

Exercise 4: Create and Amend Object Types

A product object type is a reusable entity that defines the schema and architecture of a particular category of products. They're used to apply layouts and properties, including fields and attributes, to multiple product objects—the instances of sellable products in the catalog. Object types are a great way to organize the configurations of related products and to ensure consistent application of data, behavior, and rules across a product line. You can even create subtypes of object types so that they automatically inherit the components configured in its parent object type.

Object types are designed with an IS-A inheritance type architecture. For those familiar with object-oriented programming and design, you'll 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 simply an 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 of its object type hierarchy.



BEST PRACTICE:

It's not recommended to move object types within the hierarchy. Therefore, careful consideration is required when planning your object type hierarchy and your product catalog design. Salesforce Industries Services can assist with product catalog design and modeling best practices.



NOTE:

In your training playground, an initial product object type hierarchy has already been created for you, including all of the base product object types.

Scenario

Now that Devi has the “foundation” of his product model built, he’s ready to develop the “framing”. Using the power of product object types, he can streamline the creation of his product catalog by creating classes of products that have similar characteristics.



While he’s checking out what’s already been set up, he discovers that the SmartPhone product spec is missing some important attributes, so he sorts that out.

Later, following his product model, he decides to add a new SmartWatch Product Spec as an object subtype of the BASE Product Spec object type, which will allow his new object type to inherit all of the relevant fields and attributes defined in the object hierarchy.

Goal



- Create product object types
- Review the inheritance architecture of product object types
- Assign product attributes and attribute metadata to an object type
- Modify object type layouts

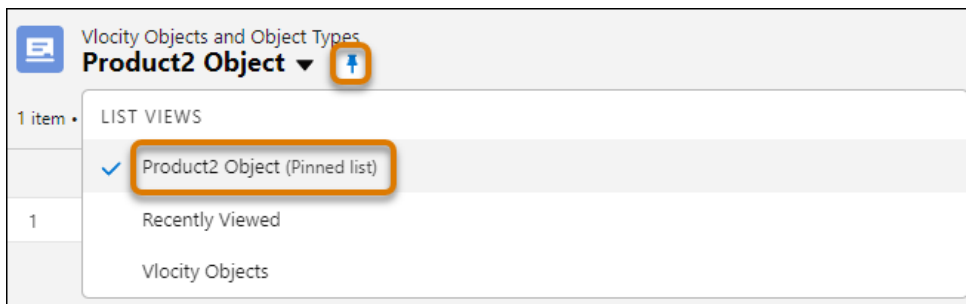
Tasks

1. Review the existing product object type hierarchy
2. Review the attribute and field assignments
3. Review an Object Type Layout
4. Create a new product object type
5. Add attributes for the product object type
6. Set attribute metadata for the product object type
7. Modify the product object type’s layout

Time: 30 mins

Task 1: Review the existing product object type hierarchy

1. If you're not already there, open **Vlocity Product Designer** from the App Launcher .
2. Select the arrow icon  from the tab at the top of the interface. In the dropdown menu, select **Vlocity Objects and Object Types**.
3. Click the arrow next to the currently selected list view (e.g. Recently Viewed) and change the view to Product2 Object. We recommend pinning this view so it shows by default for future visits.



4. Click **Product2 Object**.
5. On the Product2 Object configuration screen, notice the Object Types list on the left. This shows the hierarchy of the object types and subtypes created for you as part of the managed package.
 - a. Notice the base object types for the four SID layers: Offer, Product, Resource, and Service.
 - b. Notice that the Base Bundle Offer has no object subtypes.

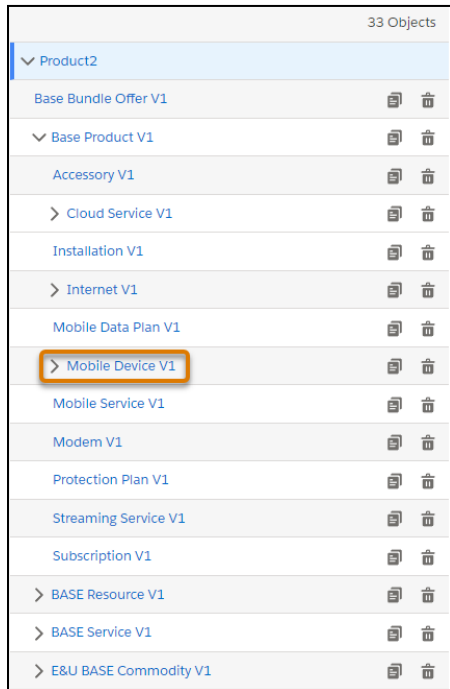


BEST PRACTICE:

It's recommended that object types in each of the SID layers are "tuned" to the product entity as it is manifested within the current layer only. Product entities can then be transformed from one layer to the next using Order Management's decomposition mappings. In keeping with this best practice, you will notice that the Base Bundle Offer object type has no object subtypes because "an offer is an offer" and should have homogenous characteristics. However, the other Base object types are not flat hierarchies because the

product entities they describe are fundamentally distinct from one another.

- Expand the **Base Product V1** row by clicking the arrow icon > next to it. Notice the product spec subtypes nested below this object type.
- Click **Mobile Device V1**.



- In the configuration pane, notice that the Mobile Device's parent object type is set to the **Base Product**. This means that the Mobile Device object type has inherited its layout and fields from Base Product.

GENERAL PROPERTIES

* Name

Parent Object Type
 ✕ 🔍

Version Label

Lifecycle Status
 ▼

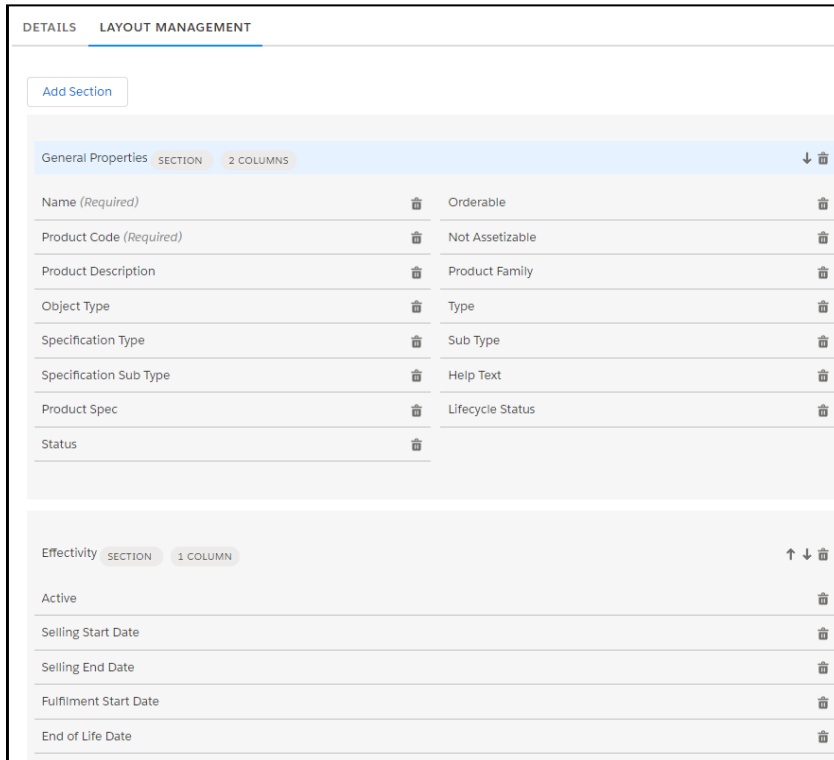


BEST PRACTICE:

As mentioned earlier, it's recommended that any data element that is common across the catalog be described using a field, and data elements that are unique to a subset of products in the catalog be described using an attribute. Following this best practice, all of the Base object types in your training org, such as Base Product and Base Service, have no attributes assigned to them, only fields. Their subtypes, such as Mobile Device, contain attributes specific to a line of products.

Task 2: Review an object type's layout, attributes, and fields

1. In the Mobile Device object type, click the **LAYOUT MANAGEMENT** tab at the top of the pane.
2. Under the **General Properties** and **Effectivity** sections, notice the fields that are used when configuring a new product when this object type is applied.



DETAILS LAYOUT MANAGEMENT

Add Section

General Properties SECTION 2 COLUMNS ↓

Name (Required)	🗑️	Orderable	🗑️
Product Code (Required)	🗑️	Not Assetizable	🗑️
Product Description	🗑️	Product Family	🗑️
Object Type	🗑️	Type	🗑️
Specification Type	🗑️	Sub Type	🗑️
Specification Sub Type	🗑️	Help Text	🗑️
Product Spec	🗑️	Lifecycle Status	🗑️
Status	🗑️		

Effectivity SECTION 1 COLUMN ↑ ↓

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

3. Now, notice the **Design Time Attributes** and **Run Time Attributes** sections. These sections hold the unique attributes for products using this object type, including Brand, Size, and Color.

Design Time Attributes	SECTION	1 COLUMN	↑ ↓ 🗑️
Brand			🗑️
Size			🗑️
Capacity			🗑️
Mobile OS			🗑️
Run Time Attributes	SECTION	1 COLUMN	↑ 🗑️
Color			🗑️
Payment Type			🗑️



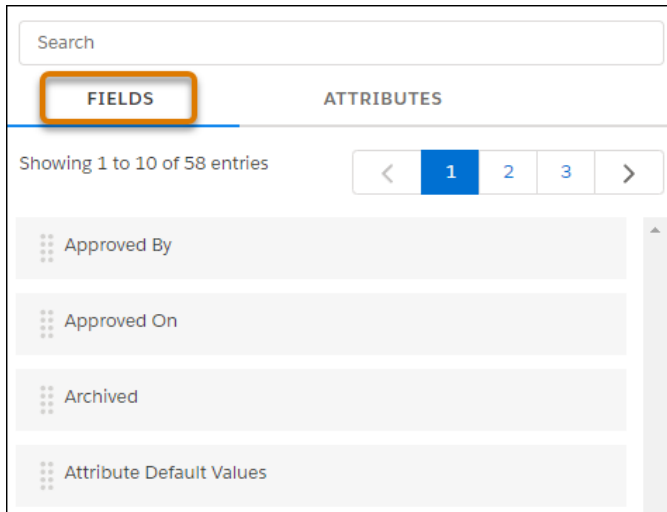
NOTE:

Design Time Attributes are those that you define behind the scenes when configuring products. Run Time attributes are those that are defined at run time in the CPQ cart during order capture. These sections don't have any unique functionality. As you'll see in a moment, you must define each attribute as Run-time Configurable in order to make them definable at run time.

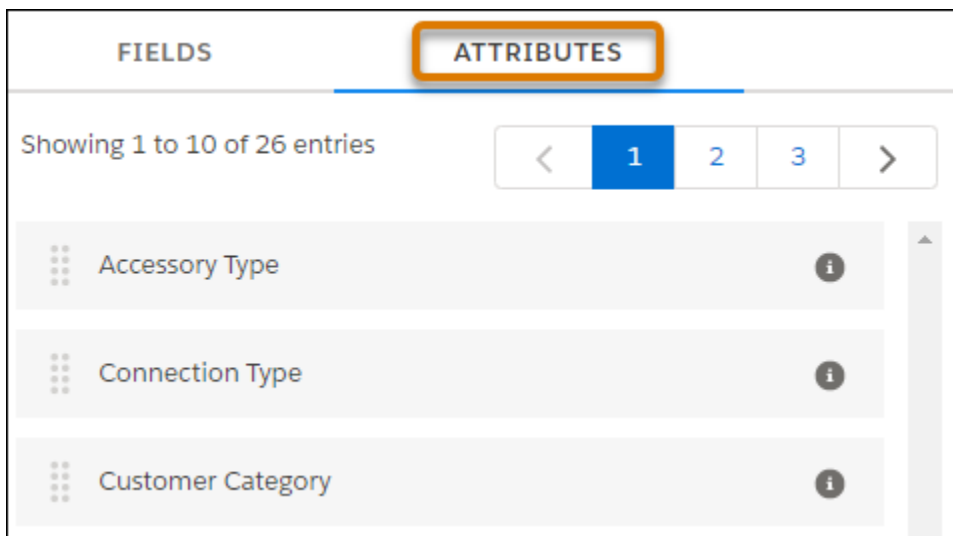
4. Scroll down and notice the Debugging Properties section which includes an attribute that holds the JSON data of the product to which the object type is applied.

Debugging Properties	SECTION	1 COLUMN	↑ 🗑️
JSONAttribute			🗑️

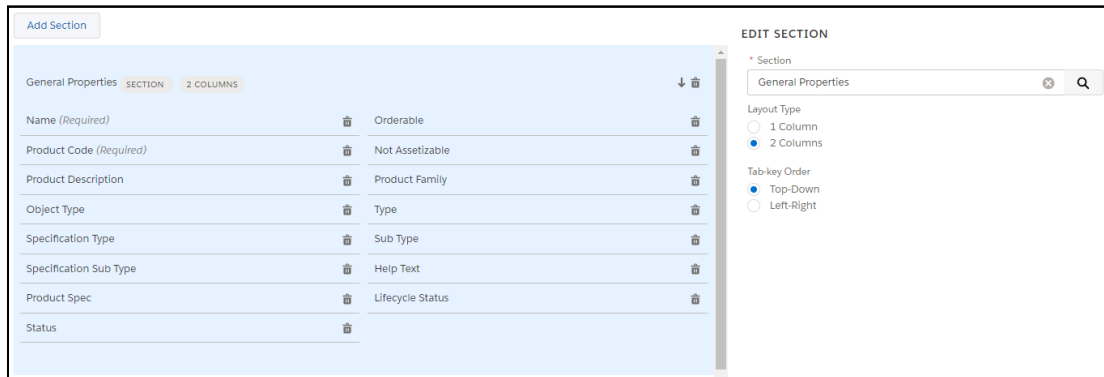
5. Under the **FIELDS** tab in the pane on the right, notice the available fields that can be clicked and dragged to the layout.



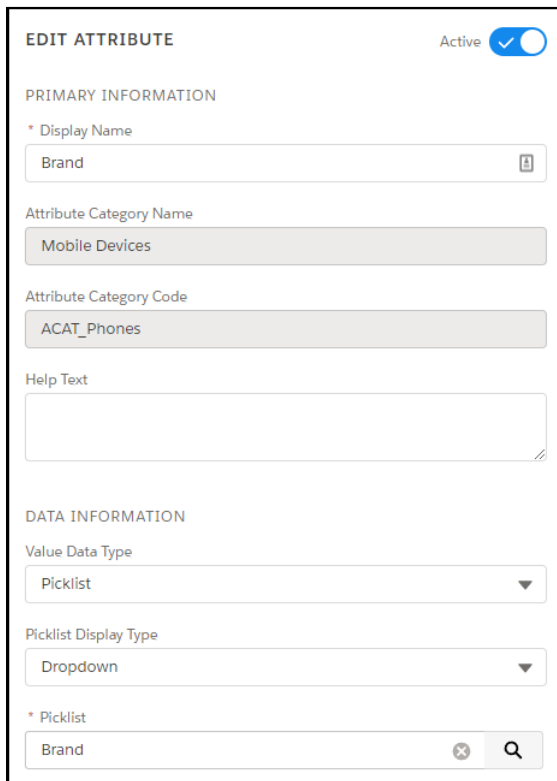
6. Click the **ATTRIBUTES** tab and scroll through the list to view the available attributes that can be added to the layout.



7. Click the **General Properties** heading at the top of the section (not any of the fields). In the pane on the right, notice the section's name, **Layout Type** and **Tab-key Order**.



8. Scroll down and click the **Brand** attribute in the Design Time Attributes section of the Mobile Device layout. In the **EDIT ATTRIBUTE** pane, notice that you can configure certain information.



You can also control the behavior of the attribute at the object type level, such as make it **Required**, **Hidden**, or **Read Only**.



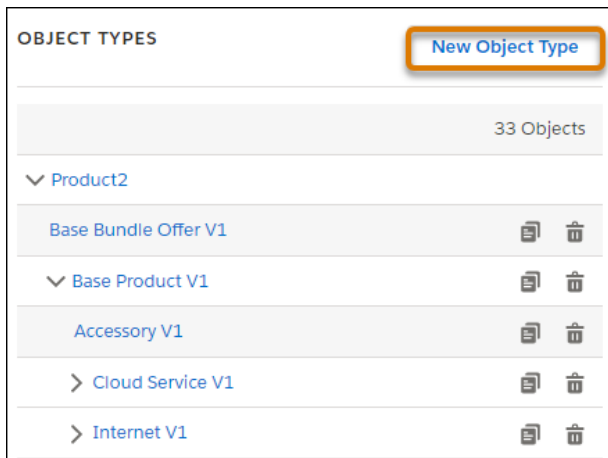
BEHAVIORS

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

Task 3: Create the SmartWatch object type

Devi's checked and there is no existing SmartWatch object type, so he decides to create a new object type which inherits from the Mobile Device object type.

1. In **OBJECT TYPES**, click **New Object Type**.



2. Enter the following information in **GENERAL PROPERTIES** and **EFFECTIVITY**.

Property	Value
Name	Smartwatch
Parent Object Type	Mobile Device
Version Label	V1
Active	<input checked="" type="checkbox"/>
Effective From	[today's date]

GENERAL PROPERTIES

* Name
Smartwatch

Parent Object Type
Mobile Device

Version Label
V1

Lifecycle Status
Draft


EFFECTIVITY

Active

Effective From
Jan 05, 2023





















Effective Until

Done Cancel

3. Click **Done**.
4. In the object types list, expand the Mobile Device V1 object type using the arrow icon  , and notice that the **Smartwatch V1** sub-type you just created is now nested under its parent object type.

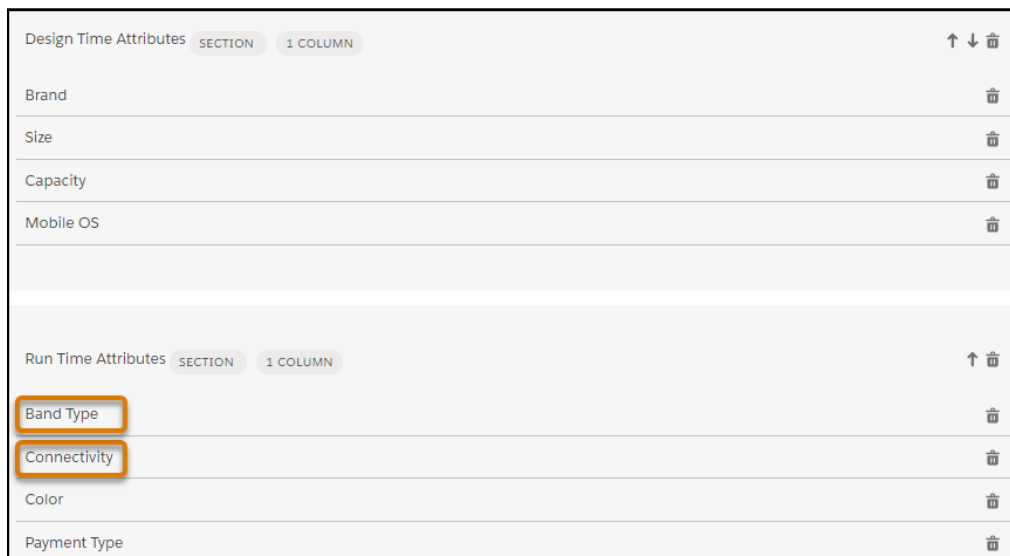
OBJECT TYPES New Object Type

34 Objects

Product2		
Base Bundle Offer V1		
Base Product V1		
Accessory V1		
Cloud Service V1		
Internet V1		
Mobile Data Plan V1		
Mobile Device V1		
Handset V1		
Smartwatch V1		
Tablet V1		

Task 4: Add attributes to the product object type

1. If you don't already have it open, click **Smartwatch V1** in the **Object Types** list to open the object type for editing.
2. Click the **LAYOUT MANAGEMENT** tab and notice that all of the fields and attributes have been inherited from the **Mobile Device** parent object type.
3. Select the **ATTRIBUTES** tab in the right pane and click and drag the **Band Type** attribute into the **Run Time Attributes**.
4. Click and drag the **Connectivity** attribute into the **Run Time Attributes** section.
5. Click and drag to rearrange the attributes in the section to match the order here:
 - Band Type
 - Connectivity
 - Color
 - Payment Type



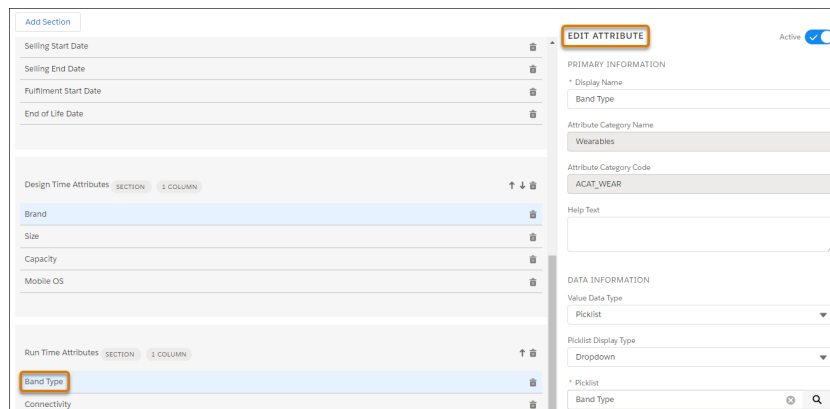
Task 5: Set attribute metadata for the product object type

Now you've added the attributes, you need to decide how you want them to behave by setting the attribute metadata. Attribute metadata assigned at the object type level is inherited by products created from the object type. Some attribute metadata, such as Has Rule, Hidden, Read Only and Required are overridable and can be changed at the product level. Non-overridable metadata includes Run-Time Configurable, Encrypted, Not Assetizable, and Not Translatable. These metadata flags must be set at the object type level and cannot be changed at the product level.

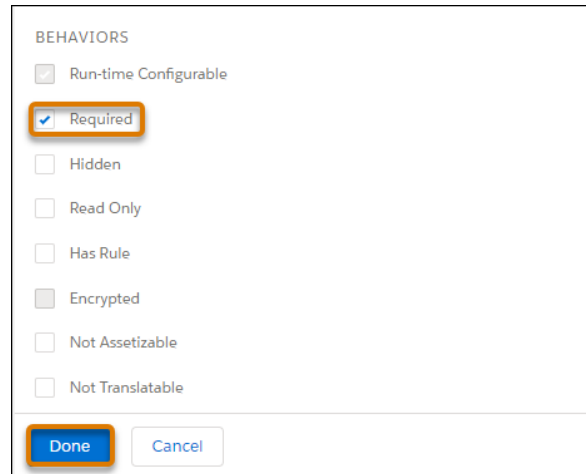
Infiwave wants the Band Type, Connectivity, and Color attributes of the SmartWatch to be required, and selectable by the customer as part of the order process.

The brand, size, and capacity of the SmartWatch should only be editable by the product designers but should be visible to the customer when they're configuring their order.

1. Set the Band Type, Connectivity, and Color attributes of the SmartWatch to be required by the customer as part of the order process.
 - a. In the Smartwatch object type Layout Management workspace, click **Band Type** to open the EDIT ATTRIBUTE pane.



- b. Type something in the **Help Text**, such as Choose the type of watch band you would like.
- c. Scroll down to the BEHAVIORS section, and select the **Required** checkbox.
- d. Click **Done** to save the changes.



- e. Next, click the **Color** attribute in the layout to open the EDIT ATTRIBUTE pane. Select the **Required** checkbox under the BEHAVIORS section. Click **Done**.
 - f. Finally, click the **Connectivity** attribute. In the EDIT ATTRIBUTE pane change the **Picklist Display Type** to **Radio button**, and select the **Required** checkbox. Click **Done**.
2. Set the value of the Brand attribute of the SmartWatch and set it to be configurable at design time only, but readable by the customer as part of the order process.
 - a. In the Layout Management workspace of Smartwatch object type, click **Brand** to open the EDIT ATTRIBUTE pane.
 - b. Infiwave sells only Infiwave smartwatches, so scroll down to the DATA INFORMATION section, and in the **Value** select **Infiwave**. Infiwave will be automatically selected as the brand for all smart watches in future.
 - c. Scroll down to the BEHAVIORS section, and select the **Read Only** checkbox.
 - d. Click **Done** to save the changes.
 3. Close the SmartWatch tab.

You've now created your first object type, modified its layout, and controlled metadata and picklist values for certain attributes. Way to go! You're now ready to create the Infiwatch product using the new Smartwatch object type.



Yay! All done!