Create property templates and classes

In this section, you learn how to create choice lists and different types of property templates. You will also create custom document and folder classes by using the property templates.

What is a Class?

A class is a model or template that is used to create objects.

The class to which an object belongs determines the object's methods, properties, and default security. The Class Description property identifies the class of an object.

Each Content Platform Engine (CPE) object is instantiated from a CPE class, which is a specification for a set of objects that have the same supported methods and properties.

Examples of CPE classes are Document, Custom Object, Folder classes.

You can create subclasses for many CPE classes and each subclass can have its own set of custom properties in addition to the properties provided by its superclass.

System default classes

When an object store is created, it is pre-populated with a set of system-created classes. They serve different functions.

- You can extend the system classes by defining subclasses.
- The most commonly extended system class is the Document class.
- You create application-specific documents by defining Document subclasses.

Examples of system default classes:

- The annotation class allows the user to link additional information to documents and other containable objects such as folders, and custom objects. Examples of annotations are text, audio, video, image, highlight, and sticky notes.
- The custom object class has no built-in behavior but has properties that pertain to a business subject. Define business object classes when you do not need to save content or use lifecycles or versions.
- The document class defines the properties of a document and is created to organize documents by type. The subclasses of the Document class is the most commonly used.
- The folder class holds a collection of objects.
- The subscription identifies an action that should occur when a specific event occurs on a specified object.

What are Document objects?

A Document object is an instance of the Document class or subclass.

- Documents can have the following elements:
  - Content elements
  - Associated annotations
  - Custom metadata or properties that are used for identification
- Document content elements can be stored locally, inside an object store or in an external repository and referenced from the object store.
- A Document object can be updated.
  - Each version of the document is assigned a version number.
- A Document object can be searched.

Folder objects

A Folder represents a container that holds containable objects such as the following ones:

- Documents
- Workflow definitions
- Custom objects

A Folder is independently securable and it can be searched.

Each object store has an automatically created root folder that represents the default root container that is associated with the object store. You cannot create or delete a root folder, but you can access it.

Folders that are directly contained under the root folder are referred to as top folders. These folders typically represent the starting points for folder navigation, because, for many applications, you might not want to display or allow users to add objects to the root folder.

A folder structure is useful to browse to documents via tools such as IBM Content Navigator. For a large volumes of content, other alternative is to store all content as unfiled and then use searches to find the documents of interest. The unfiled documents are not stored in any folders and they can be viewed in the administration console under Unfiled documents node.

Containment Concepts

Subfolders are directly contained and they exist inside a parent folder. They are deleted from the object store when they are removed from the parent folder.

Documents and other objects are referentially contained in folders.

- You can add a document to any number of folders.
- Adding a document to many folders does not duplicate the document.
- Removing a document from a folder does not delete the document from the object store.

Class Inheritance and custom classes

Class inheritance is a relationship between classes. One class inherits the structure and behavior that is defined in its superclasses.

When you create an object store, the system automatically creates class definitions for all of the system-provided classes. You can create a custom class by creating a subclass of an original class definition, and adding custom properties to the subclass that reflect your business needs. Subclasses inherit properties from their super class.

The Document class is the superclass of other document subclasses that you create. It defines the behavior of a document and contains important system properties such as Content Element, Version, and Date created. You cannot remove a property from a subclass that was inherited from its superclass.

What is a Property?

A property is a characteristic of a class. It helps identify the object that the class includes. Properties hold individual values that describe an object. Properties of a class can be of different types.

Property templates

A property template is a collection of metadata properties that globally define a property. It has no function in the object store until it is assigned to a class as a custom property. You can assign it to any number of classes in an object store. Its symbolic name must be unique within a class family.

Definition of a property includes the following metadata:

- Data type: Scalar or object-valued
- Cardinality: Single or multivalue
- Settability: Read-only, read/write

- Name: name property indicator
- Choice List assignment indicator

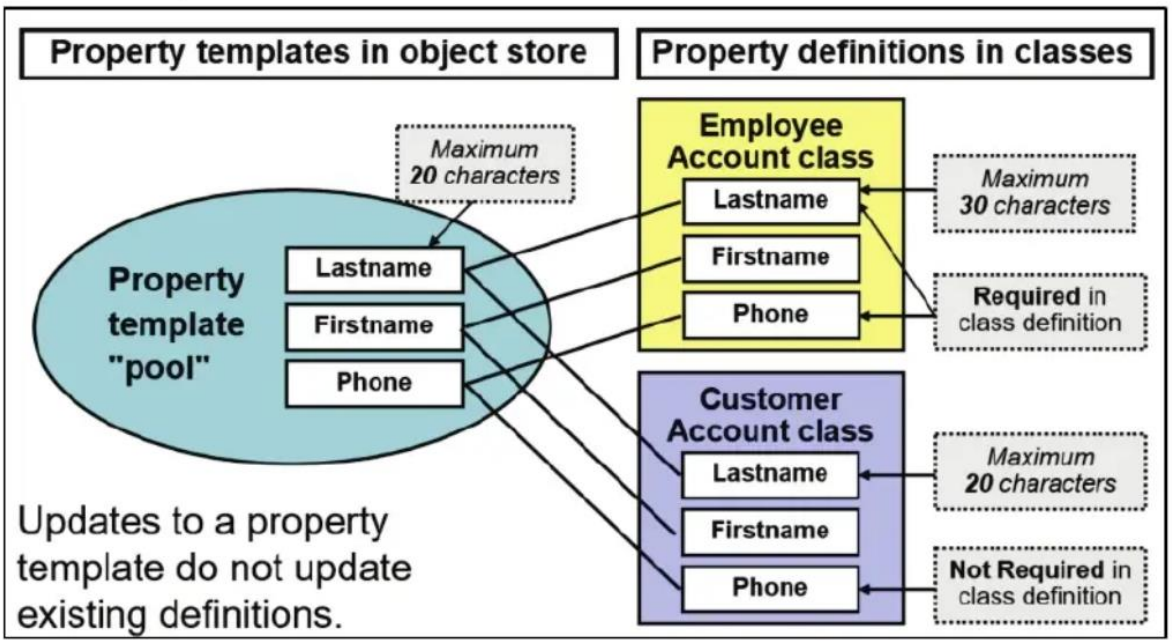
Short or long custom string properties

You can define a custom string property to be stored in either a short or long database column. Besides the maximum lengths permitted, there are differences exist between short and long strings such as how they are queried.

For more details on this topic, refer to the IBM FileNet P8 Platform V5.5.x Knowledge Center:
https://www.ibm.com/support/knowledgecenter/SSNW2F_5.5.0/com.ibm.p8.ce.admin.tasks.doc/properties/pr_select_property_descriptions.htm

What are Property Definitions?

Property definitions are based on property templates. When you are creating a class, you assign property templates to the class, which creates property definitions for that class. You can modify the attributes of the property definitions to support the class requirements. Multiple classes can use a property template and each resulting property definition is specific to the class.



The above diagram shows the available pool of property templates and their possible relationships to the property definitions in classes.

If you change a property definition of a class, then only that class is updated. When you create classes, you select the property template from the object store pool where the property template remains in its original state.

Choice lists

A choice list is a collection of predefined property values (choice items) that can be used to present values in logical groups. The users then select a value instead of typing an entry. Choice lists make data entry faster for users and also ensure that the data entered is limited to the valid choice options.

Requirements for a choice list:

- Assign a choice list to a property template to make a choice list available
- Use either a string or an integer data type for a choice list
- Ensure that each choice item within a choice list has the same data type
- Match the choice list data type to that of its associated property template

Usage options:

- You can assign one choice list to multiple property templates.
- You cannot delete the choice list object as long as it is associated with a property template.
- If the database used by CPE is case-sensitive, the display names of the choice values in the choice lists are also case-sensitive.

Groups in choice lists

When a choice list contains many values, you can group associated values (create categories) to help users find the needed value.

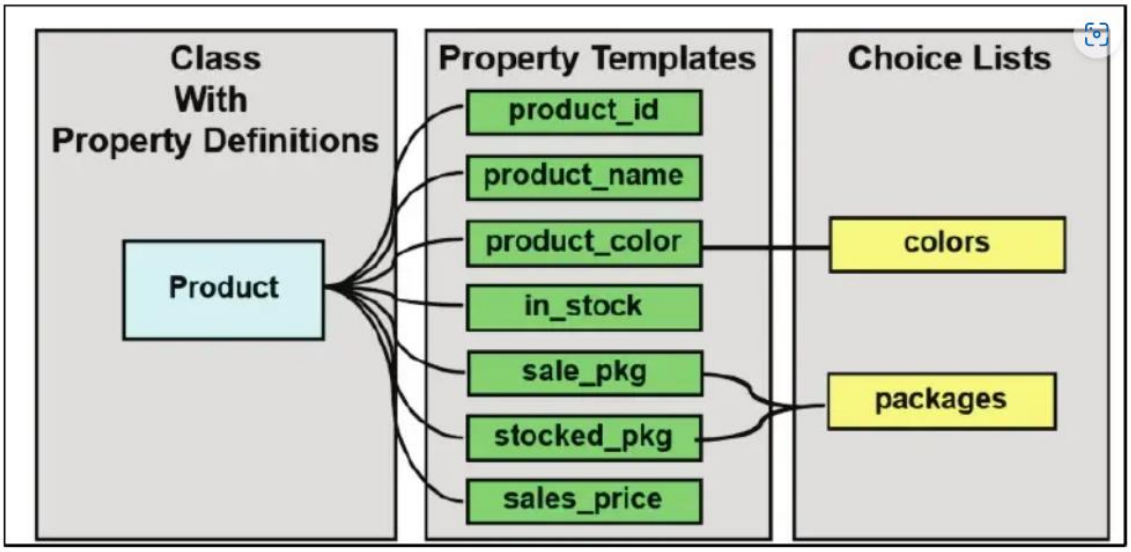
Choice lists can be hierarchical. A choice item can act as a group node and hold a choice list, allowing you to organize related choice values into nested groups. Thus a choice item can represent four types of values: a single integer value, a single string value, a group node for a list of integer-type choice items, and a group node for a list of string-type choice items.

In a hierarchical choice list, users select a category of choices and then select the choice within that category or group.

How are Classes, Properties, and Choice Lists Related?

You use Administration Console for Content Platform Engine to create and administer classes, properties, and choice lists.

The following diagram shows the relationships between classes, properties, and choice lists.



- Each class has multiple property definitions (that are based on property templates)

  A class consists of its root class properties and more custom properties.

  A property has a name and data type.
- A property template can optionally have an associated choice list.

  A choice list is a set of predefined values.
- A choice list can be associated with multiple property templates.

  For example, many properties can use a single choice list with color choice items.

Activity: Create a choice list

Choice lists help save time and prevent errors during data entry. Choice lists constrain property values to a pre-defined set values. In this activity, you create a choice list that you can later use in a property template.

In this activity, you will accomplish the following:

- Create a choice list.
- Verify the new choice list.

Create a choice list.

- Ensure that the IBM FileNet P8 Platform components are started.

  If you have not started them earlier, start the components by using the earlier activity: *Prepare your system - Start IBM FileNet P8 Platform.*
- In the **Mozilla Firefox** browser, click the **ACCE** bookmark or type the following URL: **http://vclassbase:9080/acce**
- Type **p8admin** for the **User name** field, **FileNet1** for the **Password** field, and then click **Log In**.
- On the left pane of the **EDU_P8** tab, expand the **Object Stores** folder and click the **SalesQA** object store.
- From the **SalesQA** tab, expand the **SalesQA** > **Data Design** node on the left pane and click **Choice Lists**.
- From the **Choice Lists** tab on the right pane, click **New**.

  The New Choice List tab opens.
- On the **New Choice List** tab, enter **Prospect Type** for the **Display name** field and then click **Next**.
- Select **String** for the **Choice list data type** field and then click **Next**.
- To add choice list items, click **New Items**.
- On the **New Items** page, type **Reseller** for the **Display name** field.

  When you click outside the Display Name field, the value is automatically populated for the Value field. Optionally, you can edit the value field.

  Make sure that the Value field is populated before you click Add.
- Click **Add**.

  The choice item is added to the pane.

- Repeat the steps to add another item with the name: **End user**
- Verify that both the items are listed on the pane.

- Scroll down and then click **OK** to close the **New Item** window.
- Back on the **New Choice List** tab, click **New Groups** to add a group to the choice list.
- On the **New Groups** page, type **Dealer** in the **Display name** field and then click **Add**.

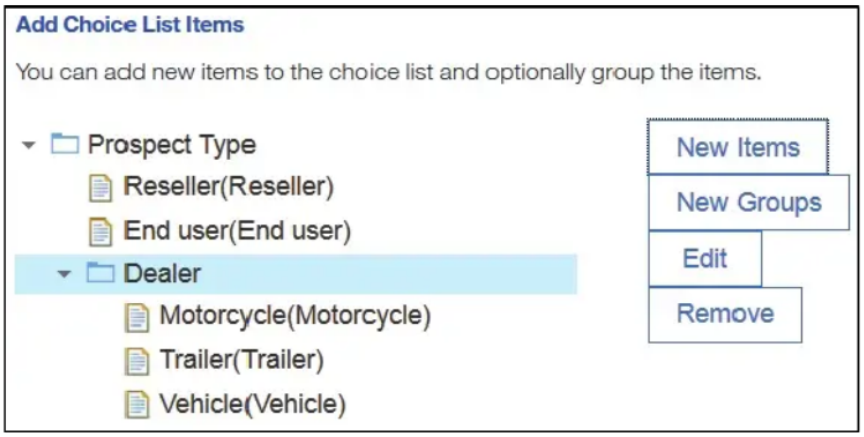
  The group name is added to the pane.

  Groups are not actual choice list items because they do not have a value property that is assigned to them. The groups are useful to organize items within a choice list. The groups can also be hierarchical.
- Click **OK**.

- Select **Dealer** and then click **New Items** to add choice items to the **Dealer** group.
- On the **New Items** page, add the following items.

  Display Name and Value are given as pairs.
  - Motorcycle, Motorcycle
  - Trailer, Trailer
  - Vehicle, Vehicle
- Click **OK**.

- Back on the **New Choice List** tab, verify that your completed choice list includes a group with vehicle choice items.



  You can edit the choice items. To edit, select the item and click Edit.

  You can also rearrange the choice items. To rearrange, click the item and use the Move Up or Move Down buttons.

- Click **Next**, and then from the **Summary** page, click **Finish**.
- On the **Success** page, click **Close**.

Verify the new choice list.

- On the **Choice Lists** tab, click **Refresh,** verify that **Prospect Type** is shown, and then click the **Prospect Type** link.
- On the **Prospect Type** tab, click the **Choice Items** subtab and then verify that the choice items that you defined are listed.
- Close the **Prospect Type** and **Choice Lists** tabs.
- Log out of the administration console and then close the browser.

Activity: Create property templates

In this activity, you create several property templates for the SalesQA object store in Administration Console for Content Platform Engine. You associate the choice list (that you created) with a property template. You will use these property templates in Folder and Document class definitions later.

In this activity, you will accomplish the following:

- Create a String property template.
- Create a Multi-valued property template.
- Create a Value-required property template.
- Create a property template with a Choice List.
- Create a Date Time Type property template.
- Create an Integer type property template.

Create a String property template.

In this task, you will create your first property template.

- In the **Mozilla Firefox** browser, click the **ACCE** bookmark or type the following URL: **http://vclassbase:9080/acce**
- Type **p8admin** for the **User name** field, **FileNet1** for the **Password** field, and then click **Log In**.
- On the left pane of the **EDU_P8** tab, expand the **Object Stores** folder and click the **SalesQA** object store.
- From the **SalesQA** tab, expand the **SalesQA** > **Data Design** node on the left pane and click **Property Templates**.
- From the **Property Templates** tab on the right pane, click **New**.
- On the **New Property Template** tab, enter **sales_prospect_name** for the **Display name** field.

  Click outside of the Display name field.
- Verify that the **Symbolic name** and **Description** fields are automatically populated.
- Optionally, edit the value for **Description** and then click **Next**.

  It is useful to provide a name prefix for property templates to track the property templates that are created for a specific application.

- Select **String** from the list for the **Data type** field and then click **Next**.
- For the **Select Choice List or Marking Set** page, leave the defaults (nothing selected) and then click **Next**.
- Select the **Single** option for the **Single or Multi-Value** field, and then click **Next**.
- On the **Summary** page, view the details and click **Finish**.
- On the **Success** page, click **Close**.
- On the **Property Templates** tab, click **Refresh**, scroll down, and then verify that **sales_prospect_name** is listed.

  You can also type first few letters of the name of your property template (sales) on the Filter field to filter the list.

Create a Multi-valued property template.

In this task, you will continue on the property templates page of the SalesQA object store.

- On the **Property Templates** tab, click **New**.
- From the **New Property Template** tab, enter **sales_contact_methods** for the **Display name** field.
- Click **Next**, select **String** from the list for the **Data type** field, and then click **Next**.
- For the **Select Choice List or Marking Set** page, leave the defaults (nothing selected) and then click **Next**.
- Select the **Multi** option for the **Single or multi-Value** field, select **Unique and ordered values** for **List order**, and then click **Next**.
- On the **Summary** page, view the details and click **Finish**.
- On the **Success** page, click **Close**.
- On the **Property Templates** tab, click **Refresh** and then verify that **sales_contact_methods** is listed.

Create a value-required property template.

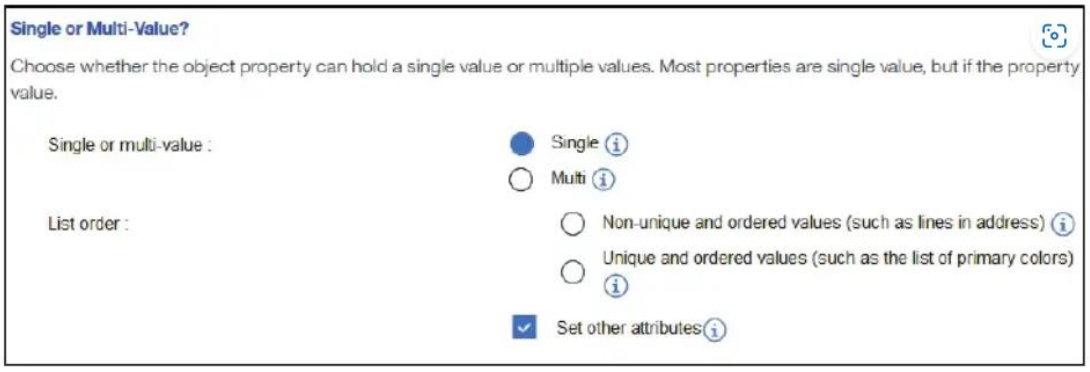
In this task, you will create a property template with a value-required status. If you want to use a property both with and without value-required status, you can set the required status as part of adding the property template to a class definition. The task is included here for you to practice setting on the property template itself.

- In the administration console, create a property template called **sales_prospect_id** by using the steps in the previous tasks and the following data.

  Accept the default values for the fields that are not listed here.

  - Name and Describe the Property Template page:
    - Display name: **sales_prospect_id**
    - Symbolic name: **sales_prospect_id**
    - Description: **sales_prospect_id**
  - Select the Data Type page:
    - Data type: **String**
  - Select Choice List Or Marking Set page:
    - Assign a choice list: **<None>**
    - Assign marking set: **<None>**
  - Single or Multi-Value? page:
    - Select the **Single** option
    - Select the **Set other attributes** option
  - Additional Property Template Attributes page:
    - Select the **Value required** option

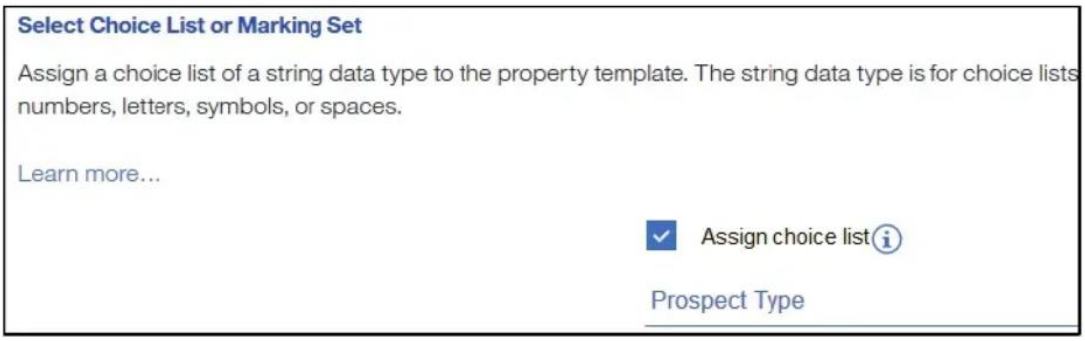
  When you select the Set other attributes option in the Single or Multi-Value page, the wizard opens with more pages to set other attributes for the property template.



- On the **Property Templates** tab, click **Refresh** and then verify that the new property template **sales_prospect_id** is listed.

Create a property template with a choice list.

- In the administration console, create a property template called **sales_prospect_category** by using the following values:
  - Name and Describe the Property Template page:
    - Name: **sales_prospect_category**
    - Symbolic name: **sales_prospect_category**
    - Description: **sales_prospect_category**
  - Select the Data Type page:
    - Data type: **String**
- On the **Select Choice List or Marking Set** page, select **Assign choice list** and then select the **Prospect Type** choice list that you created in the earlier activity.



- Click **Next**, select **Single** on the **Single or Multi-Value** page, and then click **Next**.
- On the **Summary** page, click **Finish**.
- Click **Close** on the **Success** page.
- On the **Property Templates** tab, click **Refresh** and then verify that the new property template **sales_prospect_category** is listed.

Create a Date Time Type property template.

- In the administration console, create a property template called **sales_last_contact_date** by using the following values:
  - Name and Describe the Property Template page:
    - Name: **sales_last_contact_date**
    - Symbolic name: **sales_last_contact_date**
    - Description: **sales_last_contact_date**

- Select the Data Type page:
  - Data type: **Date Time**
- Single or Multi-Value? page:
  - Select the **Single** option

Accept the default values for the fields that are not listed here.

- On the **Property Templates** tab, click **Refresh** and then verify that the new property template **sales_last_contact_date** is listed.

Create an Integer type property template.

- In the administration console, create a property template called **sales_times_contacted** by using the following values:
  - Name and Describe the Property Template page:
    - Name: **sales_times_contacted**
    - Symbolic name: **sales_times_contacted**
    - Description: **sales_times_contacted**
  - Select the Data Type page:
    - Data type: **Integer**
  - Single or Multi-value? page:
    - Select the **Single** option

  Accept the default values for the other fields that are not listed here.
- On the **Property Templates** tab, click **Refresh** and then verify that the new property template **sales_times_contacted** is listed.
- Log out of the administration console and then close the browser.

Activity: Create document and folder classes

Documents that are checked into Content Platform Engine require a class. You can organize the documents and other objects into folders. You can define the Document and Folder subclasses that are based on the needs of your organization.

In this activity, you will accomplish the following:

- Create a Folder class.
- Create a Document class.
- Test your Folder class.
- Test your Document class.

Create a Folder class.

In this task, you will create a custom Folder class and add property definitions to the class.

- In the **Mozilla Firefox** browser, click the **ACCE** bookmark or type the following URL: **http://vclassbase:9080/acce**
- Type **p8admin** for the **User name** field, **FileNet1** for the **Password** field, and then click **Log In**.
- On the left pane of the **EDU_P8** tab, expand the **Object Stores** folder and click the **SalesQA** object store.
- From the **SalesQA** tab, expand the **SalesQA** > **Data Design** > **Classes** node on the left pane.
- Right-click **Folder** and select **New Class** from the list.

  The New Folder Class tab opens on the right pane.
- On the **New Folder Class** tab, type **Sales Prospects** for the **Display name**.

  Verify that the Symbolic name and Descriptions fields are automatically populated. Click outside of the Display name field.
- Click **Next**, click **Finish** on the **Summary** page, and then click **Close** on the **Success** page.
- On the **SalesQA** tab, click **Refresh**.
- Expand the **SalesQA** > **Data Design** > **Classes** > **Folder** node on the left pane and then click **Sales Prospects**.

- From the **Sales Prospects** tab on the right pane, select the **Property Definitions** subtab and then click **Add**.
- On the **Add Properties** page, select the following properties that you want to add to this class.
  - **sales_prospect_category**
  - **sales_prospect_id**

  You can also type sales on the Filter field to filter the list. These are the property templates that you created in the previous activity.
- Scroll down and then click **OK** to close the **Add Properties** page.
- On the **Sales Prospects** tab, verify that the properties are listed, click **Save**, and then click **Close** to close the **Sales Prospects** tab.
- Leave the administration console opened for the next task.

Create a Document class.

In this task, you will create a custom Document class and add Property Definitions to the class.

- From the **SalesQA** tab, expand the **SalesQA** > **Data Design** > **Classes** node on the left pane.
- Right-click **Document** and select **New Class** from the list.

  The New Document Class tab opens.
- On the **New Document Class** tab, enter **Sales Prospect Doc** for the **Display name** field.

  Verify that the Symbolic name and Descriptions fields are automatically populated. Click outside of the Display name field.
- Click **Next,** click **Finish** on the **Summary** page, and then click **Close** on the **Success** page.
- On the **SalesQA** tab, click **Refresh**.
- Expand the **Data Design** > **Classes** > **Document** node and verify that the **Sales Prospect Doc** class is listed.
- Click **Sales Prospect Doc** in the left pane.

- From the **Sales Prospect Doc** tab on the right pane, select the **Property Definitions** subtab and then click **Add**.

- On the **Add Properties** page, type **sales** in the filter to show sales-related property templates only.
- Select the following properties that you want to add to this class.
  - sales_times_contacted
  - sales_prospect_name
  - sales_last_contact_date
  - sales_prospect_category
  - sales_contact_methods
- Scroll down, click **OK** to close the **Add Properties** page and then verify that the properties are listed on the **Sales Prospect Doc** tab.

Property	Data Type
sales_ contact_methods	String
sales_last_contact_date	Date Time
sales_prospect_category	String
sales_prospect_name	String
sales_times_contacted	Integer

- Click **Save** and then click **Close** to close the **Sales Prospect Doc** tab.
- Log out of the administration console and then close the browser.

Test your Folder class.

In this activity, you will create an instance of your Folder class in IBM Content Navigator and verify that the instance has the metadata as specified in the class specification. You will use the choice list and a required property.

- In the **Mozilla Firefox** browser, click the **Sample Desktop** bookmark or enter the following URL: **http://vclassbase:9081/navigator**
- Type **p8admin** for the **User name** field, **FileNet1** for the **Password** field, and then click **Log In**.

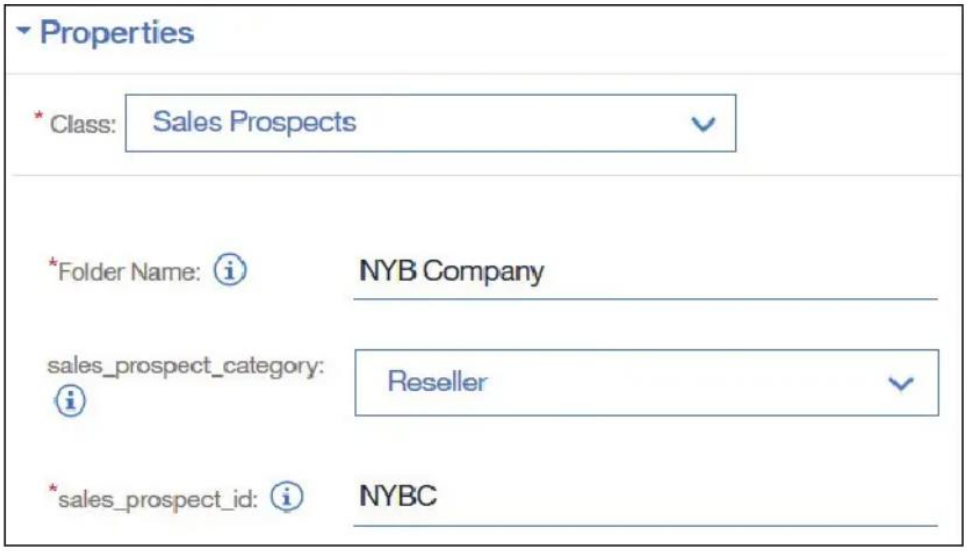
  The Content Navigator Sample Desktop opens in Browse view.
- Click the down arrow next to **LoanProcess** on the upper right and select **SalesQA** from the list.

- Click **New Folder** from the toolbar.
- On the **New Folder** page, under the **Properties** section, for the **Class** field, select **Sales Prospects** from the list and then click **OK**.
- Type **NYB Company** for the **Folder Name**.

  The name could be any text.
- For the **sales_prospect_category** field, select **Reseller** from the list and then click **OK**.

  Expand the Prospect Type node if the list is not shown.
- Enter **NYBC** for the **sales_prospect_id** field.

  The value for this field is required as you configured earlier. Notice that the required status is indicated by a red asterisk.



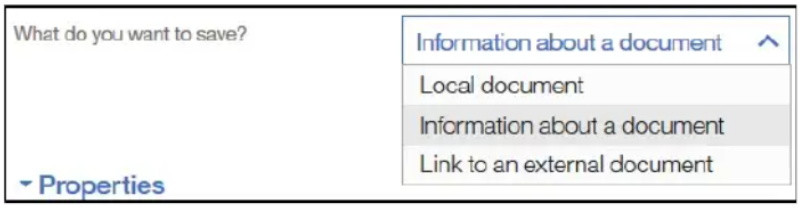
- Click **Add** on the lower right of the page.

  Back on the Browse page, verify that your new folder is listed.

Test your Document class.

In this activity, you create an instance of your Document class in IBM Content Navigator and verify that the instance has the metadata as specified by the class. You can add documents to the folder that you created.

- Double-click **NYB Company** to open the folder and then click **Add Document** from the toolbar.
- On **Add Document** page, for the **What do you want to save?** field, select **Information about the document** from the list.

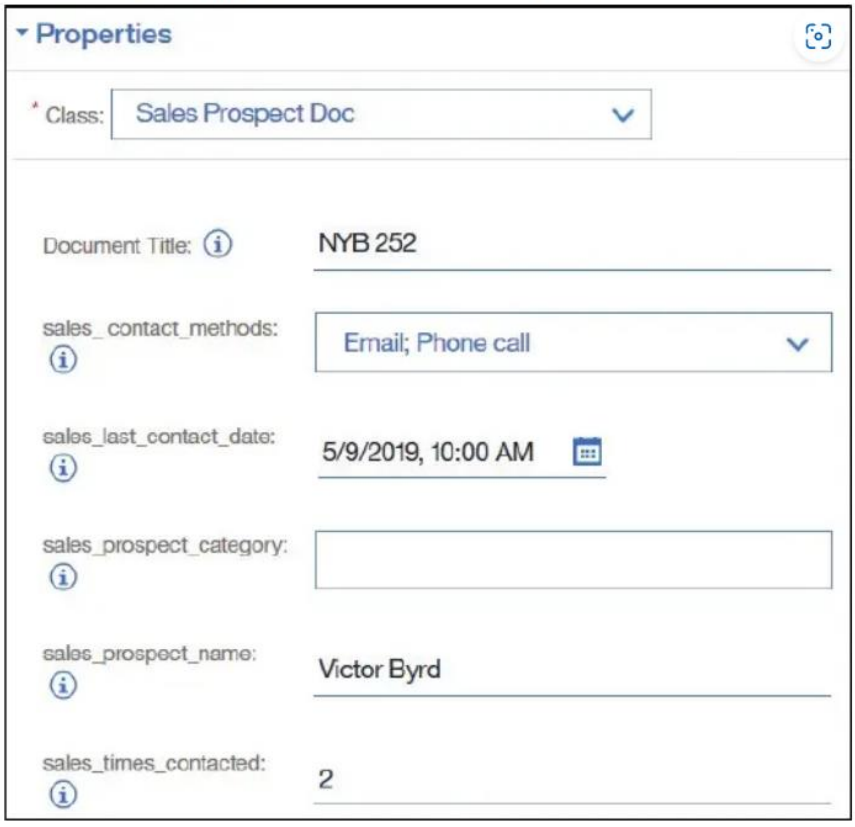


- Under the **Properties** section, for the **Class** field, select **Sales Prospect Doc** from the list and then click **OK**.
- Type **NYB 252** for the **Document Title** field.

  Title could be any text.

- Click the **sales_contact_methods** field, type **Email** on the **New** field, and then click **Add**.
- Repeat the above step to add **Phone call** and then click **OK.**
- Type or select the following values for the other fields:
  - sales_last_contact_date: **Five days before the current date**
  - sales_prospect_category: **Reseller** (Select from the choice list)
  - sales_prospect_name: **Victor Byrd**
  - sales_times_contacted: **2**

The completed page shows the values you entered:



- Click **Add** in the lower right corner.
- Back on the **Browse** page, verify that the new document is listed under your folder.
- Single-click the document and verify that the properties and their values are shown on the rightmost pane.
- Log out of ICN **Sample Desktop** and then close the browser.