it.
3. Click the Attachments tab.
   The Product Attachments page for your selected product appears.



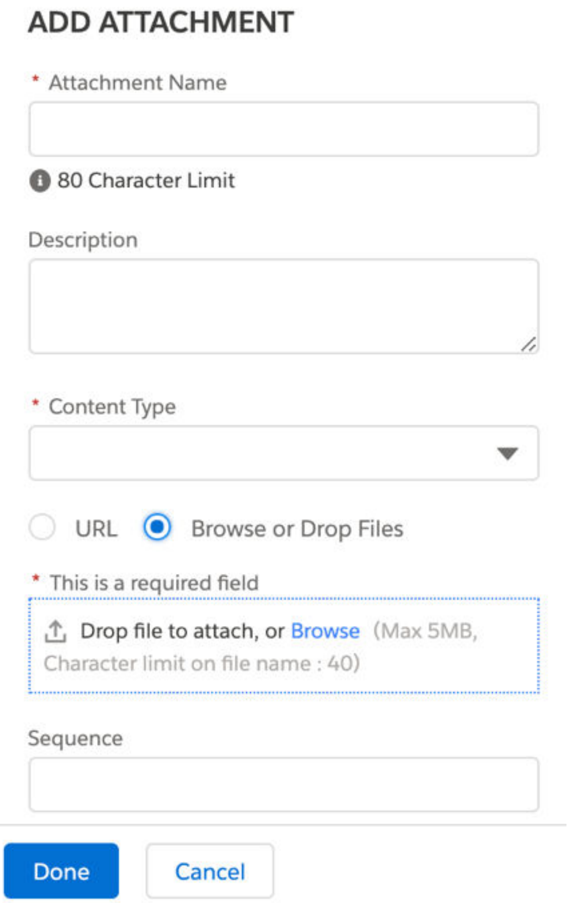
□	NAME	URL	DEFAULT IMAGE	CONTENT TYPE	CREATED DATE
□	Logo 2	/servlet/servlet.FileDownload?file=0151R000004aB0jQAE	No	Image	1/29/2020, 11:47:56 AM
□	Logo	/servlet/servlet.FileDownload?file=0151R000004aB0ZQAU	Yes	Image	1/29/2020, 11:46:35 AM

Showing all 2 entries

4. Click **New Attachment**.

TIP
In the Attachments tab, ou can also use the Browse to select a file or drop files on Drop Files area to upload the files directly.

The Add Attachment panel opens.



ADD ATTACHMENT

* Attachment Name

80 Character Limit

Description

* Content Type

○ URL ● Browse or Drop Files

* This is a required field

Drop file to attach, or Browse (Max 5MB, Character limit on file name : 40)

Sequence

5. In the Add Attachment panel, enter the following information.

Field	Description
Attachment Name	Enter a name to show in the list of attachments.
Description	Enter a description of the attachment.
Content Type	Set the type of attachment: Video, image, or file.
Default Image	Set the image as the default to display for the product. If another image is the default, it is reset as not the default. This setting is not available for files or videos.
URL	Add an image for the product as a URL or add the image
Browse or Drop Files	Browse to the file you want to attach, or drag and drop the file to the drop zone.

Field	Description
Sequence	Enter a unique number to indicate the sequence in which the file, video, or image is displayed in the list of attachments Use a sequence of 10, 20, 30 so that you can add to the sequence at a later time.

6. Click **Done**.

Results

The uploaded file is listed in the Attachments tab. You might have to refresh the preview in the Details tab to see a newly uploaded file.

In the Safari browser, the video preview does not work in the Details tab for a product or offering.

Apply Context Rules to an Offering in the Product Designer

With a qualification rule, you can control whether an offering appears in the Qualified or Disqualified tab in the Vlocity Cart. A qualification rule determines a customer's eligibility to receive a product or promotion before it is added to Vlocity Cart.

You can assign to a product specification or offering.

Example 10. Example:

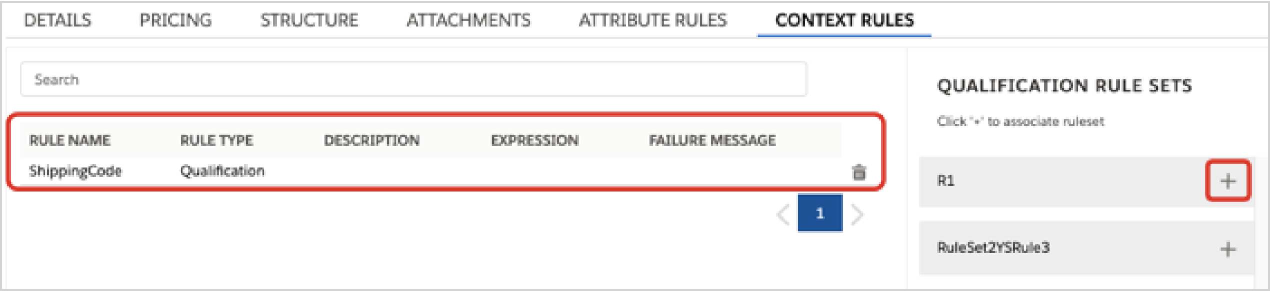
A 5G network plan is available for customers in select locations. Monique creates and applies a qualification rule to the 5G Ultra Plan offering so that only clients in those locations can get the 5G Ultra Plan.

Before You Begin
• Create a qualification rule set and the required scope, dimension, and rule in the Vlocity Product Console. • Create an offering. • Create a product specification.

1. From the Vlocity Product Designer, click the ⌄ button, and select **Products**.
2. From the Products list, click an offering or product specification.
   The product opens in a new tab.
3. In the Context Rules tab, click the + icon on a rule set in the Qualification Rule Sets panel to add it to the offering.
   All qualification rules are shown. Other types of context rules are not listed.
   The context rule is assigned to the offering and shows in the Context Rules list.
4. To remove a rule set from an offering, click the 🗑 icon for an assigned rule.

Results

The context rule is assigned to the offering and shows in the Context Rules list.



RULE NAME	RULE TYPE	DESCRIPTION	EXPRESSION	FAILURE MESSAGE
ShippingCode	Qualification			

Offer Bundles, Cardinalities, and Relationships for Product Designer

A product bundle is a set of products sold together.

Products are bundled using a “Has-A” Containment relationship as either a child or a realization. A product can have one or more product child items (or PCIs), and you can define cardinality for each. PCIs can be a single product or a product bundle, which allows you to create product bundles within product bundles.

Creating a product bundle involves the following:

- Creating a product bundle.
- Overriding the cardinality of the child product bundle.
- Creating a price list entry for a product bundle.
- Refreshing the product hierarchy cache.

Types of Product Line Items in a Bundle

There are two key types of product line items in a bundle hierarchy:

- Root product line item is the top-level line item. Root does not have a parent, so the ParentItemId is always empty.
- Product Child Item (PCI) defines the relationship between a parent product and a child product.

Types of Cardinality in Bundles

Product cardinality defines the default, minimum and maximum quantities of the product that are allowed in the product bundle. You can also override the cardinality of product bundles when they are included within another product bundle.

As a product manager, you can define two types of cardinality:

PCI cardinality quantities of the product allowed in the product bundle	**Min Quantity:** The sum of the quantities of the instances of a PCI cannot be less than the value defined in Min Quantity. **Max Quantity:** The sum of the quantities of the instances of a PCI cannot be greater than the value defined in the Max Quantity. **Default Quantity:** The value of the Quantity field when a PCI is added as a Line Item.

Group cardinality minimum and maximum allowable quantities across an item’s immediate children	**Min Children Quantity:** The sum of the quantities of an item's children cannot be less than the value defined in Min Children Quantity. **Max Children Quantity:** The sum of the quantities of an item's children cannot be more than the value defined in Max Children Quantity.

Cardinality and Inheritance

You can set cardinalities for a product bundle:

- When an offering has a structure, the root offering has Group Cardinality with a maximum and minimum number of children. You can create and delete cardinality overrides.
- You can add a simple or bundled product to the root product as a child. When doing so, the product captures local cardinalities, which you can modify. You cannot override cardinalities in this case because the base cardinalities are not shared across root products.
- When inheriting a simple or bundled product in the context of a root as grandchild at any level, local cardinality applies to both simple and bundled products as a grandchild. Base cardinality is inherited by default. You can override, create, and delete the base cardinalities if the changes are specific to the current root product. If modified, the changes reflect on all the root products that contain the grandchild products.
- Adding or inserting a simple or bundled new product in a nested bundle can be done in the context of a root as a grandchild at any level. Local cardinality applies to both simple and bundled products as a grandchild. You can override, create, and delete local cardinalities. You can modify base cardinalities. If modified, the changes are reflected in all the root products with the grand-child products.
- Group cardinality applies to a bundled product as a grandchild only. Group Cardinalities must be inherited by default. You can override, create, and delete group cardinalities if the change is specific to the current root product. You can modify base group cardinalities. If modified, then the changes reflect on all the root products.

See Also

Create Product Bundles in the Product Designer

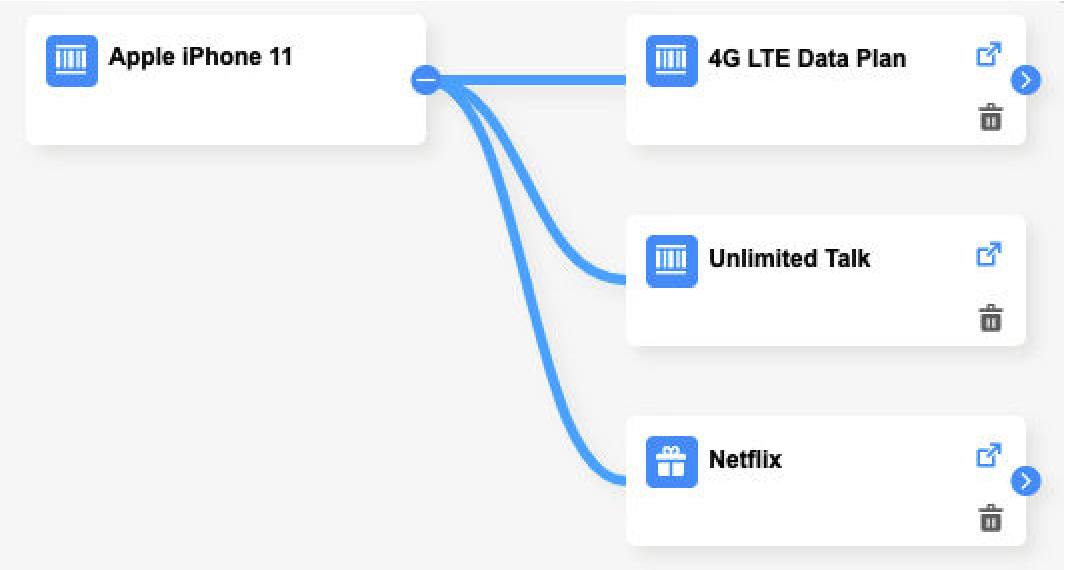
Create Product Bundles in the Product Designer

Creating a product bundle as an offer involves creating the structure of the bundle, including the main product and the product child items that comprise the bundle offer.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Create a Product Bundle.

Example 11. Example: New Product Bundle

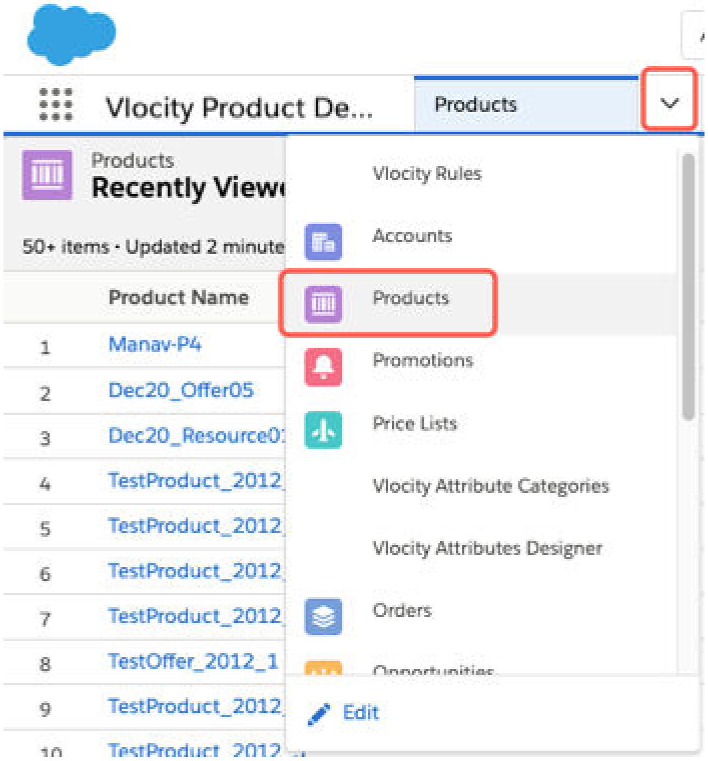
Monique creates a bundle that is available to new customers. She creates a structure for an Apple iPhone 11 offer with three child products: 4G LTE data plan, unlimited talk and text, and Netflix.



Before You Begin
• Add prices to the offers. Only offers with prices are listed in the Structure tab. • Review Offer Bundles, Cardinalities, and Relationships for Product Designer.

Creating a Bundle

1. From the Vlocity Product Designer app, click the ⌄ button, and select **Products**.



2. In the products list, click the product that will be bundled.
   The product opens on a new page.
3. From the product page, click the Structure tab.
4. Drag and drop products from the Quick Add panel onto the layout to create relationships. Only products that have prices are shown in the Quick Add panel.

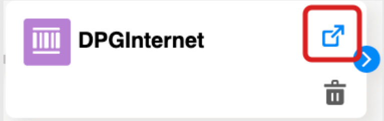
NOTE
If you have more than 200,000 PCIs records in your org, contact Salesforce to index the `Product2.GlobalGroupKey__c` field for your org.

After you drop a product into a parent node, it becomes a child of that node. You can modify product relationships.

- Update the relationship properties in the Edit Relationship form.
  After you drop a product, bundle, or service on the structure tree, the Edit Relationship form appears on the right side. Also, you can click the white space on a box to see the Edit Relationship form.
- Change the order of the nodes by dragging and dropping them into the desired order.

TIP

To open a child product in a new tab, click the **Open in a new tab** icon.



5. In the Edit Relationship form, configure the following settings.

Setting	Description
Min Quantity	Set the minimum quantity required for a valid order.
Default Quantity	Set the default number of child products that are included in the bundle when it is added to the Vlocity Cart.
Max Quantity	Set the maximum quantity required for a valid order.
Group Cardinality	If the root product has children, you can set a Min Quantity and Max Quantity.
Virtual Item	When checked, makes the product item not assetizable. This is customarily used as a grouping mechanism and to set cardinality for the parent product nodes.
Collapse Hierarchy	When checked, enables a Search Item dialog in the cart's line items pane allowing you to search items within the bundle and then add additional products to the bundle. This is intended for use with very large product bundles. For example, at run time, when you type in the name of the mobile handset that you are looking for in the search item dialog box, if the **Collapse Hierarchy** flag is set to True, the product bundle displays all the mobile handset items in a drop-down list as suggestions for you. This is useful as the Mobile handset product bundle may be large and can have several items.

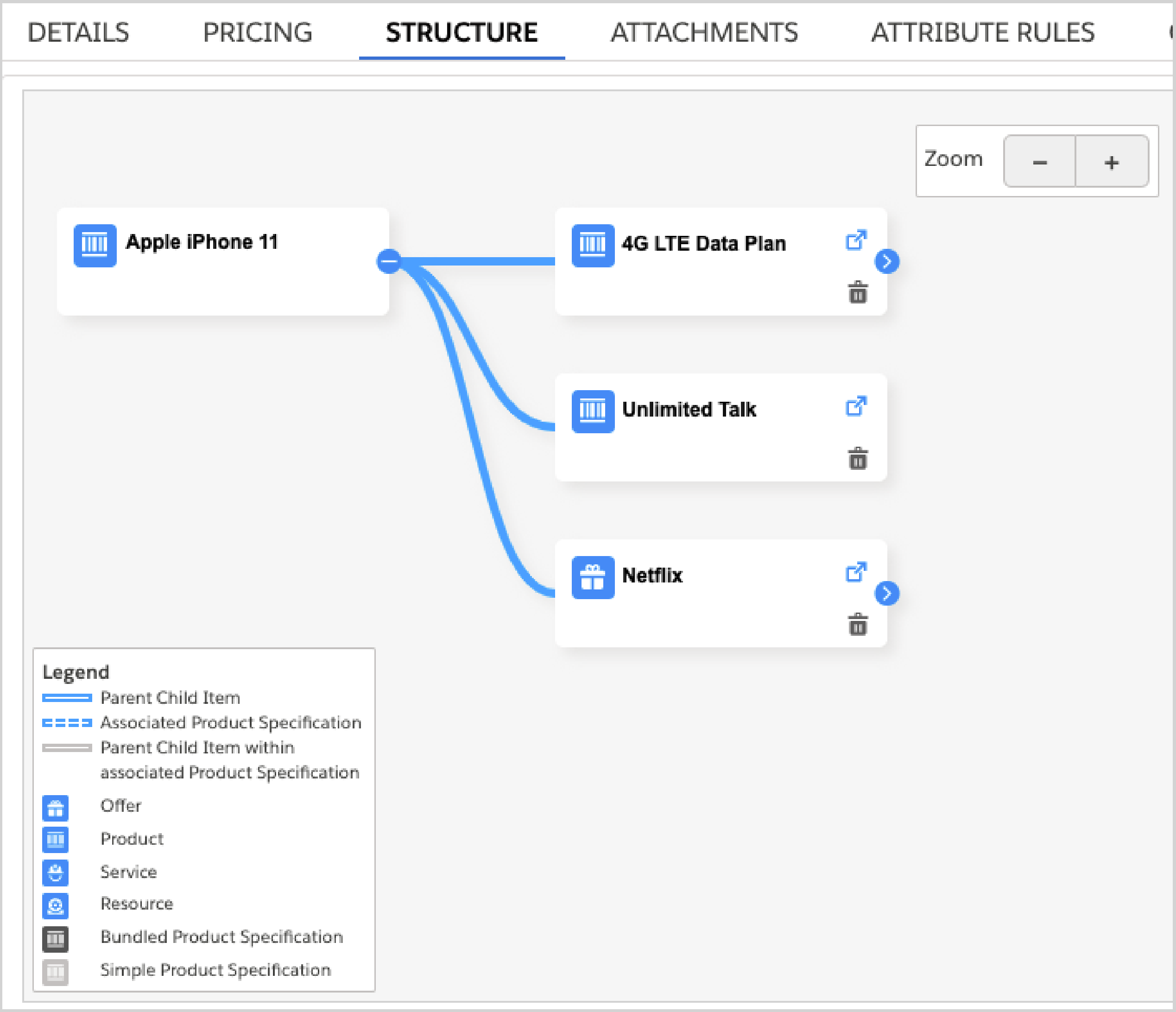
6. Click **Save**.

Results

A new or modified product bundle is saved.

With Fall '20 and later, the Structure view shows icons so that you can more easily differentiate products and their relationships without having to open each to view the details.

- Specification types (product, offers, service, and resource)
- Specification Sub Type (simple or complex product bundles)
- Relationship types between products:
  - Parent-child relationship
  - Associated product specification
  - Parent-child relationship within an associated product specification



You can delete a relationship from the product structure one node at a time. The deletion of a node is only prevented if it violates referential integrity. You can rearrange child products by dragging and dropping them.

Overriding the Cardinality of the Child Product Bundle

You can override product cardinality in a bundle. Cardinality overrides must be between the minimum and maximum limits set by the original cardinality.

For example, if the minimum original quantity is 2, you cannot create a minimum override quantity of 1; the override must be greater than or equal to 2.

The product child item window appears on the right. Notice the existing CARDINALITY settings that were set when a product child item was added to the main product bundle.

1. Click **Create Override**.
2. In the override value fields, change the values for **Min Quantity** and **Max Quantity**.
3. Click **Done**.

What's Next

Define Pricing for Offers in the Product Designer

See Also

- Overview of Product Creation in Product Designer

Override Product Attributes in Product Designer

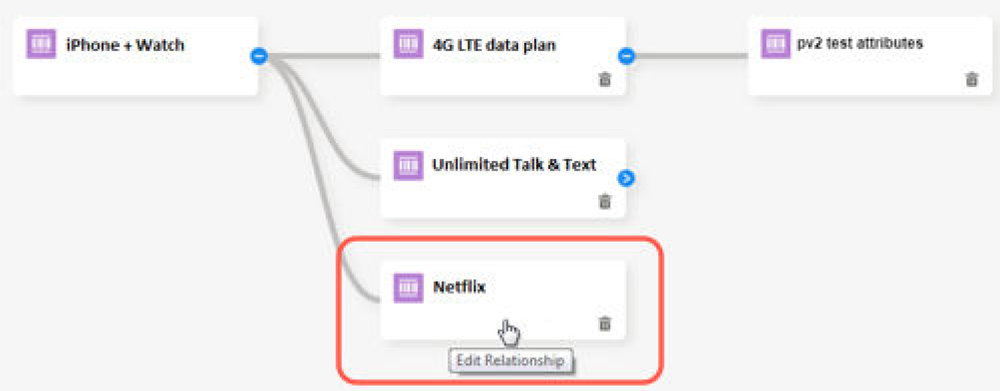
After you add a product as a child to a product bundle, you can override the existing attributes of the child product.

This information is for Vlocity Product Designer. For Vlocity Product Console, see Override Product Attributes.

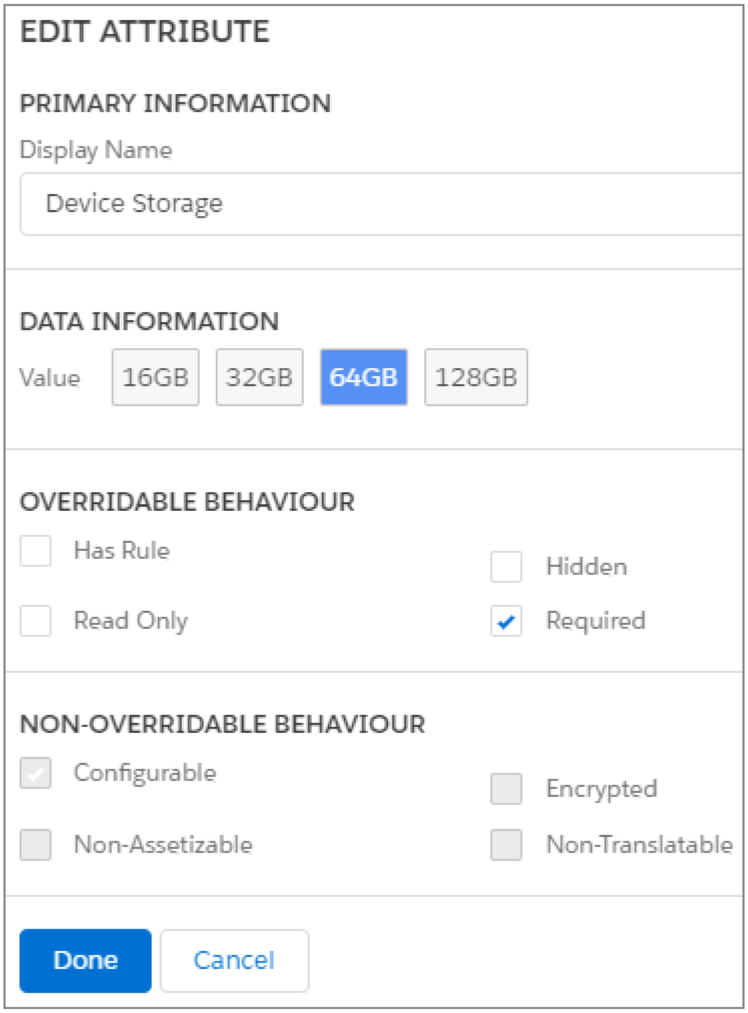
If you see empty or duplicate attributes, see Fix Duplicate and Empty Attributes.

If you don't see an attribute to override and the attribute is available on the product, run these jobs: Running the EPCProductAttribJSONBatchJob and Running the EPCFixCompiledAttributeOverrideBatchJob.

1. In the Vlocity Product Designer app, open a product or offering.
2. In the Structure tab of a product or offering, click a box.



3. In the Edit Relationship panel, expand the Attributes sections.
4. For an attribute, click **Create Override**.
   A success message indicates that the override was created.
5. Click the Edit icon for the overridden attribute.
   The Edit Attribute panel appears.



EDIT ATTRIBUTE

PRIMARY INFORMATION

Display Name: Device Storage

DATA INFORMATION

Value: 16GB | 32GB | [✓ 64GB] | 128GB

OVERRIDABLE BEHAVIOUR

□ Has Rule
□ Hidden
□ Read Only
☑ Required

NON-OVERRIDABLE BEHAVIOUR

☑ Configurable
□ Encrypted
□ Non-Assetizable
□ Non-Translatable

6. Change values for any of the following:
   - Primary information
   - Data information
   - Overridable behavior
   - Non-overridable behavior
7. In the Edit Attribute panel, click **Done**.
8. To remove an override, click the trash icon in the Actions column for the attribute.
9. In the Edit Relationship panel, click **Done**.

Create Promotions and Offers for Product Bundles in the Product Designer

You can select a child product in the structure and modify an existing promotion, add a promotion, or create an offer-level adjustment or override. From a child product, you can define contextual adjustments or overrides for bundled products.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Add a Product to a Promotion.

When you define a promotion, adjustment, or override from the **Structure** tab, it's defined within the context of the bundle for the parent product.

If you create an adjustment with a promotion in the Vlocity Product Designerand delete the promotion from the Vlocity Product Console, the promotion is deleted but not the adjustment. Instead of deleting, mark the promotion inactive if it's not required.

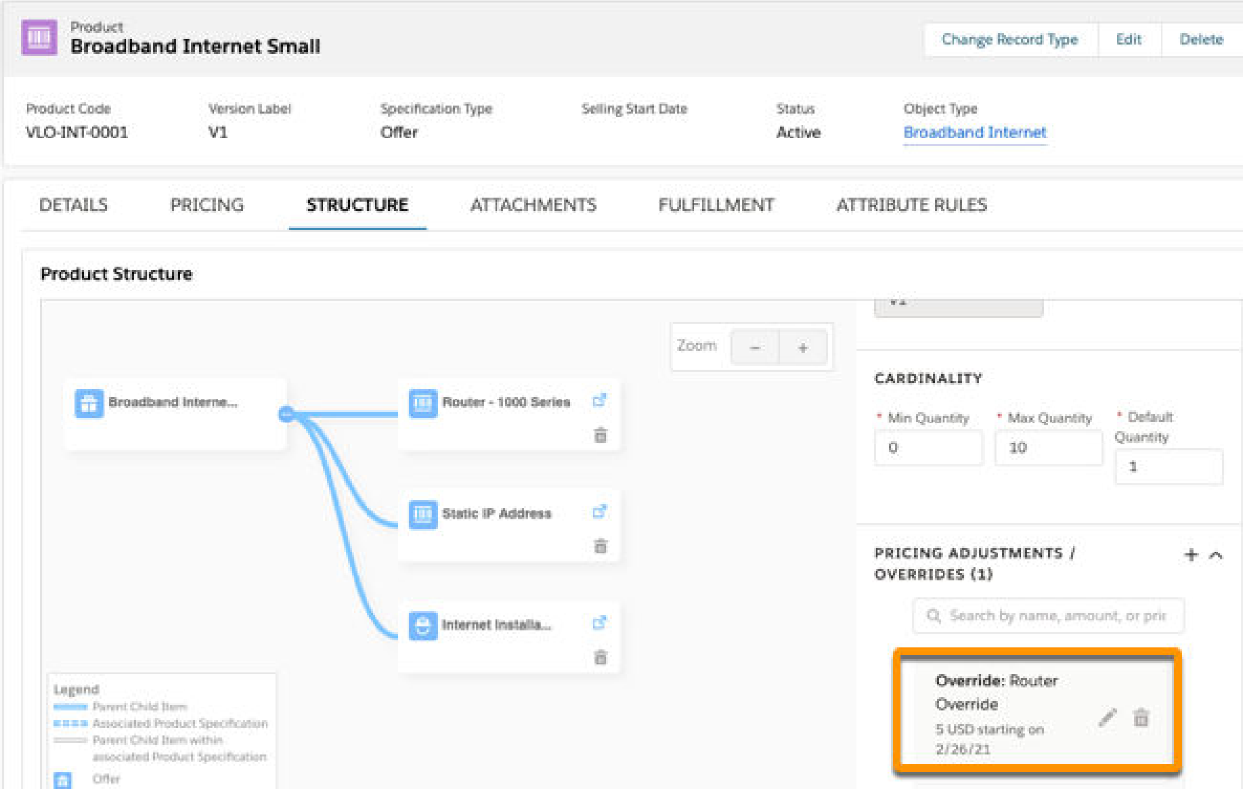
Specifying Overrides

Using the Override feature, you can create:

- A new promotion with an override
- An override on an existing promotion
- An offer-based override

You can see a newly created override in two places for a product:

- In the Structure tab of the parent product, select the child product. The PRICING ADJUSTMENTS / OVERRIDES panel shows the overrides.



Product
Broadband Internet Small

Product Code: VLO-INT-0001
Version Label: V1
Specification Type: Offer
Selling Start Date:
Status: Active
Object Type: Broadband Internet

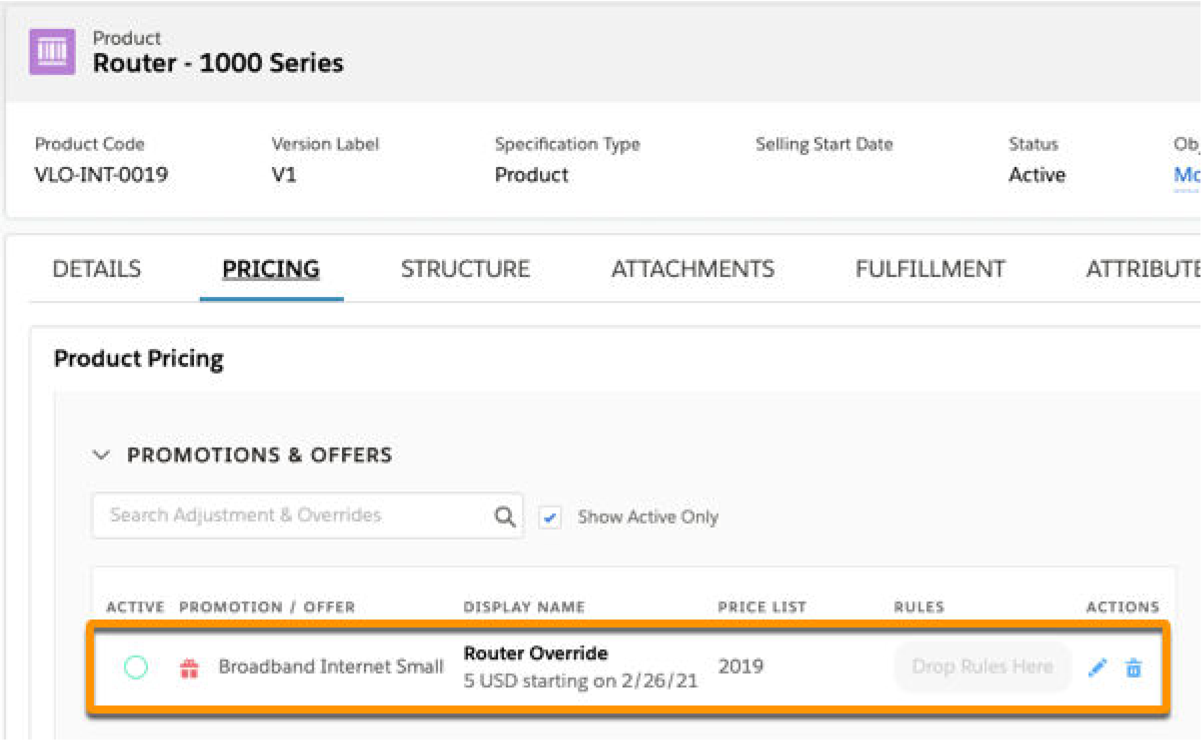
CARDINALITY

* Min Quantity: 0
* Max Quantity: 10
* Default Quantity: 1

PRICING ADJUSTMENTS / OVERRIDES (1)

Override: Router Override
5 USD starting on 2/26/21

- In the Pricing tab of the child product, the Promotions section shows the overrides and the applicable promotion.



Product
Router - 1000 Series

Product Code	Version Label	Specification Type	Selling Start Date	Status	Ob
VLO-INT-0019	V1	Product		Active	Mc

Product Pricing

PROMOTIONS & OFFERS

Search Adjustment & Overrides

☑ Show Active Only

ACTIVE	PROMOTION / OFFER	DISPLAY NAME	PRICE LIST	RULES	ACTIONS
○	Broadband Internet Small	Router Override 5 USD starting on 2/26/21	2019	Drop Rules Here	

Creating a New Promotion with an Override

1. In the Structure tab, click a product or child product.
2. In the Edit Relationship panel on the right, click the down arrow to expand the Pricing Adjustments/ Overrides section.
3. To add an adjustment click the **+** icon.
4. From the New Adjustment or Override dialog box, select **Override on a new Promotion**.
5. In the Add Override dialog box, enter the following information.

Field	Description
Price List	Select a price list.
Promotion Name	Enter a name for the new promotion.
Display Name	Enter the name of the override. This name is shown in Vlocity Cart.
Description	Enter a description that can appear in the Promotions list of the cart. As best practice, include the details of the price changes and durations for each product.
Promo Code	Enter a unique code to identify the promotion from other promotions.

6. Click **Create Promotion**.
   The promotion is saved in the product catalog.
7. Set the pricing and availability information for the adjustment.

Field	Description
Pricing Details	Set the fixed amount and currency of the override price. Select the frequency, such as **One-time** or **Monthly**. After you set the amount, the Pricing Variable list appears after **Add timing**.
Add timing	Click to specify the duration and the events that start and end the override. You can add timing if the product has a time plan/policy.
Add delay	If Add timing is enabled, click to add a delay to when the pricing override starts.
Availability	Specify the start and end dates for the promotion.

8. Click **Done**.
9. (optional) Select one or more rule sets to apply to the promotion.
10. Click **Done**.
    The new promotion override is applied to the product in the bundle.

Adding an Override to an Existing Promotion

1. In the Structure tab, click a product or child product.
2. In the Edit Relationship panel on the right, click the down arrow to expand the Pricing Adjustments/ Overrides section.
3. To add an adjustment click the **+** icon.
4. From the New Adjustment or Override dialog box, select **Override on an existing Promotion**.
5. In the Promotion field, search for and select a promotion.
6. In the Edit Override dialog box, enter the following information.

Field	Description
Display Name	Enter the name of the override. This name is shown in Vlocity Cart.
Pricing Details	Set the fixed amount and currency of the override price. Select the frequency, such as **One-time** or **Monthly**. After you set the amount, the Pricing Variable list appears after **Add timing**.
Add timing	Click to specify the duration and the events that start and end the override. You can add timing if the product has a time plan/policy.
Add delay	If Add timing is enabled, click to add a delay to when the pricing override starts
Availability	Specify the start and end dates for the promotion.

7. Click **Done**.
8. (optional) Select one or more rule sets to apply to the promotion.
9. Click **Done**.
   The new promotion override is applied to the product in the bundle.

Adding an Override to an Offer

1. In the Structure tab, click a product or child product.
2. In the Edit Relationship panel on the right, click the down arrow to expand the Pricing Adjustments/ Overrides section.
3. To add an adjustment click the **+** icon.

4. From the New Adjustment or Override dialog box, select **Override on an Offer**.
5. In the Add Override dialog box, enter the following information.

Field	Description
Price List	Select a price list.
Display Name	Enter the name of the override. This name is shown in Vlocity Cart.
Pricing Details	Set the fixed amount and currency of the override price Select the frequency, such as **One-time** or **Monthly**. After you set the amount, the Pricing Variable list appears after **Add timing**.
Add timing	Click to specify the duration and the events that start and end the adjustment. You can add timing if the product has a time plan/policy.
Add delay	If Add timing is enabled, click to add a delay to when the pricing adjustment starts.
Availability	Specify the start and end dates for the promotion.

6. Click **Done**.
7. (optional) Select one or more rule sets to apply to the promotion.
8. Click **Done**.
   The new offer override is applied to the product in the bundle.

Specifying Adjustments

You can create an adjustment to a bundled product in several ways:

- A new promotion or an offer
- An adjustment on an existing promotion
- An offer-based adjustment

Creating a New Promotion with an Adjustment

1. In the Structure tab, click a product or child product.
2. In the Edit Relationship panel on the right, click the down arrow to expand the Pricing Adjustments/ Overrides section.
3. To add an adjustment click the **+** icon.
4. From the New Adjustment or Override dialog box, select **Adjustment on a new Promotion**.
5. In the Add Adjustment dialog box, enter the following information.

Field	Description
Price List	Select a price list.
Promotion Name	Enter a name for the new promotion.
Display Name	Enter the name of the adjustment. This name is shown in Vlocity Cart.
Description	Enter a description that can appear in the Promotions list of the cart. As best practice, include the details of the price changes and durations for each product. For example, "20% off data plan for 6 months and $150 off the iPhone X".
Promo Code	Enter a unique code to identify the promotion from other promotions.

6. Click **Create Promotion**.
   The promotion is saved in the product catalog.

7. Set the pricing and availability information for the adjustment.

Field	Description
Pricing Details	Select the adjustment type (a percentage or a fixed amount). Set the percentage amount or the fixed amount and currency. Select the frequency, such as **One-time** or **Monthly**.
Add timing	Click to specify the duration and the events that start and end the adjustment. You can add timing if the product has a time plan/policy.
Add delay	If Add timing is enabled, click to add a delay to when the pricing adjustment starts.
Availability	Specify the start and end dates for the promotion.

8. Click **Done**.
9. (Optional) Select one or more rule sets to apply to the promotion.
10. Click **Done**.
    The new promotion adjustment is applied to the product in the bundle.

Adding an Adjustment to an Existing Promotion

1. In the Structure tab, click a product or child product.
2. In the Edit Relationship panel on the right, click the down arrow to expand the Pricing Adjustments/ Overrides section.
3. To add an adjustment click the **+** icon.
4. From the New Adjustment or Override dialog box, select **Adjustment to an existing Promotion**.
5. In the Promotion field, search for and select a promotion.
6. In the Add Adjustment dialog box, enter the following information.

Field	Description
Display Name	Enter the name of the adjustment. This name is shown in Vlocity Cart.
Pricing Details	Select the adjustment type (a percentage or a fixed amount). Set the percentage amount or the fixed amount and currency. Select the frequency, such as **One-time** or **Monthly**.
Add timing	Click to specify the duration and the events that start and end the adjustment. You can add timing if the product has a time plan/policy.
Add delay	If Add timing is enabled, click to add a delay to when the pricing adjustment starts.
Availability	Specify the start and end dates for the promotion.

7. Click **Done**.
8. (optional) Select one or more rule sets to apply to the promotion.
9. Click **Done**.
   The new promotion adjustment is applied to the product in the bundle.

Adding an Adjustment to an Offer

1. In the Structure tab, click a product or child product.
2. In the Edit Relationship panel on the right, click the down arrow to expand the Pricing Adjustments/ Overrides section.

3. To add an adjustment click the **+** icon.
4. From the New Adjustment or Override dialog box, select **Adjustment on an Offer**.
5. In the Add Adjustment dialog box, enter the following information.

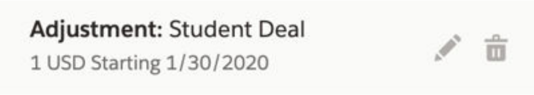
Field	Description
Price List	Select a price list. When the price list is changed, the currency updates with new currency values. If no currency values are present in the price list, the currency field is blank and a price list entry isn't created. Enter the currencies in the price list to allow the price list entry creation. No error message appears.
Display Name	Enter the name of the adjustment. This name is shown in Vlocity Cart.
Pricing Details	Select the adjustment type (a percentage or a fixed amount). Set the percentage amount or the fixed amount and currency. Select the frequency, such as **One-time** or **Monthly**.
Add timing	Click to specify the duration and the events that start and end the adjustment. You can add timing if the product has a time plan/policy.
Add delay	If Add timing is enabled, click to add a delay to when the pricing adjustment starts.
Availability	Specify the start and end dates for the promotion.

6. Click **Done**.
7. (optional) In the Context Rules, search for and select a rule set to apply.
8. Click **Done**.
   The new offer adjustment is applied to the product in the bundle.

Editing or Deleting an Adjustment or Override

When an adjustment is deleted, it doesn't delete the promotion because other products may be tied to it. Deleting the promotion adjustment or override entry doesn't delete the promotion.

1. In the Structure tab, click a product or child product.
2. In the Edit Relationship panel on the right, click the down arrow to expand the Pricing Adjustments/ Overrides section.



Adjustment: Student Deal
1 USD Starting 1/30/2020

3. To delete an adjustment or override, click the 🗑 icon, and click **Delete** in the dialog box.
4. To modify an adjustment or override, click the ✎ icon.
5. When you've finished the change, click **Done**.

What's Next

Define Pricing for Offers in the Product Designer

See Also

- Overview of Product Creation in Product Designer

Offer Pricing for Product Designer

You can price products and promotions according to your business requirements. You can also adjust prices (by a percentage or amount) or override prices (set new prices) for products.

Vlocity's pricing solution includes the following benefits:

- Reusability — A component-oriented system with reusable items that are independent of products.
- Pricing types — Categories of pricing such as penalties, charges, and adjustments to existing charges.
- Frequency — Settings that determine the frequency of the charge.
- Updates — Transition from previous prices to updated prices for less expense and disruption.

Using Vlocity's pricing capabilities, you can:

- Adjust or override a price without changing the base price.
- Offer customers the option of paying with loyalty points.
- Change the price of a product automatically over time.
- Customize the pricing process by adding steps, such as a step to accommodate tax calculation.
- Automatically reprice products, draft items in the cart, and assets.
- Set up usage pricing, which lets you charge for usage quantities over units of time.

NOTE

You must create pricing elements in the Vlocity Product Console, including price lists, pricing plans, time plans and policies, promotions, and discounts.

Then, in Vlocity Product Designer, you can set the pricing, promotions, and discounts for standalone and bundled offerings.

Pricing Components

Use pricing components to define the price aspects and assign them to the offering. This is the process of creating a price list entry (PLE).

For a product to appear in the Vlocity Cart's Products list, it must have all the following settings:

- An assigned, effective price
- An active status
- An effective date

Pricing Strategies for Product Bundles

A product bundle contains one or more child products. The prices of all child products always roll up into the total price for the bundle.

The overall bundle pricing can be accomplished in different ways:

- The parent product has a price other than zero. The parent product price is added to the child product price.
- The parent product has a price of zero. The final bundle price is the total of the price of the child products.

For example, the parent product of the bundle can be priced at $49.99. It is combined with the child products to make up the total price.

When bundles contain optional products, the display text might show the total price if all the products are purchased. However, unless all products are added to the cart, the price in the Products list in the cart won't match what you see in the cart. You can change the display text to indicate a range or a starting price for these types of bundles.

Use the two following means to change the price of a child product in a bundle without changing the base price:

- Adjustments — Percentages or amounts that reduce or increase the base price.
- Overrides — Amounts that replace the base price

Adjustments use the base price to calculate the new price. Overrides do not use any calculations. Overrides simply replace the base price.

When you create an adjustment or an override to a product's base price, you create a price list entry that is stored in the same price list as the base price. As a result, the system displays a message that multiple price list entries are in the price list for this product.

When you create product adjustments and overrides for child products, you do so in the product's Structure tab.

Price Adjustments and Overrides

You can adjust or override a product's price without changing the price itself. You can make adjustments in the following contexts:

- Manual adjustments
- Product bundles
- Promotional adjustments

Pricing Adjustments Using Product Bundles

When designing product bundles, you can add products to the bundle and then change their prices when they appear in that context. This is done at design-time using the Vlocity Product Console. When you create an adjustment, you can adjust the price by a percentage or by an amount. Adjustments use the base price to calculate the final product price. For more information, see Adjusting the Base Price for a Product in a Bundle.

Overriding the base price doesn't require the base price for calculation. It replaces the base price. The base price, however, remains unchanged.

Recording Adjustments

For any order, quote, opportunity or asset, you can review adjustments or overrides made to each line item's price in Order Pricing, Quote Pricing, Opportunity Pricing, and Account Pricing records. The adjustments that are recorded could be the result of an applied promotion, a manual adjustment made by an agent, or an adjustment or override made in the context of a product bundle.

Managing Simple Product Pricing

You can select an existing price list, parent or child, and create one-time and recurring charges as absolute amounts. You can also update or delete a price list.

Vlocity Product Console automatically detects pricing variables. If multiple pricing variables are found, then you can choose. From the Vlocity Product Console, you can search for and select qualification context rules.

Base prices appear in order of descending effective start date. You can use custom filters to search rows in the pricing list.

Product Cost

You can edit or delete the (PriceListEntry) row in the Prices table using the Cost table Delete/Edit icon. The delete action also removes the row from the table. You can also delete adjustments from Promotions and Offers.

Offer-Based Adjustments and Overrides

You can add, update, or delete an offer-based adjustment or an override for a simple product in the context of its parent or parent hierarchy.

The page shows you all the parent and grandparent offers up to the top-level offer to which the simple product belongs. A summary of adjustments and overrides defined in the context of an Offer or Promotion appears in a single list.

You can use custom filters to search rows in the Promotions/Offer Adjustment and Override list and then select an existing price list (PL), parent PL, or child PL. You can also create one-time and recurring adjustments or overrides in absolute amounts for overrides, and percentage and absolute amounts for adjustments. You can update or delete an adjustment or override *price list entries* (PLE) except for the PL name.

Vlocity automatically detects:

- Time Plan and Time Policy for recurring adjustments and overrides.
- Pricing variables. If multiple pricing variables are found, they are displayed so you can select one.
- Pricing elements. If no pricing elements are found, a new pricing element is created. If duplicate pricing elements are found, they are displayed so you can select one.

You can also find and select qualification context rules.

Promotion-Based Adjustments and Overrides for Product Designer

While you can create products and offers that have discount prices, use promotions to discount a product's offer for certain customers, limited time periods, or specific subsets of products. You can also override or adjust promotions

Promotions offer additional functionality:

- Discount the price of a product in a promotion without updating the product hierarchy or price book.
- Apply promotional discounts to products that are already assets.
- Automatically charge a penalty if the customer does not fulfill the promotion’s time commitment.
- Create promotions that begin automatically as soon as a previous promotion ends.
- Automatically continue or discontinue the subscription of a promotion’s products based on the customer’s request.

Differences Between Promotions and Products

In the shared product catalog, you can use can adjust or override the base prices of products in product bundles.

Product	Promotion
Typically does not expire after a limited time	For a limited time
Not for a limited group of customers	For a limited group of customers
Products or quantities of each product that are not necessarily limited	A limited subset of products

Products Within Promotions

A promotion applies to at least one product. You can add a promotion to both individual products and product bundles. Then you can change the pricing and cardinality of any product in the promotion.

As part of the promotion’s settings, you determine:

- Time commitment: How long the customer must subscribe to the promotion’s products
- Changes to pricing and time frame: Whether the products of the promotion are changed by an adjustment or override, and for what length of time
- Whether the promotion appears in the Promotions list of the cart

Promotion-Based Adjustment and Overrides for Simple Product Price

You can adjust or override a new or existing promotion by adding directly the simple product or its parent or parent hierarchy. The page shows all the parent and grandparent offers up to the top-level offer to which the simple product belongs. A summary of adjustments and overrides defined in the context of an offer or promotion appears in a single list. For promotion-based adjustment or overrides, show the offer name that is attached to the promotion.

You can select any offer and create a new promotion, and you can use custom filters to search rows in the Promotions/Offer Adjustment and Override list.

You can also create one-time and recurring adjustments or overrides in absolute amounts for overrides and percentage and absolute amounts for adjustments. You can update or delete an adjustment or override the price list entry (PLE) except for the associated promotion name.

The following are automatically detected:

- Time Plan and Time Policy for recurring adjustments and overrides.
- Pricing variables. If multiple pricing variables are found, they are displayed so you can select one.
- Pricing elements. If no pricing elements are found, then the Product Console creates a new pricing element. If duplicate pricing elements are found, they are displayed so you can select one.

You can also find and select qualification context rules.

Define Pricing for Offers in the Product Designer

You can define price and cost, create promotions and offers, and apply adjustments and overrides.

Each table in the Pricing tab has an arrow button which, when clicked, either expands or collapses the corresponding table. On load, only tables with records appear expanded. All other tables are collapsed. All tables only show active records by default.

This information is for the Vlocity Product Designer. For the Vlocity Product Console, see Product Pricing.

In compliance with TM Forum Frameworx standards, offerings have pricing information, but product specifications do not.

Before You Begin

- Complete the following tasks In the Vlocity Product Console.
  - Create a price list with price list entries and pricing elements.
  - Create context rules and rulesets.
  - Create time plans and policies.
  - Create promotions and discounts.

Complete each of the following tasks as required for your offering.

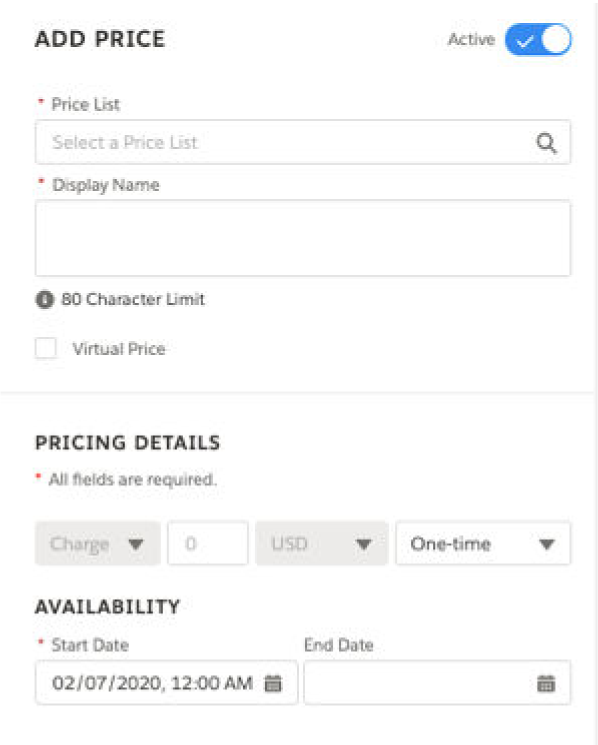
Adding a Price

This section shows product prices. In the Price table:

- Search for price list entries.
- Display all prices or only the active prices. Active prices are the default setting.
- Sort the rows by clicking the arrows beside the column headings.

To add a price:

1. From the Vlocity Product Designer app, open a product specification or offering, and click the Pricing tab.
2. In the Prices section, click **New Price**.
   The Add Price panel appears.



ADD PRICE

Active ☑

* Price List

Select a Price List

* Display Name

80 Character Limit

□ Virtual Price

PRICING DETAILS

* All fields are required.

Charge ▼ | 0 | USD ▼ | One-time ▼

AVAILABILITY

* Start Date: 02/07/2020, 12:00 AM

End Date:

3. In the Add Price panel, enter the following information.

Field	Description
Price List	Select the price list to use.
Display Name	Enter a name to show for the price.
Virtual Price	Select this option if a parent product does not have a price of its own but is the total price of all the child products.
Pricing Details	Select the charge type, quantity, and currency.
Active	Indicate that the price is currently in use.
Availability	Specify the start and end dates for the price.

4. Click **Done**.

The new price is added to the Pricing table. You can edit or delete a price by clicking the appropriate action icon. To view inactive prices, deselect **Show Active Only** in the **Prices** view.

Adding Promotions and Offers

This section shows the promotions and offers for product pricing. In the Promotions and Offers table, you can:

- Search for adjustments and overrides.
- Display only the active promotions and offers.
- Sort the rows by clicking the arrows beside the column headings.

To edit a promotion:

1. Open a product specification or offering, and click the Pricing tab.
2. In the Pricing tab, click the Edit icon beside the promotion or offer you want to edit. The Edit Adjustment panel appears.
3. Enter a display name for the price.
4. Select the adjustment type, quantity, and currency.
5. Specify the adjustment frequency and duration.
6. Select a pricing variable.
7. Specify the purchase date and end of plan duration.
8. Add a delay, if necessary, by specifying. the start and end dates.
9. Specify whether the price is active
10. Click **Done**.

You can delete a promotion by clicking the Delete action icon. To view inactive promotions and offers, deselect **Show Active Only** in the **Promotions & Offers** view.

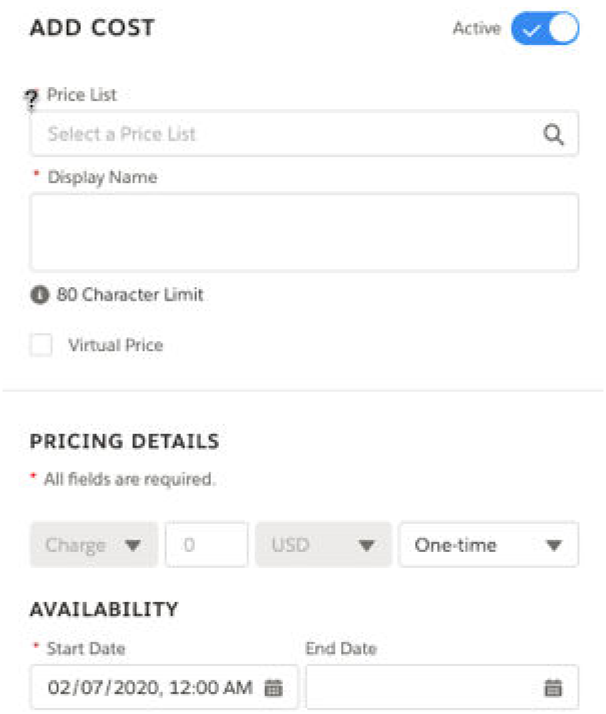
Adding a Cost

This section shows the product costs. In the Cost table, you can:

- Search for cost entries.
- Display only the active costs.
- Sort the rows by clicking the arrows beside the column headings.

To add a cost:

1. Open a product specification or offering, and click the Pricing tab.
2. Click **New Cost**.
   The Add Cost panel appears.



ADD COST

Active ✓

Price List

Select a Price List

* Display Name

80 Character Limit

□ Virtual Price

PRICING DETAILS

* All fields are required.

Charge ▼ | 0 | USD ▼ | One-time ▼

AVAILABILITY

* Start Date: 02/07/2020, 12:00 AM

End Date

3. In the Add Price panel, enter the following information.

Field	Description
Price List	Select the price list to use.
Display Name	Enter a name to show for the cost.
Virtual Price	Select this option if a parent product does not have a price of its own but is the total price of all the child products.
Pricing Details	Select the charge type, quantity, and currency.
Availability	Specify the start and end dates for the price.
Active	Indicate that the price is currently in use.

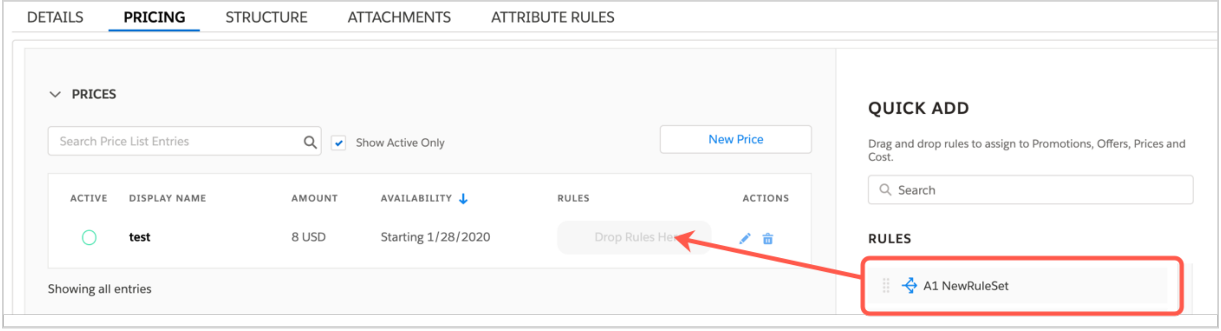
4. Click **Done**.

The new cost is added to the Cost table. You can edit or delete a price by clicking the appropriate action icon.

Assigning Context Rules

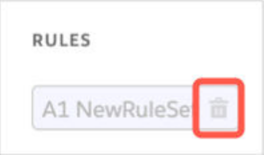
To assign a context rule to a promotion, offers, prices, or cost:

1. Open a product specification or offering, and click the Pricing tab.
2. Expand the Prices, Promotions & Offers, or Costs section that needs a rule.
3. Select a rule from the Quick Add panel.
4. Drag and drop the rule to the Rules column on the row for your item.



ACTIVE	DISPLAY NAME	AMOUNT	AVAILABILITY	RULES	ACTIONS
○	test	8 USD	Starting 1/28/2020	Drop Rules He	

5. To remove a rule, click the Delete icon for the rule.



See Also

- Overview of Product Creation in Product Designer

Track Product Catalog Changes with Projects

EPC Projects track all the changes to entities in your product catalog. You can use projects as a history for change management.

You can have as many projects as you want. Multiple projects can be active at the same time, but only one project is the default. When entities are created, modified, or deleted, that change is saved to the default project.

A project is a record type called a *work set*. The project tracks all work done to any catalog object, such as creating products, modifying promotions, creating picklists, etc.

This feature is available only for the Vlocity Product Designer.

In Spring '20 and later, you can use IDX Workbench to migrate the changes in an EPC Project to other orgs. When configuring IDX Workbench, you can specify an EPC Project object in your Source org instead of configuring a project in IDX Workbench. You can use IDX Workbench to export Projects in **Released** status in Fall '20 and **Deployed** status in Spring '20. See "Configuring a Project Object" in IDX Workbench Configuration for Migration.

Before You Begin
The Projects feature must be enabled and configured. See Enable and Set Up Projects for Vlocity Product Designer.

1. From the Vlocity Product Designer, click the ⌄ button, and select **Projects**.
2. In the Projects tab, click **New**.

3. In the New Project: Work Set page, enter the following information.

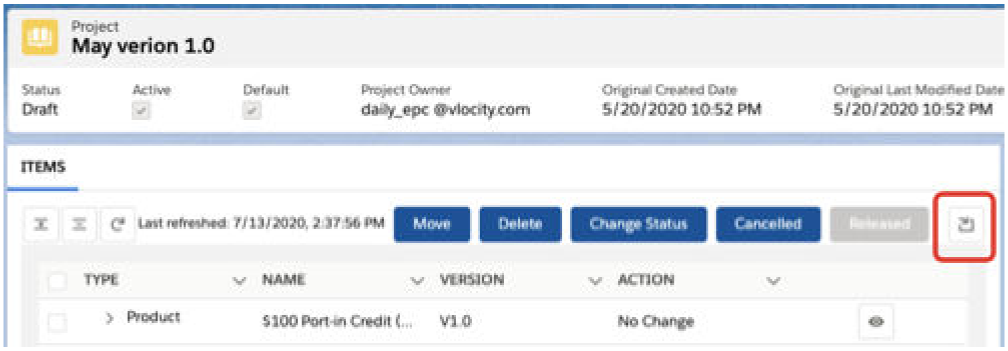
Field	Description
Project Name	Enter a unique name for a project.
Active	Indicate that the project is active and in use. By default, active projects are shown in the query results.
Description	Enter a description of a project.

The following fields are set automatically.

> **NOTE**
> These values can differ from the values of the Salesforce system fields (Created By, Created Date, Last Modified By, and Last Modified Date) if the Project record is moved to another Salesforce org.

Field	Description
Project Owner	Select the user who owns a project. A project has only one owner. Only the owner can modify a project, delete a project, or set it as the default project.
Creator	The full name of the person who created a project. This field is automatically set.
Last Modifier	The full name of the person to modify a project most recently. This field is automatically set.
Global Key	A unique system ID for a project. This field is automatically set.
Original Created Date	The date and time when a project is created. This field is automatically set.
Last Modified Date	The date and time when a project was most recently updated. This field is automatically set.

4. Click **Save**.
   The new project is saved, and a new page for the project opens.
5. To set the project as the default, click **Set as Default** ( ).
   All changes are automatically saved in the default project.



Project
May verion 1.0

Status	Active	Default	Project Owner	Original Created Date	Original Last Modified Date
Draft	☑	☑	daily_epc @vlocity.com	5/20/2020 10:52 PM	5/20/2020 10:52 PM

ITEMS

Last refreshed: 7/13/2020, 2:37:56 PM

TYPE	NAME	VERSION	ACTION
Product	$100 Port-in Credit (...	V1.0	No Change

Results

A new project is saved and available to track product catalog changes. If set as the default, the new project saves all future changes.

The Items list for a project shows information for each change:

- The type of item that changed, such as Product, Picklist, or Attribute.
- The name of the changed item that either you set when creating the item or that the system automatically sets.
- The version of the item as set in its **Version Label** field. This column is blank if the item has no version set.
- The action taken on the item: **Add**, **Delete**, or **Change**. The action can be **No Change** if the parent item, such as a product, hasn't changed but a child item has changed, such as a modified child product item.

  If an item undergoes more than one action, the Action column shows only one action of the highest precedent, according to the following order. For example, if a Region attribute is added and then deleted, the Project Items list shows Delete as the action for that attribute item.

  1. Delete
  2. Add
  3. Changed
  4. No Change

EPC Project Editing

You can modify an EPC project as the following table describes.

Task	Steps
Change a project.	From the Projects page, click the arrow on the project row, and select **Edit**.
Remove a project.	From the Projects page, click the arrow on the project row, and select **Delete**. The change itself isn't removed.
Assign the project to a new owner.	From the Projects page, click the arrow on the project row, and select **Change Owner**. In the Change Owner dialog box, search for the name of the new owner, select, and click **Submit**.
Move a change to a different project.	Move a change from one project to another. You can move only one change at a time. Only the owner of the project can move items. You can't move an item for a project if its **Lifecycle Status** set to **Released**. Only moving of whole bundles starting from root product is supported. Intermediate parents or child items can't be moved on their own without moving the whole bundle. If any items conflict with existing items, you're warned and can cancel or continue with the move. If you click **Proceed** in the dialog box, the matching root is deleted in the root's hierarchy in the target workset, and the new hierarchy inserted in its place. 1. From the list of changes in a project, select one item or multiple items to move. 2. Click **Move**. 3. In the Move Project Items panel, select the project to move the selected items to. 4. Click **Done**.
Remove a change from a project.	Delete an item from the change history in a project. Only the owner of the project can remove items. You can't delete an item for a project if its **Lifecycle Status** set to **Released**. The change itself isn't removed. 1. From the list of changes in a project, select an item or multiple items to remove. 2. Click **Delete** and confirm the deletion. The change is removed from the project's history.
Review the changed item.	From the list of changes in a project, click the arrow for a change item and select **View Item**. The object with the change opens on a new page.

Export a Product Report in Product Designer

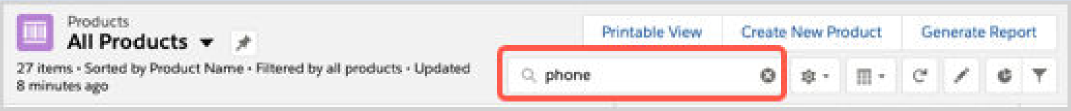
To help with audits and product reviews, export a report of products with their properties and attributes. The report is saved as a `.xslx` file.

The report is limited to 30 products, so you must use filters to limit the products. If you've more than 30 products filtered, no report is created. The report is created and downloaded to your computer as a file named `report.xslx`.

See Results for a description of the reports.

NOTE

You can't use the Search tool to limit the list of products to 30 or fewer. You must use filters.

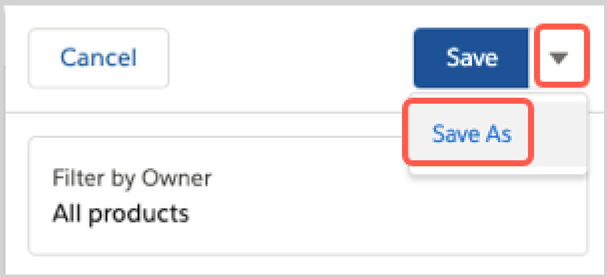


27 items • Sorted by Product Name • Filtered by all products • Updated 8 minutes ago

1. In the Vlocity Product Designer, click the ⌄ button, and select **Products**.
2. Select a view with 30 or fewer products.
   The report includes only the products that are included in a filter. Products that don't match the filter values aren't exported.

   NOTE

   Selecting the checkboxes for products doesn't change the products included in the report.
3. To create a view, click **Show Filters**, add filters to show specific products, and click **Save As**.



Filter by Owner
All products

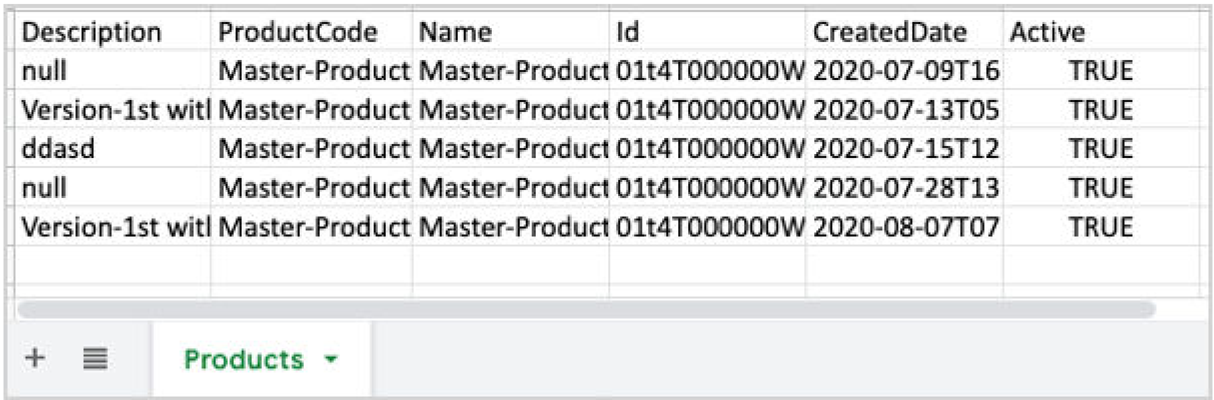
4. Click **Generate Report**.

> **NOTE**
> This step can take longer to complete depending on the number of products and attributes.

5. In the Select Report Options dialog box, choose the data to include in the export, and click **Generate**.
   - Product with Attributes — This report has two tabs: one that shows a list of products and another that shows attributes by product.
   - Products Only — This report has one tab that shows a list of products.

Results

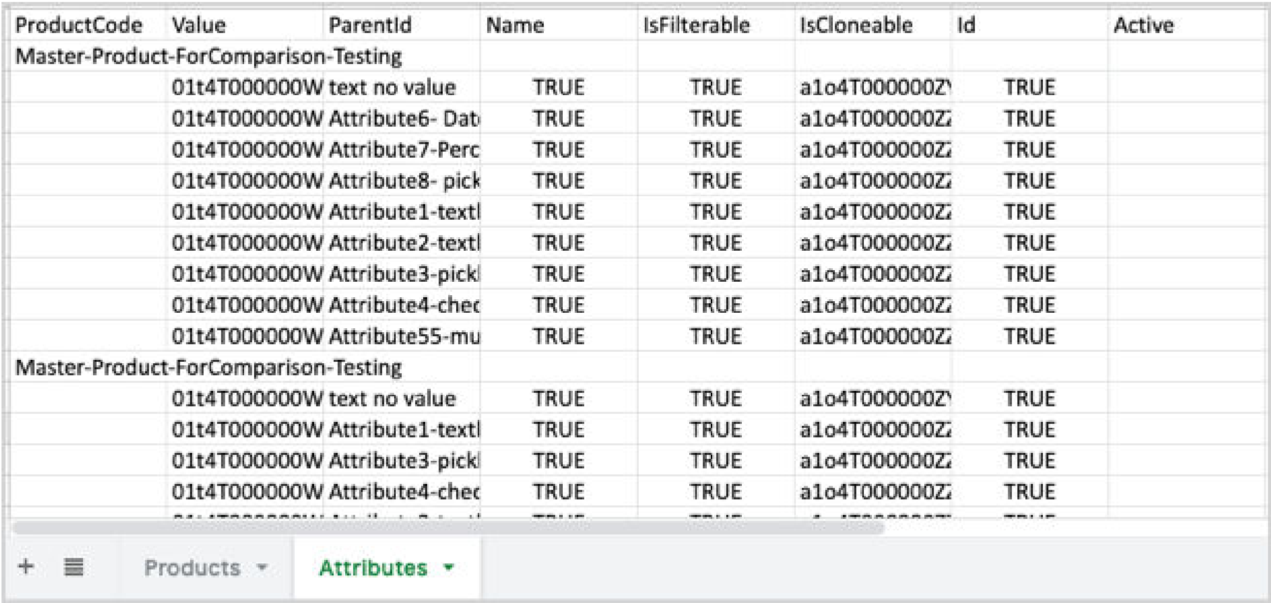
The Product Only report shows all the products and the Description, Product Code, Name, ID, Created Date, and Active columns for each.



Description	ProductCode	Name	Id	CreatedDate	Active
null	Master-Product	Master-Product	01t4T000000W	2020-07-09T16	TRUE
Version-1st witl	Master-Product	Master-Product	01t4T000000W	2020-07-13T05	TRUE
ddasd	Master-Product	Master-Product	01t4T000000W	2020-07-15T12	TRUE
null	Master-Product	Master-Product	01t4T000000W	2020-07-28T13	TRUE
Version-1st witl	Master-Product	Master-Product	01t4T000000W	2020-08-07T07	TRUE

The Products with Attribute report shows two tabs:

- Products — Each product shows the Description, Product Code, Name, ID, Created Date, and Active columns.
- Attributes — The attributes are grouped by product and include the Product Code, Value, ParentID, Name, IsFilterable, IsCloneable, and Active columns.



ProductCode	Value	ParentId	Name	IsFilterable	IsCloneable	Id	Active
Master-Product-ForComparison-Testing							
	01t4T000000W	text no value	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute6- Dat	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute7-Perc	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute8- pick	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute1-textl	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute2-textl	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute3-pickl	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute4-chec	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute55-mu	TRUE	TRUE	a1o4T000000Z	TRUE	
Master-Product-ForComparison-Testing							
	01t4T000000W	text no value	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute1-textl	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute3-pickl	TRUE	TRUE	a1o4T000000Z	TRUE	
	01t4T000000W	Attribute4-chec	TRUE	TRUE	a1o4T000000Z	TRUE	

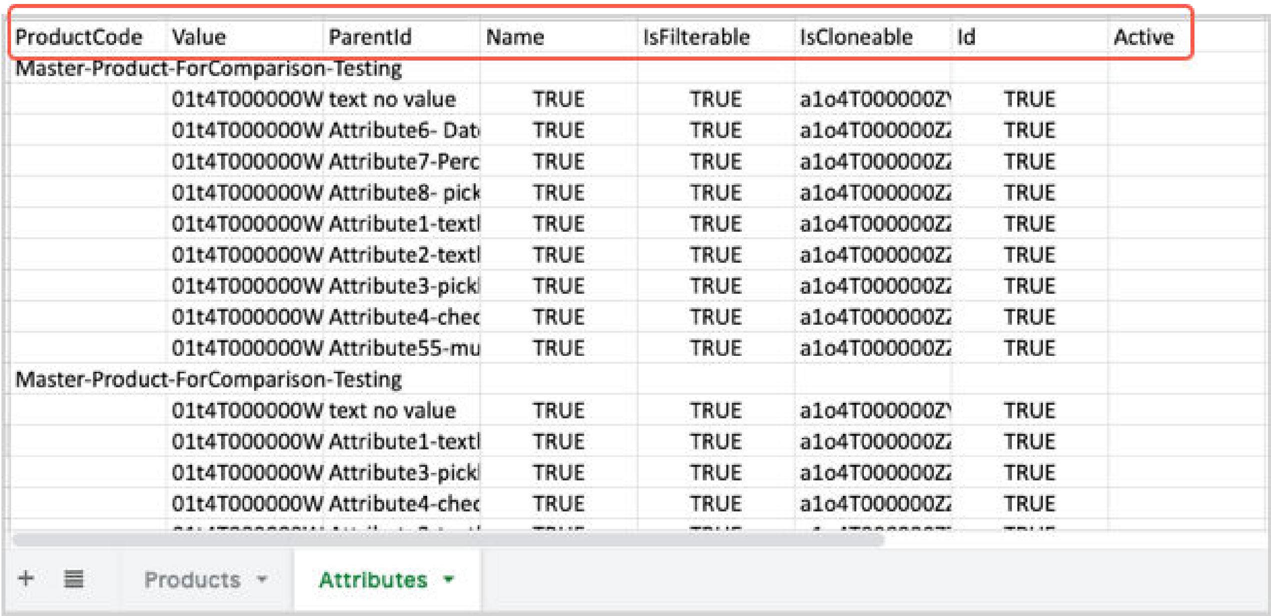
See Also

- Configure the Product Attributes Report

Configure the Product Attributes Report

For the product attribute report in Vlocity Product Designer, you can modify integration procedures to set the columns, or fields, are used for the products and attributes.

The report includes a list of the products and all their attributes with default columns for each product and attribute. To change what columns are in the report, you can edit the UPC/ExtractProductInformation Vlocity integration procedure. While you can add or remove these columns, you can't select which attributes are reported.



ProductCode	Value	ParentId	Name	IsFilterable	IsCloneable	Id	Active
Master-Product-ForComparison-Testing							
	01t4T000000W	text no value	TRUE	TRUE	a1o4T000000Z'	TRUE	
	01t4T000000W	Attribute6- Dat	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute7-Perc	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute8- pick	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute1-textl	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute2-textl	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute3-pickl	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute4-chec	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute55-mu	TRUE	TRUE	a1o4T000000Z:	TRUE	
Master-Product-ForComparison-Testing							
	01t4T000000W	text no value	TRUE	TRUE	a1o4T000000Z'	TRUE	
	01t4T000000W	Attribute1-textl	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute3-pickl	TRUE	TRUE	a1o4T000000Z:	TRUE	
	01t4T000000W	Attribute4-chec	TRUE	TRUE	a1o4T000000Z:	TRUE	

Example 12. Report Changes

Monique doesn't want the ParentID column in the report and wants to add the IsNotAssetizable and AttributeID fields as report columns.

NOTE
Don't remove the ProductCode column because it's required to match products for the report.

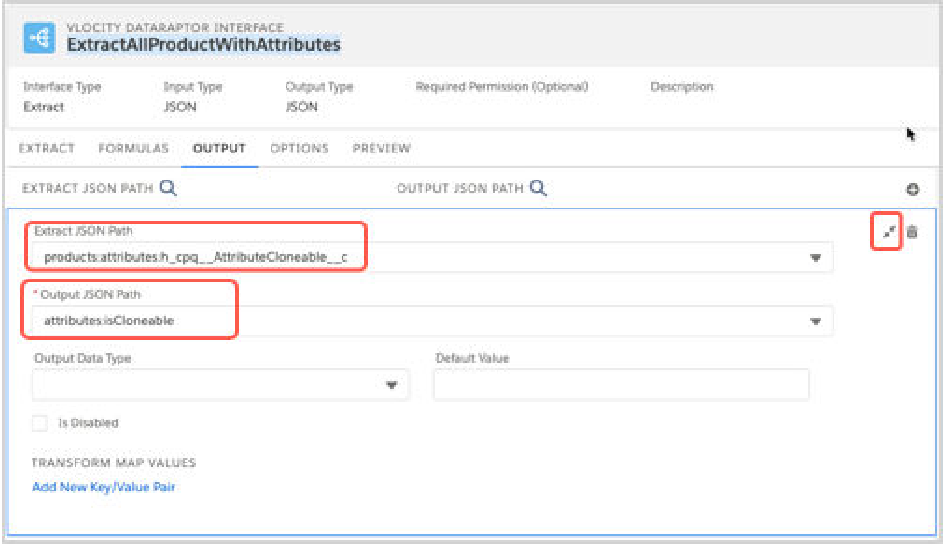
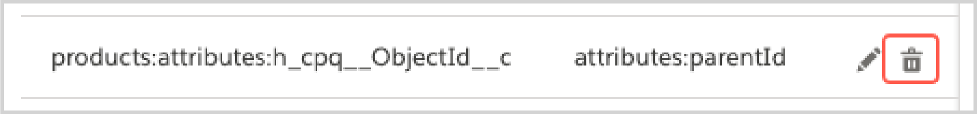
1. Open the Vlocity Integration Procedures tab.
2. Expand and open the most recent version of UPC/ExtractProductInformation.
3. Click **Create Version**.
4. Expand the FilteredProductCondition block.
5. Expand and select the corresponding blocks for the type of report to modify.

Report	Steps
Products and attributes report	1. Expand the ProductsWithAttribute block. 2. Click FilteredProductExtractor.
Products-only report	1. Expand the ProductsWithAttribute block. 2. Click FilteredProductsOnlyExtractor.

6. In the Properties tab, click the link icon to open the associated DataRaptor page.

- ExtractAllProductWithAttributes is for the products and attributes report.
- ExtractAllProducts is for the products-only report.

7. On the DataRaptor page, click the Output tab, and change the mappings.

Task	Steps
Add a report column	1. Click the **+** icon to add a mapping, and select from the **Extract JSON Path** list. 2. For the Output JSON Path, enter a value. • For attributes, you must use the `attributes:` prefix, such as attributes:IsFilterable. • Product fields don't require a prefix, such as active. • Use a descriptive name of your choice.  VLOCITY DATARAPTOR INTERFACE ExtractAllProductWithAttributes Interface Type: Extract Input Type: JSON Output Type: JSON Required Permission (Optional) Description EXTRACT FORMULAS OUTPUT OPTIONS PREVIEW EXTRACT JSON PATH OUTPUT JSON PATH Extract JSON Path: products:attributes:h_cpq__AttributeCloneable__c * Output JSON Path: attributes:isCloneable Output Data Type Default Value □ Is Disabled TRANSFORM MAP VALUES Add New Key/Value Pair 3. Click the **Collapse** button.
Remove a report column	Click the **Delete** icon. **NOTE** Don't remove the products:ProductCode extract JSON path because it's required to match products for the report.  products:attributes:h_cpq__ObjectId__c attributes:parentId

8. In the Product Information Extractor tab, click **Save** and, if ready to use, **Activate Version**.

Results

Future product attribute reports include the changes to the columns.

See Also

- Export a Product Report in Product Designer

- DataRaptor Extract Overview
- Create an Integration Procedure

Fix Duplicate and Empty Attributes

You can see empty attributes if you create an attribute for the Product2 object type and delete it. This deletion can create an empty attribute assignment under a child object type of Product2. Duplicate attributes can occasionally occur. You can run the following scripts to resolve these issues.

1. From **Setup**, click **Developer Console**.
2. From the **Debug** menu, click **Open Execute Anonymous Window**.
3. Enter the appropriate code, and click **Execute**.

Issue	Resolution

Empty attributes	`List<vlocity_cmt__ObjectFieldAttribute__c> ofaList = [SELECT Id FROM`
	`vlocity_cmt__ObjectFieldAttribute__c WHERE vlocity_cmt__AttributeId__c = null AND`
	`vlocity_cmt__FieldApiName__c = null];`
	`delete ofaList;`

Duplicate attributes	`List<vlocity_cmt__ObjectFieldAttribute__c> ofaList = [SELECT`
	`Id,vlocity_cmt__AttributeId__c FROM vlocity_cmt__ObjectFieldAttribute__c WHERE`
	`vlocity_cmt__ObjectClassId__c = 'a214x0000000BzQAAU' AND vlocity_cmt__SubClassId__c =`
	`null];`
	`Set<Id> attrIdSet = new Set<Id>();`
	`List<vlocity_cmt__ObjectFieldAttribute__c> ofaListToDelete = new`
	`List<vlocity_cmt__ObjectFieldAttribute__c>();`
	`for(vlocity_cmt__ObjectFieldAttribute__c ofa:ofaList)`
	`{`
	`    if(attrIdSet.Contains(ofa.vlocity_cmt__AttributeId__c))`
	`    {`
	`        ofaListToDelete.add(ofa);`
	`    }`
	`    attrIdSet.add(ofa.vlocity_cmt__AttributeId__c);`
	`}`
	`delete ofaListToDelete;`

Product Definition with Vlocity Product Console

Vlocity Product Consoleprovides a single, contextual interface for managing your EPC elements. Using this interface, you perform product management by defining and maintaining product entities. You can build foundational constructs like attributes and picklists, as well as metadata constructs like product object types.

The product console is designed to build fundamental constructs, such as:

- System administrators can use predefined objects, such as Salesforce Object2, to create reusable object types that define properties (attributes or fields) and layouts common for all product entities. For example, you can create offers or product specifications that are instantiated from these object types. This is typically done one time, as part of the initial setup.
- Product managers can manage product structure, pricing, rules, and offers.

Many elements of Vlocity CPQ and order management are co-located in the Vlocity Product Console to provide seamless integration between the shared product catalog, the ability to configure, price and quote the products in the catalog, and define technical specifications for order decomposition and fulfillment.

NOTE
The Vlocity Product Designer is the next generation Product Catalog administration application. See Overview of Product Creation in Product Designer.

The Vlocity Product Console Interface

The Vlocity Product Console provides a unified user experience to manage EPC elements in a single, contextual interface. The Vlocity Product Console is a web-based application for managing the Vlocity Enterprise Catalog.

It gives both business and IT users access to all the product management capabilities they need.

- Improves usability for product definition, pricing, and structure.
- Increases productivity for product marketing and admin users.
- Provides a cohesive view of shared catalog elements across Configure Price Quote, Contract Life Cycle Management, and Order Management.
- Provides omni-channel capabilities directly within the catalog.
- Decreases time to market for new promotions and products.

The Vlocity Product Console provides seamless integration between the shared product catalog and the ability to configure, price, and quote the products in the catalog. The dashboard contains workspaces for product management, pricing, rules, and more. Here, you can access many elements of Vlocity CPQ that are co-located on the Vlocity Product Console.

Product managers use the Vlocity Product Console for the following tasks:

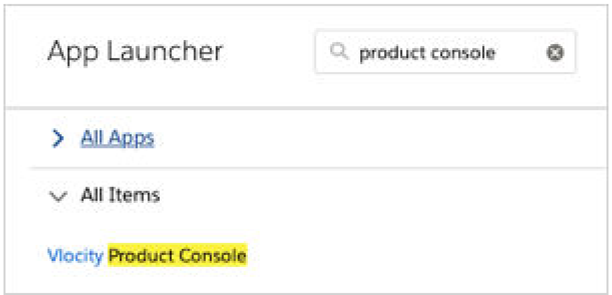
- Define and maintain product entities.
- Manage product structure, pricing, rules, and offers.
- Build foundational constructs, including attributes and picklists.
- Build metadata constructs, such as product object types.

System administrators use the Vlocity Product Console for these tasks:

- Work with predefined objects, for example, to create reusable object types that define properties, such as attributes and fields.
- Define common layouts for product entities, such as offers or product specifications, that are instantiated from the object types.
- Perform initial setup tasks at the start of Vlocity EPC implementation.

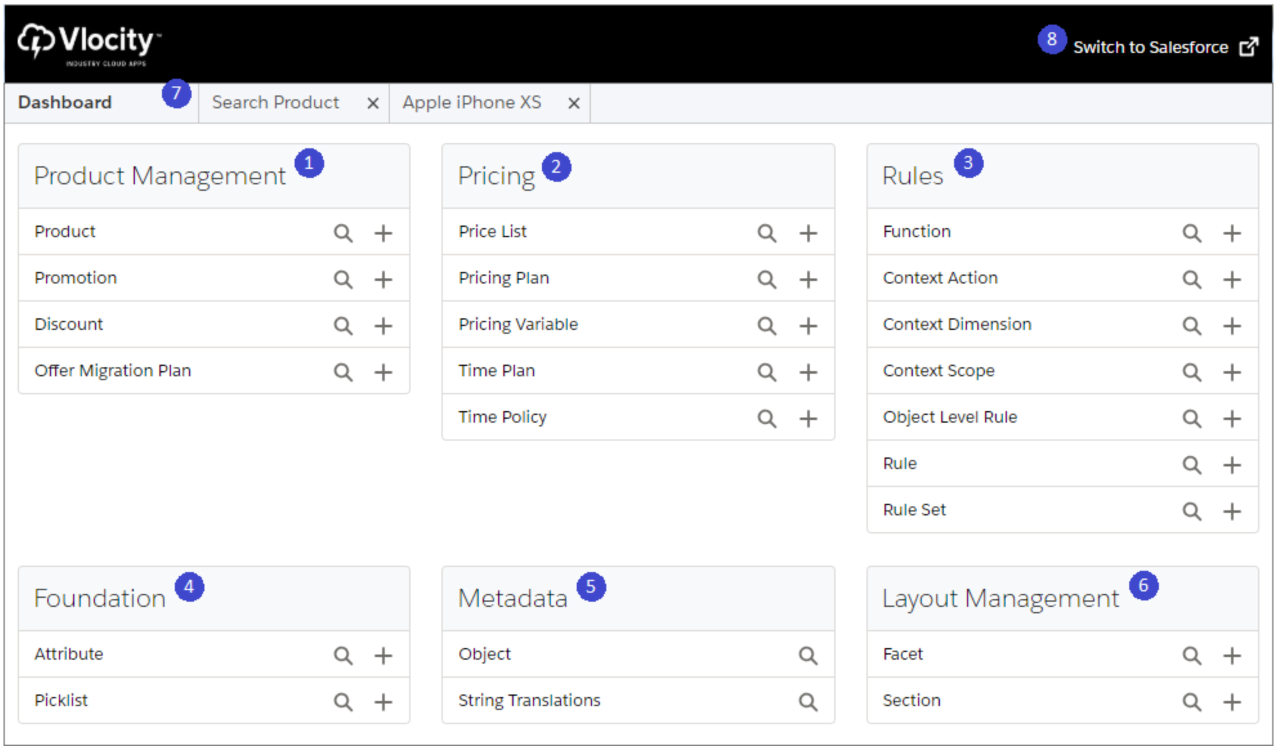
To access the Vlocity Product Console, in the Lightning interface:

1. Click the App Launcher and type **`Vlocity Product Console`** in the quick find box.
2. Click the Vlocity Product Console item.



product console

The interface has the following sections:



1. **Product Management**: Vlocity Product Console enables you to perform product management by defining and maintaining product entities. You can create products, promotions, discounts, and offer migration plans.
2. **Pricing**: Vlocity Communications, Media, and Energy enables you to price products and promotions according to your business requirements. Vlocity's pricing solution includes the following benefits.
   - Reusability
   - Categories of pricing such as penalties, charges, and adjustments to existing charges.
   - Settings that determine the frequency of the charge.
   - Updates from previous prices.
3. **Rules**: In Vlocity Communications, Media, and Energy, you can use rules to change standard product and pricing behaviors in Vlocity Cart. Vlocity Rules are essential to creating the perfect order. You control all the products and services you offer to your customer and offer them at the right prices. There are two rules frameworks that work in tandem: Context rules and advanced rules.
   - Context rules help you determine which products, promotions, and prices appear in Vlocity Cart. You can specify when to apply a penalty for cancellations and what the penalty will be.
   - Advanced rules help you design product compatibility rules based on conditions in order line items and related objects. You can also create advanced pricing, availability and eligibility rules.
4. **Foundations** contain attributes and picklists:
   - An attribute is a property or characteristic. In Vlocity apps, attributes describe profiles and products. An attribute is a property or characteristic. In Vlocity apps, attributes describe profiles and products. Product attributes must be grouped under Categories. For more information, see Product Attributes and Profile Attributes.

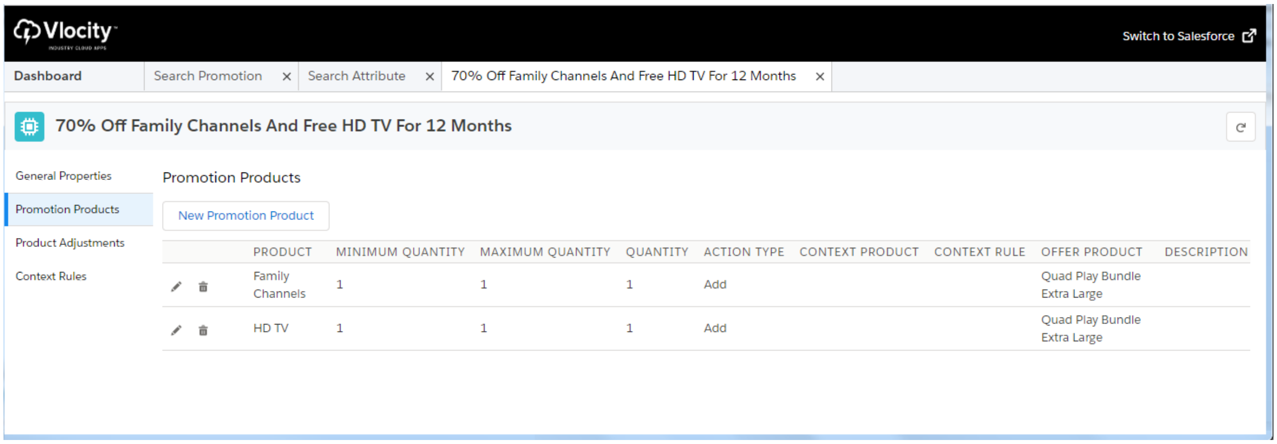
- A picklist is a list of values from which a user can choose. Used in product attributes and in context dimensions for context rules. Vlocity picklists are similar to Salesforce picklists, but are defined in a Vlocity-specific custom object.

5. **Metatdata** contains objects and string translations:
    - Metadata is the overall definition of the type of information you are storing. For example, the Case object enables you to store information regarding customer inquiries. For each object, your org will have multiple records that store the information about specific instances of that type of data. You might have a case record to store the information about someone's training inquiry and another case record to store the information about someone's configuration issue.
    - If your org has multiple languages enabled, string translations let you translate labels in your org to another language.
6. Layout Management: You can use the Vlocity Product Console to create a layout for a new product object. You create the base product object-type, assign attributes and field, and define a layout, which consists of creating a new facet and adding sections. You can then create object subtypes instantiated on the base product object-type.
7. Move between pages using breadcrumb tabs, while staying within the context of the same object.
8. Open the Salesforce interface in a new tab by clicking the **Back to Salesforce** link.

In each section, you can use the add icons to create a new item and the search icons to search for an existing item. When using the inline edit feature in the search page, if using Record Types, include a Record Type Filter for a single Record Type in the List View criteria because inline editing only works for lists that are filtered on a single record type.

Using Facets

When you find or add a new item, such as a product, attribute, or rule, the page for that item appears, and various facets, such as General Properties, Pricing, or Context Rules, are displayed on the left side as shown in the following example.



70% Off Family Channels And Free HD TV For 12 Months

Promotion Products

	PRODUCT	MINIMUM QUANTITY	MAXIMUM QUANTITY	QUANTITY	ACTION TYPE	CONTEXT PRODUCT	CONTEXT RULE	OFFER PRODUCT	DESCRIPTION
	Family Channels	1	1	1	Add			Quad Play Bundle Extra Large	
	HD TV	1	1	1	Add			Quad Play Bundle Extra Large	

These facets let you navigate between functional areas to make various configuration settings, changes, and additions.

NOTE
In the Vlocity Product Designer, only the General Properties facet is displayed and editable. See Manage Existing Object Types in Product Designer.

Navigating to the Vlocity Product Console

To use the Vlocity Product Console, you must use the Lightning interface.

1. Click the App Launcher and type `Vlocity Product Console` in the quick find box.
2. Click the Vlocity Product Console tab.

You can create product entities.

See Also

- Product Entities Overview
- Create Products in Vlocity Product Console

Product Lifecycles

Using Vlocity Shared Catalog, you can configure products at any level in the bundle, add new products, and importantly, maintain the product entities through their entire product lifecycle. Vlocity Shared Catalog supports the commercial product lifecycle by allowing you to define current, future, past and retired products.

You can consolidate products and services from different lines of business in commercial packages or bundles. Customers ordering such packages can take an advantage from special pricing applied to the bundles of products and from new functional capabilities created by combining different types of products or services.

- Current Product is a product that is currently available for sale.
- Future Product is a product that will be sold in the future. This product can be configured and ordered, but will be fulfilled once its selling period and fulfillment begins. In order to ensure that future products are not assetized until they are fulfilled, Future Date Ordering functionality must be implemented.
- Past Product is a product that is no longer for sale but still can be fulfilled for orders that have already been placed and changed for orders that have already been assetized.
- Retired Product is a product that is no longer supported, cannot be ordered or fulfilled, or changed for orders that have been assetized.

Vlocity Shared Catalog defines these stages of the product lifecycle using Product Selling Period dates. For more information, see the following topics.

This topic pertains to Vlocity Communications, Media, and Energy Fall '18 and later.

Product Selling Period Dates

Product Selling Period Dates allow you to set the selling period, the fulfillment start date, and a date to retire commercial products. The selling period is the time period between the selling start date and the selling end date of a product, and it controls when the product can be ordered or sold.

The fulfillment start date supports pre-orders for commercial products that are can be fulfilled at a later date. An end of life date is used to retire commercial products when the business decides to discontinue selling or supporting permanently.

Starting in Fall '18, you can create products in the shared catalog with the following dates:

- Selling Start Date(**vlocity_cmt__SellingStartDate__c**): This defines when the commercial product can be ordered and sold.
- Selling End Date (**vlocity_cmt__SellingEndDate__c**): This defines when the commercial product can be stopped from ordering. Changes to existing assets are supported after the end of selling and before it is retired.
- Fulfillment Start Date (**vlocity_cmt__FulfilmentStartDate__c**): This defines when the commercial product is ready to be fulfilled. By default, this is same as the selling start date but can be a future date.
- End Of Life Date (**vlocity_cmt__EndOfLifeDate__c**): The EOL date of a product is the date on which the commercial product is no longer supported.

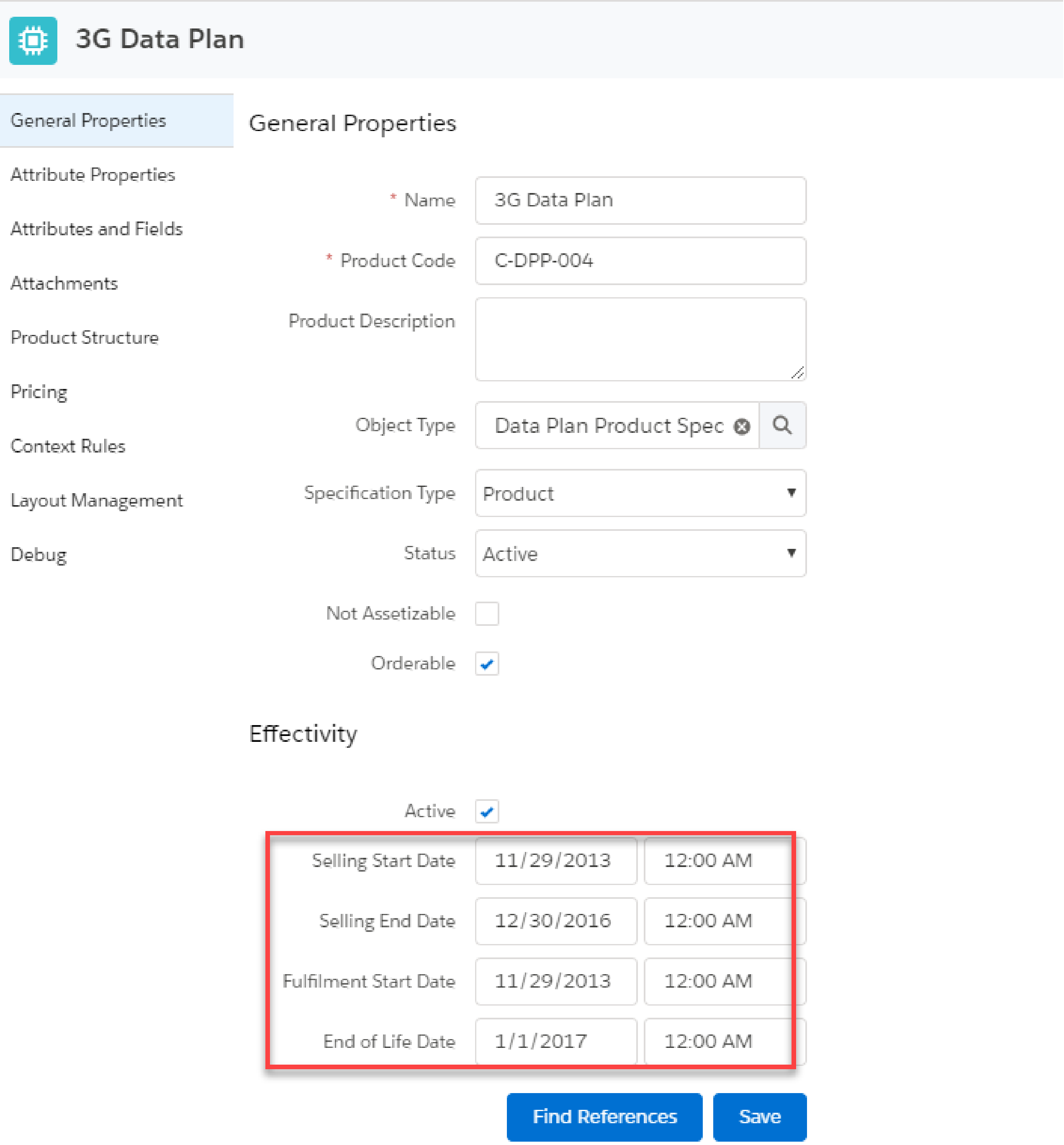
NOTE

These product selling period fields have a data type of Date/Time and are intended for commercial products only. Technical products and promotions do not use these fields and should continue to use the existing effectivity date fields. Assets do not use these fields.

This topic pertains to Vlocity Communications, Media, and Energy Fall '18 and later.

Product Selling Period Definition

You set the product selling period fields in the product dialog or in the **General Properties** facet for products in **Vlocity Product Console**.



3G Data Plan

General Properties

* Name	3G Data Plan
* Product Code	C-DPP-004
Product Description	
Object Type	Data Plan Product Spec
Specification Type	Product
Status	Active
Not Assetizable	□
Orderable	☑

Effectivity

Active	☑	
Selling Start Date	11/29/2013	12:00 AM
Selling End Date	12/30/2016	12:00 AM
Fulfilment Start Date	11/29/2013	12:00 AM
End of Life Date	1/1/2017	12:00 AM

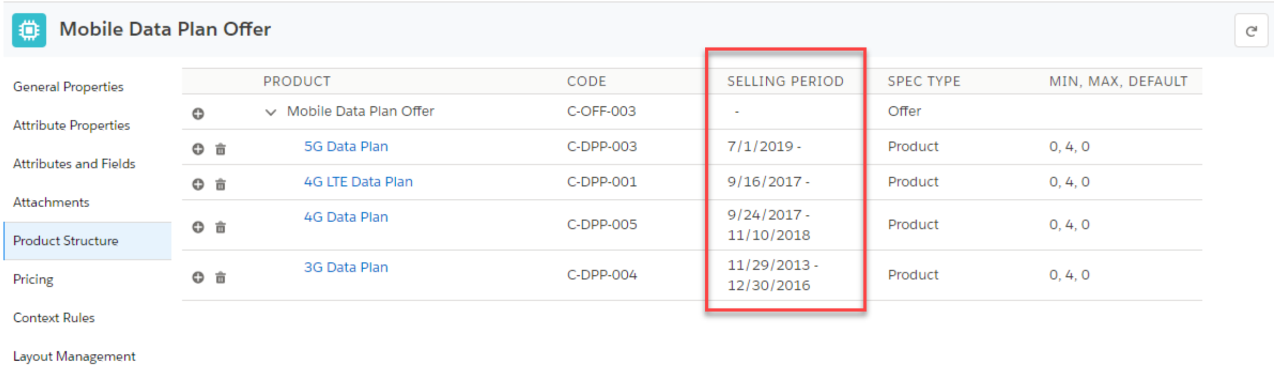
You can specify a date and a time for each field. If no time is specified, it will default to 12:00 AM. You also can choose to have no date or time, which implies that the product will be available for all of time, or you can set some of the dates but not all, as long as the following is true:

SellingStartDate <= FulfillmentStartDate <= SellingEndDate <= EndOfLifeDate

Vlocity Product Console validates all dates and times entered using this rule and will not allow you to save products until the dates are set correctly.

Product selling period data is stored in the platform cache. After you make changes to product selling period fields, you must update the product hierarchy data in the platform cache by running Product Hierarchy Maintenance, Clear Managed Platform Cache and Refresh Pricebook in this sequence.

When a product is used in bundles, it is the responsibility of the product administrator to ensure that the selling periods are consistent. To assist administrators, Vlocity Product Console has been enhanced to display the **Selling Period** dates in the **Product Structure** facet.



PRODUCT	CODE	SELLING PERIOD	SPEC TYPE	MIN, MAX, DEFAULT
Mobile Data Plan Offer	C-OFF-003	-	Offer	
5G Data Plan	C-DPP-003	7/1/2019 -	Product	0, 4, 0
4G LTE Data Plan	C-DPP-001	9/16/2017 -	Product	0, 4, 0
4G Data Plan	C-DPP-005	9/24/2017 - 11/10/2018	Product	0, 4, 0
3G Data Plan	C-DPP-004	11/29/2013 - 12/30/2016	Product	0, 4, 0

In addition, product administrators can use the **Find References** button in **Vlocity Product Console** to find references to a given product in active promotions, product bundles, assets, quote line items, order line items and opportunity line items. This button assists administrators in determining the effect of changing product selling period and end of life dates. For more information, see Finding Product References in the Shared Catalog.

This topic pertains to Vlocity Communications, Media, and Energy Fall '18 and later.

Product References in the Shared Catalog

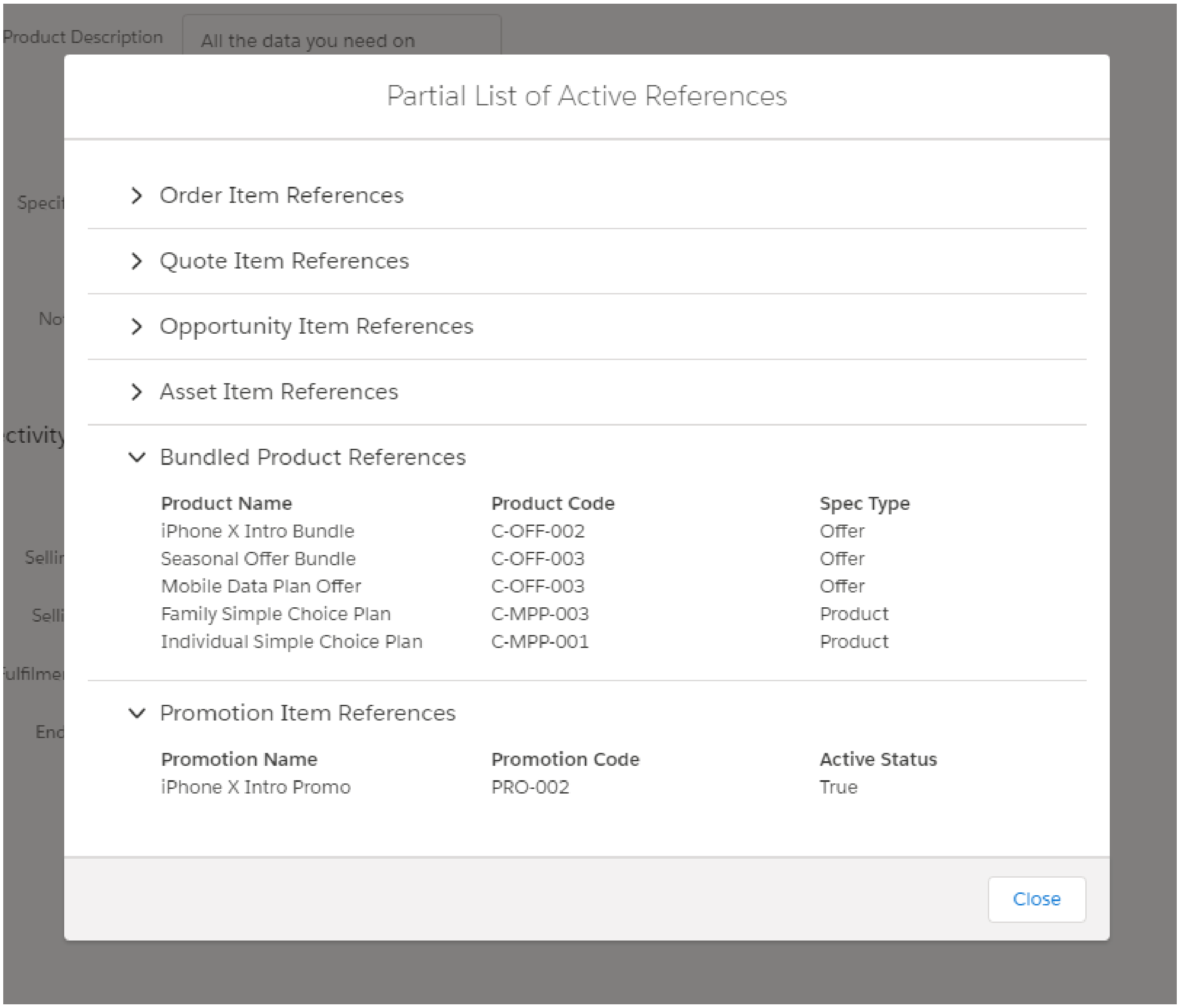
Use the **Find References** button in **Vlocity Product Console** to find references to a given product in active promotions, product bundles, assets, quote line items, order line items, and opportunity line items. This button assists you with determining the effect of changing product selling period and end of life dates.

This topic pertains to Vlocity Communications, Media, and Energy Fall '18 and later.

Using Find References

To use the **Find References** button, go to the **General Properties** facet for the product, and the button is located next to the **Save** button. Once clicked, the **Partial List of Active References** dialog box displays.

The default number of rows returned for each reference object is 20, but this limit can be increased by setting the **FindReferencesRowCountLimit** custom setting in CPQ Configuration Settings Overview. The maximum number of rows returned is 1000.



Partial List of Active References

Product Name	Product Code	Spec Type
iPhone X Intro Bundle	C-OFF-002	Offer
Seasonal Offer Bundle	C-OFF-003	Offer
Mobile Data Plan Offer	C-OFF-003	Offer
Family Simple Choice Plan	C-MPP-003	Product
Individual Simple Choice Plan	C-MPP-001	Product

Promotion Name	Promotion Code	Active Status
iPhone X Intro Promo	PRO-002	True

To find order, quote, and opportunity line item references, each line item's fulfillment status (**vlocity_cmt__FulfilmentStatus__c**) must be set to Draft, In Progress, PONR, or Pending. In order to find asset references, the asset's provisioning status (**vlocity_cmt__ProvisioningStatus__c**) must be set to Active, Changed, or Deleted. These values are normally set by order management processes. If the statuses have null or other values, the references will not display.

Product Selling Period Dates and Vlocity Cart

Vlocity Cart supports product selling periods. At runtime, users can see products in the Products list and in the Vlocity Cart based on the Order Start Date.

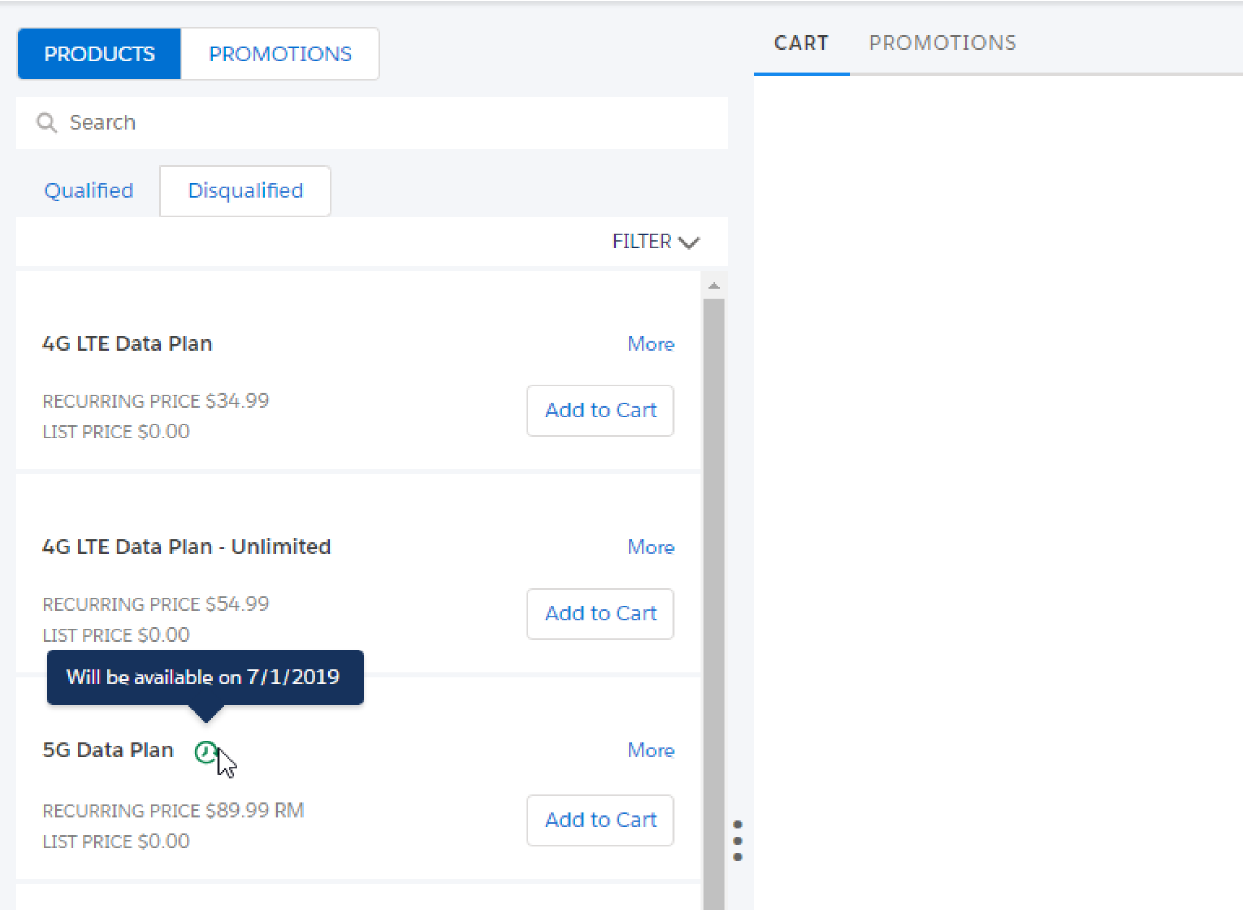
This topic pertains to Vlocity Communications, Media, and Energy Fall '18 and later.

Product Selling Period Dates and the Products List

The Products list selection pane in Vlocity Cart now displays current and future products based on the Order (or Quote or Opportunity) Start Date (**Order.EffectiveDate**).

Product selling period functionality does not override other standard controls for displaying products in the Products list. In order to display, products must be marked Orderable, Active, and have a price list entry. The price list entry **Effective From** date must be current as well.

- Current products are products with dates that are within the selling period based on the Order **Start Date** or Quote/Opportunity **CreatedDate**. Products that have reached the end of their selling period or are at their end of life date will not display in the product list and can't be added to the cart.
- Future products with a **Selling Start Date** that is later than the Order **Start Date** or Quote/Opportunity **CreatedDate** also display in the product list, and they are indicated with a green clock icon. You can hover over the icon and see the **Selling Start Date** or the product will be available. If you enter a **Fulfillment Start Date**, you can add the future product to the cart and submit the order normally. Future Dated Ordering functionality must be implemented in order to manage the fulfillment of future products.



4G LTE Data Plan
More
RECURRING PRICE $34.99
LIST PRICE $0.00
Add to Cart

4G LTE Data Plan - Unlimited
More
RECURRING PRICE $54.99
LIST PRICE $0.00
Add to Cart

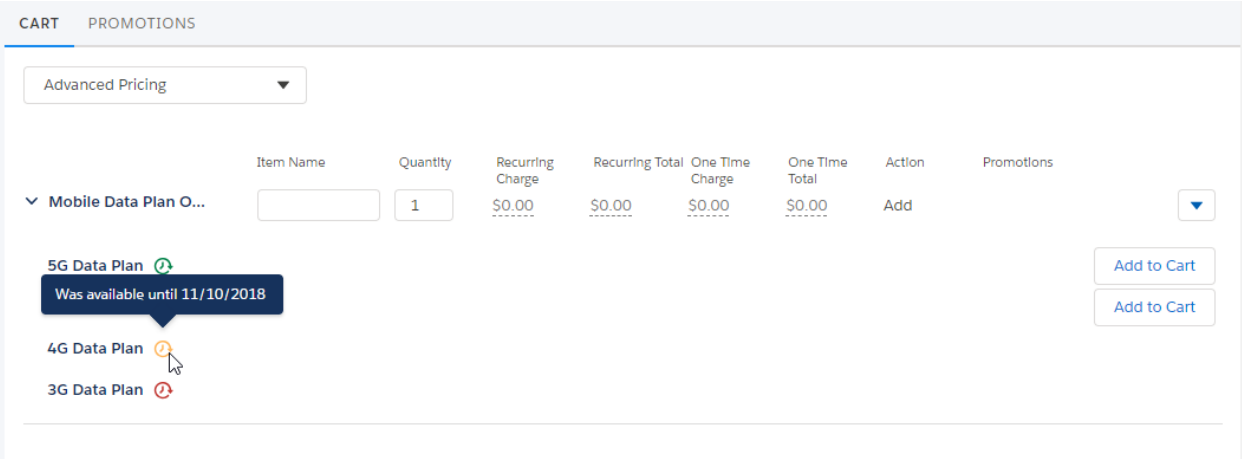
Will be available on 7/1/2019

5G Data Plan
More
RECURRING PRICE $89.99 RM
LIST PRICE $0.00
Add to Cart

Product Selling Period Dates and the Cart

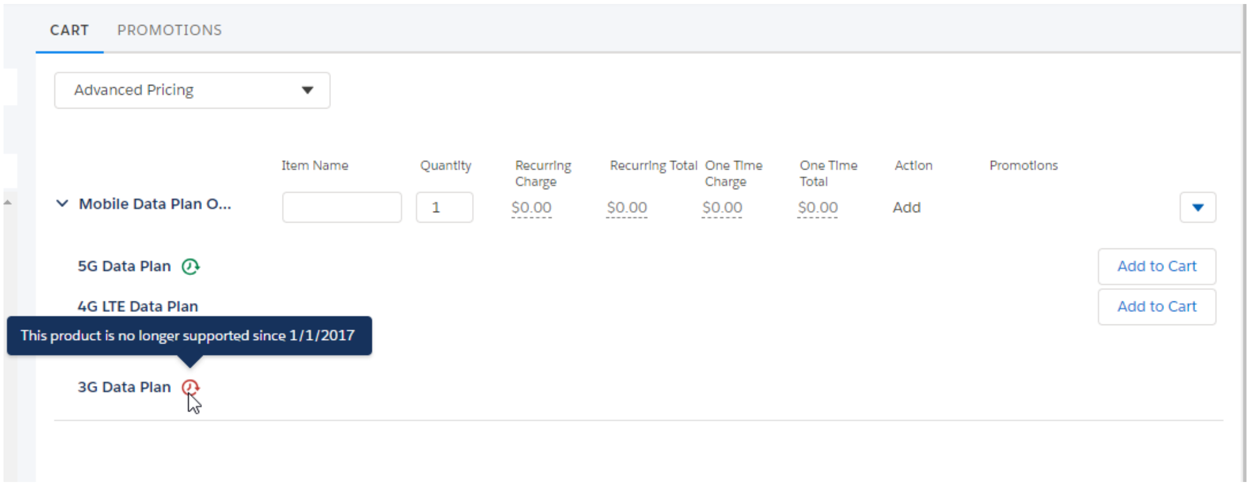
To support *MACD* and asset-based ordering (ABO), Vlocity Cart can display current, future, past and retired products based on the Order (or Quote or Opportunity) Start Date (**Order.EffectiveDate**).

- Current and future products display in the cart normally, but future products are indicated with a green clock icon like they are in the Products list. Both can be added to the cart and configured normally.
- Past products are products that have passed their **Selling End Date** but have not passed their **End of Life** date. They are indicated by a yellow clock icon. They cannot be added to the cart or configured for new orders, but they can be configured or deleted during MACD or asset-based ordering. You can hover over the icon to see the **Selling End Date**.



Was available until 11/10/2018

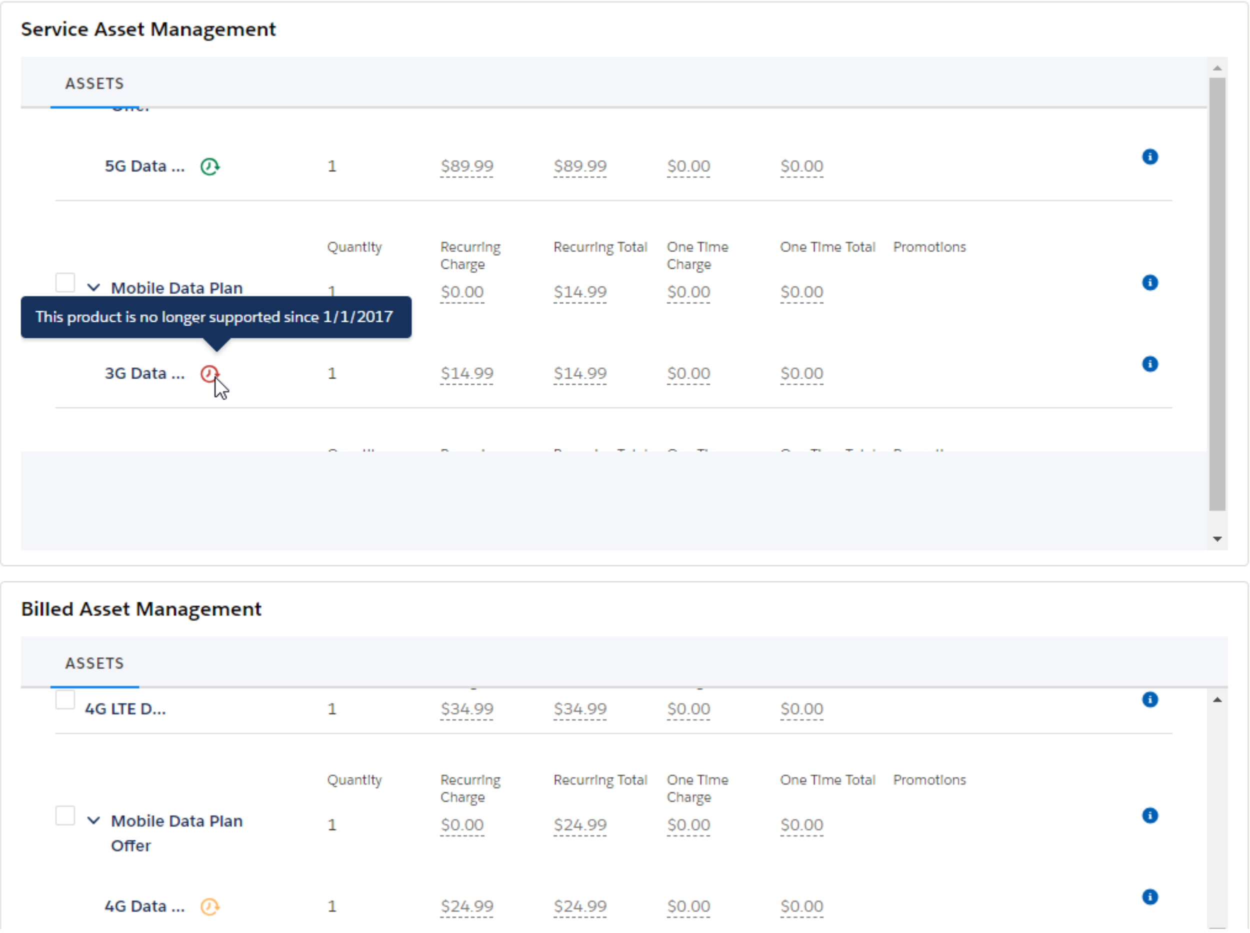
- Retired products are products that have passed their end of life date, and they are indicated by a red clock icon. In a new order, they cannot be added to the cart or configured, but during MACD or asset-based ordering, they can be deleted. You can hover over the icon to see the date the **End of Life** date.



This product is no longer supported since 1/1/2017

Product Selling Period Dates and MACD / Asset-based Ordering

Asset Management card layouts display icons for future, past, and retired products. Hover over each icon to see the exact date. Regardless of their state, you can select all assets for **Change to Quote**, **Change to Order**, and **Move** operations. When they are in the Cart, the Cart rules noted above apply.



Service Asset Management

ASSETS

	Quantity	Recurring Charge	Recurring Total	One Time Charge	One Time Total	Promotions
5G Data ...	1	$89.99	$89.99	$0.00	$0.00	
Mobile Data Plan	1	$0.00	$14.99	$0.00	$0.00	
3G Data ...	1	$14.99	$14.99	$0.00	$0.00	

This product is no longer supported since 1/1/2017

Billed Asset Management

ASSETS

	Quantity	Recurring Charge	Recurring Total	One Time Charge	One Time Total	Promotions
4G LTE D...	1	$34.99	$34.99	$0.00	$0.00	
Mobile Data Plan Offer	1	$0.00	$24.99	$0.00	$0.00	
4G Data ...	1	$24.99	$24.99	$0.00	$0.00	

Upgrade to Product Selling Period Dates

If your deployment is on an earlier release of Vlocity Communications, Media, and Energy, you must upgrade to Fall '18 before you can set up your shared catalog to use product selling period dates.

To set up your shared catalog:

1. Update existing products in your catalog to use the new fields. The previous effectivity dates will not be deleted. Use the Populate Product Selling Period batch job to copy the values from the existing Effective Date field to the new Selling Start Date(time) and the existing End Date field into the Selling End Date(time). For example, a product could have an Effective Start Date of 7/7/2018. After running the batch job, this date is converted to the Selling Start Date, and it is a DateTime of 7/7/2018 00:00:00.

NOTE
The Populate Product Selling Period batch job processes all products, both commercial and technical, in your shared catalog.

2. Add the new fields to your existing commercial product object type layouts. For more information, see Adding New Product Selling Period Fields to Object Types in the Shared Catalog.
3. Update existing customizations or integrations. Customers should ensure that any customizations that evaluate the legacy effectivity date fields are updated to use both the new product selling period fields and the DateTime data type. The legacy effectivity dates continue to be supported and will automatically be kept in sync with the new product selling period dates for backward compatibility. New installations will receive the new product selling period dates automatically and no conversion is necessary.

Product Entities Overview

Before creating products in Vlocity Product Console, you must understand the following product entities, their dependencies, and the order in which you must create them.

Product Object Types

Product object types help you streamline the creation of your product catalog by creating classes of products that have similar characteristics. A product object type is a reusable entity that describes a common set of fields and attributes as well as the layouts for all linked products. You can use object types to group products with similar characteristics, to ensure consistent behavior and application of rules.

Object types are designed with an IS-A inheritance type architecture. For those of you familiar with object-oriented programming and design, you recognize that object types create relationships between abstractions where one object type is a subtype of another object type. The first level of abstraction is called a base object type, and the subsequent levels are called subtypes. Then, when you apply this relationship to a product, the object type allows the product to inherit all of the characteristics (attributes, fields, and layouts) of its object type hierarchy.

For example, a company may offer a set of different types of products: electronics, clothing, and furniture. Products in each category share similar characteristics that are not applicable to products in a different category; such as screen size and network speed are not applicable to clothing or furniture.

Instead of creating a separate electronic product, you can define the Electronics object type that contains all characteristics related to electronics. All child object types of Electronics inherit these properties. You can modify the properties of child object types by adding new properties. A Smartphone object type can be a child of the Electronics object type. It inherits all properties from Electronics but also has additional properties specific to Smartphones: IMEI Number or Bluetooth Capability.

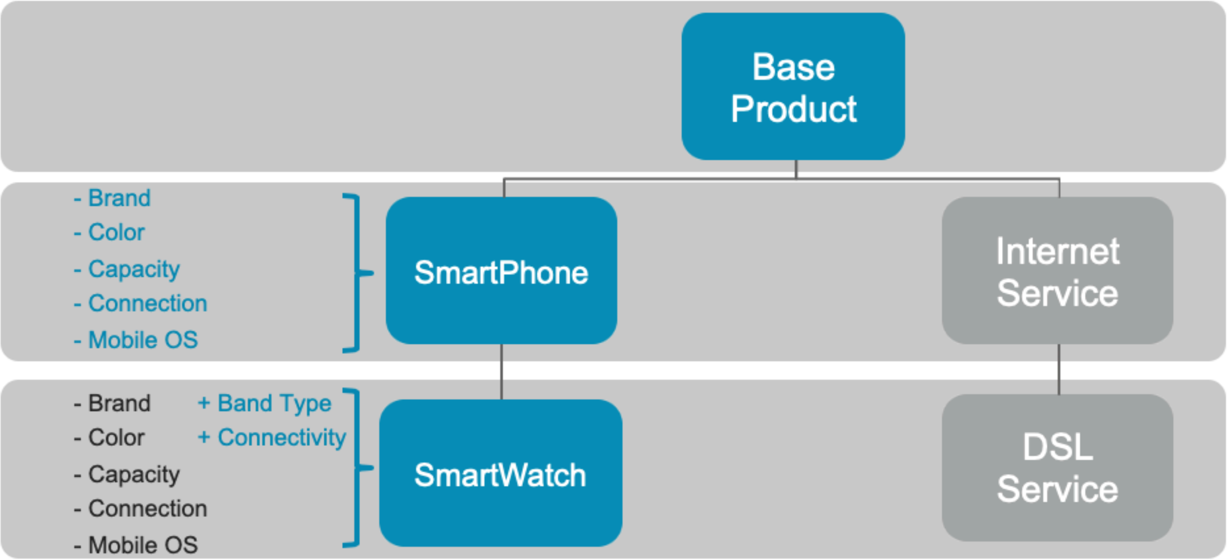
After you create an object type, like a smartphone, you can instantiate it to create a hundred smartphone product specifications to ensure consistent behavior and application of rules, such as fulfillment rules.

Object types are created during an initial setup by a system administrator. For instruction on how to create an object type, refer to Create an Object Type.

Inheritance Within the Object Type Hierarchy

A product object type is a reusable entity that has a common set of fields, attributes, and layouts for all the linked products. Vlocity object types are designed with an inheritance type architecture where a child object type inherits the layout of the parent product, along with all the associated fields and attributes.

In the object type hierarchy, attribute inheritance is dynamic where new attributes and fields assigned to an object type are inherited dynamically by the nested object types. The following image illustrates the attribute inheritance in the object type hierarchy. In this example, the SmartPhone attributes, Brand, Color, Capacity and so on are inherited by the subtype SmartWatch.



You should not delete the inherited attributes and fields from the nested object types. Instead, you must unassign or deactivate them in the parent object type.

The object type layouts use a different inheritance model. Layouts help you organize attributes, fields, and custom elements into facets and sections. When you create a new nested object type, Vlocity Product Console creates a new layout for the nested object type via a deep copy. The new layout is a copy of the parent object type layout and contains all the associated fields and attributes. However, any subsequent changes to the parent object type layout will not be replicated down the hierarchy. That is, the nested object type layouts remain unchanged. This allows you to modify the object type layout, without having to worry about the nested object type layouts being altered with the addition or deletion of fields from the parent object type layout.

Layout Inheritance within the Object Type Hierarchy

Similar to Salesforce page layouts, object type layouts allow you to control the layout and organization of an object type's attributes and fields. New objects subtypes and products inherit the layout from the object that they're associated with.

Object Type Layouts

Using object type layouts, you can customize layouts for each class of product, so that the attributes and fields displayed at design-time are different. For example, displayed attributes and fields for a SmartPhone are different than for a backpack.

An object type layout is made up of facets, sections, and section elements:

- A facet appears on the left navigation menu, and it contains sections and section elements.
- A section is essentially a container for a Visualforce page, and a section can have either a standard field layout view type or a custom view type. Vlocity provides several common custom view types, and customers can create their own custom view types by creating custom Visualforce pages.
- Section elements are fields or attributes that are grouped in sections.

Layout Inheritance

Unlike the attribute and field dynamic inheritance model, object type layouts use a “hybrid” inheritance model. When a new object subtype is created, Vlocity Product Console creates a new layout for the object subtype via a deep copy. What this means is that any subsequent changes to the object supertype’s layout are not replicated down the hierarchy after initial creation. This allows you to change the layout, adding or deleting fields for each object type, without having to worry about your layouts being altered with the additions of any fields on the object supertype.

Product Object Types Best Practices

Vlocity recommends creating a base object types for each layer of your SID model (Offer Specs, Product Specs, Service Specs and Resources Specs) during initial setup. These are essentially your product catalog’s “primordial objects” from which all other products descend.

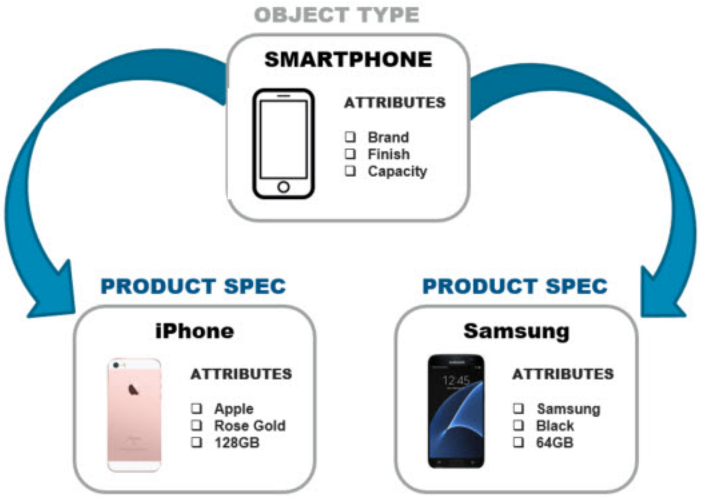
You assign all of the fields in the product catalog to the base product object type, and then you can create object subtypes for all the product classes and products in your catalog. They will then inherit the base product’s layout, fields, and any attributes.

The base object type layouts must be created manually, all the subsequent object subtypes inherit the layout when they are created.

At this time, to ensure data integrity, it is not recommended to change the object type hierarchy once it has been defined by changing parent object type for a child. Instead, delete the child object type and recreate it under the new parent.

Product Specifications

A product (or product specification) is a fundamental entity for which you define parameters (attributes or fields), such as bundling, pricing rules, fulfillment rules, and compatibility rules.



Before you can define a product specification, you must have a corresponding object type. A product specification inherits all the attributes of the object type.

For example, iPhone has all attributes defined for the smart phone object type (such as brand, finish, and capacity).

You can create many product specs from the same object type, but each product specification can only be associated with one object type.

The specification types are:

- Product Specification
  An example of a product specification is a Smart phone with parameters, such as phone model, color, size, etc.
- Service Specification
  An example of a service specification is an internet service for which you can define parameters, such as upload or download speed, broadband type (ADSL or Fiber), and so on.
- Resource Specification
  Resource specification defines resources needed to implement the Customer Facing Services (CFS). For example, IP port address, or a router.

Vlocity managed package Product2 record types are:

- Product: Use for Vlocity products.
- Product Template: Use for Vlocity product templates to ease creation of similar products.
- Class: Not in use, but part of the Vlocity package.

Vlocity Picklists

If an attribute includes multiple values, a picklist represents all of the defined values for that attribute. A picklist is an ordered list of values that customers can choose from when buying a product. You must define picklists before creating attributes.

For example, iPhone has three attributes, finish, capacity, and display. Each of these attributes has several possible values, such as:

- Phone finish (Rose Gold, Gold, Silver, Black, etc.)
- Capacity (32GB, 128GB, 256GB)
- Display (4.7-inch, 5.5-inch)

You must define a separate picklist for each of the three attributes. When defining a picklist, you can optionally nominate a default value for each picklist.

Picklists in Vlocity EPC perform a similar function as Salesforce picklists, but they are custom objects called Vlocity Picklists (API name: `vlocity_cmt__Picklist__c`).

Picklists can be created with the following data types:

- Text
- Number
- Boolean
- Date
- Datetime

Picklists are designed to be globally reusable across attributes assigned at the object type level or at the product level.

IMPORTANT

Picklist attribute values are stored as text, even if the picklist you select has another **Data Type**, such as **Number**. If you use a Picklist attribute in a Set Value or Set Default Value rule, verify that the rule expresses the attribute's data type as text for the Value to Set and in the rule's Expression box.

For instructions on how to create a picklist, see Create a Picklist.

Product Attributes

Products may or may not have attributes, which you can use in Vlocity CPQ to customize a product, to filter products to find the right one for customers, or to set pricing for a product.

Products may or may not have Product Attributes. A product that does not have attributes comes only in one form, and customers do not make any decisions about the features of the product. For example, a headset or a computer cord that come in one color.

If a customer can choose from a selection of product features, you must define these features as attributes before creating an object type. Also, if an attribute includes multiple values that a customer can choose from, you must define picklists before defining the attribute.

Product attributes provide a powerful and flexible way to extend product entities. Product attributes are stored in an aggregate JSON blob on the Product object and are used extensively throughout Vlocity CPQ and Vlocity Order Management.

In Vlocity CPQ, you can use product attributes to filter products at runtime, displaying the best fit products to customers. You can use them to configure product specifications, allowing users to specify product colors, sizes, and other options. You can also use attributes to change product pricing using attribute-based pricing rules.

Finally, in Vlocity Order Management, you can use attributes to map commercial products, or the products that customers see, with technical products that communicate with backend systems to fulfill the products that are purchased.

Attributes vs. Fields

When defining product properties, you can use attributes or fields. The choice depends on how you plan to implement property data during order capture.

- *Attributes*: A list of all attributes and their associated Attribute Codes assigned to the Product2 Object.
  If a data element, such as a SKU number, is intended to be common to all products across your product catalog, then you should create a new field on your product object. Typically, adding new fields requires Salesforce system admin privileges.
- *Fields*: A list of all fields and their associated Field API name for the Product2 Object.
  If the data element is intended to be specific to a product, or class of products, then you should create an attribute, such as download speed. At runtime, attributes are visible to customers in Vlocity Cart and can be used to filter the catalog and provide configuration capabilities. Adding attributes requires access to the Product Console.

The fundamental differences between fields and attributes are:

	Fields	Attributes
Where the value is created?	Created in Salesforce Setup on a Salesforce object. Field example: Orderable.	In Vlocity attribute/category interface. Attribute example: Color.
Where the value is stored (including metadata, such as default value, picklist, etc.)?	Each field value is stored in a separate Salesforce field. Related metadata is stored separately in Salesforce. For example, if you create 15 fields, there will be 15 field records in Salesforce and each record will store its value.	Each attribute value and its' metadata (including picklists, default value, etc.) is stored in a JSON blob (i.e. in a FIELD in JSON). For example, if you create 15 attributes, all 15 attributes and all their metadata will be stored in one JSON blob which is stored in one Salesforce field.

	Fields	Attributes
Who can create?	Requires Salesforce System Administrator privileges.	A Product Manager with sufficient privileges.
Is searchable at runtime?	Each field is stored in a separate Salesforce database field and is easily searchable.	Since all attribute data is stored in one JSON blob, expect performance implications when using search.
When to use at design-time to define Product2 properties	Use for commonly used properties that are searchable and runtime (order time) configurable.	Use for properties that are sparely used.
When to use at runtime to define asset properties	Do not use.	Use attributes only. For example, IMEI number, VIN #, etc.

JSONAttribute v2 Schema Upgrade Considerations

The upgraded attribute schema provides multiple performance and storage enhancements. For example, the new schema extracts values from the vlocity_cmt__JSONAttribute__c field into separate fields: Metadata components, default runtime values, and metadata changes.

This separation allows metadata to remain on the design-time object (the product) while runtime values and metadata changes are maintained on the runtime object (the XLIs), which results in smaller XLI payloads. The data shape of the XLI attribute fields is aligned with the API response, which results in fewer translation operations and better runtime performance.

For deployments prior to the Fall '18 release, you can run an upgrade process to update product attributes to the latest schema. However, any custom Apex code, configurations, and DataRaptors that depend on the vlocity_cmt__JSONAttribute__c field must be updated to use the new fields. The new schema is optional: the previous JSONAttribute field will continue to be supported and your existing processes can remain the same.

⚠ **CAUTION**

The migration is one-way only; the action is not reversible. Vlocity does not provide utilities to migrate back to the old schema model.

Custom implementations using the JSONAttributeSupport class must use the new v2 methods outlined in JSONAttributeSupport.

The Asset Attribute Visualforce component on the Asset page is not supported by the v2 attribute model. If your org needs the Assets Attribute component, it cannot be upgraded to the v2 attribute model. The Card Assets Visualforce component will support this functionality for orgs using the v2 attribute model.

The following table provides the newly added fields by object:

Object	Field	Description
Product2	AttributeMetadata__c	Stores productAttribute JSON without default values.
Product2	AttributeDefaultValues__c	Stores the default values for product attributes.
OpportunityLineItem	AttributeSelectedValues__c	Stores the attribute code and the value selected by the user.
OpportunityLineItem	AttributeMetadataChanges__c	Stores the attribute code and any metadata changes from the default product configuration.
OrderItem	AttributeSelectedValues__c	Stores the attribute code and the value selected by the user.
OrderItem	AttributeMetadataChanges__c	Stores the attribute code and any metadata changes from the default product configuration.
QuoteLineItem	AttributeSelectedValues__c	Stores the attribute code and the value selected by the user.
QuoteLineItem	AttributeMetadataChanges__c	Stores the attribute code and any metadata changes from the default product configuration.
Asset	AttributeSelectedValues__c	Stores the attribute code and the value selected by the user.
Asset	AttributeMetadataChanges__c	Stores the attribute code and any metadata changes from the default product configuration.
CompiledAttributeOverride__c	AttributeDefaultValues__c	Stores the attribute code and the value selected by the user.
CompiledAttributeOverride__c	AttributeMetadataChanges__c	Stores the attribute code and any metadata changes from the default product configuration.
EntityFilter__c with attribute set as the entity filter condition	AttributesMetadataFieldName__c	• The value for OrderItem/OpportunityLineItem/ QuoteLineItem is PriceBookEntry.Product2.{*namespace*}__AttributeMetadata__c. • The value for Product2 is {*namespace*}__AttributeMetadata__c.
EntityFilter__c with attribute set as the entity filter condition	AttributesValuesFieldName__c	• The value for OrderItem/OpportunityLineItem/ QuoteLineItem is {*namespace*}__AttributeSelectedValues__c. • The value for Product2 is {*namespace*}__AttributeDefaultValues__c.

API Considerations

Prior to the Fall '18 release, all of the Cart-Based APIs return the new JSON attribute structure. Any changes in the API services are transparent and require no modifications or updates. Any external integration processes that directly access or modify the pre-Fall '18 JSON fields will be impacted.

Starting with the Winter '20 release, the Digital Commerce APIs support the JSONAttribute v2 schema.

Upgrading the Attribute Schema

The schema upgrade steps are listed in Upgrading the Attribute Schema.

Encrypted and Not Assetizable Attributes

Starting in the Fall '18 release, attributes for technical products can now be set as not assetizable (to identify temporary data) and encrypted (to protect personally identifiable information or similar data).

Is Not Assetizable

The **Is Not Assetizable** attribute metadata flag is useful for transitory information that shouldn’t persist or doesn’t need to persist beyond completion of order management where assetization would usually occur. This metadata flag is stored in Order Management, not in the Salesforce platform.

An example of transitory information is the initial password that a customer uses to log in to the provider's website for the first time. During fulfillment, the initial password is generated within an orchestration order item, then saved on the technical fulfillment request line, which is then passed to the next orchestration item that emails the password to the customer. The password doesn’t need to persist at the end of order management. Other information that doesn’t need to persist can be the result of back-end services connected to order items that don’t become customer assets, such as a service needed to fulfill a customer’s order. As an example, there might be a check to see whether a required modem is in stock and on the delivery vehicle. This information is not needed after that check has been made.

Is Encrypted

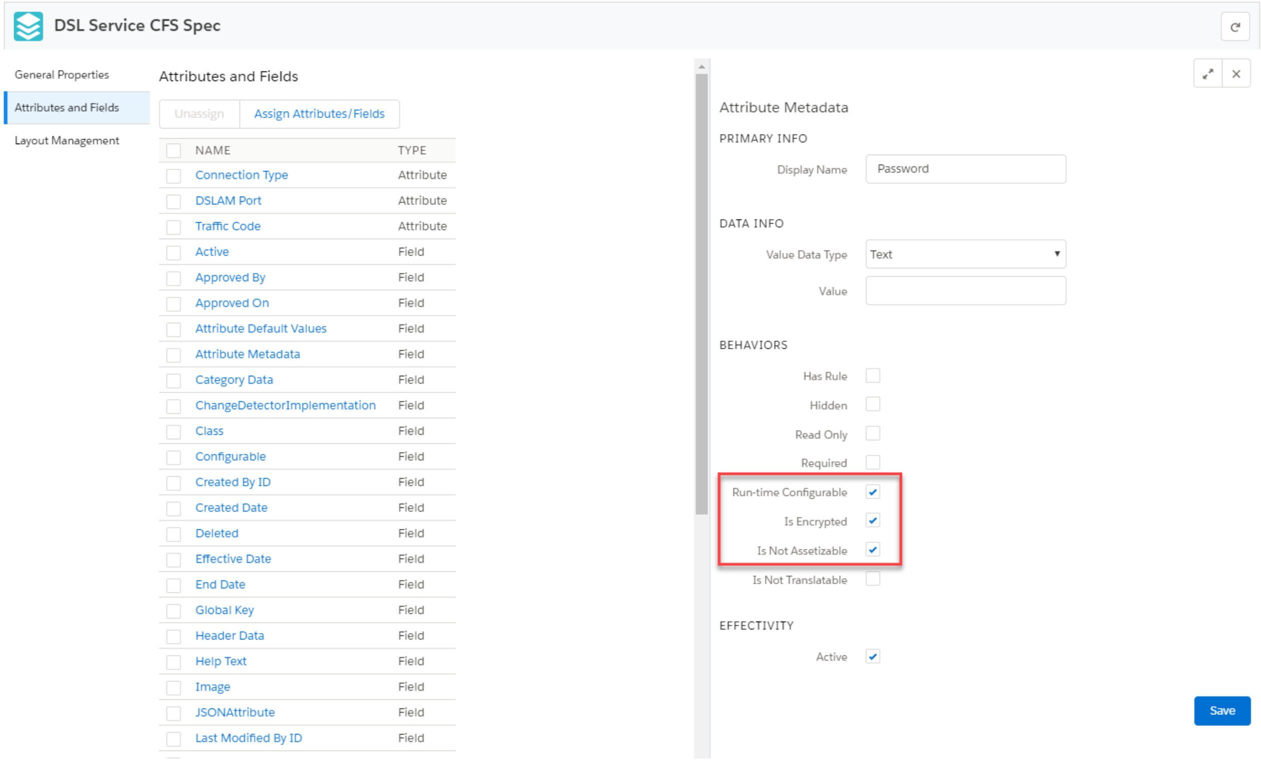
The **Is Encrypted** attribute metadata flag is used to protect personally identifiable information (PII) and other data such as the customer name, phone number, social security number, and so on. These pieces of information can be captured as attributes and should be sent downstream only for legitimate purposes. You can use the encrypted flag to indicate the PII or other information must be stored securely.

Setting Is Not Assetizable and Is Encrypted

Is Not Assetizable and **Is Encrypted** are attribute metadata flags. You set these flags when an attribute is assigned to an object type in Vlocity Product Designer or Vlocity Product Console. These flags can only be set when the **Run-time Configurable** flag is also set. After the attribute is marked as **Run-time Configurable**, you can then set **Is Not Assetizable** or **Is Encrypted** or both.

If none are selected, attribute values are stored in the technical inventory after decomposition. If both are selected, the attribute is encrypted during processing and not stored with the inventory items. However, if you set the **Is Encrypted** flag to True, and **Is Not Assetizable** to False, the attribute is stored in the technical inventory, but the attribute is encrypted. Technical assetization stores the output of decomposition into the technical tables in Order Management, not in the Salesforce platform.

Figure 2. Setting Is Not Assetizable and Is Encrypted Attribute Metadata



DSL Service CFS Spec

Attributes and Fields

NAME	TYPE
Connection Type	Attribute
DSLAM Port	Attribute
Traffic Code	Attribute
Active	Field
Approved By	Field
Approved On	Field
Attribute Default Values	Field
Attribute Metadata	Field
Category Data	Field
ChangeDetectorImplementation	Field
Class	Field
Configurable	Field
Created By ID	Field
Created Date	Field
Deleted	Field
Effective Date	Field
End Date	Field
Global Key	Field
Header Data	Field
Help Text	Field
Image	Field
JSONAttribute	Field
Last Modified By ID	Field

Attribute Metadata

PRIMARY INFO

Display Name: Password

DATA INFO

Value Data Type: Text

Value:

BEHAVIORS

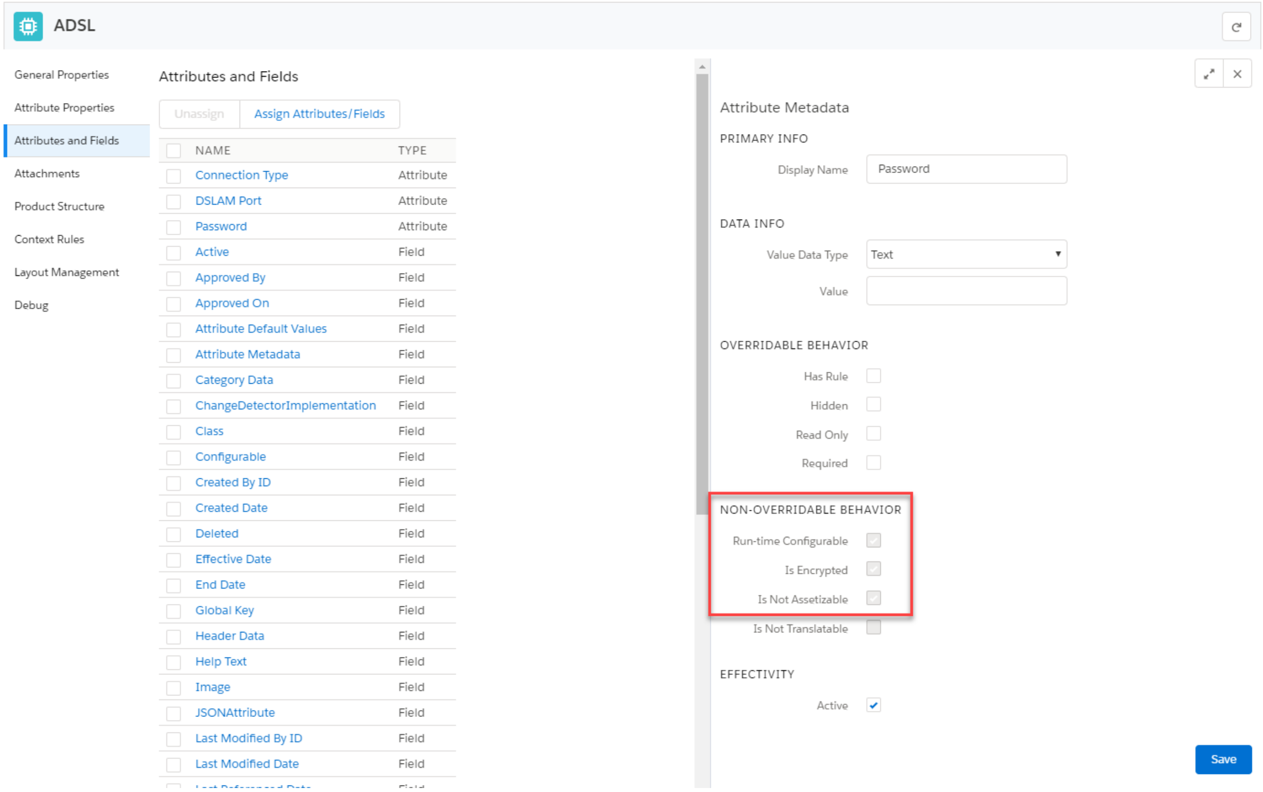
Has Rule □
Hidden □
Read Only □
Required □
Run-time Configurable ☑
Is Encrypted ☑
Is Not Assetizable ☑
Is Not Translatable □

EFFECTIVITY

Active ☑

After assigned to the object type, every product that is created in that object type inherits these metadata flags, with the other attribute and metadata. **Is Not Assetizable**, **Is Encrypted** and **Run-time Configurable** are then non-overridable at the product level. This prevents any product manager or EPC administrator from removing these flags during product creation.

Figure 3. Non-Overridable Attribute Metadata on Products



ADSL

Attributes and Fields

□	NAME	TYPE
□	Connection Type	Attribute
□	DSLAM Port	Attribute
□	Password	Attribute
□	Active	Field
□	Approved By	Field
□	Approved On	Field
□	Attribute Default Values	Field
□	Attribute Metadata	Field
□	Category Data	Field
□	ChangeDetectorImplementation	Field
□	Class	Field
□	Configurable	Field
□	Created By ID	Field
□	Created Date	Field
□	Deleted	Field
□	Effective Date	Field
□	End Date	Field
□	Global Key	Field
□	Header Data	Field
□	Help Text	Field
□	Image	Field
□	JSONAttribute	Field
□	Last Modified By ID	Field
□	Last Modified Date	Field

Attribute Metadata

PRIMARY INFO

Display Name: Password

DATA INFO

Value Data Type: Text ▾

Value:

OVERRIDABLE BEHAVIOR

Has Rule □

Hidden □

Read Only □

Required □

NON-OVERRIDABLE BEHAVIOR

Run-time Configurable □

Is Encrypted □

Is Not Assetizable □

Is Not Translatable □

EFFECTIVITY

Active ☑

If you set these flags on existing attributes, it does not affect attributes that are already stored in existing technical inventory.

Is Not Assetizable and **Is Encrypted** metadata flags are intended to trigger functions in downstream order management systems, such as Vlocity Order Management Plus. There is no encryption to these attributes when used on commercial products in Vlocity CPQ or in Vlocity Order Management, and the information remain stored and accessible on the originating order. Consequently, these attribute flags are intended for use in technical products only.

Using Is Not Assetizable and Is Encrypted

Once these attribute metadata flags have been set on an object type and inherited by a product, you must create decomposition rules to map the attributes to the downstream fulfillment requests, which are the basis for the technical inventory. See Partial Assetization of Product Attributes.

Fix Duplicate and Empty Attributes

You can see empty attributes if you create an attribute for the Product2 object type and delete it. This deletion can create an empty attribute assignment under a child object type of Product2. Duplicate attributes can occasionally occur. You can run the following scripts to resolve these issues.

1. From **Setup**, click **Developer Console**.
2. From the **Debug** menu, click **Open Execute Anonymous Window**.
3. Enter the appropriate code, and click **Execute**.

Issue	Resolution

Empty attributes	`List<vlocity_cmt__ObjectFieldAttribute__c> ofaList = [SELECT Id FROM vlocity_cmt__ObjectFieldAttribute__c WHERE vlocity_cmt__AttributeId__c = null AND vlocity_cmt__FieldApiName__c = null];`
	`delete ofaList;`

Duplicate attributes	`List<vlocity_cmt__ObjectFieldAttribute__c> ofaList = [SELECT Id,vlocity_cmt__AttributeId__c FROM vlocity_cmt__ObjectFieldAttribute__c WHERE vlocity_cmt__ObjectClassId__c = 'a214x0000000BzQAAU' AND vlocity_cmt__SubClassId__c = null];`
	`Set<Id> attrIdSet = new Set<Id>();`
	`List<vlocity_cmt__ObjectFieldAttribute__c> ofaListToDelete = new List<vlocity_cmt__ObjectFieldAttribute__c>();`
	`for(vlocity_cmt__ObjectFieldAttribute__c ofa:ofaList)`
	`{`
	`    if(attrIdSet.Contains(ofa.vlocity_cmt__AttributeId__c))`
	`    {`
	`        ofaListToDelete.add(ofa);`
	`    }`
	`    attrIdSet.add(ofa.vlocity_cmt__AttributeId__c);`
	`}`
	`delete ofaListToDelete;`

Product Bundles

A product bundle is a set of products sold together. Products are bundled using a “Has-A” Containment relationship as either a child or a realization. A product can have one or more product child items (or PCIs), and you can define cardinality for each. PCIs can be a single product or a product bundle, which allows you to create product bundles within product bundles.

Types of Product Line Items in a Bundle

There are two key types of product line items in a bundle hierarchy:

- Root product line item is the top-level line item. Root does not have a parent, therefore the ParentItemId is always empty.
- Product Child Item (PCI) defines the relationship between a parent product and its children.

Product Cardinality

Product cardinality defines the default, minimum and maximum quantities of the product that are allowed in the product bundle. Most order line items have a minimum and maximum value allowed, which Vlocity Cart enforces in the Quantity field for each line item. In the Vlocity EPC database, the product data model defines the cardinality of product child items, which are displayed as order line items in Vlocity Cart. Cardinality is like a rule that ensures you specify the right quantities in the cart: not too many, and not too few.

For example, a business account may be allowed to purchase up to two Complete TV products, with the purchase of the product optional. In this case, the underlying Vlocity EPC data model defines the following cardinality for the Complete TV product, where the minimum cardinality is 0 and the maximum is 1. With this cardinality defined, if you try to specify a quantity of "3", the cart notifies you that you can only add up to "2" Complete TV products.

You can override the cardinality of product bundles when they are included within another product bundle.

As a product manager, you can define two types of cardinality:

PCI cardinality (quantities of the product that are allowed in the product bundle)	Min Quantity: The sum of the quantities of the instances of a PCI cannot be less than the value defined in Min Quantity.
	Max Quantity: The sum of the quantities of the instances of a PCI cannot be greater than the value defined in the Max Quantity.
	Default Quantity: The value of the Quantity field when a PCI is added as a Line Item.
Group cardinality (defines the minimum and maximum allowable quantities across an item's immediate children)	Min Children Quantity: The sum of the quantities of an item's children cannot be less than the value defined in Min Children Quantity.
	Max Children Quantity: The sum of the quantities of an item's children cannot be more than the value defined in Max Children Quantity.

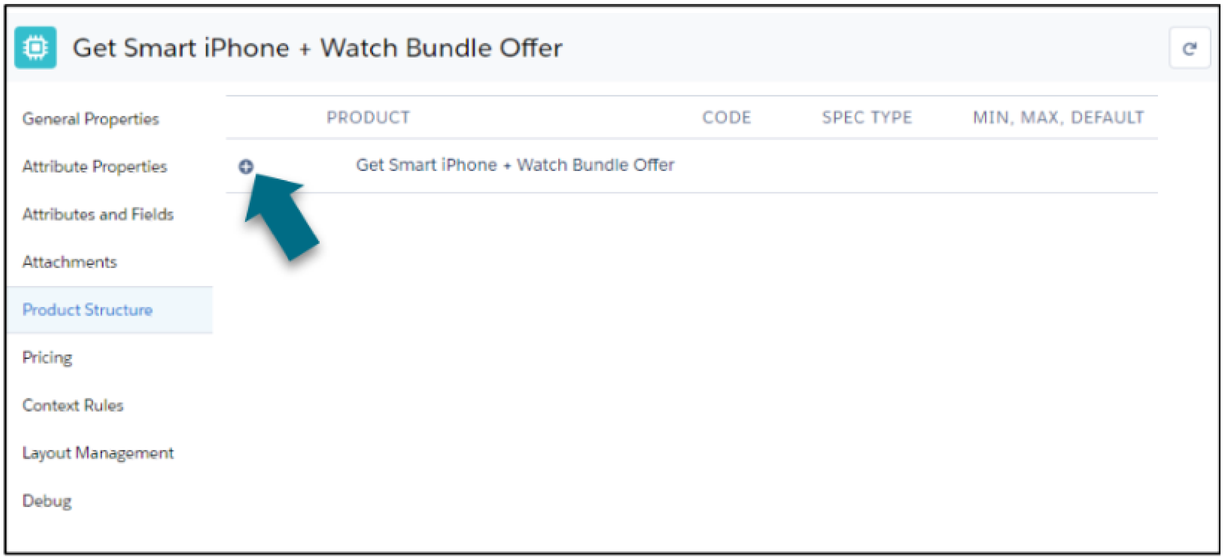
Cardinality Overrides

You can override product cardinality in a bundle. Cardinality overrides must be equal or greater than the minimum and maximum limits set by the original cardinality. For example, if the minimum original quantity is 2, you cannot create a minimum override quantity of 1; the override must be greater than or equal to 2.

Create a Product Bundle

Creating a product bundle as an offer involves creating the structure of the bundle, including the main product and the product child items that comprise the bundle offer.

1. In the Console's tab navigation bar, click **Dashboard**.
2. Under Product Management next to Product, click the **+** icon.



3. In the New Product dialog, enter information for the following:
   - Object Type
   - Product Name
   - Specification Type
4. Click **Save**.
5. Click the **General Properties** facet and enter the following:
   - Product Code
   - Product Description
   - Status: Active
   - Orderable: ✓
   - Active: ✓
   - Effective Date
6. Click **Save**. A green *Product saved!* success message appears.
7. Click the **Product Structure** facet. This allows you to create the structure of the product bundle.
8. Click the **Add** icon to add a product child item (PCI) to the product bundle.
9. Enter the following GENERAL PROPERTIES information:
   - Child Product: Use the lookup dialog to select the child product.
   - Relationship Type
   - Sequence: This value determines the order the child products will appear in the bundle. Use a sequence such as 10, 20, 30 to ensure additions can be easily inserted.
10. In the Cardinality section, enter the following information:
    - Min Quantity: Determines the minimum quantity required for a valid order.
    - Default Quantity: Determines the default number of child products to include in the bundle when it is added to the cart.
    - Max Quantity: Determines the maximum quantity required for a valid order.
11. The Virtual Item flag, when checked, makes the product item not assetizable. This is customarily used as a grouping mechanism, and to set cardinality for the parent product nodes.
12. The **Collapse Hierarchy** flag, when checked, enables a Search Item dialog in the cart's line items pane allowing you to search items within the bundle and then add additional products to the bundle. This is intended for use with very large product bundles. For example, at run time, when you type in the name of the mobile handset that you are looking for in the search item dialog box, if the **Collapse Hierarchy**flag is set to True, the product bundle displays all the mobile handset items in a drop-down list as suggestions for you. This is useful as the Mobile handset product bundle may be large and can have several items.
13. Click **Save**.
14. Click the Add icon next to the parent product.
15. Using the steps above, add child products to the bundle.

Override the Cardinality of the Child Product Bundle

You can override product cardinality in a bundle. Cardinality overrides must be equal to or greater than the minimum and maximum limits set by the original cardinality.

For example, if the minimum original quantity is 2, you cannot create a minimum override quantity of 1; the override must be greater than or equal to 2.

1. In the product structure window, expand the product bundle by clicking the toggle arrow. Click in the white space of the line for a product child item. Do NOT click the product name link because that will navigate you to the product record.
   The product child item window appears on the right. Notice the existing CARDINALITY settings that were set when the product child item was added to the main product bundle.
2. Click **Override Mode**.
3. Click **Create Override**.
   Cardinality overrides must be equal or greater than the minimum and maximum limits set by the original cardinality. For example, if the minimum original quantity is 2, you cannot create a minimum override quantity of 1; the override must be greater than or equal to 2.
4. In the override value fields, change the values for Min Quantity and Max Quantity.
5. Click **Save**.

Excluding a Product Child Item from the Product Bundle

You can exclude a product child item from the main product bundle.

1. Click in the white space of the line for the product child item. Do NOT click the product name link because that will navigate you to the product record.
2. Click **Override Mode**.
3. Click **Exclude Item**.

When you refresh the page and expand the product bundle, there are cardinality override notations beside the product child item you excluded.

Override Product Attributes

With Vlocity EPC, you create attributes one time and assign them to an object type, which products inherit. Within a bundle, you can override attributes for a product.

NOTE

This procedure is for the Vlocity Product Console. For the Vlocity Product Designer procedure, see Override Product Attributes in Product Designer.

For example, Google and Apple smartwatches could both have the object type SmartWatch, each with a different set of attributes. Having this structure in place helps the CSR support customers by giving them the correct product details and place consistent orders.

Although you can’t delete any inherited attributes or fields, you can override them at the product level, such as overriding the color picklist inherited from the SmartWatch object type to include colors that are specific to the Series 3 Apple Watch. You can also add additional attributes directly at the product level.

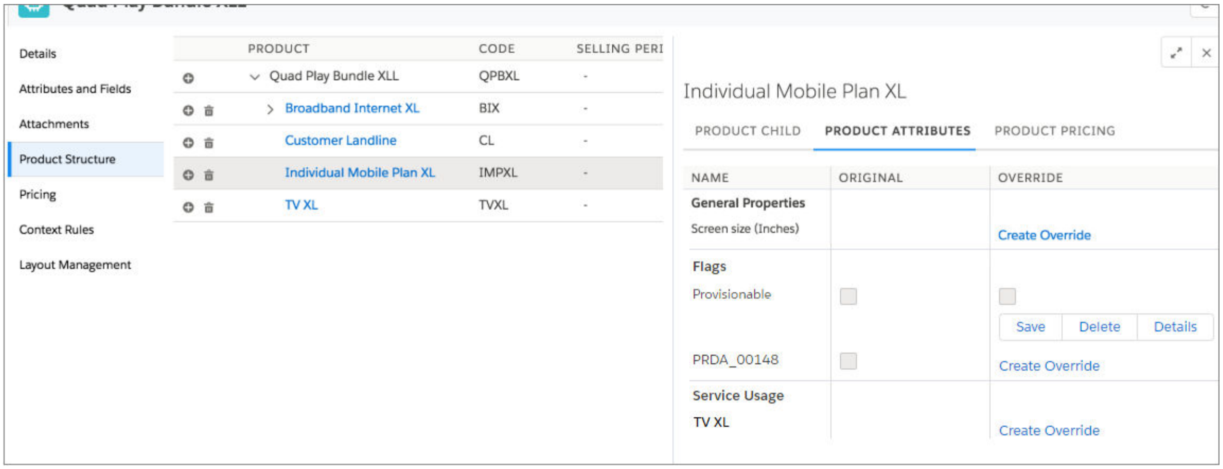
If you see empty or duplicate attributes, see Fix Duplicate and Empty Attributes.

If you don’t see an attribute to override and the attribute is available on the product, run these jobs:

- Running the EPCProductAttribJSONBatchJob
- Running the EPCFixCompiledAttributeOverrideBatchJob

To override the attributes of a product in a bundle:

1. In the product structure window, expand the product bundle by clicking the toggle arrow. Click in the white space of the line for a product child item. Do NOT click the product name link because that navigates you to the product record.
   The product child item window appears on the right.
2. Click the Product Attributes tab. Notice the existing product attribute settings that were set when the product child item was added to the main product bundle.
3. For any of the product attributes, click **Create Override**. The override options appear.



PRODUCT	CODE	SELLING PERI
Quad Play Bundle XLL	QPBXL	-
Broadband Internet XL	BIX	-
Customer Landline	CL	-
Individual Mobile Plan XL	IMPXL	-
TV XL	TVXL	-

NAME	ORIGINAL	OVERRIDE
General Properties		
Screen size (Inches)		Create Override
Flags		
Provisionable	☐	☐ Save Delete Details
PRDA_00148	☐	Create Override
Service Usage		
TV XL		Create Override

4. Click **Details**. The appropriate field or dialog appears. This detail can be, for example, an Attribute Metadata dialog, a drop list, or a number-picker.
5. Select the values for the product attributes.
6. Click **Save**.

If you decide not to override the attribute, click **Delete**.

Create a Price List Entry for a Product Bundle

You can add a price list entry to your new product bundle so that it rolls up prices for the child products when it’s displayed in Vlocity Cart.

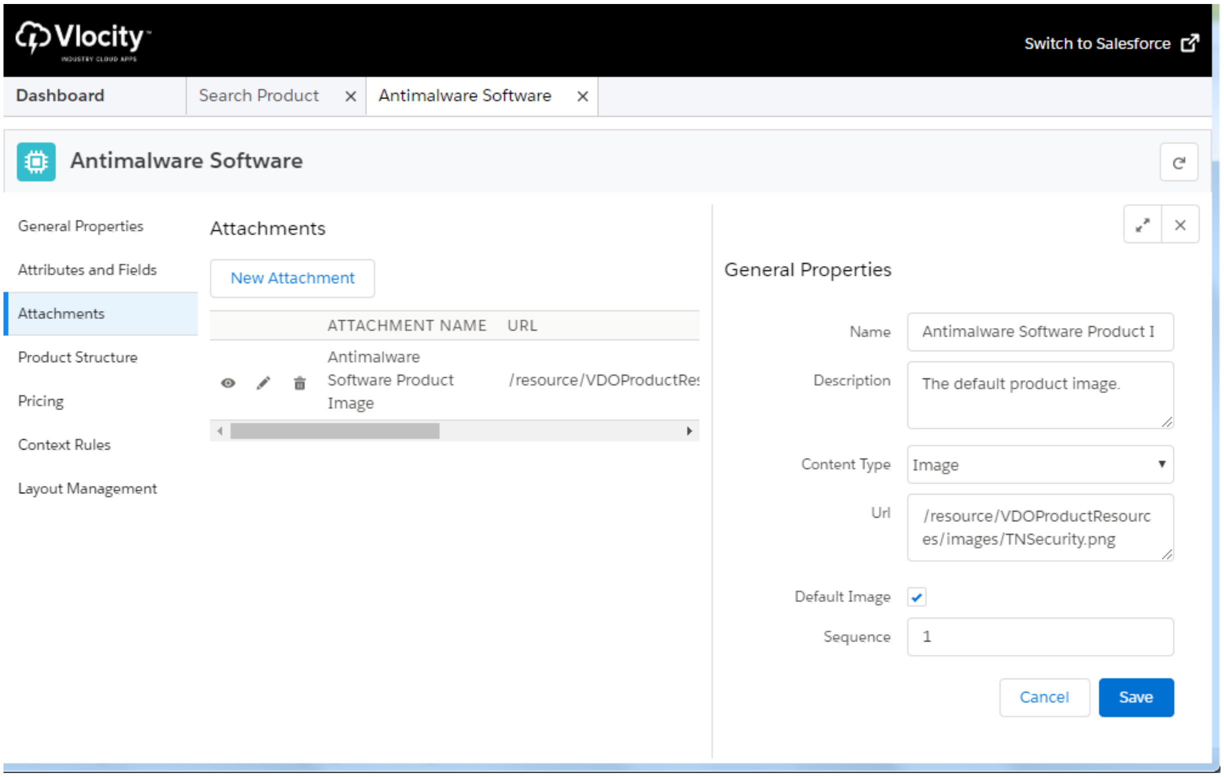
1. Click the **Pricing** facet.
2. Click **New**.
3. In the PRICE LIST ENTRY General Properties dialog, enter information for:
   - Price List
   - Display Text
   - Base Price

4. In the PRICE LIST ENTRY Pricing Variable dialog, enter information for:
   - Charge Type
   - Sub-Type
   - Type
5. In the PRICE LIST ENTRY Pricing Variable dialog, click **Search**.
6. From the search results, select an option.
7. In the PRICE LIST ENTRY Pricing Element dialog, click an option.
8. In the PRICE LIST ENTRY Effectivity dialog, select a date and set it to Active.
9. Click **Save**.

Add an Attachment or an Image to a Product

Vlocity Product Console provides the ability to upload various types of attachments, including images and documents. On the Attachments tab, you indicate the product image to show across different channels.

1. From the Vlocity Product Console, select the product to which to add the image.
2. Click the **Attachments** facet.
3. Click **New Attachment**. The General Properties panel appears.



	ATTACHMENT NAME	URL
	Antimalware Software Product Image	/resource/VDOProductRes

Name: Antimalware Software Product I

Description: The default product image.

Content Type: Image

Url: /resource/VDOProductResources/images/TNSecurity.png

Default Image: ☑

Sequence: 1

4. In the Attachment General Properties window, enter the following information:
   - Name: This value is only visible at design-time in the Attachments facet of the Console.
   - Description: This value is only visible at design-time in the Attachments facet of the Console.
   - Content Type: Click the down arrow and select an option, for example, Image.

- Url: Select the URL to the image, video, or file. Vlocity does not have the capability to upload files, so it only supports product images through an absolute or relative URL.
- Default Image: If the default flag is checked, the image appears as the default image in the Product Details window in Vlocity Cart.
- Sequence: The sequence value is used to order multiple images in the Gallery in the Product Details window in Vlocity Cart.

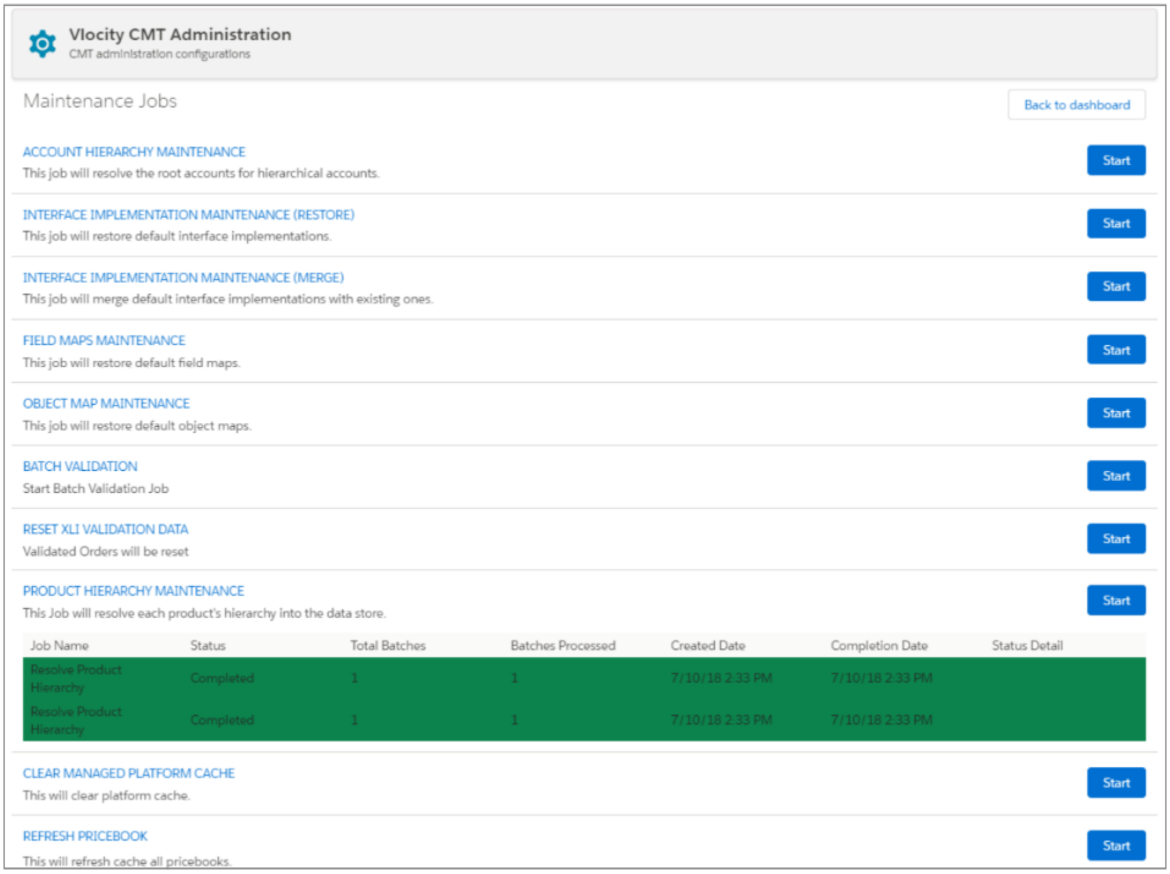
5. In the General Properties window, click **Save**.

In the attachment list, you can use the icons to preview, edit, or delete attachments.

Refresh the Product Hierarchy Cache

After you create a product bundle, you must update the product hierarchy cache, or the new product bundle does not display properly in Vlocity Cart.

1. Build the product hierarchy data store by running the Product Hierarchy Maintenance job:
   a. From the App Launcher in Lightning or the + icon in Salesforce Classic, click **Vlocity CMT Administration**.
   b. Under Admin Console, click **Maintenance Jobs**.
   c. On the Maintenance Jobs page, find the PRODUCT HIERARCHY MAINTENANCE job and click **Start**.
      For Fall '20 and later, see Product Hierarchy Maintenance about the **Delete Old Data** option.
   d. Click **OK** to accept the notification that the job may affect your org settings and items.
      Messages indicate that the job completed successfully. Salesforce also emails a successful job completion report to the email address registered.



Vlocity CMT Administration

CMT administration configurations

Maintenance Jobs

ACCOUNT HIERARCHY MAINTENANCE

This job will resolve the root accounts for hierarchical accounts.

INTERFACE IMPLEMENTATION MAINTENANCE (RESTORE)

This job will restore default interface implementations.

INTERFACE IMPLEMENTATION MAINTENANCE (MERGE)

This job will merge default interface implementations with existing ones.

FIELD MAPS MAINTENANCE

This job will restore default field maps.

OBJECT MAP MAINTENANCE

This job will restore default object maps.

BATCH VALIDATION

Start Batch Validation Job

RESET XLI VALIDATION DATA

Validated Orders will be reset

PRODUCT HIERARCHY MAINTENANCE

This Job will resolve each product's hierarchy into the data store.

Job Name	Status	Total Batches	Batches Processed	Created Date	Completion Date	Status Detail
Resolve Product Hierarchy	Completed	1	1	7/10/18 2:33 PM	7/10/18 2:33 PM	
Resolve Product Hierarchy	Completed	1	1	7/10/18 2:33 PM	7/10/18 2:33 PM	

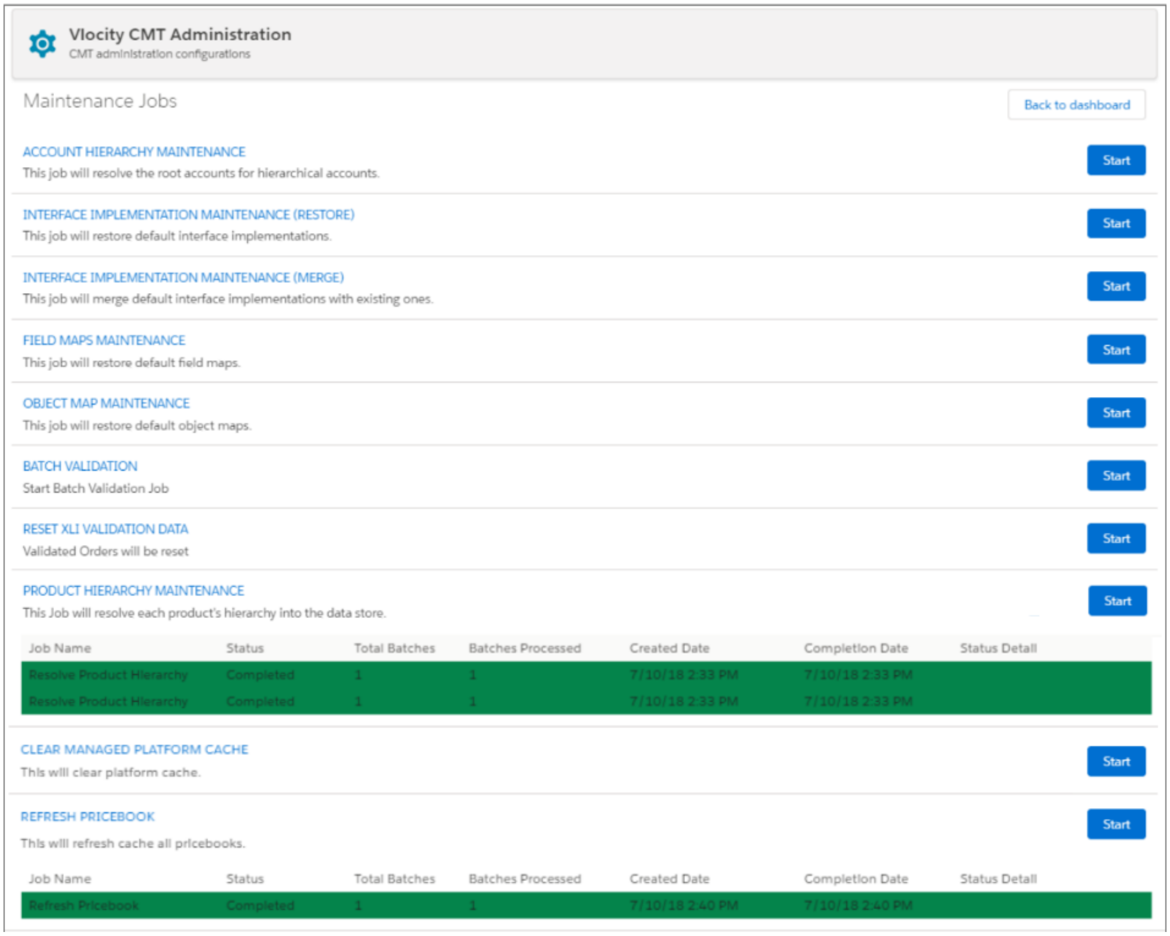
CLEAR MANAGED PLATFORM CACHE

This will clear platform cache.

REFRESH PRICEBOOK

This will refresh cache all pricebooks.

2. Populate the platform cache and the attribute cache:
   a. On the Maintenance Jobs page, find the REFRESH PRICEBOOK job and click **Start**.
   b. Click **OK** to accept the notification that the job may affect your org settings and items. Messages should display indicating that the job completed successfully.



Vlocity CMT Administration

CMT administration configurations

Maintenance Jobs

ACCOUNT HIERARCHY MAINTENANCE

This job will resolve the root accounts for hierarchical accounts.

INTERFACE IMPLEMENTATION MAINTENANCE (RESTORE)

This job will restore default interface implementations.

INTERFACE IMPLEMENTATION MAINTENANCE (MERGE)

This job will merge default interface implementations with existing ones.

FIELD MAPS MAINTENANCE

This job will restore default field maps.

OBJECT MAP MAINTENANCE

This job will restore default object maps.

BATCH VALIDATION

Start Batch Validation Job

RESET XLI VALIDATION DATA

Validated Orders will be reset

PRODUCT HIERARCHY MAINTENANCE

This Job will resolve each product's hierarchy into the data store.

Job Name	Status	Total Batches	Batches Processed	Created Date	Completion Date	Status Detail
Resolve Product Hierarchy	Completed	1	1	7/10/18 2:33 PM	7/10/18 2:33 PM	
Resolve Product Hierarchy	Completed	1	1	7/10/18 2:33 PM	7/10/18 2:33 PM	

CLEAR MANAGED PLATFORM CACHE

This will clear platform cache.

REFRESH PRICEBOOK

This will refresh cache all pricebooks.

Job Name	Status	Total Batches	Batches Processed	Created Date	Completion Date	Status Detail
Refresh Pricebook	Completed	1	1	7/10/18 2:40 PM	7/10/18 2:40 PM	

Types of Products in a Bundle

When adding products to a bundle, you can mark them as optional, required, or virtual.

- An optional item is an item that is not required by the product definition and is defined by having Min Quantity = 0 in the PCI record.
  An optional item may have Default Quantity = 1 and therefore show up by default when added to cart. This does not make it mandatory. It is still an optional item and can be deleted from the cart.
- A required item is an item which has the Min Quantity > 0. A special case of required items is when Min = Max (= Def). In this case, not only is the item required but users cannot change the quantity at run time.
- A virtual item is used as a grouping mechanism for like products using the minimum and maximum children quantity to define the allowable items in the group. Virtual items only appear as products during the product design phase. In the cart, virtual items appear as grouping categories, no line items are created for virtual products. Virtual products are not assetized. When migrating assets, you must take virtual products in consideration when you populate the product hierarchy path field for the assets (for example, CHILD_OFFER).

Create Products in Vlocity Product Console

With Vlocity EPC, you can create an offering to sell to customers, such as an iPhone 11. Before creating an offering, you must create other entities, such as object types, attributes, and product specifications. You can then create bundled offerings and set pricing.

Before You Begin
1. Create a Price Book 2. Create New Product Attribute Categories

NOTE

Complete the following tasks in the specified order.

1. Create a Picklist
2. Create Attributes
3. Create an Object Type
4. Assign Attributes and Fields to an Object Type
5. Define Page Layout for an Object Type
6. Create a Product Specification

Create a Price Book

Every price list must be associated with a price book in Salesforce. You can associate multiple price lists to the same price book. If you don't have a price book to use, you must create one.

1. Click the **Price Books** tab. If the tab is not displayed, click the **All Tabs** plus icon and click the **Price Books** link.
2. Click the **New** button.
3. Enter a unique name for the price book.
4. For example, **`Standard Price Book`**.
5. Select the **Active** checkbox.
6. Click **Save**.

What's Next

Create New Product Attribute Categories

See Also

- Creating Products in Vlocity Product Console

Create New Product Attribute Categories

Attribute categories group attributes. They also appear as headings at run time in Vlocity Cart's Filter and Configuration windows. Attribute categories are required to create product attributes.

The type of attribute is defined at the Attribute Category level. You must create Attributes Categories before adding new attributes to the category.

For more information about housing listings or filters, see Configure Housing Listing Filters.

To create a product attribute category:

1. On the Vlocity Attribute Designer tab, click **New**.
2. In the **Name** field, enter the attribute category name (for example, `Color`).
3. In the **Display Sequence** field, enter a number to indicate the sequence in which the category is to be displayed on the record detail page or Housing Listing.
4. In the **Code** field, enter a unique code for the category.

   TIP
   The code is not visible in the user interface.

5. Select the **UI Control Type** for the attributes in the new category.
   Each attribute category has a specific user interface control type for its attributes:
   - Select **On-Off** to add or remove attributes from categories.
   - Select **3-State** to set attributes to positive, neutral, or negative.
   - Select **1-5 Scale** to set attributes to a number from 1 to 5.
   - Select **Text Comment** to add text to attributes.
   - Select **Multi-Select** for housing filters.
6. From the **Applicable Types** list, select **Product2**.
   When creating housing filters, select **Any**.
7. From the **Applicable Subtype** picklist, select one of the following:
   - **Product Attribute**: For Product records.
   - **Profile Attribute**: For Profile records. Profile attributes are used with Vlocity Intelligence.
   - **Inventory Item**: For attributes to set up housing filters.
8. Select **Filterable** to be able to use the attribute category to filter products or housing.
9. Update the **Color Code** is used to determine the color of the category tag. Enter the color code using a hex value.
10. Click **Save**.

Create a Picklist

A picklist is an ordered list of values that customers can choose from when buying a product. If an attribute includes predefined values, a picklist represents all available values for that attribute. You can reuse picklists across attributes assigned at the object type level or the product level.

If using attributes with picklists, this is the first step in the Create Products in Vlocity Product Console process. Otherwise, you can skip to the Create Attributes step. All steps must be performed in the specified order.

1. In the **Vlocity Product Console**, in the **Foundation** panel, click the plus icon next to **Picklist**.
2. Enter the following values for a new picklist:

Field Name	Required?	Description
Picklist Name	Yes	Names are used to identify picklists in search results.
Code	Yes	A unique code for the picklist. Typical naming convention for picklist code is PKL_<alphanumeric-value>. For example, PKL_100900. The unique code is not enforced, but recommended.
Description	No	Purpose of the picklist.
Data Type	Yes	Data type for the picklist items. Available values include: • Text • Number • Date/Time • Date • Boolean
Active	Yes	Activates the picklist.

3. Click **Save**.
4. Click the **Picklist Items** tab.
5. Click **New Item**.
6. Enter values for the following fields:

Field Name	Required?	Description
Text Value	Yes	Value for the picklist item. This value must be unique within a picklist. For example, for a Color picklist, you cannot have one item labeled **`Silver`** with a value of **`Silver`** and another item labeled **`Default`** with a value of **`Silver`**. The Default item must either have a different, unique value or be removed. The data type of this field is set by the picklist data type. Text value is important if picklist item's data type is other than text. For example, if the data type is numeric, and the value is 1, the text value is one (or 1 as text).
Label	Yes	Label for the item as it appears in the picklist.
Code	Yes	Unique code for the picklist item (unique within this picklist - not enforced but recommended)
Sequence	Yes	Order in which the item appears in the picklist.
Abbreviation	No	Abbreviated version of the label.
Active	Yes	Activates the picklist item.
Effectivity	Yes	Date on which the item goes into effect

7. Click **Save**.
8. Follow Steps 5-7 to create additional picklist values.

What's Next

Create Attributes

See Also

- Create Products in Vlocity Product Console

Create Attributes

A product attribute is a characteristic that defines a particular product and affects a consumer's purchase decision. Attributes can include product characteristics.

To be compatible with the Summer ’17 release and later, you must create product attributes only through Vlocity Product Console and never through the Salesforce Product tab.

1. In the Vlocity Product Console, in the Foundation panel, click the plus icon next to Attributes.
2. Complete the following fields:

> **NOTE**
> Certain fields appear after you click **Save**.

Name	Required?	Description
Name	Yes	Names are used to identify attributes in search.
Code	Yes	A unique code for the attribute. Typical naming convention for code is `ATT_<alphanumeric-value>`. For example, `ATT_10040`.
Description	No	Descriptions help users identify the purpose of the attribute.
Active	Yes	Activates the attribute.
Attribute Category	Yes	Attributes are organized into predefined categories.
Value Type	Yes	Available values include: • Currency • Percent • Text • Number • Checkbox • Date • Datetime • Picklist • Multi Picklist • Lookup

3. If you selected the **Picklist** Value Type, select a previously created picklist.

IMPORTANT

Picklist attribute values are stored as text, even if the picklist you select has another **Data Type**, such as **Number**. If you use a Picklist attribute in a Set Value or Set Default Value rule, verify that the rule expresses the attribute's data type as text for the Value to Set and in the rule's Expression box.

4. Click **Save**.
5. Click the **Applicable Objects** link.
6. Select checkbox next to **Product2 Object**.
7. Click **Save**.

What's Next

Create an Object Type.

See Also

- Creating Products in Vlocity Product Console

Create an Object Type

A base object type that you create serves as a template that stores configuration of attributes, fields, and layouts. When you create object subtypes, they inherit configuration of the base object's layout, fields, and any attributes.

This configuration can be modified for each object type. However, you must modify it before you create object subtypes, so that they inherit the new modified configuration.

1. In the **Vlocity Product Console**, in the **Metadata** section, click the search icon next to **Object**.
2. In the **Search Object** page, click search.
3. Click the pen icon next to the **Product2 Object**.
4. In the left menu, click **Object Types**.
5. Click **New Object Type**.
6. In the **General Properties** area, enter the name of the base object, for example, Base Object Type.
7. When creating a base object type, leave the **Parent Object Type** field empty. For an object subtype, from the **Parent Object Type** field, select the base object type that you have previously created.
8. In the **Effectivity** area, select the **Active** checkbox and optionally define the effective from and effective until values.
9. Click **Save**.

NOTE

Do not create child product types until you define attributes, fields and layouts for the Base Product type, so that the children inherit the basic settings, which you can later modify.

What's Next

Assign Attributes and Fields for an Object Type

See Also

- Create Products in Vlocity Product Console

Assign Attributes and Fields for an Object Type

You can create attributes to define the characteristics of products. Each attribute is associated with an attribute category. The association of an attribute to an Object type or product is called Attribute Assignment. You assign attributes to a product Object type and these attributes are later inherited by the child object types when you add them.

You can only assign attributes and fields that you have previously created. For guidelines on when to use fields vs. attributes, refer to Attributes vs. Fields.

1. In the Product Console, under **Metadata**, click the search icon next to Object. Click the **Base Object type**.
2. Select the object. For example, Product2 object.
3. Click the **Object types** facet.
4. Select the object type to which you want to assign attributes and fields. For example, SmartPhone Product spec.
5. Click the **Assign Attributes/Fields** tab. The attributes and fields that can be assigned to the Object type appear on the right side.
6. Select the attributes and fields by selecting the corresponding checkboxes.
7. Click **Assign**. Vlocity assigns the selected attributes and fields to the Base object type.
8. When you add an attribute to an existing object type that is associated with products, run the EPCProductAttribJSONBatchJob job.
   See Running the EPCProductAttribJSONBatchJob.
9. If an associated product has at least one attribute with an override, run the EPCFixCompiledAttributeOverrideBatchJob.
   See Running the EPCFixCompiledAttributeOverrideBatchJob.

What's Next

Define the Layout for an Object Type

See Also

- Create Products in Vlocity Product Console
- Overview of Attributes

Encrypted and Not Assetizable Attributes

Starting in the Fall '18 release, attributes for technical products can now be set as not assetizable (to identify temporary data) and encrypted (to protect personally identifiable information or similar data).

Is Not Assetizable

The **Is Not Assetizable** attribute metadata flag is useful for transitory information that shouldn’t persist or doesn’t need to persist beyond completion of order management where assetization would usually occur. This metadata flag is stored in Order Management, not in the Salesforce platform.

An example of transitory information is the initial password that a customer uses to log in to the provider's website for the first time. During fulfillment, the initial password is generated within an orchestration order item, then saved on the technical fulfillment request line, which is then passed to the next orchestration item that emails the password to the customer. The password doesn’t need to persist at the end of order management. Other information that doesn’t need to persist can be the result of back-end services connected to order items that don’t become customer assets, such as a service needed to fulfill a customer’s order. As an example, there might be a check to see whether a required modem is in stock and on the delivery vehicle. This information is not needed after that check has been made.

Is Encrypted

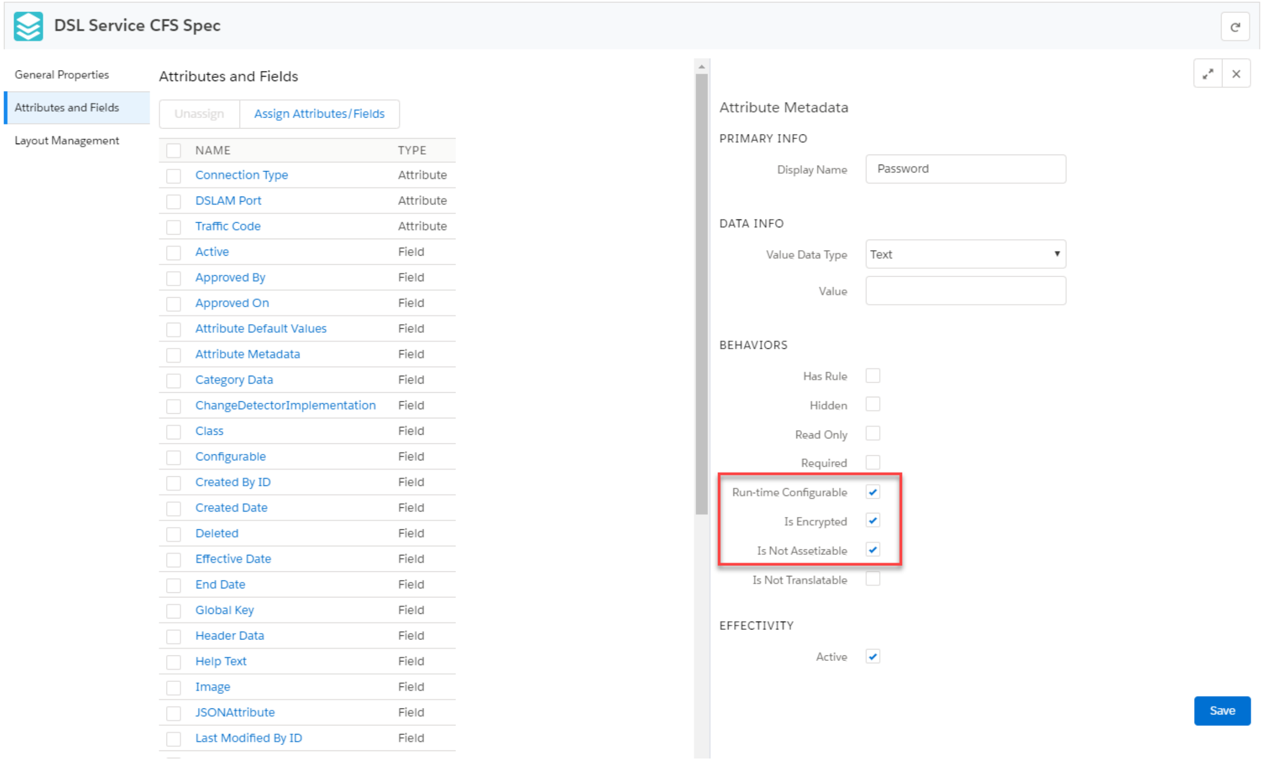
The **Is Encrypted** attribute metadata flag is used to protect personally identifiable information (PII) and other data such as the customer name, phone number, social security number, and so on. These pieces of information can be captured as attributes and should be sent downstream only for legitimate purposes. You can use the encrypted flag to indicate the PII or other information must be stored securely.

Setting Is Not Assetizable and Is Encrypted

Is Not Assetizable and **Is Encrypted** are attribute metadata flags. You set these flags when an attribute is assigned to an object type in Vlocity Product Designer or Vlocity Product Console. These flags can only be set when the **Run-time Configurable** flag is also set. After the attribute is marked as **Run-time Configurable**, you can then set **Is Not Assetizable** or **Is Encrypted** or both.

If none are selected, attribute values are stored in the technical inventory after decomposition. If both are selected, the attribute is encrypted during processing and not stored with the inventory items. However, if you set the **Is Encrypted** flag to True, and **Is Not Assetizable** to False, the attribute is stored in the technical inventory, but the attribute is encrypted. Technical assetization stores the output of decomposition into the technical tables in Order Management, not in the Salesforce platform.

Figure 4. Setting Is Not Assetizable and Is Encrypted Attribute Metadata



DSL Service CFS Spec

- General Properties
- Attributes and Fields
- Layout Management

Attributes and Fields

NAME	TYPE
Connection Type	Attribute
DSLAM Port	Attribute
Traffic Code	Attribute
Active	Field
Approved By	Field
Approved On	Field
Attribute Default Values	Field
Attribute Metadata	Field
Category Data	Field
ChangeDetectorImplementation	Field
Class	Field
Configurable	Field
Created By ID	Field
Created Date	Field
Deleted	Field
Effective Date	Field
End Date	Field
Global Key	Field
Header Data	Field
Help Text	Field
Image	Field
JSONAttribute	Field
Last Modified By ID	Field

Attribute Metadata

PRIMARY INFO

Display Name: Password

DATA INFO

Value Data Type: Text

Value:

BEHAVIORS

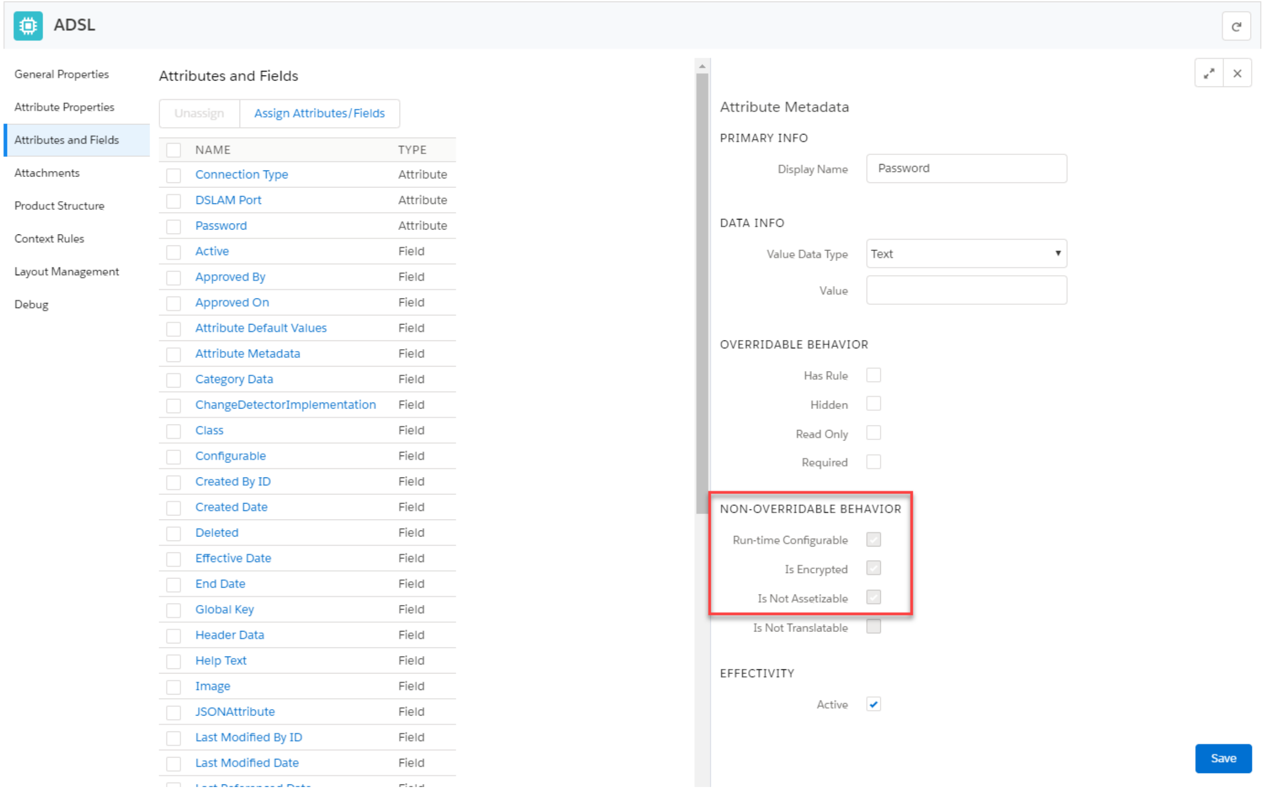
Has Rule ☐
Hidden ☐
Read Only ☐
Required ☐
Run-time Configurable ☑
Is Encrypted ☑
Is Not Assetizable ☑
Is Not Translatable ☐

EFFECTIVITY

Active ☑

After assigned to the object type, every product that is created in that object type inherits these metadata flags, with the other attribute and metadata. **Is Not Assetizable**, **Is Encrypted** and **Run-time Configurable** are then non-overridable at the product level. This prevents any product manager or EPC administrator from removing these flags during product creation.

Figure 5. Non-Overridable Attribute Metadata on Products



ADSL

Attributes and Fields

NAME	TYPE
Connection Type	Attribute
DSLAM Port	Attribute
Password	Attribute
Active	Field
Approved By	Field
Approved On	Field
Attribute Default Values	Field
Attribute Metadata	Field
Category Data	Field
ChangeDetectorImplementation	Field
Class	Field
Configurable	Field
Created By ID	Field
Created Date	Field
Deleted	Field
Effective Date	Field
End Date	Field
Global Key	Field
Header Data	Field
Help Text	Field
Image	Field
JSONAttribute	Field
Last Modified By ID	Field
Last Modified Date	Field

Attribute Metadata

PRIMARY INFO

Display Name: Password

DATA INFO

Value Data Type: Text

Value:

OVERRIDABLE BEHAVIOR

Has Rule □
Hidden □
Read Only □
Required □

NON-OVERRIDABLE BEHAVIOR

Run-time Configurable □
Is Encrypted □
Is Not Assetizable □
Is Not Translatable □

EFFECTIVITY

Active ☑

If you set these flags on existing attributes, it does not affect attributes that are already stored in existing technical inventory.

Is Not Assetizable and **Is Encrypted** metadata flags are intended to trigger functions in downstream order management systems, such as Vlocity Order Management Plus. There is no encryption to these attributes when used on commercial products in Vlocity CPQ or in Vlocity Order Management, and the information remain stored and accessible on the originating order. Consequently, these attribute flags are intended for use in technical products only.

Using Is Not Assetizable and Is Encrypted

Once these attribute metadata flags have been set on an object type and inherited by a product, you must create decomposition rules to map the attributes to the downstream fulfillment requests, which are the basis for the technical inventory. See Partial Assetization of Product Attributes.

Change the Product Attribute Display Sequence

In the product catalog, you can override or update the display sequence for the attribute and the attribute category in the context of the product.

An attribute category contains a set of attributes with a default display sequence. When the attributes are associated with a product, the attribute display sequences are inherited. You can choose to either accept or override the default display sequence of the attributes within the attribute category.

For a standalone product, the ability to change the attribute display sequence is on the attribute assignment. The display sequence of an attribute represents the order of an attribute within an attribute category.

iPhone 8 and Samsung Galaxy Example

- Product: iPhone8:
  - Attr.Category: Mobile_Handset (Seq:11): This sequence represents the sequence of attribute category for the iPhone8.
    - Attr: 1. Color (Red, White, Black): 1 represents the sequence of attributes within the Mobile_Handset category.
    - Attr: 2. Storage (32 G, 64 G).
  - Attr.Category: Mobile_Handset_Manufacturer (Seq:12): This sequence also represents the sequence of attribute category for the iPhone8.
    - Attr: 1. Manufacturer: 1 represents the sequence of attributes within the Mobile_Handset category.
    - Attr: 2. Model.

When adding the above product to the CPQ cart, the attribute and attribute category appear in the cart according to the sequence defined above.

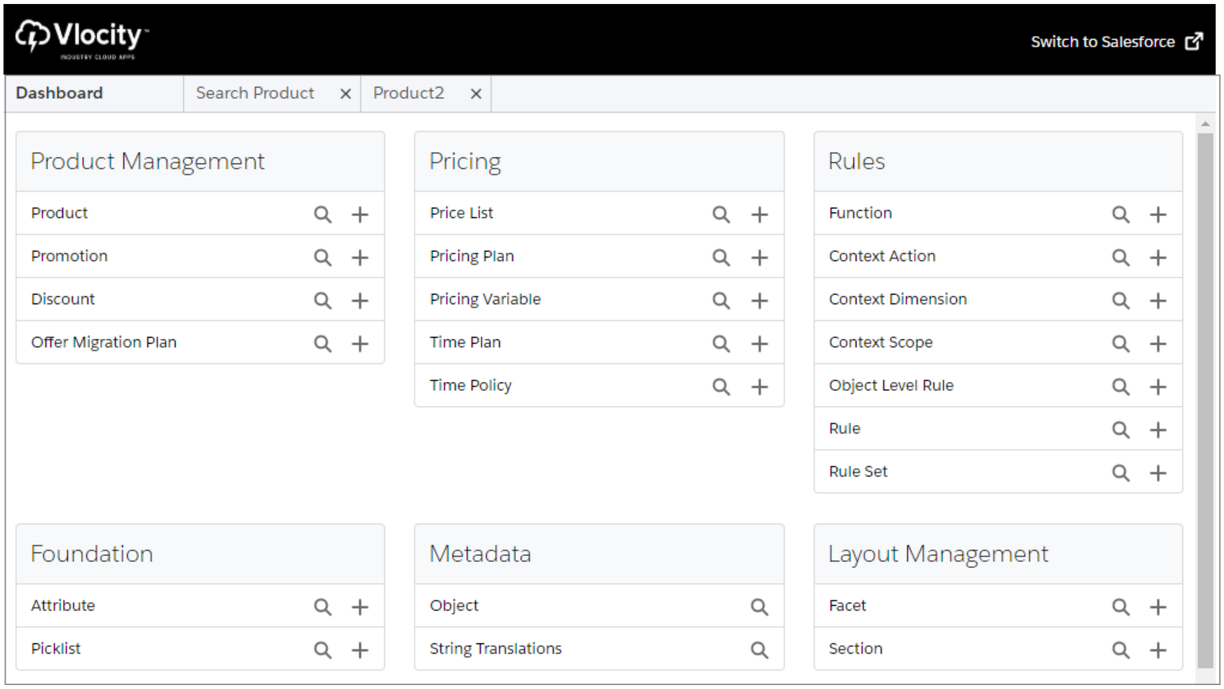
Now, for a Samsung Galaxy Phone, you can use the same attributes as you did for the iPhone8 but in a different sequence, as follows:

- Product: Samsung Galaxy:
  - Attr.Category: Mobile_Handset (Seq: -11- 12): This sequence represents the sequence of attribute category for the iPhone8.
    - Attr: 1>2. Color (Red, White, Black): 2 represents the sequence of attributes within the Mobile_Handset category.
    - Attr: -2 -1. Storage (32 G, 64 G).
  - Attr.Category: Mobile_Handset_Manufacturer (Seq:12>11): This sequence also represents the sequence of attribute category for the iPhone8.
    - Attr: 1>2. Manufacturer: 1 represents the sequence of attributes within the Mobile_Handset category.
    - Attr: 2>1. Model.

When adding the Samsung Galaxy product to the CPQ cart, the attribute and attribute category appear in the overridden display sequence.

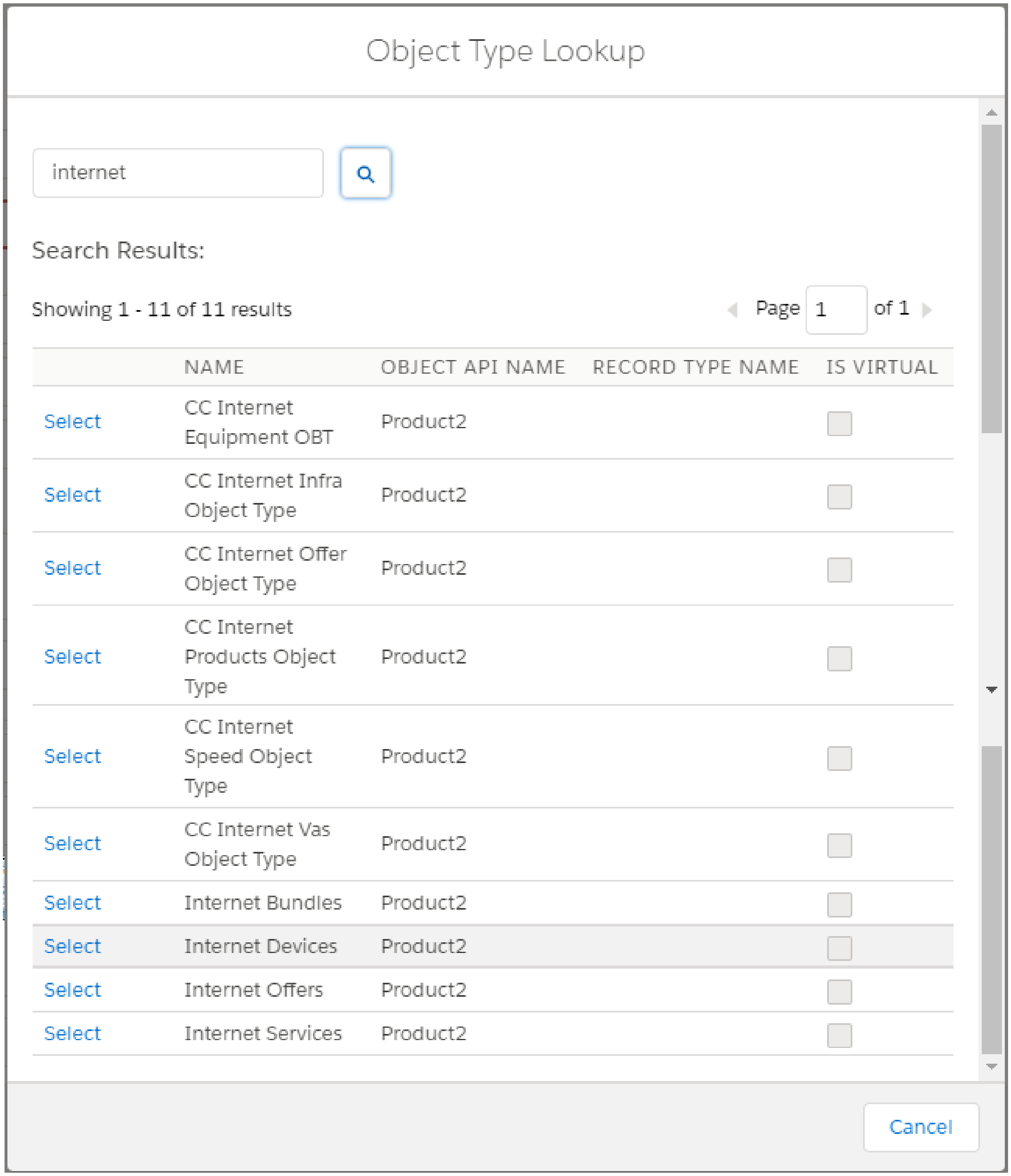
To change the product attribute display sequence as it will appear in Vlocity Cart:

1. In the Vlocity Product Console, click the search icon beside Product and search for any given product.



The Search Results page appears.

2. Click a product from the list of search results. The page for that product appears.
3. Click the **General Properties** facet on the left.
4. In the **Object Types** field, search for an object type and click **Select**.



Object Type Lookup

internet

Search Results:

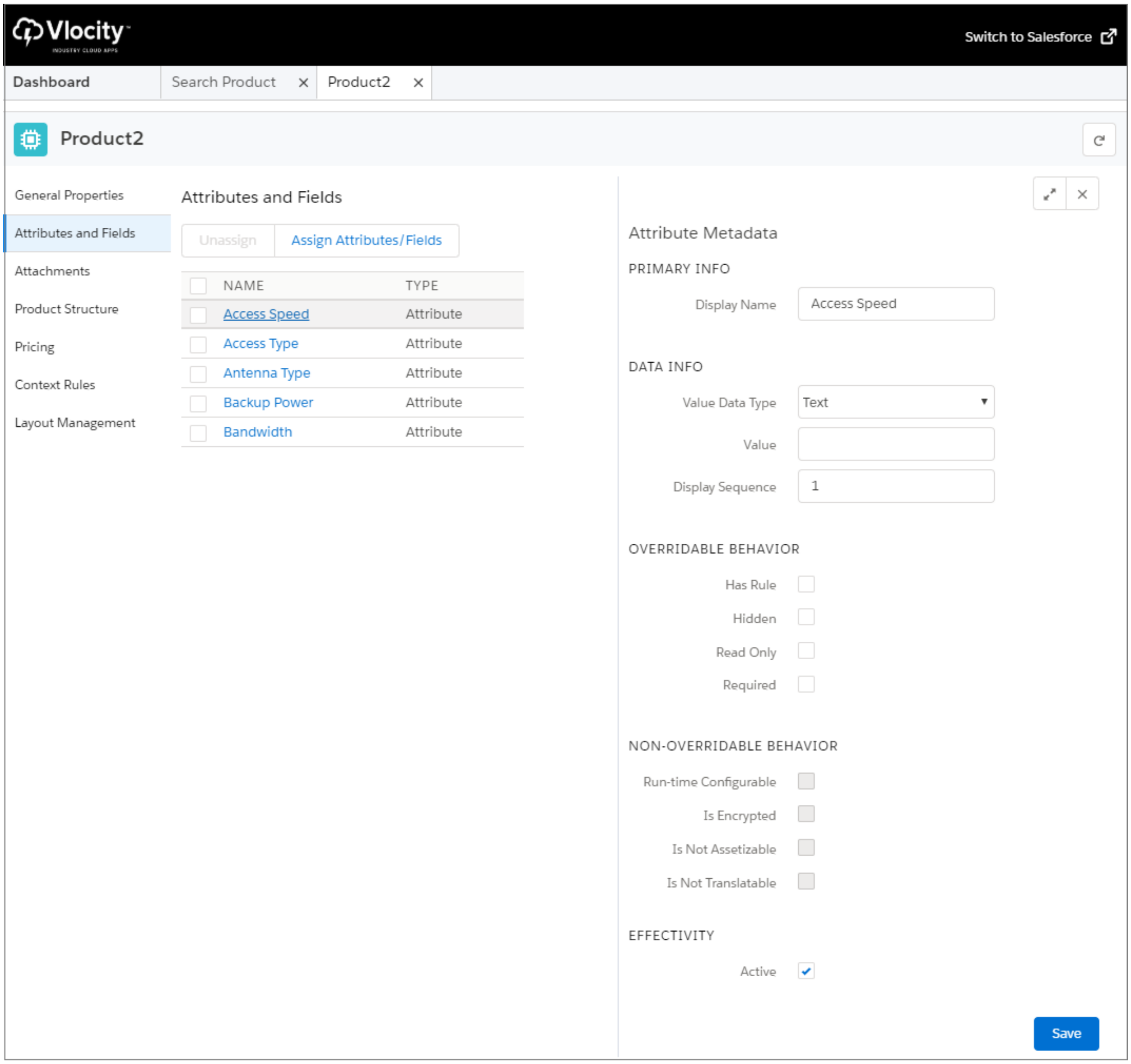
Showing 1 - 11 of 11 results

Page 1 of 1

	NAME	OBJECT API NAME	RECORD TYPE NAME	IS VIRTUAL
Select	CC Internet Equipment OBT	Product2		□
Select	CC Internet Infra Object Type	Product2		□
Select	CC Internet Offer Object Type	Product2		□
Select	CC Internet Products Object Type	Product2		□
Select	CC Internet Speed Object Type	Product2		□
Select	CC Internet Vas Object Type	Product2		□
Select	Internet Bundles	Product2		□
Select	Internet Devices	Product2		□
Select	Internet Offers	Product2		□
Select	Internet Services	Product2		□

5. Enter any other required information, such as the Product Code.
6. Click the **Attributes and Fields** facet.
7. Click any attribute.

8. Change the number in the **Display Sequence** field.



Attributes and Fields

NAME	TYPE
Access Speed	Attribute
Access Type	Attribute
Antenna Type	Attribute
Backup Power	Attribute
Bandwidth	Attribute

Attribute Metadata

PRIMARY INFO

Display Name: Access Speed

DATA INFO

Value Data Type: Text

Value:

Display Sequence: 1

OVERRIDABLE BEHAVIOR

Has Rule □
Hidden □
Read Only □
Required □

NON-OVERRIDABLE BEHAVIOR

Run-time Configurable □
Is Encrypted □
Is Not Assetizable □
Is Not Translatable □

EFFECTIVITY

Active ☑

9. Click **Save**.
   The `attributeDisplaySequence` is passed as part of the response to the product and to the ObjectType calls. The `attributeDisplaySequence` is also sent as a parameter when other attribute properties are modified through the product. The JSON on the product is updated and the new display sequence appears at runtime in the CPQ cart.

NOTE

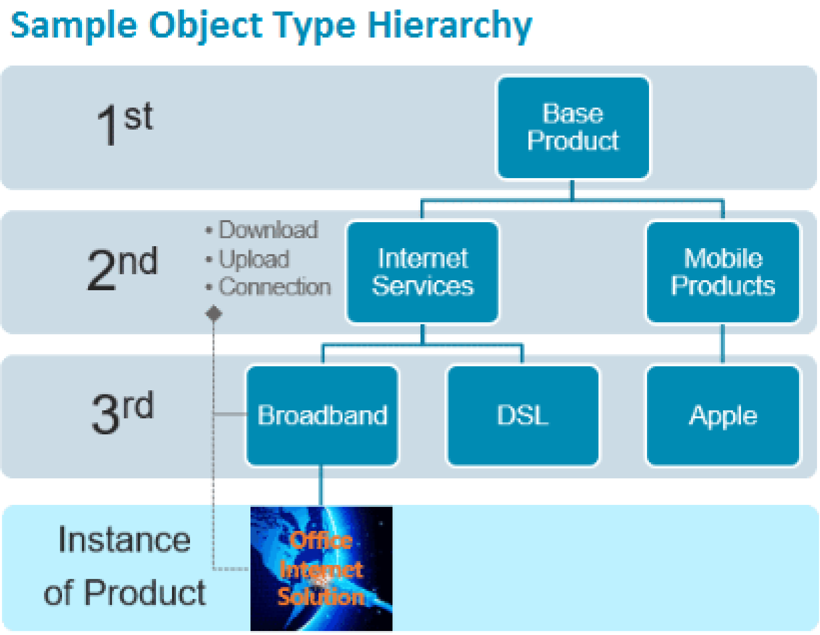
If the attributes belong to different attribute categories, then sorting occurs according to the attribute category that already exists. Then, inside the attribute category, attributes are sorted according to your change.

Define the Layout for an Object Type

An object type is a reusable entity that defines properties, such as fields and attributes, and layouts for all product instances. Objects types are like Salesforce record-types, but with enhanced capabilities. You can use object types to group products with similar characteristics and ensure consistent behavior and application of rules.

Object types support hierarchy inheritance, which allows you to create such objects as a mobile device with attributes such as brand, capacity, and color that are common among many mobile devices. From the object type, you can create a child object type, such as an iPhone, and assign iPhone-specific attributes, such as color. Any product linked to the iPhone object type will automatically inherit the iPhone attributes from the mobile device parent object type as well as any iPhone-specific attributes of the iPhone child object type.

Vlocity’s object types are designed as an "Is A" inheritance architecture, meaning you can create relationships between abstractions, where on object type is a subtype of another object type. The first level of abstraction is known as an object type, and the subsequent levels are known as subtypes. When you apply this relationship to a product, you create a relationship consisting of a product instance.



When defining object types and their inheritance:

- Vlocity recommends that you create a base, or prime, product during the initial setup. This is your product catalog's base object from which all the other objects descend. The base object should include the fields that are related to the base object and that apply to all products created from the base object.

  You should also include fields that are configured at design time rather than runtime. For example, the **displayUrl** field is useful to add because you can define it at design time in the Product Catalog Console.

  NOTE
  Attributes are stored in a product's JSON blob and can affect the performance of runtime searches. Fields are not stored in a JSON blob. See Attributes vs. Fields.

- You can then create child object types for all the product classes and products in your catalog. They inherit the layout of the base product with all the associated fields and attributes.
- You create the layout for the Base Product manually. Layouts help you organize attributes, fields, and custom elements into facets and sections. Facets are pages with corresponding links that appear in the left navigation menu. Layout elements, such as fields or custom URI-based elements, are grouped into sections.
- An object subtype inherits the layout via deep copy, but any subsequent changes to the object super type’s layout is not replicated down the hierarchy after initial creation. You can add or delete fields for each object type without altering the layout with the addition of any fields on the parent object type.
- Inherited attributes and fields cannot be deleted, and instead must be unassigned from the object supertype. Attributes are specific to products and classes of products and cannot be removed.
- New attributes and fields assigned to an object type are inherited dynamically by its subtypes.

NOTE
You must add attributes or fields to the layout. Otherwise, they are not displayed when you create new products from the object type.

When creating object types in the Product Console, you:

1. Create the base product object-type.
2. Assign attributes and field.
3. Define a layout, which consists of creating a new facet and adding sections.

You can then create object subtypes instantiated on the base product object-type.

To define a layout:

1. In the left-hand-side menu, click **Layout Management**.

2. To add a facet:
   a. Click **Add Facet**.
   b. From the **Facet** picklist, select a desired facet.
   c. In the **Sequence** field enter a numeric value for the order in which the facet appears on the page.

NOTE

In the Vlocity Product Designer, only the General Properties facet is displayed and editable. See Manage Existing Object Types in Product Designer.

3. To add a section:
   a. In a facet, click **Add Section**.
   b. From the **Section** picklist, select a name for the section.
   c. In the **Sequence** field enter a numeric value for the order in which the section appears in the facet.
   d. From the **View Type** picklist, select **Filed Layout** or **Custom View**. Field layout allows you to add desired fields and attributes. Custom view is a predefined view that you can use. For custom view, you must specify a value for the **View URI** field. For more information on which URI values to use for custom view, refer to Custom View URIs.
   e. Click **Save**.
4. To add a field:
   a. In a section, click **Add Section Element**.
   b. From the **Field** picklist, select the field name.
   c. In the **Display Label** field, enter the label for the field as you want it to appear in the UI.
   d. In the **Sequence** field, enter a numeric value for the order in which the field appears in the section.
   e. Click **Save**.

Custom View URIs

When defining a layout for an object type, you can use predefined custom URIs to create custom views.

Facet	Section Type	URI Name
Generic	Layout Management	LayoutManagement
Generic	Attachments	Attachments
Attributes	Applicable Objects	ApplicableObjects
Picklists	Picklist Items	PicklistItems
Object Class	Attribute-Field Bindings	AttributeFieldBindings
Object Class	Attributes and Fields	ObjectClassAttrsFields
Product Class	Attributes and Fields	VlocObjAttrsFields
Product	Product Details	ProductDetails
Product	Product Structure	ProductStructure
Product	Pricing	ProductPricing
Context Rules	ContextRules	ContextRules

Create a Product Specification

You create product specifications from the object types. All fields, attributes, and layouts defined at the object type level are inherited by product specifications. For product specifications, you set values for the attributes and fields defined for the object type.

An object type has one-to-many relationship with a product specification. In other words, you can create many product specifications from the same object type.

For example, for the Smartphone object type, you can create iPhone 6S Plus Product Spec and Samsung Galaxy S8 Product Spec.

After you define product specifications, you can associate them with prices, promotions, and context rules to create offers or commercial products that customers can purchase.

1. In the **Vlocity Product Console**, in the **Product Management** section, click the plus icon next to **Product**.
2. In the **New Product** page, complete the required fields:

Field Name	Required?	Description
Object Type	Yes	Select the object type you have previously created. For example, Smartphone type.
Product Name	Yes	Enter a name for the product. Product names appears to customers in the product catalog. Example of a name: iPhone 6S Plus.
Product Code	Yes	A unique code for the product. For example, VLO-MOB-0003.
Product Description	Yes	Description is displayed to customers in the product catalog.
Active	Yes	Activates the product so it can be associated with an offer. Deactivating a product after it's Active and available to sales channels can have multiple impacts, such as product hierarchies, rules, and existing orders. For example, a deactivated product is removed from Vlocity Cart, but the Vlocity Cart price isn’t changed.
Specification Type	Yes	Select the desired specification type.

3. Click **Save**.
4. Use the menu links on the left to complete additional fields that you have defined in **Layout Management** for the object type.

See Also

- Creating Products in Vlocity Product Console

Product Definition with the Salesforce Interface

A new product must have a description and usually includes attributes. You can exclude and update fields and attributes. A new product may also be a part of a product family. You define the product relationships and product and group cardinality. Decide if the product or offer is a line item. For example, you may want to offer individual television channels, but you don't want each to be available as a separate line item. You can create order line items that are not necessarily individual assets.

To set up a new product in Vlocity Communications, Media, and Energy, start with your company's model of what a product is. You're likely to load several products into Vlocity Communications, Media, and Energy at once. In some cases it may be possible to import or integrate your existing product catalog. You probably have a core set of picklist values and attributes that are reusable and are used over a variety of products and offers. After you have loaded the existing products into Vlocity, you must create new products and offers as your inventory expands.

NOTE

Starting with Vlocity Communications, Media, and Energy Summer '17 release, all new customers must define products using the Vlocity Product Console. For more information on how to define products in the Vlocity Product Console, refer to Defining Products Using Vocity Product Console.

For existing customers, to take advantage of the new functionality, Vlocity recommends upgrading to the latest release of Vlocity Communications, Media, and Energy which leverages the Vlocity Product Console. For information about the latest product release, see Vlocity Communications, Media, and Energy Release Summary.

When creating products, you start by creating attribute categories. Use product attributes to differentiate products instead of creating a separate product for each configuration. You can also price products based on attributes. For more information about attributes, see Product Attributes Overview.

There is one instance of every product. You configure products by selecting attributes. Attributes drive provisioning and pricing. Users and administrators can select attributes, or external systems can populate attributes.

Rules determine what's available. There are hard and soft rules. With a hard rule, if a product is not available, it is not displayed. With a soft rule, if a product is not available, it is displayed, but cannot be selected. Rules run when the cart changes. Nonsaleable products, such as services and resources, can be included to help with provisioning, but are not selectable. For more information about rules, see Set Up Rules and Entity Filters.

Vlocity supports product bundles. You can create a bundle that includes multiple phone choices, then create a rule to specify that there must be a minimum of one and a maximum of one. Child group cardinality specifies the minimum and maximum child products in a bundle. Bundles can also override cardinality, price, and attributes. The following is a sample bundle:

Sample Bundle

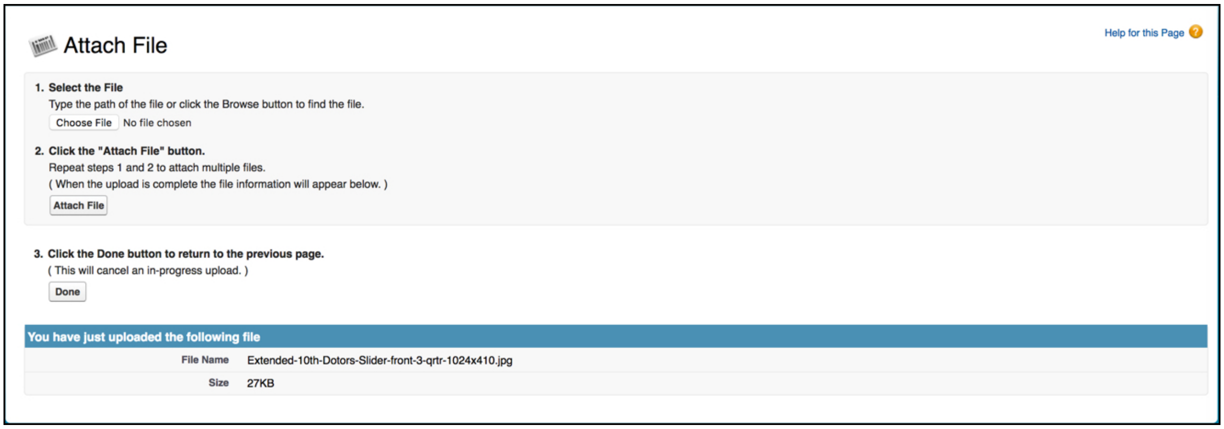
- Product (offer) = Internet 50 Mbps, SLA Best Effort, Router, 4 IP
- Product (specification) = Copper VDSL, SLA B, Cisco modern model, IP address
- Internet speed can be upgraded to 100 Mbps and 500 Mbps

Every product must be in the default price book.

Add an Image to a Product

You can upload files for an offering or specification, including images and documents. Previews of images and videos are shown in the Details tab for a product.

1. On the Products tab, click the product to which to add the image.
2. On the Product record detail page, scroll down to the **Notes & Attachments** related list, and click **Attach File**.



Attach File

1. Select the File
Type the path of the file or click the Browse button to find the file.
No file chosen

2. Click the "Attach File" button.
Repeat steps 1 and 2 to attach multiple files.
(When the upload is complete the file information will appear below.)

3. Click the Done button to return to the previous page.
(This will cancel an in-progress upload.)

You have just uploaded the following file

File Name	Extended-10th-Dotors-Slider-front-3-qrtr-1024x410.jpg
Size	27KB

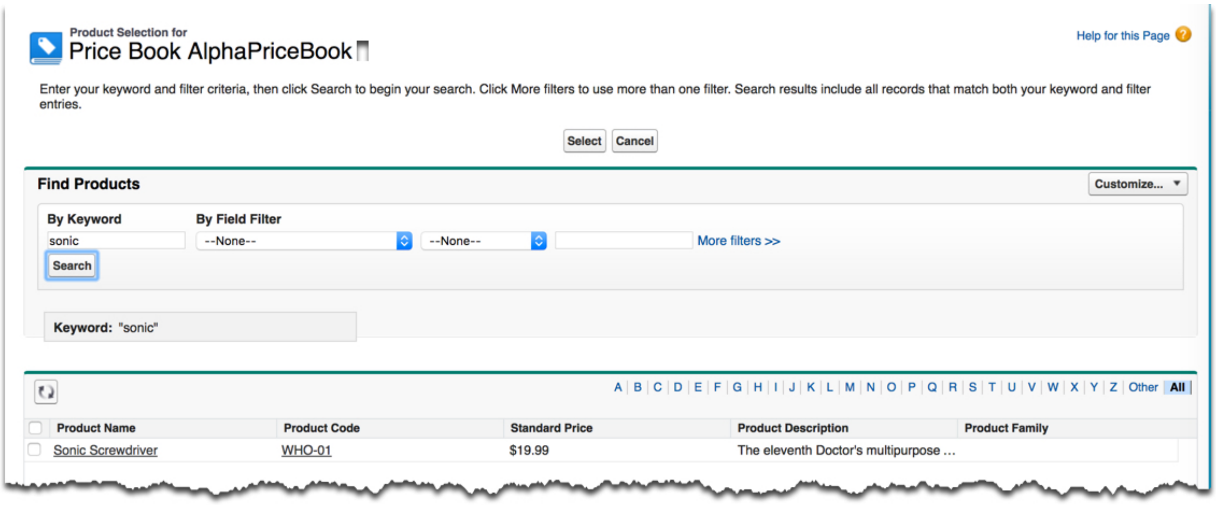
3. Click **Choose File**.
4. Find the file to upload.
5. Click **Open**.
6. Click **Attach File**.
7. To add more images, repeat steps 2 through 6.
8. When you are finished uploading images, click **Done**.

Associate a Product with a Price Book

Products must be associated with a price book.

To associate a product with a price book:

1. On the Price Books tab, from the **View** list, select **All**.
2. Click **Go!**.
3. Click the price book with which to associate the product.
4. In the **Products** related list, click **Add**.



Product Selection for

Price Book AlphaPriceBook

Enter your keyword and filter criteria, then click Search to begin your search. Click More filters to use more than one filter. Search results include all records that match both your keyword and filter entries.

Find Products

By Keyword: sonic

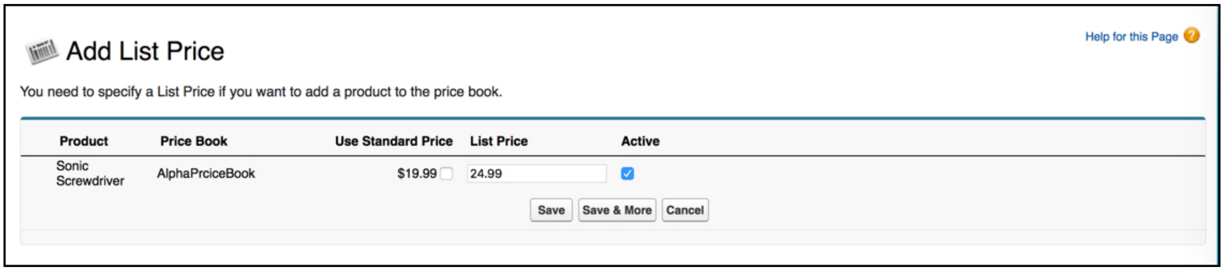
By Field Filter: --None-- --None--

Keyword: "sonic"

□	Product Name	Product Code	Standard Price	Product Description	Product Family
□	Sonic Screwdriver	WHO-01	$19.99	The eleventh Doctor's multipurpose ...	

Help for this Page
Select
Cancel
Customize...
More filters >>
Search
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All

5. Find and select the product to add to the price book.
6. Click **Select**.



Add List Price

You need to specify a List Price if you want to add a product to the price book.

Product	Price Book	Use Standard Price	List Price	Active
Sonic Screwdriver	AlphaPrciceBook	$19.99 □	24.99	☑

7. Select **Use Standard Price** or enter a **List Price**.
8. Click **Save**.

Configure Attributes and Fields for an Object Type

Attributes and fields you assign to the Base Object object type are later inherited by the child object types that you create. You can only assign attributes and fields that you have previously created.

For guidelines on when to use fields vs. attributes, refer to When to Use Attributes vs. Fields.

To assign attributes and fields:

1. Click the Base Object type.
2. In the left-hand-side menu, click **Attributes and Fields**.
3. Click **Assign Attributes/Fields**.
4. Select a checkbox for each attribute and field that is applicable to the base object.

5. Click **Assign**.

What's Next

Define Page Layout for an Object Type

See Also

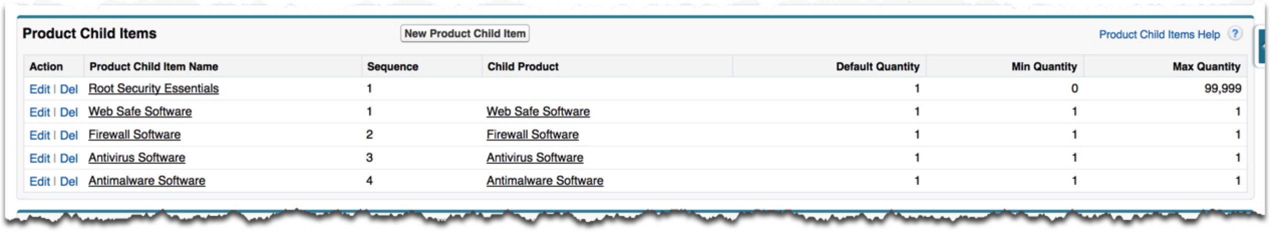
Creating Products in Vlocity Product Console

Add Product Child Items to a Product

Product child items (PCIs) are products associated with a bundle of products or services. A product can have one or more PCIs. Add PCIs by editing the Product record detail page. Every product contains a root PCI that defines the minimum and maximum number of child items for the root, or parent, product.

For more information about the root PCI, see Configure the Root Product Child Item.

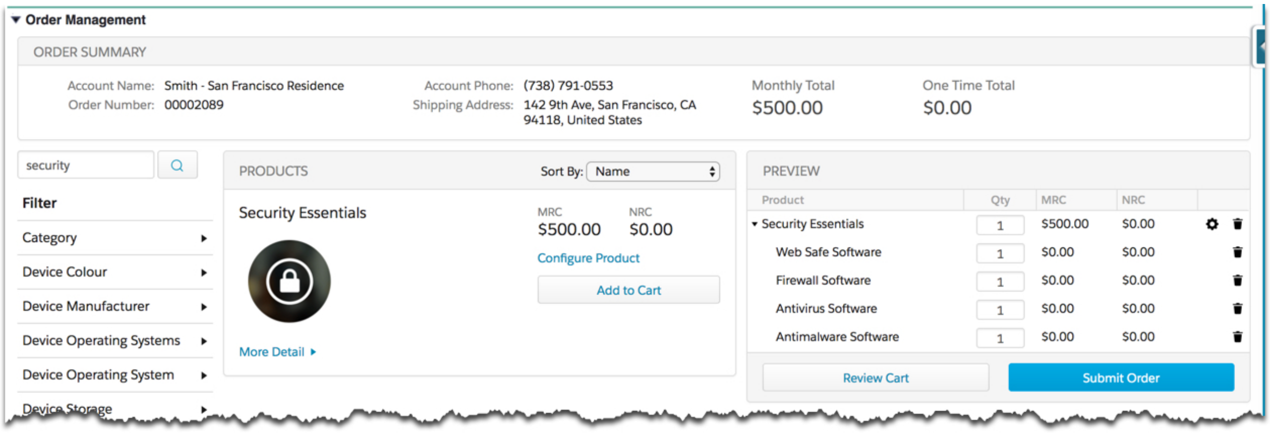
Figure 6. PCIs in the Security Essentials Bundle



Product Child Items — New Product Child Item — Product Child Items Help

Action	Product Child Item Name	Sequence	Child Product	Default Quantity	Min Quantity	Max Quantity
Edit | Del	Root Security Essentials	1		1	0	99,999
Edit | Del	Web Safe Software	1	Web Safe Software	1	1	1
Edit | Del	Firewall Software	2	Firewall Software	1	1	1
Edit | Del	Antivirus Software	3	Antivirus Software	1	1	1
Edit | Del	Antimalware Software	4	Antimalware Software	1	1	1

Figure 7. Product and Its Child Items in the Order Manager and Cart



▾ Order Management

ORDER SUMMARY

Account Name: Smith - San Francisco Residence
Order Number: 00002089

Account Phone: (738) 791-0553
Shipping Address: 142 9th Ave, San Francisco, CA 94118, United States

Monthly Total $500.00

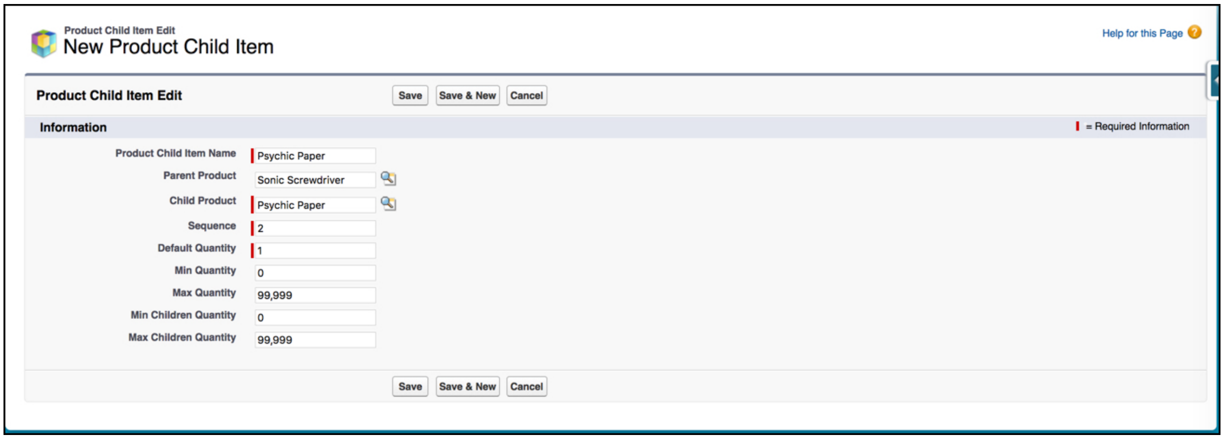
One Time Total $0.00

PREVIEW

Product	Qty	MRC	NRC
▾ Security Essentials	1	$500.00	$0.00
Web Safe Software	1	$0.00	$0.00
Firewall Software	1	$0.00	$0.00
Antivirus Software	1	$0.00	$0.00
Antimalware Software	1	$0.00	$0.00

To add a PCI to a product:

1. Go to the Product record detail page.
2. In the **Product Child** Items related list, click **New Product Child Item**.



Product Child Item Edit
New Product Child Item

Product Child Item Edit

Information

= Required Information

Product Child Item Name	Psychic Paper
Parent Product	Sonic Screwdriver
Child Product	Psychic Paper
Sequence	2
Default Quantity	1
Min Quantity	0
Max Quantity	99,999
Min Children Quantity	0
Max Children Quantity	99,999

3. On the **Product Child Item Edit** page, enter the following information:
   - **Product Child Item Name** is a name for the PCI.
   - **Parent Product** is a lookup to the parent product.
   - **Child Product** is a lookup to the child product's Product record.
   - **Sequence** is the placement of the PCI in the sequence of product child items.

   NOTE

   The **Sequence** box is meant to contain a number that indicates how the child appears within the parent product. However, the **Sequence** box allows non-numeric values. Make sure to enter only numeric values when using this feature.

   - **Default Quantity** is the default number of this PCI this product has.
   - **Min Quantity** is the minimum number of this product that the parent product can include. The minimum quantity must be less than or equal to the default quantity and the maximum quantity.
   - **Max Quantity** is the maximum number of this product that the parent product can include. The maximum quantity must be greater than or equal to the default quantity and the minimum quantity.
   - **Min Children Quantity** is the minimum number of child items all of the PCIs can include. That is, the sum of the quantities of an item's children cannot be less than the value defined in **Min Children Quantity**.
   - **Max Children Quantity** is the maximum number of child items all of the PCIs can include. That is, the sum of the quantities of an items children cannot be more than the value defined in **Max Child Quantity**.
4. Click **Save** or click **Save & New** to add more PCIs.

Cardinality of Child Products in a Bundle

A bundle is a set of products sold together, and it is created by adding Product Child Items under a product, on the Product page.

Product Child Items are component products associated with a package of products or services. A product can have one or more Product Child Items, which are added by accessing the product record.

Product Child Items must be added in their correct order hierarchy. All line items have a Line Number which controls the hierarchy of products, but this is not displayed in the Order Manager. The Child Line Number for the Product Child Item must follow the format 000x.000.

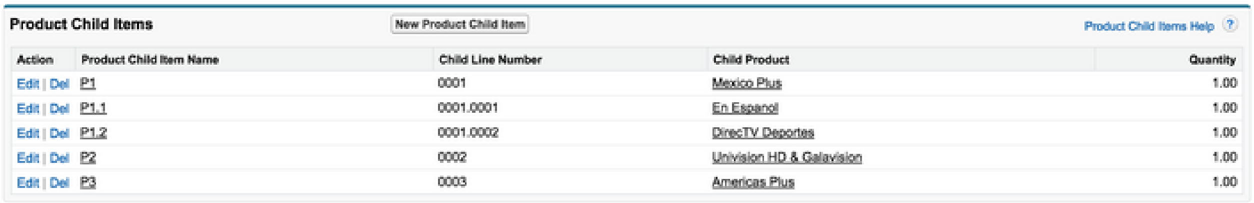
To add Product Child Items to a Product:

1. From the Product record, click New Product Child Item.
2. In the Product Child Item Name field, enter a name that indicates the order of the item—for example, P1, 1, or 1.0.
3. In the Child Product field, enter a child product or click the lookup icon to select the child product.
4. In the Child Line Number field, enter the correct number of the item in the hierarchy; for example, 0001.
5. If the quantity is different to 1, enter a new quantity.
6. Click Save or Save & New to add more products.
7. Repeat steps 1 through 6, adding products in the correct hierarchical order

Minimum and Maximum

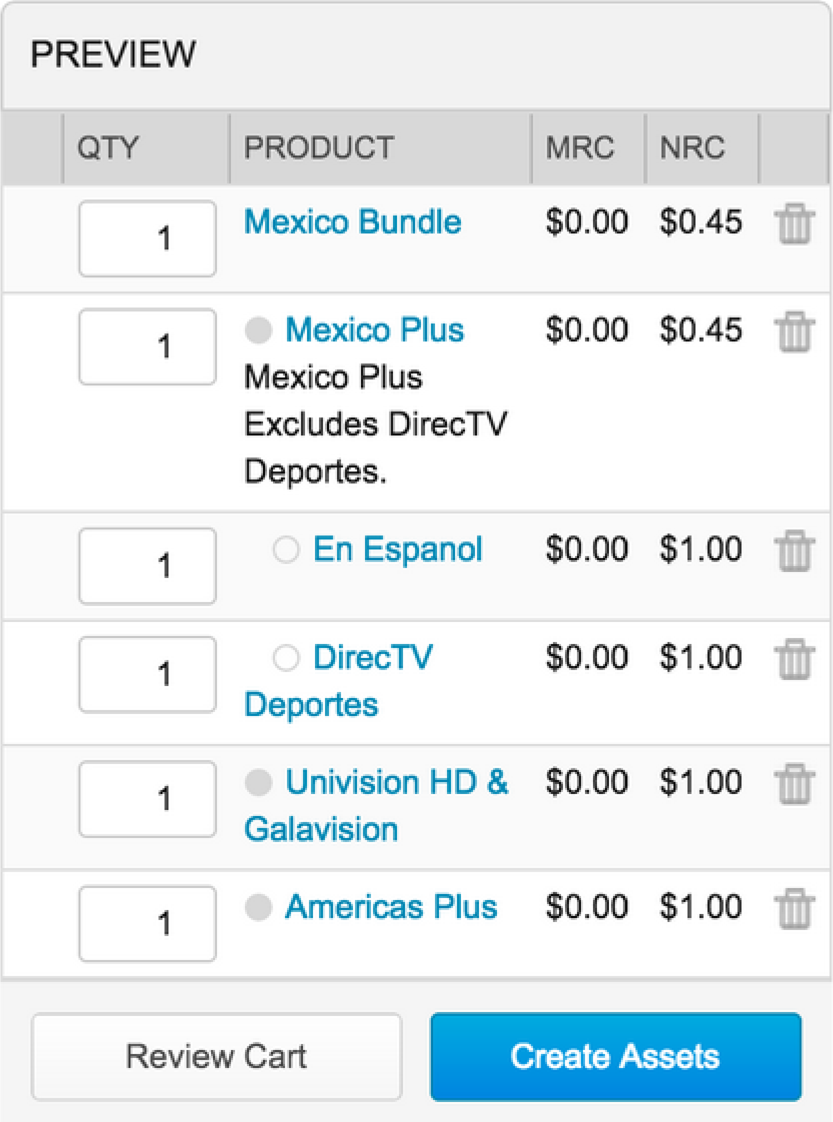
For each child item, there is a Min and a Max, which is set by the Admin who sets up the bundle. For instance, the Admin adds bundle A, which brings along Child1, Child2, and Child3 (say, Triple-Play as the parent and you get internet, catv, and phone). Let's say that the default is that you get one of each. But you might want 2 or 3 set-top boxes for your CATV. The Min and Max parameters would say that you must have at least the Min number and you cannot have more than the Max number. The code will roll back any quantity change outside of the Min and Max.

Figure 8. Example of Product Child Items in the Mexico Bundle Product



Action	Product Child Item Name	Child Line Number	Child Product	Quantity
Edit | Del	P1	0001	Mexico Plus	1.00
Edit | Del	P1.1	0001.0001	En Espanol	1.00
Edit | Del	P1.2	0001.0002	DirecTV Deportes	1.00
Edit | Del	P2	0002	Univision HD & Galavision	1.00
Edit | Del	P3	0003	Americas Plus	1.00

This will appear on an order, quote, opportunity as the following:



PREVIEW

QTY	PRODUCT	MRC	NRC
1	Mexico Bundle	$0.00	$0.45
1	● Mexico Plus Mexico Plus Excludes DirecTV Deportes.	$0.00	$0.45
1	○ En Espanol	$0.00	$1.00
1	○ DirecTV Deportes	$0.00	$1.00
1	● Univision HD & Galavision	$0.00	$1.00
1	● Americas Plus	$0.00	$1.00

Code to Enforce Cardinality of ProductChildItems

```
DefaultHandleActionImplementation.updateProduct() when update is to Quantity
if LineNumber__c length() < 5 this is top level item skip constraint logic
query to get Product2.Id and Name from PricebookEntry which allows query
for ProductChildItem Min / Max
if Min < >null - need to enforce contraint
if Qty >= Min && Qty <= Max update Qty
else create message which posts to exception
else - update Qty
else - update Qty
```

Add Multiple Price Types for a Product

Vlocity Communications supports multiple price types for Products in a Price Book, including Recurring Price and Overage Charge. After a Product has been added to a Price Book, you can add these and other price types for that Product.

For more information about price books, see Products Concepts in the Salesforce Help.

To modify price types for a product:

1. From a Product page, click **Edit** next to a Price Book.
2. Enter any additional pricing information for the product and click **Save**.

NOTE

To modify the Recurring Unit of Measure (UOM) and Overage UOM picklist values, from Setup, click **Customize** and then click, **Price Books**, and then click **Price Book Entries**, and then click **Fields**. Select **Recurring UOM** or **Overage UOM** to add, modify, or delete values.

Update Products Designed in Aloha to Use the Product Console

The IsConfigurable__c flag was used differently in the Aloha interface than it is used in Product Console. To update the flag to manage products in the Vlocity Product Console, you must run the UpdateAttributeConfigurableBatchJob. The input includes a date/time. All attribute assignments created before that date are updated.

The UpdateAttributeConfigurableBatchJob changes the flag according to the following logic:

If IsReadOnly__c is False and IsConfigurable__c is False, then set IsConfigurable__c to True (Otherwise don't change IsConfigurable__c)

The UpdateAttributeConfigurableBatchJob updates the IsConfigurable__c flag for products with attributes created in Aloha.

To run the UpdateAttributeConfigurableBatchJob:

1. From the **User** menu, choose **Developer Console**.
   The **Force.com Developer Console** window opens.
2. From the **Debug** menu, choose **Open Execute Anonymous Window**.
   The **Enter Apex Code** dialog box opens.
3. Delete any code that appears in the **Enter Apex Code** dialog box.
4. Paste the following statement into the **Enter Apex Code** dialog box, replacing the date and time with the appropriate date and time for your data and replacing **vlocity_cmt** with your org namespace:

```
Datetime createdBeforeDate = Datetime.newInstanceGmt(2016, 1, 1, 0, 0, 0);
//example, change to reflect your date
vlocity_cmt.UpdateAttributeConfigurableBatchJob epcBatch = new
vlocity_cmt.UpdateAttributeConfigurableBatchJob(createdBeforeDate);
Database.executeBatch(epcBatch);
```

5. Click **Execute** .
6. Close Developer Console.

Product Attributes

Product attributes define product characteristics and can be categorized. You can use attributes to filter products. Pricing is based on product attributes.

> **NOTE**
>
> In Vlocity Communications, Media, and Energy Summer '17, use the EPC Product Console to create attributes. For more information, see Create Products in Vlocity Product Console.

Vlocity has extended the Product object to support product attributes. Attributes are organized using *attribute categories*. Product attributes define product specifications. For example, a phone might come in four different colors, three different sizes, and so on. You define attribute categories, such as color and size, to accommodate these options. Using product attributes, you can filter products to display only those available to a specific customer. You can also base pricing on product attributes. For more information, see Use Cases for Calculation Procedures.

Products can be categorized and filtered by custom attributes. The product administrator can add these attributes to the products in a price book. All product records support attributes. Product attributes can be filterable and configurable.

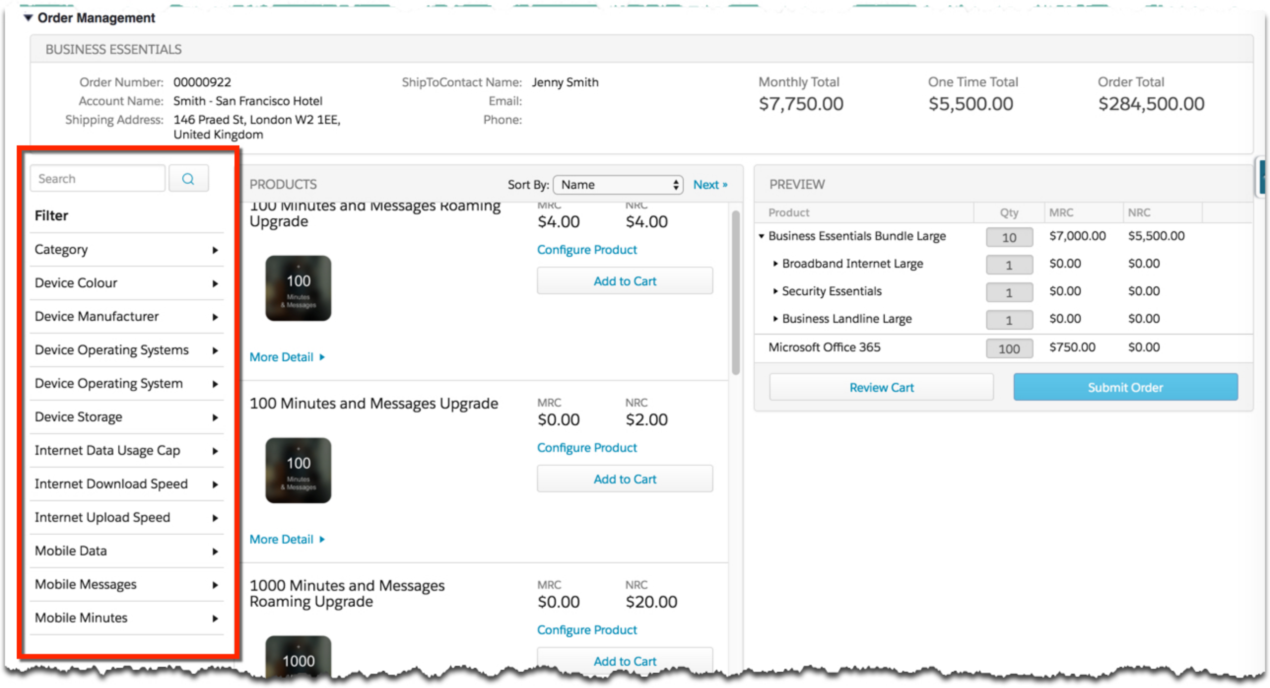
- If a product attribute is filterable, you can use it as a filter in the Opportunity Manager, Order Manager, or Quote Manager.
- If a product attribute is configurable, the user can configure the attribute after adding the product to the cart.

Pricing can depend on rules based on attributes. The attributes drive the pricing. You can offer a product with pricing based on its attributes. For example, you can offer broadband Internet service at different prices for 50 Mbps, 100 Mbps, and so on. You can also define rules that specify one price for a product by itself, but a different price when the Product is part of a bundle.

When searching for products in the CPQ user interface, you can filter the product list based on the product attributes that have been applied. When you add a product to your cart, you can click the product in the cart to configure the item attributes.

Attributes are flexible. There are no limits to the number of attributes a product can have. Each attribute corresponds to one field. Using attributes avoids some performance, search, and security issues. Pre- and post-processing steps can use attributes.

Figure 9. Order Manager with Attributes Displayed



▼ Order Management

BUSINESS ESSENTIALS

Order Number:	00000922	ShipToContact Name:	Jenny Smith	Monthly Total	One Time Total	Order Total
Account Name:	Smith - San Francisco Hotel	Email:		$7,750.00	$5,500.00	$284,500.00
Shipping Address:	146 Praed St, London W2 1EE, United Kingdom	Phone:				

Search

Filter

- Category ▸
- Device Colour ▸
- Device Manufacturer ▸
- Device Operating Systems ▸
- Device Operating System ▸
- Device Storage ▸
- Internet Data Usage Cap ▸
- Internet Download Speed ▸
- Internet Upload Speed ▸
- Mobile Data ▸
- Mobile Messages ▸
- Mobile Minutes ▸

PRODUCTS — Sort By: Name — Next »

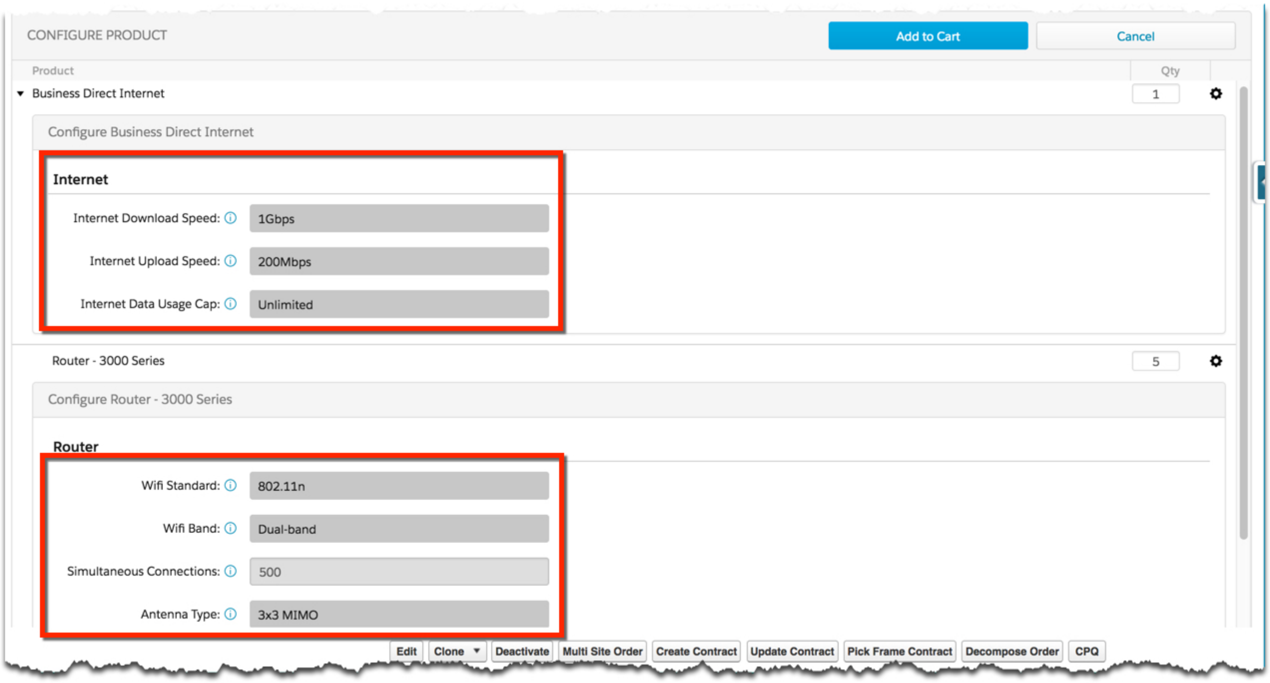
Product	MRC	NRC
100 Minutes and Messages Roaming Upgrade	$4.00	$4.00
100 Minutes and Messages Upgrade	$0.00	$2.00
1000 Minutes and Messages Roaming Upgrade	$0.00	$20.00

PREVIEW

Product	Qty	MRC	NRC
▾ Business Essentials Bundle Large	10	$7,000.00	$5,500.00
▸ Broadband Internet Large	1	$0.00	$0.00
▸ Security Essentials	1	$0.00	$0.00
▸ Business Landline Large	1	$0.00	$0.00
Microsoft Office 365	100	$750.00	$0.00

You can use attributes to configure products before or after they have been added to the cart.

Figure 10. Product with Configurable Attributes



Business Direct Internet — 1

Configure Business Direct Internet

Internet

Internet Download Speed: 1Gbps

Internet Upload Speed: 200Mbps

Internet Data Usage Cap: Unlimited

Router - 3000 Series — 5

Configure Router - 3000 Series

Router

Wifi Standard: 802.11n

Wifi Band: Dual-band

Simultaneous Connections: 500

Antenna Type: 3x3 MIMO

Create New Product Attribute Categories

Attribute categories group attributes. They also appear as headings at run time in Vlocity Cart's Filter and Configuration windows. Attribute categories are required to create product attributes.

The type of attribute is defined at the Attribute Category level. You must create Attributes Categories before adding new attributes to the category.

For more information about housing listings or filters, see Configure Housing Listing Filters.

To create a product attribute category:

1. On the Vlocity Attribute Designer tab, click **New**.
2. In the **Name** field, enter the attribute category name (for example, `Color`).
3. In the **Display Sequence** field, enter a number to indicate the sequence in which the category is to be displayed on the record detail page or Housing Listing.
4. In the **Code** field, enter a unique code for the category.

> **TIP**
>
> The code is not visible in the user interface.

5. Select the **UI Control Type** for the attributes in the new category.
   Each attribute category has a specific user interface control type for its attributes:
   - Select **On-Off** to add or remove attributes from categories.
   - Select **3-State** to set attributes to positive, neutral, or negative.
   - Select **1-5 Scale** to set attributes to a number from 1 to 5.
   - Select **Text Comment** to add text to attributes.
   - Select **Multi-Select** for housing filters.
6. From the **Applicable Types** list, select **Product2**.
   When creating housing filters, select **Any**.
7. From the **Applicable Subtype** picklist, select one of the following:
   - **Product Attribute**: For Product records.
   - **Profile Attribute**: For Profile records. Profile attributes are used with Vlocity Intelligence.
   - **Inventory Item**: For attributes to set up housing filters.
8. Select **Filterable** to be able to use the attribute category to filter products or housing.
9. Update the **Color Code** is used to determine the color of the category tag. Enter the color code using a hex value.
10. Click **Save**.

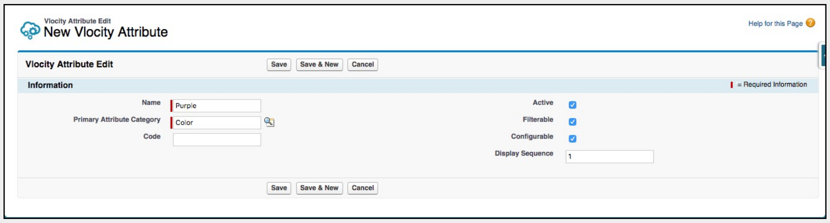
Create a New Product Attribute

Product attributes further define products. For example, an iPhone might have attributes for model, color, storage capacity, and so on. You add attributes to attribute categories that you have previously defined.

Before You Begin
Create an attribute category. See Create a New Product Attribute Category.

To create a new product attribute:

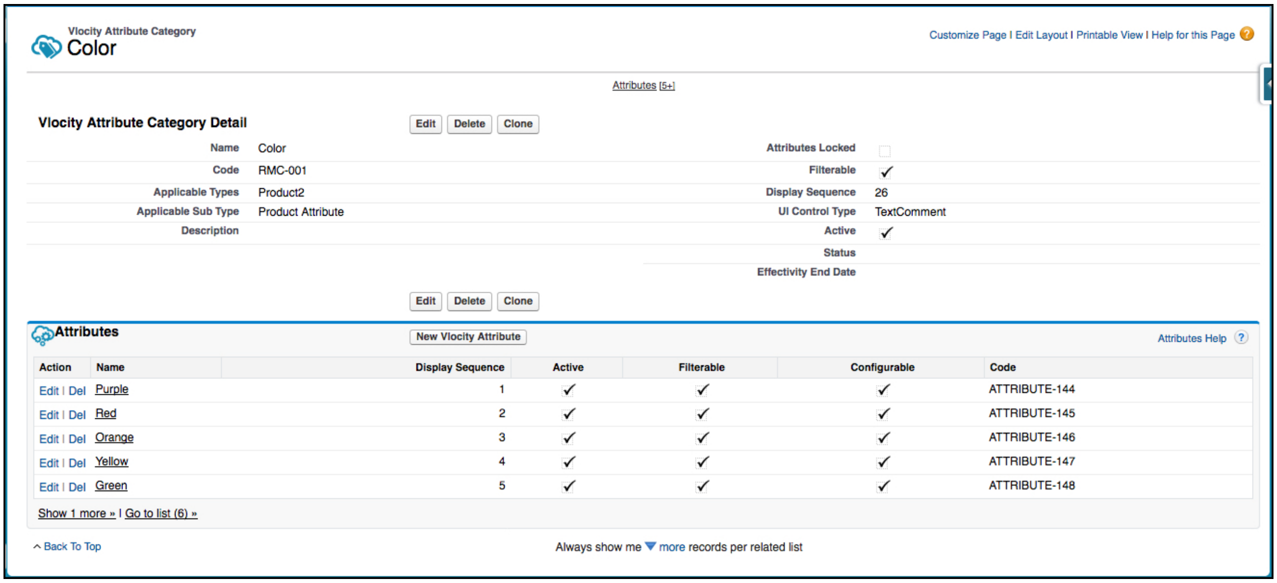
1. On the Vlocity Attribute Categories tab, display all attributes.
2. Click the category to which you want to add the attribute.
3. To create the new attribute, go to the platform-appropriate page or dialog box, as follows:

If you're using	Do this
Salesforce Lightning Experience	1. Click **Related**. 2. Click **Attributes**. 3. Click **New**.
Salesforce Classic (Aloha)	In the **Attributes** related list, click **New Vlocity Attribute**.  Vlocity Attribute Edit New Vlocity Attribute Help for this Page Vlocity Attribute Edit Save / Save & New / Cancel Information = Required Information Name: Purple Primary Attribute Category: Color Code: Active ☑ Filterable ☑ Configurable ☑ Display Sequence: 1 Save / Save & New / Cancel

4. Enter the following information:

- **Name** is the attribute name.
- **Code** is a unique code for the attribute. DataRaptors use the code instead of the name.
- **Primary Attribute Category** is the category to which the attribute belongs.
- **Active** indicates if the attribute is currently used.
- **Filterable** indicates if customers can use the attribute to filter search results.
- **Configurable** indicates if the attribute can be configured.
- **Display Sequence** is the sequence in which the attribute should appear within the attribute category.

5. Click **Save**.


Vlocity Attribute Category
Color
Customize Page | Edit Layout | Printable View | Help for this Page
Attributes [5+]
Edit
Delete
Clone
New Vlocity Attribute
Attributes Help
Back To Top

Vlocity Attribute Category Detail

Name	Color	Attributes Locked	□
Code	RMC-001	Filterable	✓
Applicable Types	Product2	Display Sequence	26
Applicable Sub Type	Product Attribute	UI Control Type	TextComment
Description		Active	✓
		Status	
		Effectivity End Date	

Attributes

Action	Name	Display Sequence	Active	Filterable	Configurable	Code
Edit | Del	Purple	1	✓	✓	✓	ATTRIBUTE-144
Edit | Del	Red	2	✓	✓	✓	ATTRIBUTE-145
Edit | Del	Orange	3	✓	✓	✓	ATTRIBUTE-146
Edit | Del	Yellow	4	✓	✓	✓	ATTRIBUTE-147
Edit | Del	Green	5	✓	✓	✓	ATTRIBUTE-148

Show 1 more » | Go to list (6) »

Always show me more records per related list

Set Up Product Attributes

After creating attribute categories and attributes, you can apply those attributes to products. End users can find and filter products based on attribute values.

When setting up product attributes, you can select the **Customizable** checkbox. Doing so reveals more options. You can select the user interface element for the attribute.

The following table lists valid formats for product attributes. Based on the attribute settings, such as required, minimum, maximum, minimum length, and maximum length, some input elements will display standard error messages. For example, if an attribute is required, the input element may read "Please enter a value for this field."

Attribute Format	User Interface Elements	Restrict Length or Value
Currency	Slider or single-value text box	Value, Min and Max
Percent	Slider or single-value text box	Value, Min and Max
Text	Text box or text area	Length, Min Length and Max Length

Attribute Format	User Interface Elements	Restrict Length or Value
Number	Slider or single-value text box	Value, Min and Max
Checkbox	Selected or cleared (not selected)	n/a
Date	Calendar	n/a
Datetime	Calendar, time	n/a
Picklist	Radio button or drop down	n/a
Multipicklist	Checkbox or drop down	n/a

You can use a format mask to constrain attribute values. Do not use a format mask to restrict the length or value of attributes. For more information about the attribute format mask, see Constrain Product Attributes Using the Format Mask.

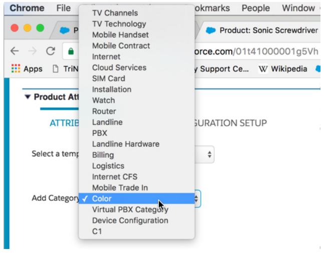
You can also set up an attribute to use a custom action. For more information, see Adding a Custom Action to a Card.

If you change the attribute setup on a product, and that product is in use in order line items elsewhere in the system, the attributes on those line items do not change.

Attributes are stored in the JSONAttribute object. Vlocity Omnichannel APIs can use the code in this object. You can edit the JSON directly, but, if you do, do not edit the attribute using the Vlocity user interface.

To apply attributes to a product:

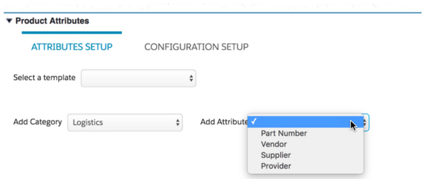
1. On the **Product** record detail page, in the **Product Attributes** section, from the **Add Category** picklist, select the attribute category.



If the attribute category has more than one attribute associated with it, the **Add Attribute** picklist is displayed.

> **NOTE**
>
> If you have created product attribute templates, you can apply them to other products. From the **Select a template** picklist, select the product attribute template. For more information about creating product attribute templates, see Create a Product Attribute Template.

2. From the **Add Attributes** picklist, select the attribute.



Add Category: Logistics

3. Select **Customizable** to reveal more attribute properties.
4. From the first **Attributes Setup** picklist, select the information format for the attribute.
5. From the second **Attributes Setup** picklist, select how the attribute initially appears in the user interface.
6. In the **Value** box, enter the default value for the attribute.
7. For attributes that are numbers or text, enter the minimum and maximum values for this attribute.
   - For attributes that require text, the **Min Length** and **Max Length** boxes control how many characters can be entered.
   - For attributes that are numbers—currency, percent, and number—the **Min** and **Max** boxes control the numeric value.
8. In the **Description** box, enter a description for the attribute. This text is displayed when a user hovers over the attribute.
9. Select **Required** to require this attribute for product configuration. Required attributes appear in the cart with asterisks (*) next to them.
10. Select **Read Only** if the user cannot change the value of this attribute.
11. Select **Hidden** to hide this attribute in the user interface.
12. To optimize performance when a rule pertains to an attribute, select **Has Rule**.

    For example, you could have a rule that specifies when a processor has a certain speed, offer a specific amount of RAM. For more information about rules, see Set Up Rules and Entity Filters.
13. To apply a format mask, see Constrain Product Attributes Using the Format Mask.
14. To apply a custom cards-based template, in the **Custom Template** box, enter the template name.
15. To use a custom action, click **More Properties** and perform the following steps
    a. Select **Use custom action**.
    b. Enter a label for the button.
    c. Enter the Apex code for the custom action.
16. To add more attributes, select them from the **Add Attribute to Logistics** picklist.
17. Click **Save Product Attributes**.
18. To configure attribute rules, see Configure Attribute Rules.

Configuring the Attribute Values Sort Order in Picklists

To control the order in which attribute values are displayed in a picklist in Vlocity carts, set the `AttributeValueDisplaySortMode` custom setting to the name of the field to be used for sorting. For example, if your attributes are composed of three fields, `DisplaySequence`, `Id`, and `Value`, and you want attribute values listed alphabetically, set `AttributeValueDisplaySortMode` to `"Value"`.

If you prefer to configure the sort order by specifying a numeric index, set `AttributeValueDisplaySortMode` to `"DisplaySequence"` and the the sort mode will use the Sequence property of the picklist items to sort the picklist item values.

Constrain Product Attributes Using a Format Mask

You can constrain attribute values using a format mask. For example, if a field contains a telephone number, you can define an attribute mask that constrains the value to an appropriate set of data, such as `(999) 999-9999`. By applying a format mask, you ensure that a user can type only in a predetermined pattern. The format mask is a part of setting up product attributes on a product.

NOTE

Do not use a format mask to restrict the length or numeric value of an attribute. Use the **Min** and **Max** or **Min Length** and **Max Length** boxes instead. See Create a New Product Attribute.

An attribute format mask can use the following characters:

- The letter A represents any letter.
- The number 9 represents any number.
- An asterisk (`*`) represents any letter or number.
- A question mark (?) means that part of the mask is optional.

You can also use any character or a space as a default placeholder character. If you omit a placeholder character, the underscore (_) is used.

Configure Attribute Rules

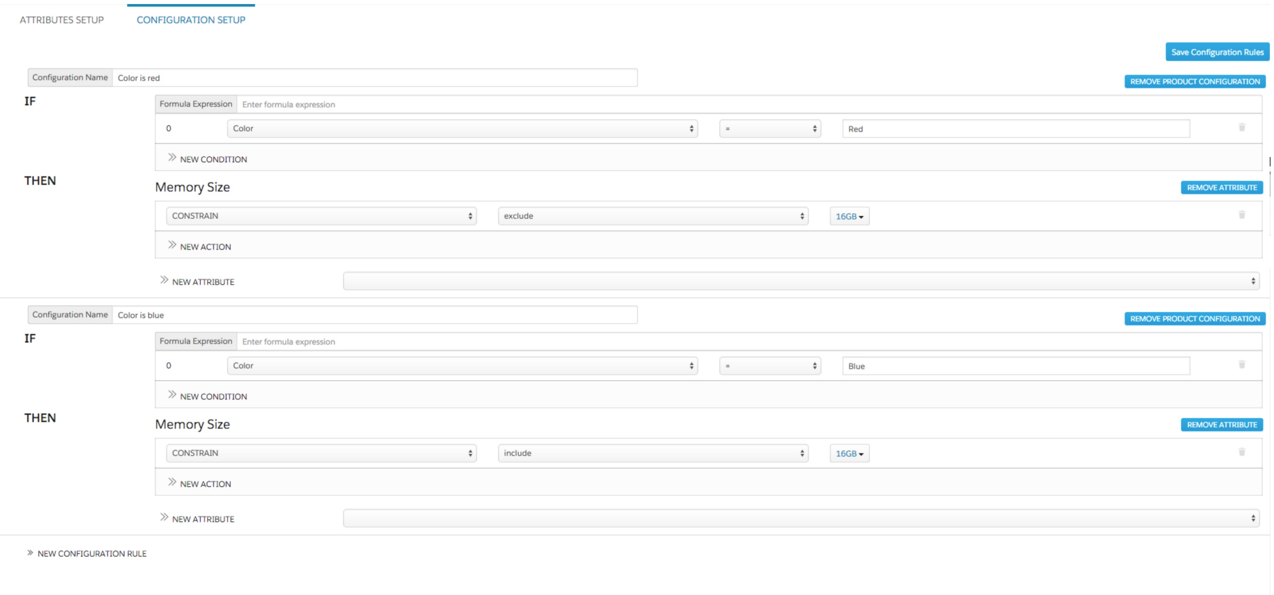
You can display attributes and attribute values that correspond to specific product configurations. Product compatibility rules support attribute configuration. Configuration evaluations are server-side and actions are client-side. Changing an attribute triggers attribute configurations.

You must explicitly define all required behavior. After an action is applied, only another configuration can change it. For example, you can define a product configuration where changing the color attribute affects the memory attribute choices.

Each configuration requires two sections—the IF section and the THEN section, also known as the condition section and the action section.

- Use the IF section to create the conditions that trigger the configuration.
- Use the THEN section to define the actions.

The configuration in the following image defines that, If the Color attribute is Red, then the Memory Size attribute must exclude 16GB. If the Color attribute is Blue, then the Memory Size attribute must include 16GB.



Configuration Name: Color is red

IF

Formula Expression: Enter formula expression

0 Color = Red

THEN

Memory Size

CONSTRAIN exclude 16GB

Configuration Name: Color is blue

IF

Formula Expression: Enter formula expression

0 Color = Blue

THEN

Memory Size

CONSTRAIN include 16GB

The configuration setup is similar to setting up rules, which include, from left to right:

- index number
- attribute
- operator
- value

In the example above, 0 is the index number, Color is the attribute, the equal sign (=) is the operator, and Red is the value.

To configure attribute rules:

1. On the Product record detail page, in the **Product Attributes** section, click **Configuration Setup**.
2. Click **New Configuration Rule**.
3. In the **Configuration Name** box, enter a name for the rule.
4. Click **New Condition**.
5. Select the attribute.
6. Select the operator.
7. Enter the value.
8. Repeat steps 4 through 7 to enter more conditions.
9. If the action should occur when any one of several conditions is met, in the **Formula Expression** box, enter the appropriate expression.
   A formula expression indicates if the rule action occurs when all conditions are met or when one condition is met. By default, rules use and. Using a formula expression, you can specify that a rule uses or, for example, `{index` *`x`*`} OR {index` *`y`*`}`.
10. From the **New Attribute** picklist, select the attribute.

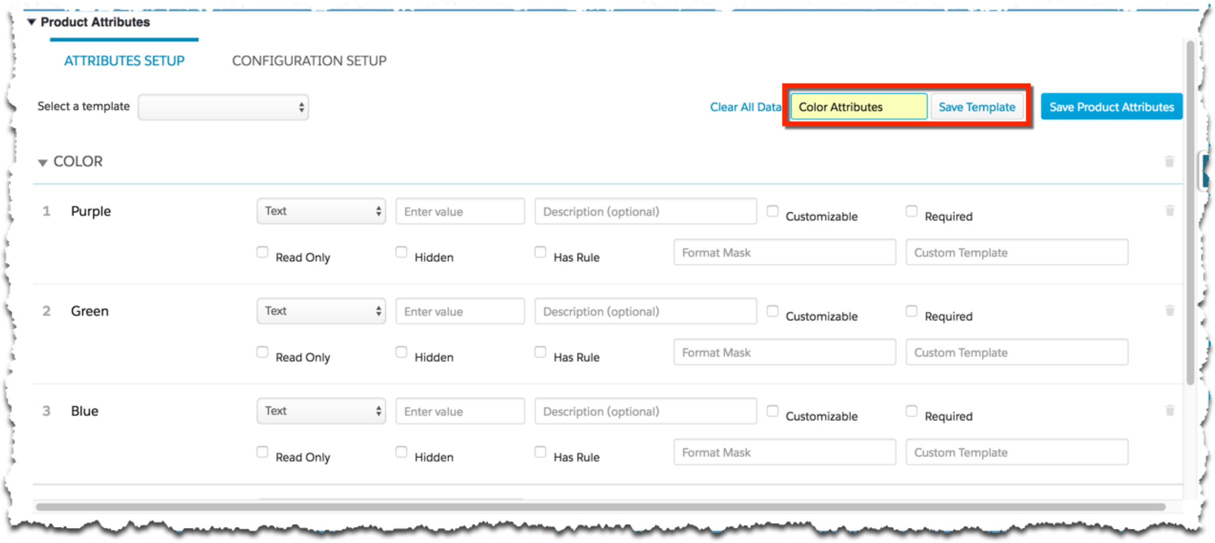
11. Select the appropriate action.
    The following actions are available:
    - **Constrain** includes or excludes specific picklist choices. It is only applicable for attributes in picklists.
    - **Assign** assigns values to any type of attribute.
    - **Disable** disables any attribute.
    - **Require** indicates that an attribute is required.
    - **Hide** hides any attribute in the user interface. The attribute definition is in the JSON but is hidden from the user.
12. To add more actions, click **New Action** and repeat step 11.
13. When you are finished, click **Save Configuration Rules**.

Create a Product Attribute Template

If you have a set of attributes that you can apply to multiple products, create a *product attribute template* to speed the assignment of attributes to a product. You can create templates using attribute settings that you have already defined on a product.

To create a product attribute template:

1. On any product, set up product attributes. See Create a New Product Attribute.



2. In the **Enter Template Name** box, enter a name for the product attribute template.
3. Click **Save Template**.

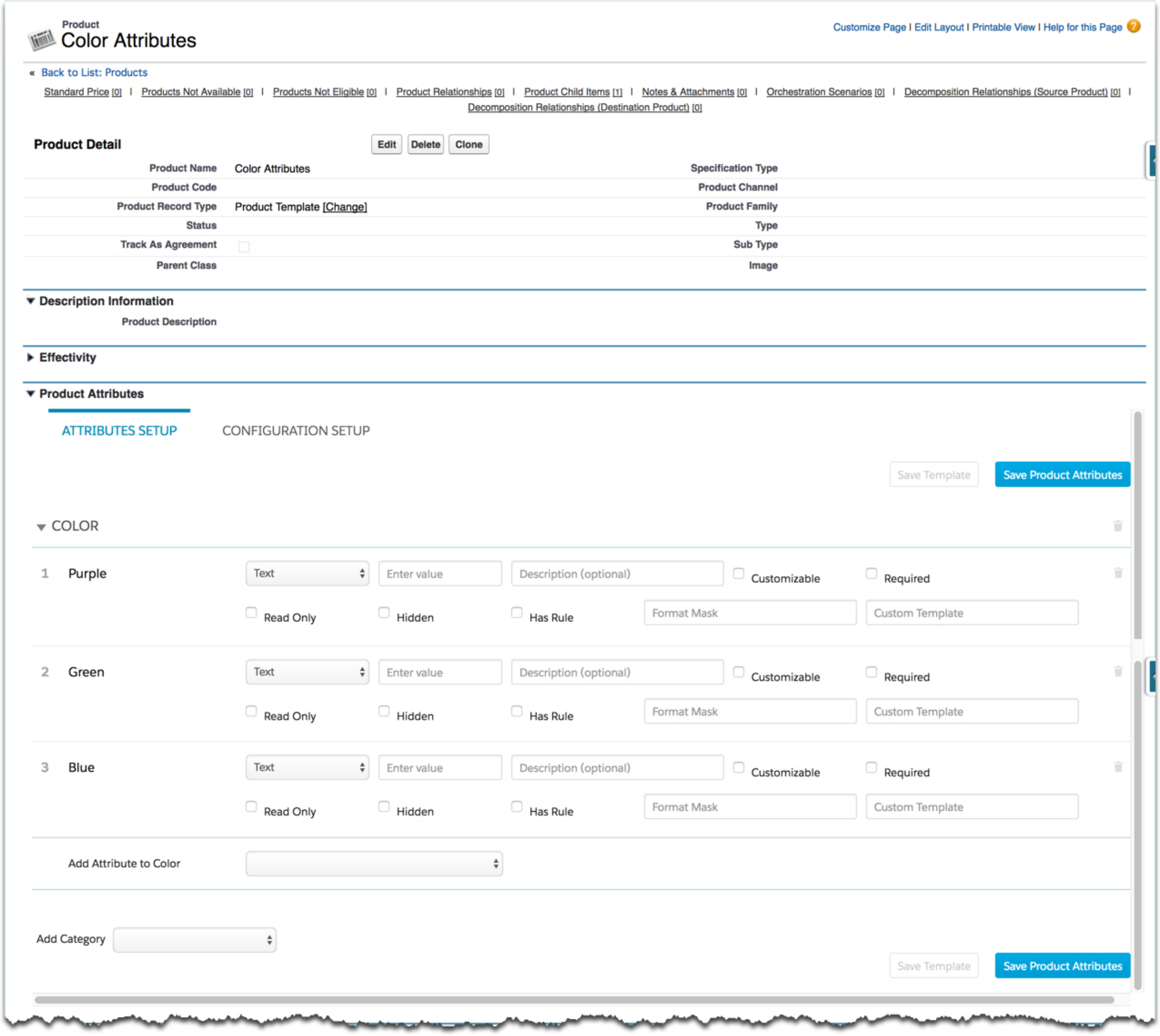
Product attribute templates appear in the products list. To apply a product attribute template to a product, when you are setting up attributes for that product, from the **Select a template** picklist, select the product attribute template to use.

Edit a Product Attribute Template

Product attribute templates are displayed in the products list. You can edit them in the same way that you edit product attributes on a Product record detail page.

To edit a product attribute template:

1. On the **Products** tab, click the product attribute template name.
2. Edit as desired. For more information, see Create a New Product Attribute.



Product
Color Attributes

Customize Page | Edit Layout | Printable View | Help for this Page

« Back to List: Products

Standard Price [0] | Products Not Available [0] | Products Not Eligible [0] | Product Relationships [0] | Product Child Items [1] | Notes & Attachments [0] | Orchestration Scenarios [0] | Decomposition Relationships (Source Product) [0] | Decomposition Relationships (Destination Product) [0]

Product Detail Edit Delete Clone

Product Name	Color Attributes	Specification Type	
Product Code		Product Channel	
Product Record Type	Product Template [Change]	Product Family	
Status		Type	
Track As Agreement	□	Sub Type	
Parent Class		Image	

▼ Description Information

Product Description

▶ Effectivity

▼ Product Attributes

ATTRIBUTES SETUP CONFIGURATION SETUP

Save Template Save Product Attributes

▼ COLOR

1	Purple	Text	Enter value	Description (optional)	□ Customizable	□ Required
		□ Read Only	□ Hidden	□ Has Rule	Format Mask	Custom Template
2	Green	Text	Enter value	Description (optional)	□ Customizable	□ Required
		□ Read Only	□ Hidden	□ Has Rule	Format Mask	Custom Template
3	Blue	Text	Enter value	Description (optional)	□ Customizable	□ Required
		□ Read Only	□ Hidden	□ Has Rule	Format Mask	Custom Template

Add Attribute to Color

Add Category

Save Template Save Product Attributes

Fixing JSON Attributes

In Vlocity, each Product object contains a JSONAttributes field. In some cases, the JSONAttributes fields can be out of sync with the attributes that are assigned to the products. When this happens, there are batch jobs that you can run to sync up the Product JSONAttributes fields with the product attributes.

The batch job you run depends on how and where you defined your products.

- For products that are managed in the Vlocity Product Designer or Vlocity Product Console, run the EPCProductAttribJSONBatchJob. Run this job when you add an attribute to an existing object type that is associated with products.
- For products that are managed in the Aloha interface using the attributes set up, run the FixProductAttribJSONBatchJob.

There is one additional batch job that fixes a different issue that may occur with the JSONAttribute field. When products have attribute overrides, run the EPCFixCompiledAttributeOverrideBatchJob to regenerate the JSONAttribute fields for those overrides. Run this job when you add an attribute to an existing object type that is associated with products and an associated product has at least one attribute with an override.

Fixing Filterable Attributes In Vlocity Cart

To use filter products in Vlocity Cart, you must create filterable product attributes, assign them to products, and run the ProductAttribJSONBatchJob Update All to populate the filterable attributes cache. If your org has products with attributes that were created in another org—perhaps you have a sandbox copy or you have loaded products and their attributes using DataRaptor—when you run the ProductAttribJSONBatchJob Update All job by clicking the **Refresh Pricebook** button, you will find that the Filter button in Vlocity Cart is blank or products are missing attributes because the batch job failed with the error:

```
First error: Insert failed. First exception on row 0: first error
INSUFFICIENT_ACCESS_ON_CROSS_REFERENCE_ENTITY, insufficient access rights on
cross-reference id: id.
```

You receive this error because each product’s attributes are stored in a JSONAttribute field. The JSONAttribute field contains attribute record IDs, to enable faster direct access for CPQ processes. However, when creating new orgs, the Salesforce copy process is unable to detect those record IDs embedded in the JSONAttribute field. The record IDs remain coded as the original attribute record IDs from the original org. Left untouched, these record IDs will create errors in the ProductAttributesBatchProcessor.

If you created and manage products in the Salesforce Aloha interface, there are two steps. First, you must define the ProductAttribJSONBatchJob Update All custom setting. Then, you must run the FixProductAttribJSONBatchJob. For more information, see Defining the ProductAttribJSONBatchJob Update All Custom Setting and Running the FixProductAttribJSONBatchJob.

If you created and manage products in the Vlocity Product Console, you must run the EPCProductAttribJSONBatchJob. For more information, see Running the EPCProductAttribJSONBatchJob.

After running the appropriate batch job, refresh the price book and check the Apex Jobs log. The errors should be resolved. For more information about refreshing the price book, see Refresh the Price Book.

After the ProductAttribJSONBatchJob has been run once, the ProductAttribJSONBatchJob custom setting displays the timestamp of the last time the batch job was run.

Running the FixProductAttribJSONBatchJob

For products that are managed in the Aloha interface using the attributes set up, run the FixProductAttribJSONBatchJob to regenerate the Product object JSONAttribute fields. Run the

FixProductAttribJSONBatchJob to sync the Product object JSONAttribute fields with the attributes that are assigned to the products.

By default, the FixProductAttribJSONBatchJob creates the JSONAttribute field only for the products for which the JSONAttribute field is null. To update all products, you must define the ProductAttribJSONBatchJob Update All setting.

NOTE
If you receive the error INSUFFICIENT_ACCESS_ON_CROSS_REFERENCE_ENTITY before you run the FixProductAttribJSONBatchJob, you must define the ProductAttribJSONBatchJob Update All custom setting. For more information, see Defining the ProductAttribJSONBatchJob Update All Custom Setting.

To run the FixProductAttribJSONBatchJob:

1. From the **User** menu, choose **Developer Console**.
   The **Force.com Developer Console** window opens.
2. From the **Debug** menu, choose **Open Execute Anonymous Window**.
   The **Enter Apex Code** dialog box opens.
3. Delete any code that appears in the **Enter Apex Code** dialog box.
4. Paste the following statement into the **Enter Apex Code** dialog box, replacing `vlocity_cmt` with your namespace:

```
vlocity_cmt.FixProductAttribJSONBatchJob batch = new
vlocity_cmt.FixProductAttribJSONBatchJob();
Id batchJobId = Database.executeBatch(new
vlocity_cmt.FixProductAttribJSONBatchJob());
```

5. Click **Execute**.
6. Close Developer Console.

Defining the ProductAttribJSONBatchJob Update All Custom Setting

NOTE
This topic is applicable to Vlocity Communications, Media, and Technology V15. For Vlocity Communications, Media, and Energy Summer '17 and later, see Running the EPCProductAttribJSONBatchJob.

By default, the ProductAttribJSONBatchJob Update All job queries for products with attributes with blank JSONAttribute fields and creates a JSON blob for each of those records. However, when the INSUFFICIENT_ACCESS_ON_CROSS_REFERENCE_ENTITY error occurs, you must replace all JSONAttribute fields. You must define the ProductAttribJSONBatchJob Update All custom setting to change its behavior.

To define the ProductAttribJSONBatchJob Update All Custom Setting:

1. From Setup, in the **Quick Find** box, enter **`Custom Settings`**.
2. Click **Custom Settings**.
3. Next to CPQ Configuration Setup, click **Manage**.
4. Click **New**.
5. Enter the following:
   - The **Name** is **`ProductAttribJSONBatchJob Update All`**.
   - The **Value** is **`True`**.

   > **IMPORTANT**
   >
   > The name is case-sensitive. Check your typing.

6. Click **Save**.
7. Click **Back to List**.
8. Go to Running the FixProductAttribJSONBatchJob.

Running the EPCProductAttribJSONBatchJob

The `EPCProductAttribJSONBatchJob` regenerates the Product object JSONAttribute fields for products that were created and are managed in Product Console. Run the `EPCProductAttribJSONBatchJob` to sync the Product object JSONAttribute fields with the attributes that are assigned to the products.

Run this job when you add an attribute to an existing object type that is associated with products.

When running the `EPCProductAttribJSONBatchJob`, you enter a script to get a list of Product2 IDs, or `prod.Ids`.

To run the `EPCProductAttribJSONBatchJob`:

1. From the **User** menu, choose **Developer Console**.
2. From the **Debug** menu, choose **Open Execute Anonymous Window**.
3. Delete any code that appears in the **Enter Apex Code** dialog box.
4. Paste the following statement into the **Enter Apex Code** dialog box, replacing *`vlocity_cmt`* with your namespace:

```
List<Id> productIds = new List<Id>();
for (Product2 prod : [ Select Id from Product2 where
vlocity_cmt__ObjectTypeId__c != null ])
{
```

```
productIds.add(prod.Id);
}
Database.executeBatch(new
vlocity_cmt.EPCProductAttribJSONBatchJob(productIds), 1);
```

5. Click **Execute**.
6. Close Developer Console.

When you import data that contains product attributes, such as when creating a sandbox environment, you must correct the product attribute record IDs. If your org contains migrated product attribute overrides from promotions, you must also run EPCFixCompiledAttributeOverrideBatchJob to correct the attribute record IDs.

Running the EPCFixCompiledAttributeOverrideBatchJob

When products have attribute overrides, run the EPCFixCompiledAttributeOverrideBatchJob to regenerate the JSONAttribute fields for those overrides. The EPCFixCompiledAttributeOverrideBatchJob regenerates the JSONAttribute field on CompiledAttributeOverride records for attribute overrides.

Run this job when you add an attribute to an existing object type that is associated with products and an associated product has at least one attribute with an override.

To run the EPCFixCompiledAttributeOverrideBatchJob:

1. From the **User** menu, choose **Developer Console**.
2. From the **Debug** menu, choose **Open Execute Anonymous Window**.
3. Delete any code that appears in the **Enter Apex Code** dialog box.
4. Paste the following statement into the **Enter Apex Code** dialog box, replacing `vlocity_cmt` with your org namespace:

```
Database.executeBatch(new
vlocity_cmt.EPCFixCompiledAttributeOverrideBatchJob (), 1);
```

5. Click **Execute**.
6. Close Developer Console.

Guided Selling

Vlocity Digital Interaction Platform includes guided selling. Another name for guided selling is "sales configuration." Guided selling involves a dynamic questionnaire that recommends opportunities for upselling and cross-selling. Using guided selling prevents selling invalid product and service combinations.

Vlocity Actions launch OmniScript guided processes. Any information stored on the record from which the process is launched can be passed to the OmniScript. OmniScript automates processes, such as guided selling, troubleshooting scripts, and structured data collection. Guided industry interactions are prebuilt, omnichannel process flows that help customer service to efficiently address common customer inquiries. They enable customer and partner self-service, reduce average call handling time (AHT), and deliver better customer experience.

Guided sales and ordering are prebuilt process flows that guide customers, partners, and customer service representatives through the recommendation, selection, and ordering process. The guided billing service includes prebuilt process flows guide customers, partners, and customer service representatives through timely resolution of billing service inquiries, including pay bill, dispute bill, request adjustment, resume suspended service, and more.

Guided service troubleshooting includes prebuilt process flows to guide customers, partners, and customer service representatives through timely resolution of service issues, including troubleshooting service, resetting equipment, requesting service appointments, and more.

Creating Guided Selling OmniScript Using CPQ APIs

Guided selling can be used to lead the customer to the most appropriate products. These APIs enable you to create an order, query and display products, add and update products in a cart, and submit an order, all within an OmniScript.

After products have been configured for use within the OmniScript, you can create a customer-facing or internal guided selling script. This OmniScript must be launched from an order or account record (the Context ID must be from an account or order). If the OmniScript is launched from an account, the script will create an order. If it is launched from a draft order, it will add items to the order.

NOTE

When upgrading from releases prior to CME Spring '17, see CPQ APIs To Replace OmniCPQServiceWrapper.

Refreshing the Price Book Prior to Winter '18

NOTE

In Vlocity Communications, Media, and Energy Winter '19 and later, use the Refresh Platform Cache job on the Vlocity CMT Administration page to refresh price books. In Winter '18 and Fall '18, use the Refresh Pricebook job on the Vlocity CMT Administration page. For more information, see Administration Jobs Reference for Vlocity Communications, Media, and Energy.

You must refresh all price books that opportunities or orders use. While refreshing, Vlocity Communications, Media, and Energy processes the complete hierarchy, including the details of all product child items, and caches that information in the org level cache. Refreshing a price book also updates stored filterable attributes. If you do not refresh the price books after an upgrade, Vlocity Communications, Media, and Energy caches the hierarchy at run time when a product is accessed for the first time.

NOTE
You must refresh the price book in Salesforce Classic, not Lightning Experience.

Refreshing the price book stores filterable attributes and caches product bundles when the cache is enabled. You must refresh the price book in the following situations:

- When you add or modify an attribute or attribute value for any product in the price book
- When you add any new product to the price book, whether the product includes attributes or not
- When you add, modify, or delete the product child items for any bundle in a price book

  NOTE
  This is applicable only if you are using the platform cache. For more information, see Configure CPQ Platform Cache.

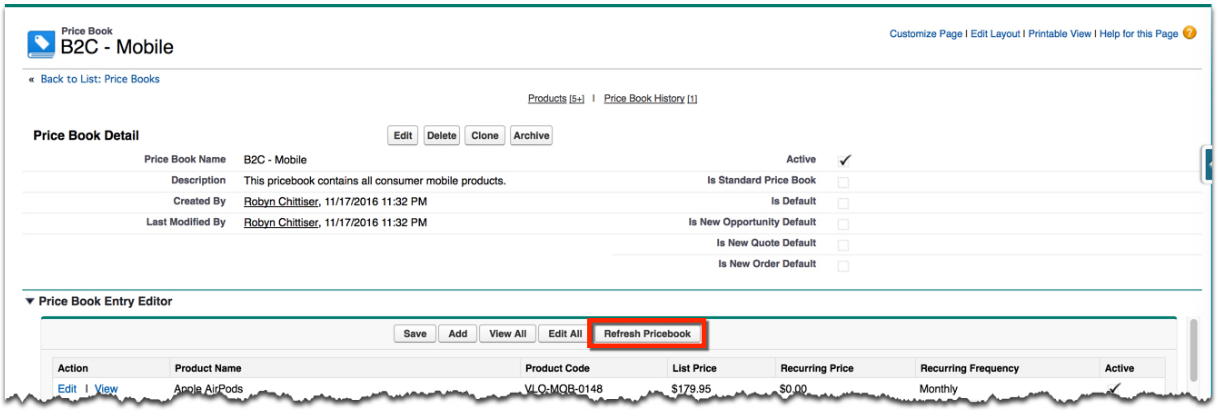
Refreshing the price book runs the following batch jobs:

- The **ProductAttributesBatchProcessor** batch job copies the product attributes from the JSONAttribute field to the *sObject* Cached Filterable Attribute, enabling queries such as the filter in the Product List in Vlocity Cart.
- The **ProductHierarchyBatchProcessor** batch job copies the product hierarchies created in the **ResolveProductHierarchyBatchJob** to the CPQ partition platform org cache, leveraging the Salesforce platform cache for optimized product hierarchy operations and enables support for product hierarchies.

Do not refresh the price book while orders are being created. Invalid orders and runtime errors may result.

To refresh the price book:

1. Go to the Price Book tab.
2. Click the price book to refresh.
3. Click **Refresh Pricebook**.


Price Book
B2C - Mobile
Customize Page | Edit Layout | Printable View | Help for this Page
« Back to List: Price Books
Products [5+] | Price Book History [1]
Price Book Detail
Edit
Delete
Clone
Archive
Price Book Entry Editor
Save
Add
View All
Edit All
Refresh Pricebook

Price Book Name	B2C - Mobile	Active	✓
Description	This pricebook contains all consumer mobile products.	Is Standard Price Book	□
Created By	Robyn Chittiser, 11/17/2016 11:32 PM	Is Default	□
Last Modified By	Robyn Chittiser, 11/17/2016 11:32 PM	Is New Opportunity Default	□
		Is New Quote Default	□
		Is New Order Default	□

Action	Product Name	Product Code	List Price	Recurring Price	Recurring Frequency	Active
Edit | View	Apple AirPods	VLO-MOB-0148	$179.95	$0.00	Monthly	✓

NOTE

If the **Refresh Pricebook** button is not visible, you must expose it. For more information, see Expose the Refresh Pricebook Button

4. Repeat for each price book in your org.

Configuring OmniScript for Guided Selling

To create the OmniScript you plan to use with guided selling:

1. **+Add New Key/Value Pair** under the Element Type To HTMLTemplate Mapping section in the OmniScript. The Horizontal Mode Persistent Component's HTML template is vlcHCartHyb.html. See Overwrite OmniScript Element Template for more information.
2. Add an OmniScript Remote Action to call the createCart API of CpqAppHandler class from OmniScript. The createCart method takes the following arguments:
   - objectType (String)
   - inputFields (JSON Array)
     - AccountId (String)
     - effectivedate (String)
     - status (String)
     - vlocity_cmt_PriceListId__c (String)

   To create an order:

   a. Add a remote action component, either in-between steps or in a step (in which case it will display as a button).

   b. In the **Remote Class** field enter `vlocity_cmt.CpqAppHandler`.

   c. In the **Remote Method** field enter `createCart`.

   d. In the Extra Payload section click on the **+ Add New Key/Value Pair** button and enter objectType in the left hand field and **Order** in the right hand column.

NOTE

The Extra Payload section holds the arguments, modeled in JSON, that will be sent to the CpqAppHandler method being called.

e. In order to add a JSON array we need to edit the component's JSON directly, to do this click the **Edit as JSON** link in the top right hand corner of the screen.

f. In the **extraPayload** node, copy and paste the following JSON just under the objectType key/value pair:

```
"inputFields": [

    {
        "effectivedate": "2018-09-20"
    },
    {
        "status": "draft"
    },
    {
        "AccountId": "0010O00001qJU3a"
    },
    {
        "vlocity_cmt__PriceListId__c": "a3O0O000000bOXu"
    }
]
```

NOTE

The AccountId and vlocity_cmt_PriceListId__c IDs must be copied from Salesforce records in your org. These can be extracted with DataRaptor or Apex before the step containing the cart runs.

g. (Optional) Test the remote action by previewing the OmniScript and checking the Data JSON for additional data.

For more information on the createCart method, see Create Cart.

3. To get the ID from the JSON array we can use a Set Values action to turn the Id into a String.
   a. Drag a Set Values action after the step containing the createCart method call.
   b. In the **Element Name** column add **`OrderId`**.
   c. Under the **Value** column **add `=STRING(%records:Id%)`**
4. Use OmniScript Remote Action to call getCarts API of the CpqAppHandler class from OmniScript. This action populates the top of the cart. For details, seeGet Cart Summary. Required inputs to this API are *`cartId`*. This parameter should be input in the **Extra Payload** section.
   - Under the **Send /Response Transformation** section, add vlcCart_Top as the Response JSON Node. Leave the Response JSONPath blank. This will set Response JSON Node as parent node.

5. Add an OmniScript Remote Action to call getCartsItems API of CpqAppHandler class from OmniScript. For details, see Get Cart Items. This action populates the bottom of the cart. The *`cartId`* parameter must be input in the **Extra Payload** section. *`Pagesize`* is also supported. If you want a different pagesize, use an OmniScript Set Values element to set separate values for cart pagesize and selectable items pagesize. For more information on Set Values, see Setting Values in OmniScript.
   a. Change `HTML Template Id` to **`vlcSelectableItemHyb.html`** in Selectable Items.
   b. vlcCart is the persistent component ID mentioned in script configuration. It should match. Under the **Send /Response Transformation** section, leave the Response JSONPath blank. This will set Response JSONNode as parent node.
   c. Under the **Extra Payload** section, input the OrderId in the value of the cartId parameter (the record ID that was returned in step 3).
6. Use OmniScript Remote Action to call the Get List of Products API of CpqAppHandler class from OmniScript. This action populates selectable items for product selection. Required inputs to this API are `cartId` and `pagesize`. These parameters must be input in the **Extra Payload** section.
   - Enter `vlcSelectableItemHyb.html` for the HTML Template ID.
   - Under the `Send/Response Transformation` section, leave the Response JSONPath blank for your product. This will set Response JSONNode as parent node.
   - Under the **Extra Payload** section, input the following parameters:
     - **cartId**: Enter OrderId in the value. This is the record Id that was returned in step 3.
     - **pagesize**: determines the number of records retrieved in getCartsProducts API call. For example, `pagesize = 10`. A **Load More** button displays if there are more products than the page size.
7. Set Values in an OmniScript by adding a query parameter for querying products on name and description.
8. In order to query by attribute, use an OmniScript Remote Action. For each query before the getCartsProducts call, add the parameters in the **Extra Payload** section. (For details, see Get List of Products.) For example, an entry to the Extra Payload section looks like `attributes` and `CPQProductAttributeRed:Yes`.
   - To filter by attribute, use the following format:

     ```
     $attributeCode:$value1_$value2_$value3,$attributeCode:$value1
     ```
   - For example, CPQProductAttributeRed is attribute code and iOS, Samsung, Microsoft, Motorola, and LG are the attribute values.
9. To query by Name and Description, use an OmniScript Remote Action. For each query, before the getCartsProducts call, add the parameters in the Extra Payload section. For example, an entry to the Extra Payload section looks like `query` and `iPhone`. This entry searches all products with "iPhone" in Name and Description.
10. To submit the order, use an OmniScript Remote Action to invoke CpqAppHandler.checkout. (For details, see Checkout Items in Cart.) Add the following parameters in the Extra Payload section:
    a. *`cartId`*: Same value as Order ID
    b. *`ContextId`*: Same value as Order ID

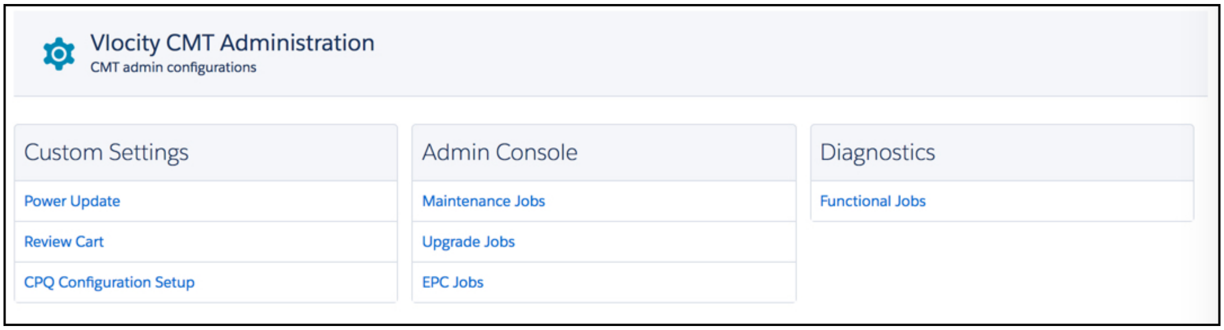
Running the Product Hierarchy Maintenance Job

The Product Hierarchy Maintenance job populates the product child item IDs related to each product in the data store object. The CPQ process uses this information to represent bundle hierarchies.

This job needs to be run after creating or modifying products. For more information about the Product Hierarchy Maintenance job and other Vlocity CMT maintenance jobs, see Administration Jobs Reference for Vlocity Communications, Media, and Energy.

To run the Product Hierarchy Maintenance job:

1. Go to the Vlocity CMT Administration tab.



2. Click **Maintenance Jobs**.
3. Next to the Product Hierarchy Maintenance job, click **Start**.

Refreshing the Price Book Prior to Winter '18

> **NOTE**
> In Vlocity Communications, Media, and Energy Winter '19 and later, use the Refresh Platform Cache job on the Vlocity CMT Administration page to refresh price books. In Winter '18 and Fall '18, use the Refresh Pricebook job on the Vlocity CMT Administration page. For more information, see Administration Jobs Reference for Vlocity Communications, Media, and Energy.

You must refresh all price books that opportunities or orders use. While refreshing, Vlocity Communications, Media, and Energy processes the complete hierarchy, including the details of all product child items, and caches that information in the org level cache. Refreshing a price book also updates stored filterable attributes. If you do not refresh the price books after an upgrade, Vlocity Communications, Media, and Energy caches the hierarchy at run time when a product is accessed for the first time.

> **NOTE**
> You must refresh the price book in Salesforce Classic, not Lightning Experience.

Refreshing the price book stores filterable attributes and caches product bundles when the cache is enabled. You must refresh the price book in the following situations:

- When you add or modify an attribute or attribute value for any product in the price book
- When you add any new product to the price book, whether the product includes attributes or not
- When you add, modify, or delete the product child items for any bundle in a price book

> **NOTE**
> This is applicable only if you are using the platform cache. For more information, see Configure CPQ Platform Cache.

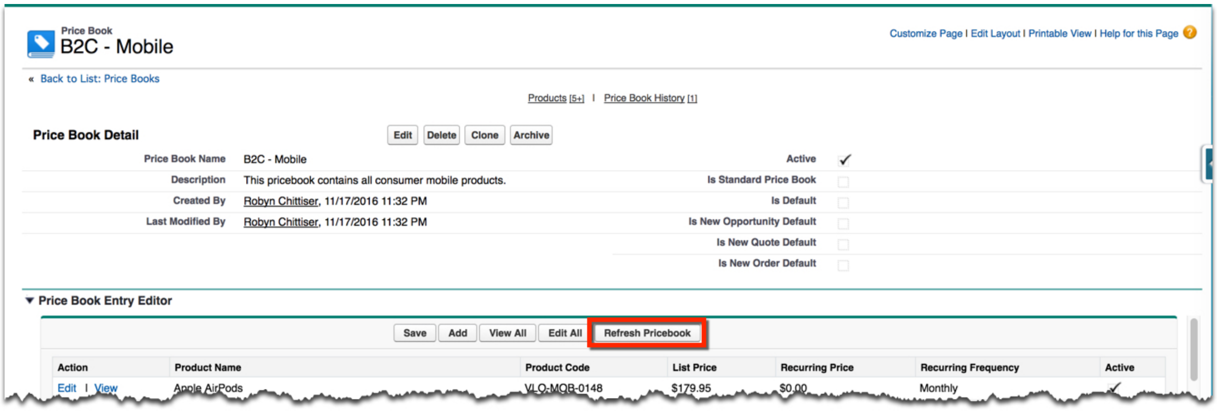
Refreshing the price book runs the following batch jobs:

- The **ProductAttributesBatchProcessor** batch job copies the product attributes from the JSONAttribute field to the *sObject* Cached Filterable Attribute, enabling queries such as the filter in the Product List in Vlocity Cart.
- The **ProductHierarchyBatchProcessor** batch job copies the product hierarchies created in the **ResolveProductHierarchyBatchJob** to the CPQ partition platform org cache, leveraging the Salesforce platform cache for optimized product hierarchy operations and enables support for product hierarchies.

Do not refresh the price book while orders are being created. Invalid orders and runtime errors may result.

To refresh the price book:

1. Go to the Price Book tab.
2. Click the price book to refresh.
3. Click **Refresh Pricebook**.


Price Book
B2C - Mobile
Customize Page | Edit Layout | Printable View | Help for this Page
« Back to List: Price Books
Products [5+] | Price Book History [1]
Edit
Delete
Clone
Archive
Save
Add
View All
Edit All
Refresh Pricebook

Price Book Detail

Price Book Name	B2C - Mobile	Active	✓
Description	This pricebook contains all consumer mobile products.	Is Standard Price Book	□
Created By	Robyn Chittiser, 11/17/2016 11:32 PM	Is Default	□
Last Modified By	Robyn Chittiser, 11/17/2016 11:32 PM	Is New Opportunity Default	□
		Is New Quote Default	□
		Is New Order Default	□

Price Book Entry Editor

Action	Product Name	Product Code	List Price	Recurring Price	Recurring Frequency	Active
Edit | View	Apple AirPods	VLO-MOB-0148	$179.95	$0.00	Monthly	✓

> **NOTE**
> If the **Refresh Pricebook** button is not visible, you must expose it. For more information, see Expose the Refresh Pricebook Button

4. Repeat for each price book in your org.

Exposing the Refresh Pricebook Button

When you assign attributes to a product, you have to refresh the price book. PricebookEntryEditor is a Visualforce page that includes the **Refresh Pricebook** button. It is not enabled in the release package in Lightning Experience. You must use Salesforce Classic.

To expose the **Refresh Pricebook** button:

1. Go to a Price Book record detail page.
2. In the upper right corner of the page, click **Edit Layout**.
3. In the object palette, select **Visualforce Pages**.
4. Drag **PricebookEntryEditor** to the layout.
5. Click the **Properties** icon.
   The **Visualforce Page Properties** dialog box opens.
6. Enter the following information:
   - **Width (in pixels or %)** is `100%`.
   - **Height (in pixels)** is at least `500`.
   - Select **Show scrollbars**.
7. Click **OK**.
8. In the object palette, click **Save**.

Creating Catalogs and Catalog Relationships

In Vlocity, you can create a hierarchy or catalogs to configure complex relationships and dependencies between products and promotions. The Root catalog (created in Catalog__c) is the top most node in the catalog hierarchy. Catalogs added as child nodes (created in CatalogRelationship__c) of the Root catalog are the catalogs that appear as the first level of the catalog category panel.

Once you have created catalogs, you must add all sellable products and promotions to a catalog.

To create catalogs, perform the following steps in the specified order:

1. Create the Root Catalog
2. Create Product Catalogs
3. Create Catalog Relationships
4. Add Products to a Catalog

Creating the Root Catalog

This is the first step in the Creating Catalogs and Catalog Relationships process. During the initial setup, all steps must be performed in the specified order.

To create the Root catalog:

1. Click the **Catalogs** tab.

2. Click **New**.
3. Complete the following fields:

Field Name	Description/Value
Catalog Name	Root
Is Catalog Root	Check the checkbox to mark the catalog as root.
Is Active	Select the checkbox to activate the catalog.
Start Date Time	Select the start date when the catalog becomes effective.

4. Click **Save** or **Save and New**, if creating other product catalogs.

Creating Product Catalogs

This is the second step in the Creating Catalogs and Catalog Relationships process. During the initial setup, all steps must be performed in the specified order.

To create a product catalog:

1. Click the **Catalogs** tab.
2. Click **New**.
3. Complete the following fields:

Field Name	Description/Value
Catalog Name	Enter a descriptive name, for example, Voice, Data or Service.
Is Catalog Root	Do *not* select the checkbox.
Is Active	Select the checkbox to activate the catalog.
Start Date Time	Select the start date when the catalog becomes effective.

4. Add a unique catalog code with no spaces. This is required to use a catalog for digital commerce.
5. Specify a price list. This is required to use a catalog for digital commerce.
6. Click **Save** or **Save and New**, if creating other product catalogs.

Creating Catalog Relationships

This is the third step in the Creating Catalogs and Catalog Relationships process. During the initial setup, all steps must be performed in the specified order.

To create a catalog three hierarchy:

1. Click the **Catalogs** tab.
2. Open the Root catalog that you have previously created.
3. In the **Catalog Relationships** section, click the **New Catalog Relationship** button.
4. Complete the following fields:

Field Name	Description or Value
Catalog Relationship Name	Use the name of the product catalog you plan to add as a child.
Product Catalog	Root
Child Catalog	Select the desired product catalog that you have previously created.

5. Click **Save**.

Adding Products to a Catalog

This is the fourth step in the Creating Catalogs and Catalog Relationships process. During the initial setup, all steps must be performed in the specified order.

To add a product to a catalog:

1. Click the **Catalogs** tab.
2. Click the name of the desired product catalog.
3. In the **Catalog Product Relationships** section, click the **New Catalog Product Relationship** button.
4. Complete the following fields:

Field Name	Description or Value
Catalog Product Relationship Name	Use the name of the product you plan to add to the catalog.
Product	Select the product you wish to add to the catalog.
Effective Date	Select start date of the time period when the product is available.
End Date	Select the end date of the time period when the product is available.
Is Active	Select the checkbox.

5. Click **Save**.