Course Guide

IBM FileNet Content Manager 5.2.1: Work with Object Metadata

Course code F282 ERC 1.0

IBM Training

June 2016 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Contents

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

DB2® FileNet® Tivoli® WebSphere®

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other product and service names might be trademarks of IBM or other companies.

Course description

IBM FileNet Content Manager 5.2.1: Work with Object Metadata

Duration: 1 day

Purpose

Learn how to create document and folder classes and properties, and how to create event subscriptions on those classes to run automated actions.

Audience

An IBM FileNet Content Manager application builder is responsible for developing an IBM FileNet Content Manager application in a development environment.

Prerequisites

- Basic computer skills: Able to use computers, authenticate using a name and password, follow hyperlinks, and use common user interface elements such as buttons, menus, and tabs.First prerequisite
- Another prerequisite

Objectives

- Create a document class.
- Create a folder class.
- Create property templates.
- Create choice lists.
- Modify classes, properties, choice lists.
- Create an event subscription with an action.
- Update an existing event subscription with a new code module.
- Use a workflow subscription.

Contents

- Create document and folder classes
- Modify classes and properties
- Create an event subscription

Curriculum relationship

Go to http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=Y678448H04759K32 for more information about IBM ECM training paths.

Agenda

Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

(00:15) Course introduction
(00:30) Unit 1. Create document and folder classes
(00:30) Exercise 1. Create document and folder classes
(00:30) Unit 2. Modify classes and properties
(00:30) Exercise 2. Modify classes and properties
(00:30) Unit 3. Create event subscriptions
(00:30) Exercise 3. Create event subscriptions

Unit 1. Create document and folder classes

Estimated time

00:30

Overview

This unit describes how to create custom document and folder subclasses and their properties. You also learn about property inheritance and using choice lists to make data entry easier.

How you will check your progress

- Successfully complete the exercises.

References

IBM Knowledge Connection URL:

http://www.ibm.com/support/knowledgecenter/SSNW2F/welcome

IBM Training

Unit objectives

- Create a document class.
- Create a folder class.
- Create a property template with a choice list.

Figure 1-1. Unit objectives

IBM Training

What is a Class?

- A class is a model or template that is used to create objects.
  - The objects have common definition and common properties.
  - All objects belong to a class.
  - Each object of a class is identical in behavior but contains different property values.
  - The Class Description property identifies the class of an object.
- When an object store is created, it is pre-populated with a set of system-created classes.
  - 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.

Figure 1-2. What is a Class?

Help path

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Adding documents and objects>Classifying documents>Content Platform Engine classes>Class overview

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/classes/cl_concepts_classes.htm

Examples of system default classes:

They serve different functions.

The **annotation class** allows the user to link additional information to documents and other containable objects such as snapshots, 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 Document class is the most commonly used class and is the default.

The **folder class** holds a collection of objects that are related to each other by common properties.

The **referential containment relationship class** is used when storing documents in folders. An instance of this class connects exactly one document to exactly one folder.

The **subscription class** defines properties that specify the class, instance, or workflow that an event affects and the action to take in response to the event.

IBM Training

What are Document Objects?

- A Document object is an instance of the Document class or a subclass
- A Document object can have the following elements:
  - Content elements
  - Associated annotations
  - Custom metadata or properties that are used for identification
- Document content elements are stored
  - Locally, inside an object store
  - 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

Figure 1-3. What are Document Objects?

IBM Training

Folders

- A Folder represents a container that holds *containable* objects:
  - Child folders or subfolders
  - Documents, and their subclasses
  - Workflow definitions, Publish templates, others
- A Folder
  - Has a parent folder.
  - Has zero or more annotations that are associated with it.
  - Is independently securable.
  - Can be searched.

Figure 1-4. Folders

Help path

FileNet P8 Platform 5.2.1>Developing FileNet P8 applications>Content Engine Development>Content Engine Java and .NET Developer's Guide>Containment

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.dev.ce.doc/containment_concepts.htm%23fldr_folders

Each object store has an automatically created root folder that represents the default root container 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.

IBM Training

Containment Concepts

- Folders are directly contained.
  - They exist inside a folder.
  - They are deleted from the object store when they are removed from a folder.
- Document 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.

Figure 1-5. Containment Concepts

IBM Training

Class Inheritance

- Class inheritance is a relationship between classes.
  - One class inherits the structure and behavior that is defined in one or more other classes.
  - A subclass always inherits all of the properties of its superclass.
- Create a custom class.
  - Create a subclass of an original class definition that is provided in the system.
  - Add custom properties to the subclass that reflect your business needs.
- The Document class is the superclass of other document classes that you create.
  - Defines the behavior of a document.
  - Contains important system properties.
    - Example: Content Element and Version.

Figure 1-6. Class Inheritance

Help path

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Adding documents and objects>Classifying documents>Content Platform Engine classes>Class overview>Inheritance between classes

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/classes/cl_inheritance.htm

When you create an object store, the system automatically creates class definitions for all of the system-provided classes. To create custom classes, you create subclasses of these original class definitions. You can also add custom properties to these classes that reflect your business needs.

Custom properties of superclass and subclass

You can add custom properties to the default superclass, such as document class, but you cannot remove a property from a subclass that was inherited from its superclass. These inheritance rules can be important when you design your object stores and determine your superclass and subclass hierarchy. For example, at an insurance company, you might create a class named Policyholders. Subclasses of Policyholders might include a class named Claims. Additionally, changes to the superclass are inherited by its subclasses and are applied to any new versions of documents based on those subclasses.

You can add custom properties to a limited set of system classes (for example, the Document class, the Folder class, the Custom Object class).

IBM Training

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.
  - Subclasses inherit properties that are defined on a class.
- Definition of a property includes the following items:
  - Data type: Scalar or object-valued
  - Cardinality: Single or multivalue
  - Settability: Read-only, read/write
  - “Name” property indicator
  - ChoiceList assignment indicator.
  - Required or optional value indicator.

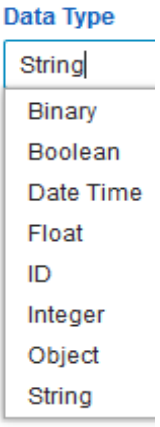


Figure 1-7. What is a Property?

Help paths

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Adding documents and objects>Classifying documents>Creating properties>Properties overview

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/properties/pr_concepts_properties.htm

FileNet P8 Platform 5.2.1>Developing FileNet P8 applications>Content Engine Development>Content Engine Java and .NET Developer's Guide>Properties

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.dev.ce.doc/property_concepts.htm

The screen capture shows the data types for a property.

The system classes have a set of basic properties that define the class. For example, a class has a property that indicates the creation date. When an object is created, the creation date property is populated with the current date. Creation date is an example of a system property. System properties have values that are supplied by the IBM FileNet Content Manager system.

You can edit some system properties; other properties are read-only. In addition to the system properties, you can assign custom properties to each class that reflect the content organization

needs for your business. For instance, a subclass of document class that is named Contracts can have custom properties for all parties involved in a contract.

Scalar data types:

- Binary, Boolean, Date Time, Float, ID, Integer, String

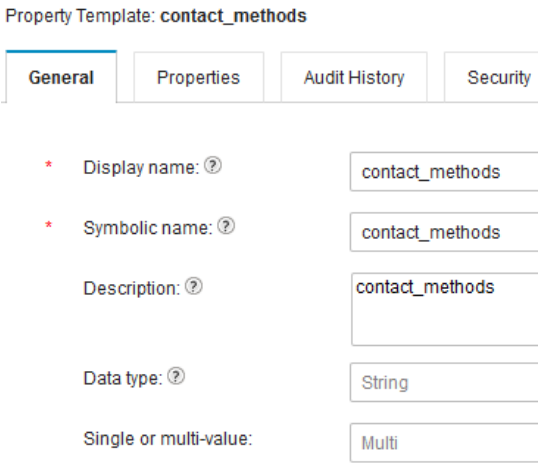
A single object-valued type is used to represent an object. For example, the annotations that are associated with a document can be represented as object type properties for the document.

IBM Training

What is a Property Template?

- 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.

- Examples of metadata properties:
  - Data type
  - Cardinality (single or multi-values).



Create document and folder classes

Figure 1-8. What is a Property Template?

Help path

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Adding documents and objects>Classifying documents>Creating properties>Creating a property template

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/properties/pr_create_property_template.htm

The screen capture shows the metadata for a property template.

A property template has no function or meaning in the object store until it is assigned to a class as a custom property. During property template creation, you assign values to metadata properties such as data type and cardinality (determines whether a property holds a single value or multiple values).

When you create a property template, do not assign it the same display name as an existing property in the same class family. Although you can create such a property template without causing an apparent error, you cannot later assign it to a class as a custom property.

The symbolic name of a property template is its programmatic identifier and is required to be unique within a class family only. A class family is defined by a root class (for example, Document, Folder, and CustomObject) and all of its subclasses.

Data types

String data types can hold numbers, but they are treated as characters rather than as numbers. For example, you might use a String data type for a numerical ID field because you can perform searches on the ID field and the results are ordered alphanumerically.

However, integer and Float data types are numbers that are used in calculations. Either data type can be used to calculate values such as *age greater than 55*. The Float data type is used for numbers that include decimal places.

The ID data type is used only for a globally unique identifier, or Microsoft GUID.

Example of a multi-value property: a phone number field with three entries (Work, Cell, Home)

If properties are defined with multiple values, these values can be defined as nonunique and ordered (such as an address) or as unique and ordered (such as a phone number).

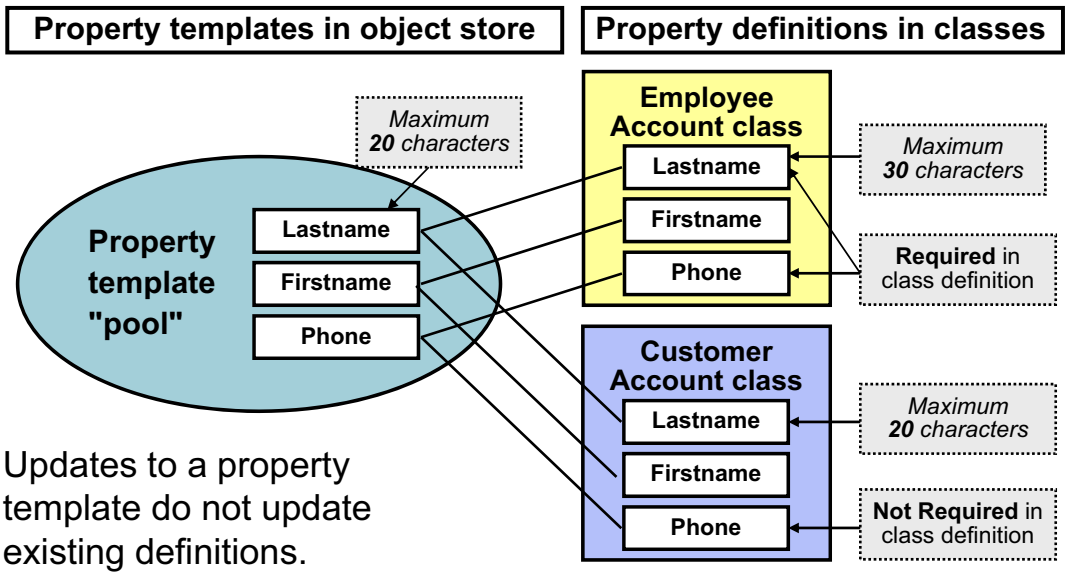
Multi-value options: select *Non-unique, ordered values* or select *Unique and ordered values* to specify the type of multi-value property.

Users have a different interface when they add values to a property that accepts multiple values. You can experiment with properties to observe the differences if you have time, to become familiar with these differences.

IBM Training

What is a Property Definition?

- Use a property template to create a property definition.
  - Multiple classes can use a property template.
  - Each resulting property definition is specific to the class.



Create document and folder classes

Figure 1-9. What is a Property Definition?

This diagram shows the available pool of property templates and their possible relationships to the property definitions in classes. The property templates in an object store can be used to define classes in that object store. One property template can be used within multiple class definitions. When you create a class, you can select property templates from the set of existing property templates in the object store.

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.

The distinction between property templates and definitions is subtle but important. If you change a property template, the change does not automatically cascade to property definitions. As a result, class definitions retain the original property definitions until you remove the original property definition and associate the revised property template with the class. This revised association creates a revised property definition for the class.

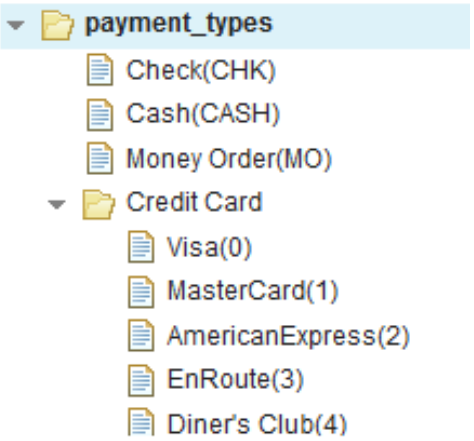
If you change a property template, only classes that are created after the change use the revised template. Classes that were assigned that property template before the revision continue to use the template as it was defined before the revision.

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.

IBM Training

What are Choice Lists?

- A choice list is a preset list of property values (choice items).
  - The user selects a value instead of typing an entry.
- Choice lists are used to do the following tasks:
  - Ensure valid entries.
  - Simplify and speed data entry.
  - Present elements in logical groups.



- Choice list features
  - Choice lists can be hierarchical.
  - Choice lists can have groups to organize the items.
  - Choice lists can be associated with multiple properties.

Figure 1-10. What are Choice Lists?

Help path

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Adding documents and objects>Classifying documents>Choice lists

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/choicelists/chl_concepts_choice_lists.htm

The screen capture shows an example of a nested choice list.

A choice list is a collection of predefined property values that can be used to present users with a list of values from which to choose.

Groups

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

Groups are not actual choice list items because they are not assigned a value. You add choice items to the group.

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

Example of a choice list

Colors have a Blue group and a Green group. The Blue group consists of Royal Blue and Navy Blue elements. The Green group consists of Pine Green and Emerald Green elements. Consider using choice lists when you are designing a custom class. Is a limited number of choices available for the property values? If so, then use choice lists. Choice lists make data entry faster for users and also ensure that the data entered is limited to the valid choice options.

IBM Training

Using Choice Lists

- Choice list requirements.
  - Assign a choice list to a property template.
  - Use either a string or an integer data type for a choice list.
  - Match the choice list data type to that of its associated property template.
- Choice list restrictions
  - A choice list cannot be used within another choice list.
- Usage options
  - You can assign one choice list to multiple property templates.
- Guidelines for Creating Choice Lists
  - Group choice list elements logically.
  - Limit the number of elements in each group.
  - Make sure that the values for the choice items are unique.
  - Plan the choice items before creating a choice list.

Figure 1-11. Using Choice Lists

Uniqueness of names

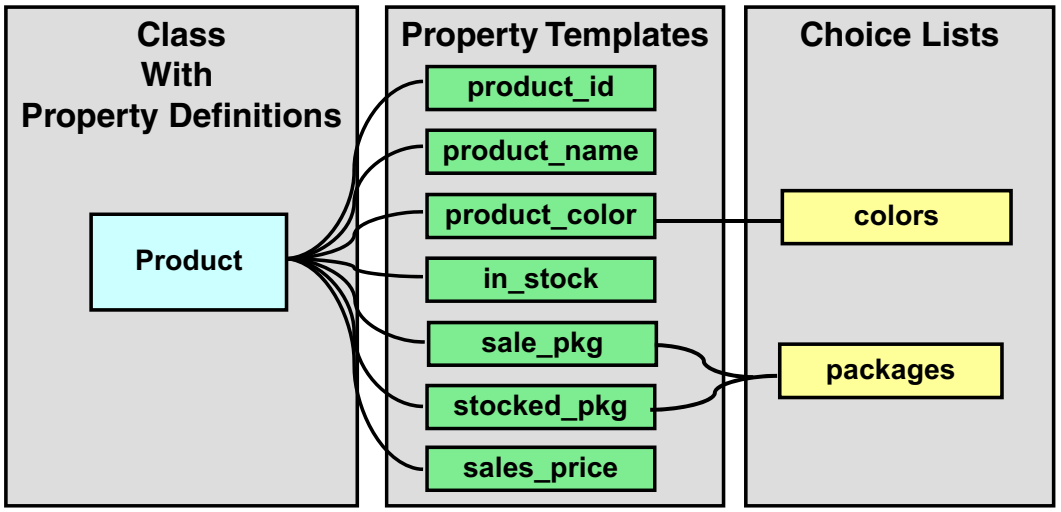
The Administration Console for Content Platform Engine tool enforces uniqueness of group names within a single choice list, and also uniqueness of display names within a single group. However, it does not enforce the uniqueness of item values. Sometimes, uniqueness is not wanted.

Refer to the documentation for an example of a use case where uniqueness is not desirable and also for the implications of changing existing choice list items.

IBM Training

How are Classes, Properties, and Choice Lists Related?

- Each class consists of its root class properties and more properties.
- Each property has a name and data type.
  - Each property can also have a choice list, a set of predefined values.



Create document and folder classes

Figure 1-12. How are Classes, Properties, and Choice Lists Related?

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

- Each class has multiple properties.
- Each property can have an associated choice list.
- It is optional for a property to have a choice list.
- Multiple properties can use the same choice list. For example, many properties can use a single choice list with Yes and No choice items.

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

IBM Training

Guidelines for Creating or Modifying a Class

- Add a property only to a user-defined custom class:
  - Adding a property to a system-created class applies to all subclasses.
- Create separate classes to match your business strategy.
- Minimize database row sizes.
  - Some databases have a maximum byte limit for the row length.
  - Each property that you create becomes a table column.
    - You can exceed the row size limitation and receive an error when adding more property definitions to a class.
- The number of levels of subclasses in a class is unlimited.
  - However, consider fewer levels to avoid complex content management design.

Figure 1-13. Guidelines for Creating or Modifying a Class

Help paths

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Adding documents and objects>Classifying documents>Content Platform Engine classes>Class overview

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/classes/cl_concepts_classes.htm

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Defining the repository infrastructure>Defining object stores

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/objectstores/os_creating_object_stores.htm

Database row size

Some databases have a maximum byte limit for the row length. Because each property that you create becomes a table column, you can exceed the row size limitation and receive an error when you add more property definitions to a class. Each column data type takes some bytes from the row length.

Example: DocVersion Table. When you add a custom property to the Document class, it adds a column to this table.

Table overflow support

Table overflow support applies to DB2 only. If table overflow support is enabled, Content Platform Engine creates an overflow table if the row-size limit of a base DB2 table is exceeded. The default value is false. Enabling table overflow support can slightly degrade database performance. When you plan your system, consider metadata design options before enabling this support.

IBM Training

Working with Properties

- Root classes
  - Do not add properties to the root classes
  - Do not change properties in the root classes in any way.
- Property templates
  - Name property templates carefully, after analysis and design.
  - Reuse property templates whenever possible.
  - Minimize the number of property templates that you use.
  - Use the same prefix in the names of property templates that are used together.
  - List the property templates that use a choice list in the description of the choice list.
    - Identify the property templates dependent on the choice list.
  - You can set default values for property definitions on the More tab.

Figure 1-14. Working with Properties

Metadata Object Names

- Names must be unique for the object type within the object store.
- Names must have 1 - 64 characters.
- Do not use spaces or numbers at the beginning of Content Platform Engine object names. Spaces and Leading numbers are removed in the symbolic name.

IBM Training

Unit summary

- Create a document class.
- Create a folder class.
- Create a property template with a choice list.

Figure 1-15. Unit summary

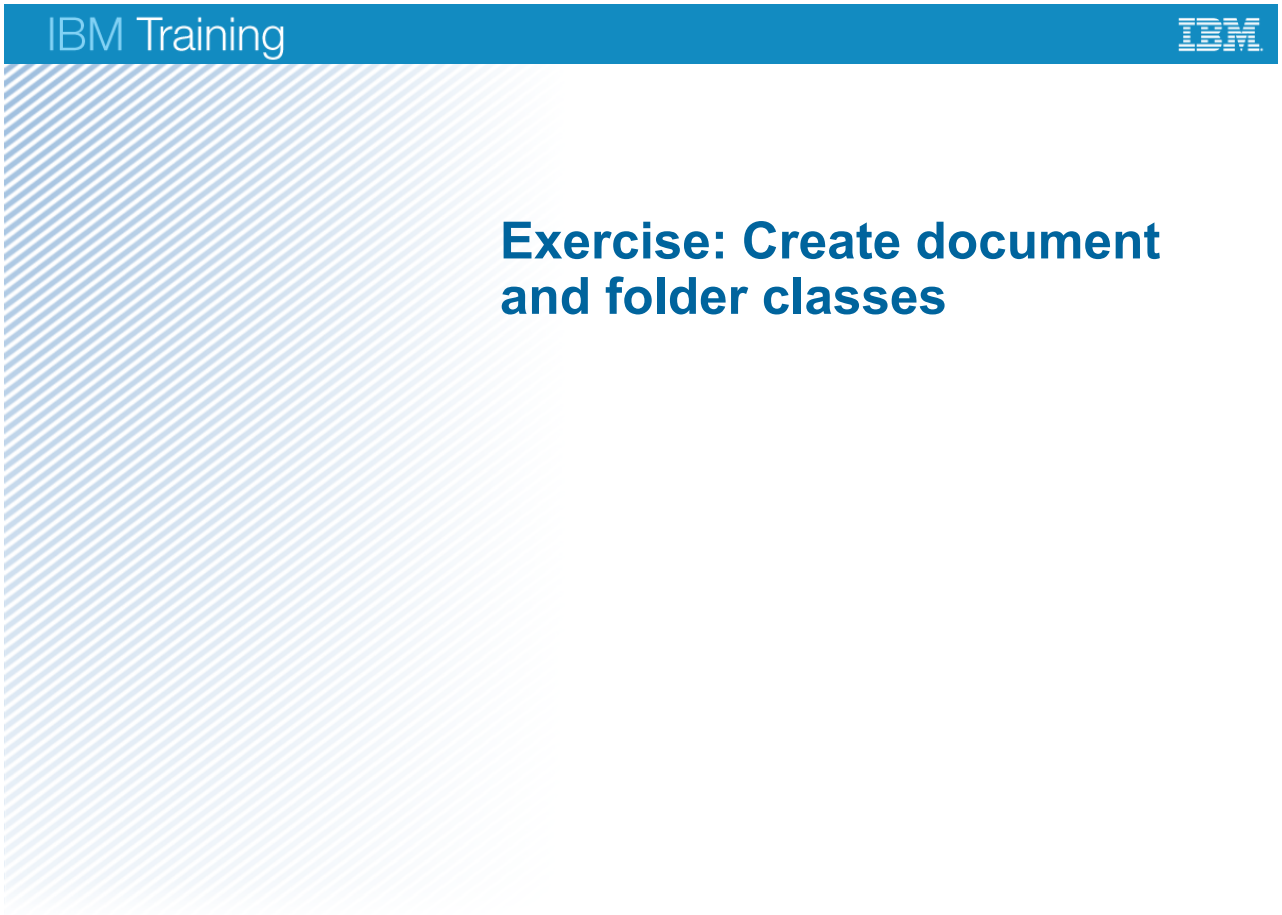


Figure 1-16. Exercise: Create document and folder classes

IBM Training

Exercise objectives

- Create a document class.
- Create a folder class.
- Create a property template with a choice list.

Figure 1-17. Exercise objectives

Unit 2. Modify classes and properties

Estimated time

00:30

Overview

This unit describes the relationships between classes, properties, choice lists, and objects. You learn about their dependencies, as well as how to update metadata within your application.

How you will check your progress

- Successfully complete the exercises.

References

IBM Knowledge Connection URL:

http://www.ibm.com/support/knowledgecenter/SSNW2F/welcome

IBM Training

Unit objectives

- Change property templates, choice lists, and classes.
- Work with metadata dependencies

Figure 2-1. Unit objectives

IBM Training

Changing a Display Name for a Metadata Object

- Display name or symbolic name?
  - If a name change for a metadata object is needed, avoid changing the symbolic name.
  - The display name for each object is visible in Administration Console and Content Navigator.
  - The symbolic name does not change when you change the display name.
  - Because APIs use the symbolic name, they continue to function after the display name is changed.
- What objects can be renamed?
  - You can modify the Display Name of the following objects:
    - Class definitions
    - Property templates
    - Choice lists

Modify classes and properties

Figure 2-2. Changing a Display Name for a Metadata Object

Example Scenario for changing a display name for a metadata object

You originally create a property template with a display name: product_type_178

The Graphical User Interface (GUI) team suggests that you change the display name to a more user-friendly name. The display name is shown in the user client.

You must change the name from product_type_178 to Product Type.

Property template modifications

- You can rename property templates and property definitions.
- A change to the property template name propagates to the property definitions based on the property template. Other changes are not propagated.
- You can change the description.
- You cannot change the data type or cardinality.

IBM Training

Modifying a Choice List

- You can modify a choice list in the following ways:
  - Change the choice list display name or description.
  - Add new choice items and groups.
  - Edit choice items and groups.
- You cannot delete the choice list object when it is associated with a property template.
- Choice list item changes
  - Changing the value of an existing choice list item of a choice list affects only the objects that are created after the change.
  - The objects that are created before the change retain the old value of the choice item.

Figure 2-3. Modifying a Choice List

Help path

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Adding documents and objects>Classifying documents>Choice lists

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/choicelists/chl_concepts_choice_lists.htm

Refer to the documentation for the implications of changing existing choice list items.

Example Scenario for adding a choice to group to an existing choice list

Your company deals with hardware-related products. You created a choice list that is called Product Type that contains different hardware items such as servers, laptops, and desktops. Your company decided to add software products to its portfolio. You must add a choice group for software items to the existing choice list (Product Type).

IBM Training

Assigning a Different Document Class

- You can change the class of an existing object.
  - Security permissions remain the same as the original class.
  - The content stays in the original storage and is not moved.
  - Properties that are not in the new class are removed from the object, along with their values.
- Preserve object instances with important data
  1. Identify a similar class with properties for the data that must be saved.
  2. If important properties are missing, add them to this class.
  3. Change the important instances to the similar class.
  4. Delete the original class definition of the objects.
- Examine the existing document versions for the history of the class assignments to the document.

Modify classes and properties

Figure 2-4. Assigning a Different Document Class

Help path

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Adding documents and objects>Working with documents>Working with document metadata>Assigning a different document class

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/docsandfolders/df_about_changing_doc_class.htm

Example Scenario for changing the Document class

You add a document and assign it a document class of `Manuals` because you plan to use it as a chapter in an installation manual.

You decide to use the same document in your training materials and change the document class to `Courseware`.

When you want to include this document as Technical Notice, you change it to a `Tech Notes` document class.

Preserving the data

When you change the class of an existing object, sometimes you want to delete object instances with their data. In other times, you want to keep data that is deleted when the instances are deleted.

Assigning a different class does not change the following items:

- Security permissions that the original document class directly applied to that document object. You can change the security by editing the security lists of the document object.
- The default storage of the content of existing document objects. The default storage area and storage policy of the new document class apply only to new instances of the class.
- Saved searches. If your saved searches use the former document class as a search parameter, you might no longer find the document.

Deleting a Document class

The FileNet P8 system prevents an administrator from removing any class that has objects that are instantiated on it.

Make sure to move instantiated objects to a class that supports the metadata with custom properties of same name and data type.

IBM Training

Metadata Dependencies

- An object instance depends on its subclass or class definition.
- A subclass definition depends on its class definition.
- A class definition depends on its property definitions.
- A property definition depends on its property template.
- A property template depends on its choice list.

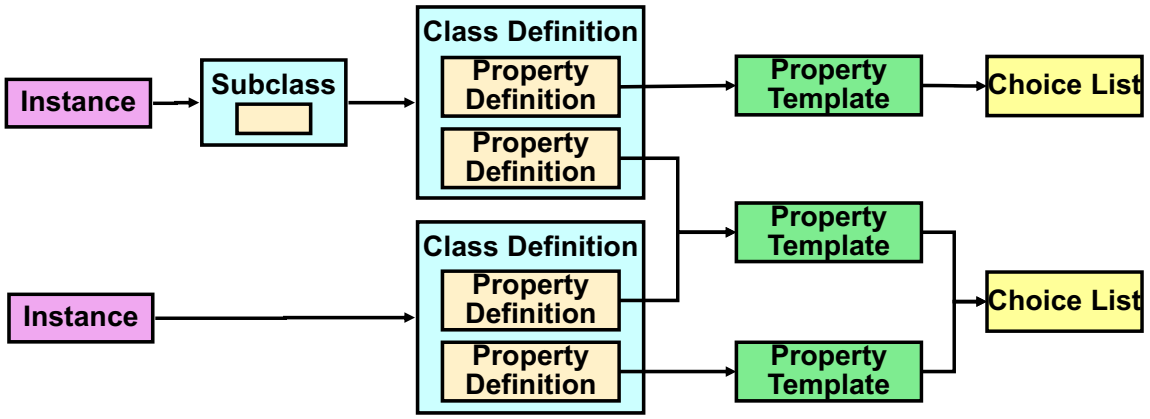


Figure 2-5. Metadata Dependencies

The diagram on this page provides a model of how class metadata is interrelated. Sometimes the dependencies are complex because the metadata objects are reusable. The dependencies result in the following constraints:

- If the class has subclasses or instances, you cannot delete it.
- If the property template has dependent property definitions, you cannot delete it.
- If a choice list is used by property templates, you cannot delete it.

To remove a dependency, you can either delete the object or edit the object to remove the dependency. If alternatives exist, avoid deleting an object to avoid data loss.

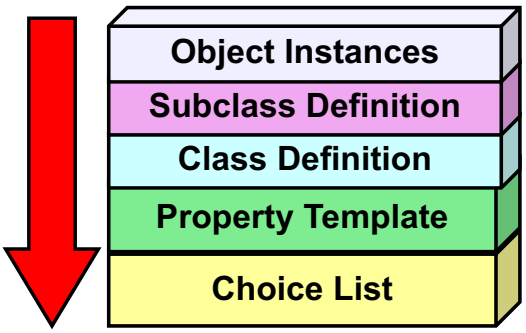
This diagram does not show the dependencies between the choice lists and property definitions. For example, if you remove the choice list from the property template without also removing the property definition from the class definition, you cannot delete the choice list.

Always use a test object store before you make metadata changes to a production system to identify and create solutions for any issues that might occur.

IBM Training

Deleting metadata

- Object dependencies
  - When one object references another object, the first object has a dependency on the second object.
  - You cannot delete an object when other objects refer to it.
  - Delete the referenced object or remove the reference to remove a dependency.
- Remove dependencies in the following sequence:
  1. Object instance
  2. Subclass definition
  3. Class definition
  4. Property template
  5. Choice list



Modify classes and properties

Figure 2-6. Deleting metadata

This diagram shows dependencies of various objects in an object store.

You must consider object dependencies before deleting metadata. This page introduces dependencies and a basic guideline for handling dependencies: delete the referenced object or remove the reference in the dependent object so that you can remove a dependency. The most direct sequence to remove dependencies is shown in the diagram and is specified in the sequence for removing dependencies.

Do not delete metadata that is referenced by custom code. Keep the metadata referenced by custom code to prevent breaking references in custom code.

IBM Training

Delete a property template

- To delete the property template (blue), what is removed first?
  - The associated property definition (orange) must be removed first.
    - This property definition is based on the blue property template.
    - **Note:** All instantiated objects with that property lose that property and its values.
  - It is not necessary to delete the whole class.

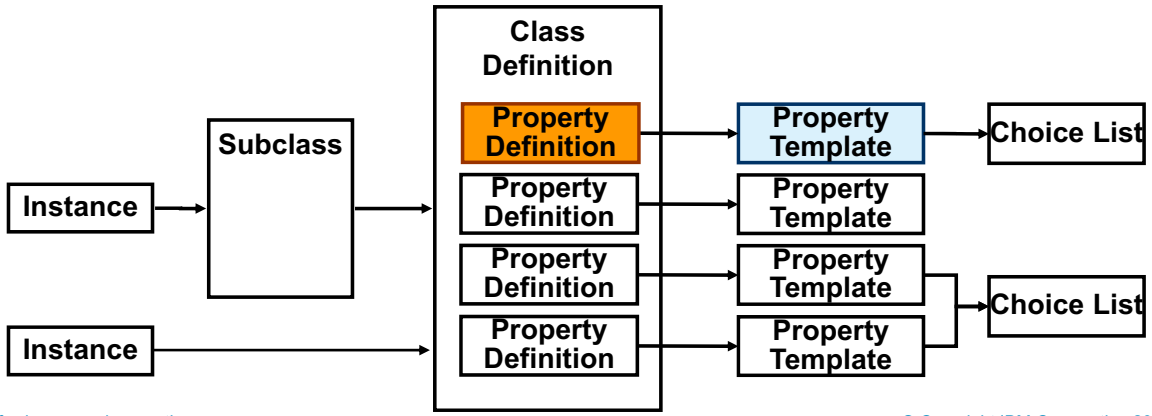


Figure 2-7. Delete a property template

This diagram shows two instances of a class. One instance belongs to a subclass of the class and the other instance belongs to the class. The class is shown with four property definitions, each based on a different property template. The first property template uses a choice list. The second property template does not use a choice list. The third and fourth property templates use the same choice list.

You can remove a property from a class. When you remove the property, all instantiated objects with that property lose it and its property values. You must carefully evaluate the potential data loss from metadata changes. When you remove the property definition from the class, you delete all the instantiated property values for that class.

IBM Training

Remove property definitions from a class

- You can remove custom properties from a class definition.
  - Use the Property Definitions tab of the Class page.
  - IBM Content Navigator does not immediately reflect the removal of a property from a class.
    - Refresh metadata cache to verify the removal.
- You cannot remove system properties or inherited properties.
  - You can hide these properties.
- Superclass property removal
  - If you remove a custom property from a superclass, you can choose to retain the property on child classes as a custom property.

Figure 2-8. Remove property definitions from a class

If you remove a property from a superclass, you cannot add the property back to the superclass until the property is deleted from all the subclasses.

The addition of a property to a class or removal of a property from a class are not immediately reflected in IBM Content Navigator. The metadata information must be refreshed in the web application server.

Note

During the lesson activities, you are going to refresh the metadata by restarting the Content Navigator in the WebSphere Application Server.

In production environment, this procedure causes the system to be unavailable for a few minutes, so it is generally avoided.

IBM Training

How do you find metadata dependencies?

- When you try to delete a property template that is referenced by other objects, you get an error message.

- To identify the objects that refer to the property template:
  - Open the Properties page for the property template.
  - The Used in Classes field lists the objects.

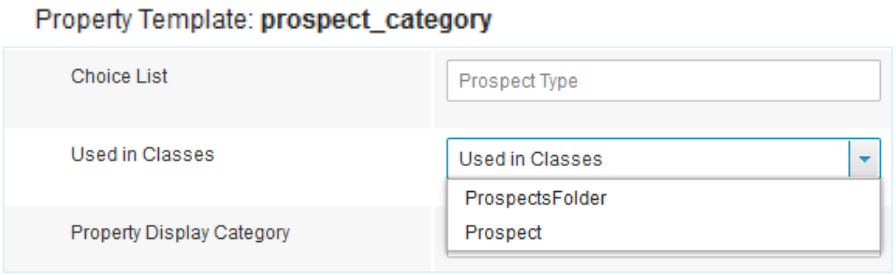


Figure 2-9. How do you find metadata dependencies?

The screen capture shows an error message. When you delete a property template that is referenced by other objects, you get an error message that states that the property template is referenced by other objects. The error message does not identify the objects that reference the property.

The second screen capture shows a list of objects that refer to the property template.

IBM Training

Delete a class

- To delete a class with dependencies, do the following steps:
  1. Remove its dependent subclasses and their instances.
  2. Remove its dependent instances.
  3. Delete the class.
- When you remove the dependent instances of a class, you have the following choices:
  - Preserve the instances by changing their class to a similar class.
  - Or, delete the object instances.

Figure 2-10. Delete a class

IBM Training

Unit summary

- Change property templates, choice lists, and classes.
- Work with metadata dependencies

Figure 2-11. Unit summary

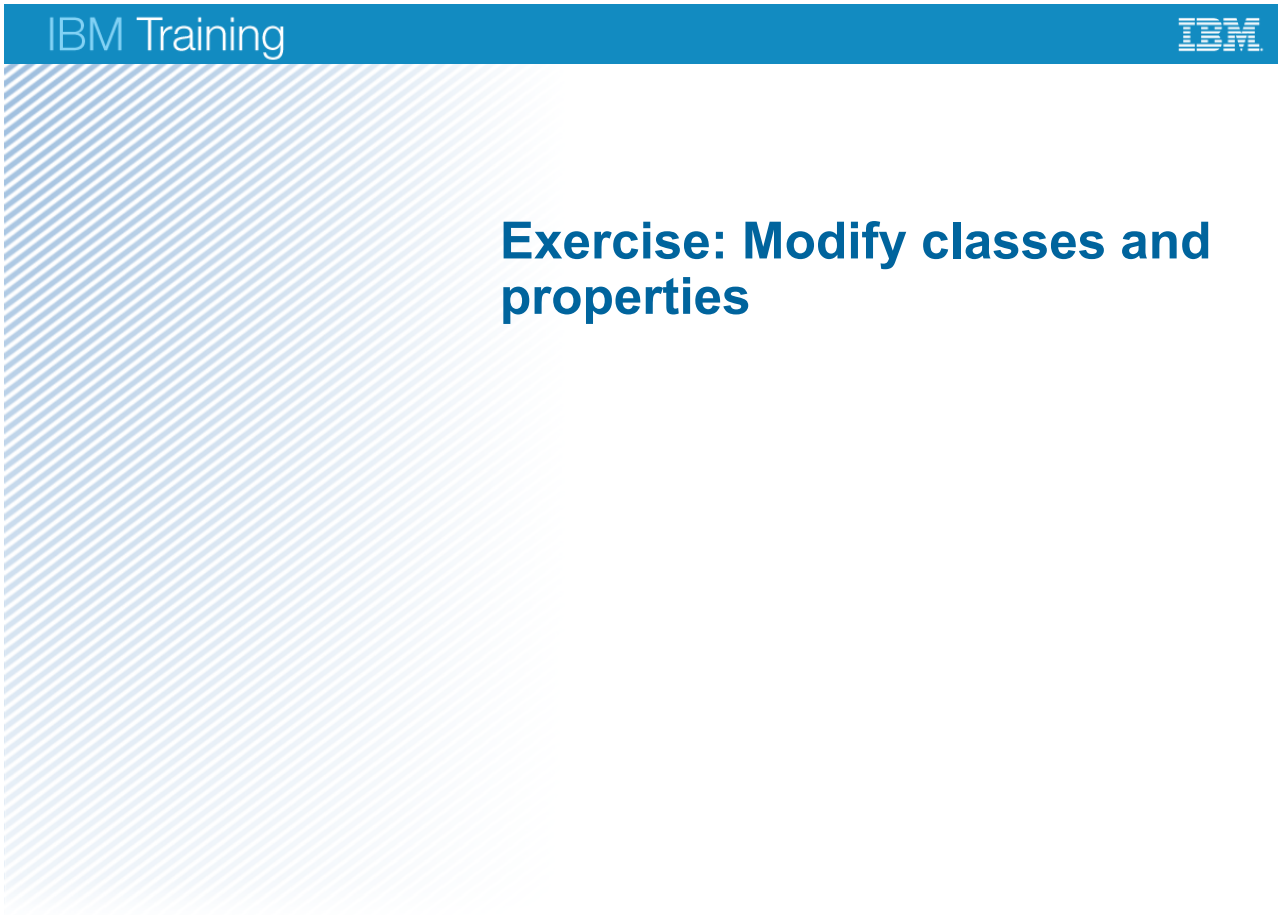


Figure 2-12. Exercise: Modify classes and properties

IBM Training

Exercise objectives

- Change a property template name.
- Modify a choice list.
- Change the class of an object.
- Remove a choice list from a class.

Figure 2-13. Exercise objectives

Unit 3. Create event subscriptions

Estimated time

00:30

Overview

This unit describes how you can use events (things that happen to objects) to automatically run actions, such as Java scripts or workflows.

How you will check your progress

- Successfully complete the exercises.

References

IBM Knowledge Connection URL:

http://www.ibm.com/support/knowledgecenter/SSNW2F/welcome

IBM Training

Unit objectives

- Create event subscriptions

Figure 3-1. Unit objectives

IBM Training

About Server Extensions

- You can extend Content Platform Engine functions in the following ways with your own server-based action handlers.
  - Events and Subscriptions
  - Change Preprocessors
  - Document Lifecycle Policies
  - Automatic Document Classification
- Java interfaces are provided with the product.
  - You create your action handlers by implementing them.
- The solution developer for your company provides the Java code for the action handlers.
  - As the solution builder, you create required Content Platform Engine objects that use the code to do the required actions.

Figure 3-2. About Server Extensions

Help path

FileNet P8 Platform 5.2.1>Developing FileNet P8 applications>Content Engine Development>Content Engine Java and .NET Developer's Guide>Server Extensions

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.dev.ce.doc/server_extensions.htm

This lesson describes the Events and Subscriptions topic.

Document Lifecycle Policies

A document lifecycle policy is an object that defines the "life" of a document in terms of phases or states through which the document cycles. The states are user-defined, and, when one state changes to another, a lifecycle-related system event fires. A lifecycle action handles the event. A lifecycle action is a user-implemented, server-side component. For example, you have a lifecycle policy that defines five states for loan application documents: Application, Approval, Funding, Servicing, and Closed. The policy also defines a lifecycle action. Each time a user changes the state of a loan application, the lifecycle action is started and do something based on the application's new state.

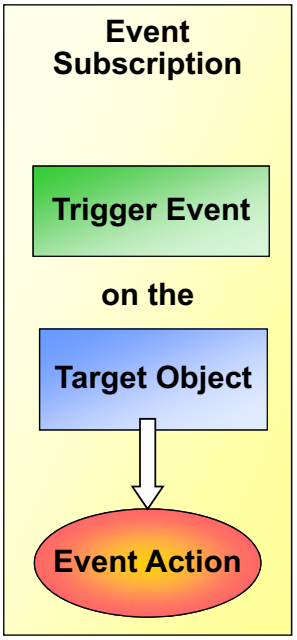
Automatic Document Classification

Documents that are added to an object store require a class. A document can be classified by a user, or it can be classified automatically when the document is checked in. The Content Platform Engine provides an extensible framework that enables an incoming document of a specified MIME type to be automatically assigned to a target document class. This framework also sets selected properties based on values found in the incoming document. A classification component, or classifier, does the work of assigning a document class. One such classifier that is packaged with the Content Platform Engine is XML Classifier.

IBM Training

Events and Subscriptions

- A subscription has the following elements:
  - One or more Trigger Events
    - A specified action on an object in an object store
    - Examples: Add Document, check-out, or delete
  - Subscription Target Object
    - A Content Platform Engine object upon which the events can be triggered.
    - Can include subclasses.
    - Examples: Document, Folder
  - Event Action (or workflow or both)
    - Identifies an event action handler (a Java class) that runs when an event is triggered on a target object.



Create event subscriptions

Figure 3-3. Events and Subscriptions

Help paths

FileNet P8 Platform 5.2.1>Administering>Administering Content Platform Engine>Changing objects>Subscribing to events

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.ce.admin.tasks.doc/p8pcc038.htm

FileNet P8 Platform 5.2.1>System overview>Features>Content management>Events and subscriptions

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.sysoverview.doc/p8sov006.htm

The diagram shows that an event subscription with a trigger event on a target class that results in an event action.

The trigger event can be a Content Platform Engine event or a user-defined custom event.

Types of subscriptions

- Event Subscription: runs user-defined code.
- Workflow Subscription: launches an IBM FileNet workflow.

The following actions on a Document object are some actions that can be used as an event trigger: create, check in, classify complete, update, check out, delete, promote version, demote version.

The following event actions are examples: launch a workflow, send an email message, file an object in a folder, create or associate related objects, query or update external databases.

Example

When an event is triggered on the target object, the event action is run. For example, you can have a subscription that notifies you by email (event action) when documents of the "Code Module" class (target object) are created (triggered event).

IBM Training

Define Subscription Filter

- Create a filter to restrict the application scope of a subscription.
  - Example: Filter out Creation events that Check-out triggers.

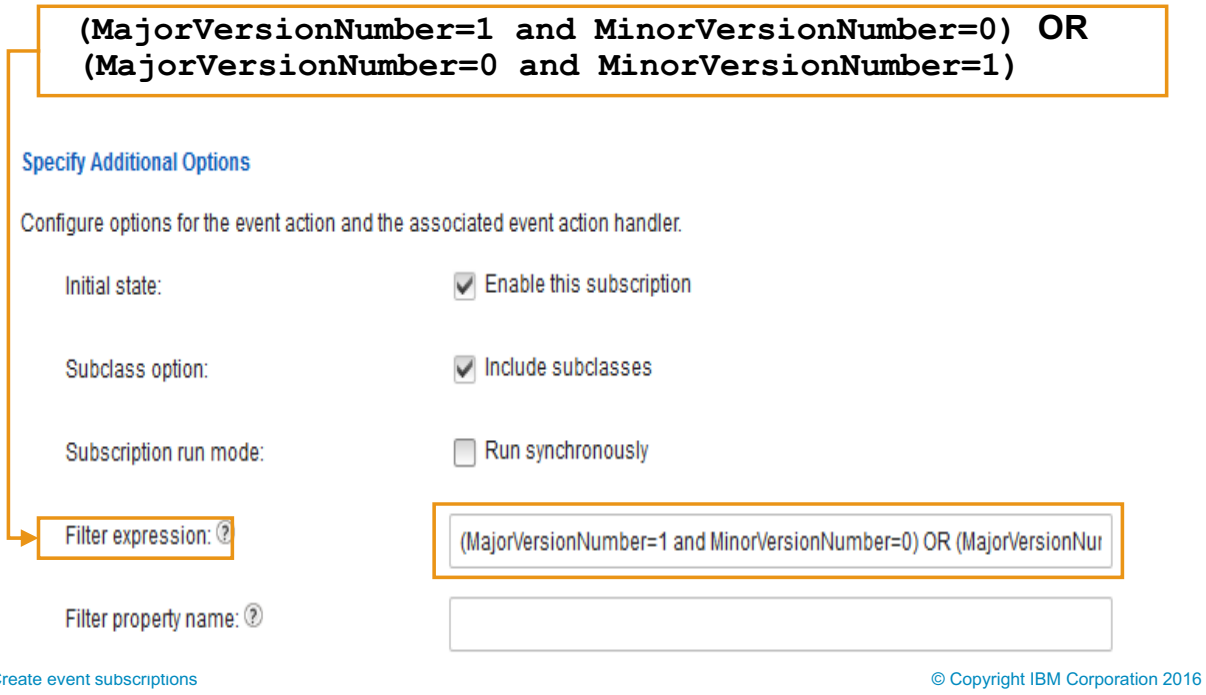


Figure 3-4. Define Subscription Filter

The screen capture shows the Create a Subscription page with a filter expression.

Creation event occurs when a user adds a document or checks out a document.

If you want to do something only when adding a document, you must filter out the Creation events that are caused by a check-out.

Checked out event

The document check-out operation triggers the Creation event (a new reservation object is created) in addition to the expected Check-out event. Therefore, if you are subscribing to the Creation event, you might need to create a filter in the subscription so that your event handler doesn't run when a check-out occurs.

The filter in the preceding example applies to the new document object (the source object) that is passed into the event handler. As a new document, it has a version number of 1.0 or 0.1.

Subscription scope

To restrict the application scope of a subscription, create a filter.

For example, restrict the scope to documents in class Memorandum with the security level setting: SecurityLevel = 'Confidential'

IBM Training

Workflow Subscription

- Workflow subscription
  - Starts the Workflow event action, which starts a workflow.
  - Specifies a workflow in addition to specifying the trigger event, target object, and event action.
  - The workflow definition must exist in the object store.
  - The workflow definition must be transferred.
- Example
  1. Trigger event: A document is added.
  2. Event subscription: Triggers an event action.
  3. Event action: Run by the Content Platform Engine.
  4. Specified workflow: Started by Content Platform Engine.

Create event subscriptions

Figure 3-5. Workflow Subscription

Before you create a subscription, the following objects must already be created:

- An Object or class of objects that the subscription applies
- A workflow – It must be transferred.

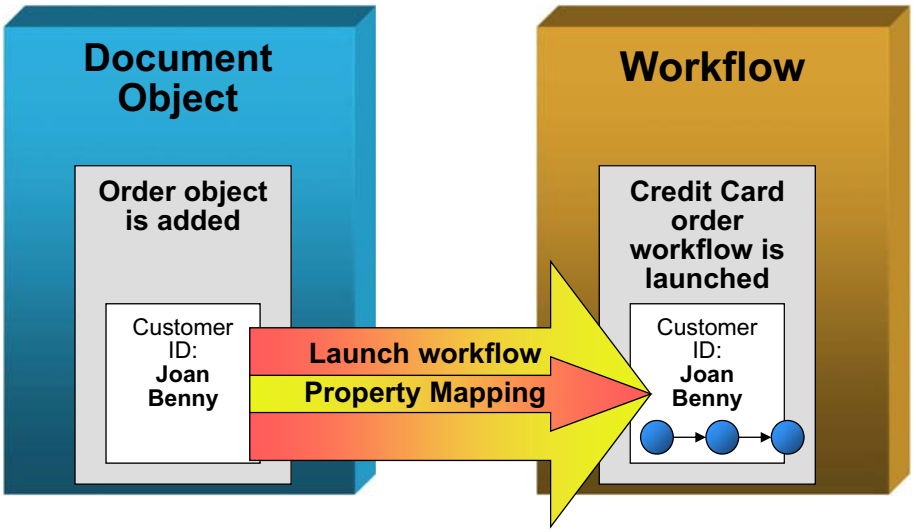
Important

A workflow subscription applies to a specific version of a workflow definition. If the workflow definition is updated, then the workflow subscription must be updated as well.

IBM Training

Launching Workflows: Property Mapping

- Event Subscription
  - When an event subscription starts a workflow, property mapping provides the workflow with information.
  - The object properties are mapped to workflow properties.



Create event subscriptions

Figure 3-6. Launching Workflows: Property Mapping

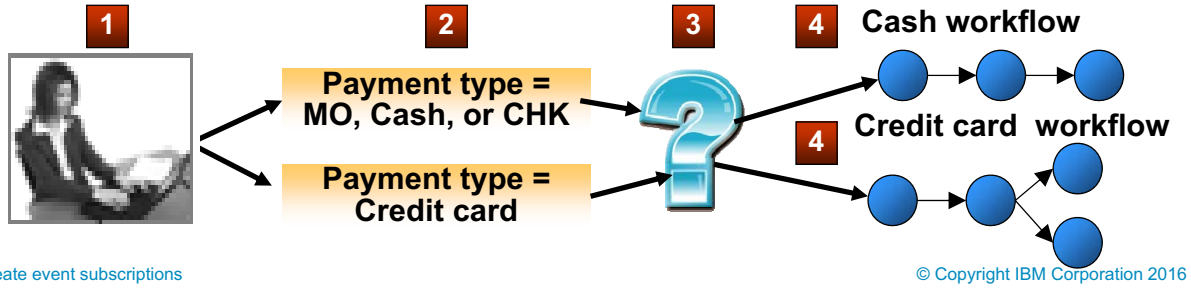
The diagram shows a Content Platform Engine event subscription launching a workflow.

The document property name does not have to match the property of the workflow but their data types must match.

IBM Training

Expressions in Event Subscriptions

- Filter Expression
  - Determines whether the subscription runs.
  - If the expressions are complementary, one workflow is started and the other workflow is not started.
- Order example:
  - Step 1: A customer orders product with credit card, payment type 1.
  - Step 2: Two event subscriptions exist for Order object creation.
  - Step 3: Match payment type to expression conditions.
  - Step 4: Expression condition starts the correct workflow.



Create event subscriptions

Figure 3-7. Expressions in Event Subscriptions

Help path

FileNet P8 Platform 5.2.1>Working with documents and other content>Working with documents with FileNet Workplace XT>Tools>Workflow subscriptions>Workflow Subscription wizard>Set expression

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.xt.user.doc/ae_help/subscription/wp_subscription_expression.htm

Filter expressions determine whether the subscription needs to run. This determination is more generic than launching workflow. For example, you can have an event action that updates an external database. In the subscription for the Loan document class, you can create a filter condition LoanAmount > 100000, which means that when a user submits a loan that is less than 100000, the action to update the database is not started.

The diagram is explained in the Order example text. The Order object example uses the following expressions:

If Payment type = MO, Cash, or CHK, then launch the cash workflow. (MO is money order. CHK is check.)

If Payment type = Credit card type, then launch the credit card workflow. (You can provide a number that can be used as a code for the type of credit card.)

Steps to create an Event Action

1. Create an event handler Java class.
   a. Compile the Java code to get the class file.
   b. Optionally, create the JAR file.
   c. You can also write the code in JavaScript.
2. Create an Event Action Object.
   a. Specify the Event Handler Java class name.
   b. Load Java class files or JAR files as code module content elements.
3. Assign the event action to a subscription.

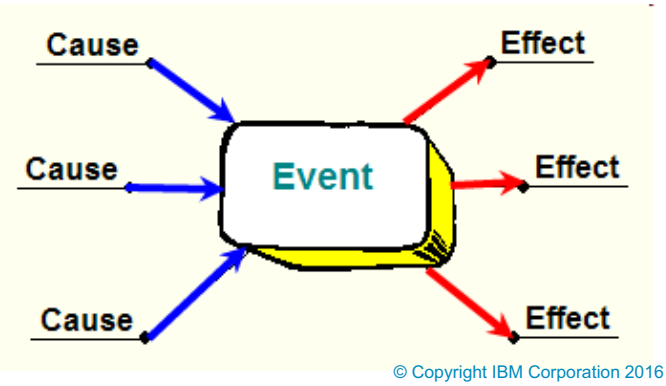


Figure 3-8. Steps to create an Event Action

After you create an event action that uses a Java class, you must subscribe to an event on an object (such as a document class). You can then test the code by triggering the actions on the object.

Code module

The code module represents a user-implemented Java event handler that is run when an action-based object event is raised. To represent a Java event handler as a code module, the Java class or JAR file must be on the Content Platform Engine.

To locate the event handler implementation, you have two options:

You can set the location in the class path of the application server on which the Content Platform Engine runs.

You can check it in to an object store as a CodeModule object.

Code modules are automatically available when the Content Platform Engine is deployed to multiple application server instances, or when you move your content metadata from one system to another. If you reference event handlers in the class path of an application server, you must manually distribute them to new systems.

Adding External JAR files to the Content Platform Engine

Some event actions might require external JAR files.

Add the JAR files as content elements of the code module.

These JAR files are imported with the Content Platform Engine objects.

Copy to the application server or set in the class path.

The JAR files are globally available for the other event actions.

IBM Training

Update Event Action with new code module version

- If you modify the code for a Java event action handler that is contained within a code module:
  - You must update any event action that references the code module.
- Steps to update:
  1. Check out the code module.
  2. Modify the Java event action handler source and compile.
  3. Check in the code module with the new version of the Java class.
  4. Copy the object reference for the code module.
  5. Update the Code Module property of the event action by pasting the object reference.

Figure 3-9. Update Event Action with new code module version

An event action identifies an event action handler, a Java class that runs when an event is triggered on a target object. For example, if a message is sent to an administrator when a document is deleted, sending the message is the event action. An event action, trigger event, and target object are defined in a subscription assigned to an object or a class.

IBM Training

Disabling Subscriptions

- You can disable a subscription without deleting it.
  - Use disable for testing.
    - Disable the event subscription while you fix the problem.
    - After you change the event action, re-enable the subscription.
  - Deleting is permanent, but disabling is temporary.
- Disabled subscriptions
  - Enabled column displays the value *False*.

Figure 3-10. Disabling Subscriptions

IBM Training

Subscription Run Mode

- Event subscriptions can be run synchronously or asynchronously.
- Synchronous Subscription
  - The operations of the object and the event actions are completed as a single transaction.
  - Failure in either results in rollback of both operations.
- Asynchronous Subscription
  - The operations of the object and the event actions are completed as separate transactions.
  - Object operation can succeed independently of the event action operation.

Figure 3-11. Subscription Run Mode

Help path

FileNet P8 Platform 5.2.1>System overview>Features>Content management>Events and subscriptions

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.sysoverview.doc/p8sov006.htm

When an asynchronous event is queued in an object store, any server within the same site as that object store can process that queued event, not just the server that generated the event.

Use case for Synchronous event:

A synchronous event might be applied to a Claim folder class that returns an error if a document that does not belong to the Claim Document class is filed in the folder.

Triggering event: Filing a Claims document in the Claims folder

Target class: Claims Folder class

Event Action: Send an email to the Claims Department for further processing.

When a wrong document is filed in the folder, it returns an error and subscription process does not complete the action of sending an email.

IBM Training

Unit summary

- Create event subscriptions

Figure 3-12. Unit summary

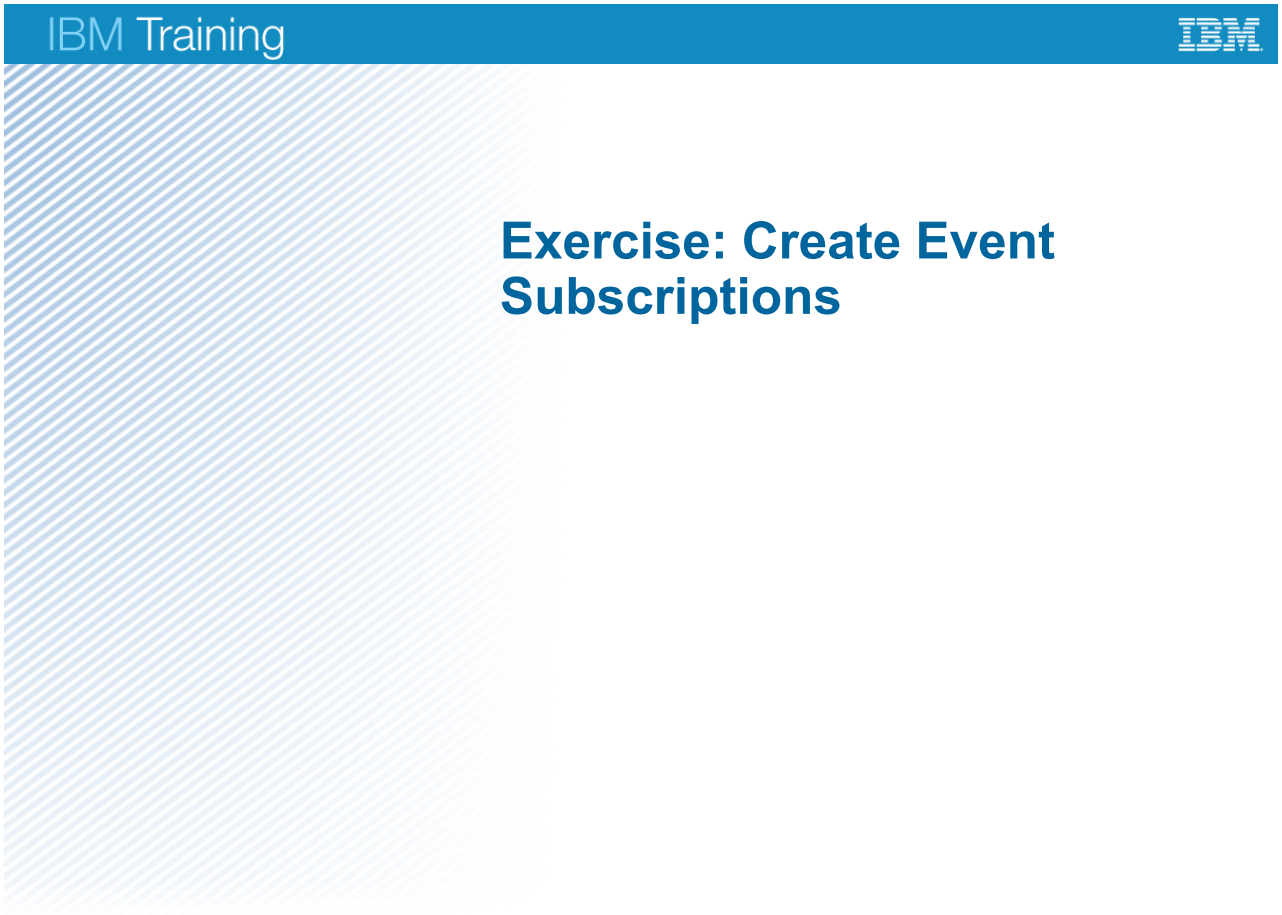


Figure 3-13. Exercise: Create Event Subscriptions

IBM Training

Exercise objectives

- Create a subscription with an event action.
- Update an event action with a new code module.
- Examine a workflow subscription.

Figure 3-14. Exercise objectives

IBM Training