IBM Enterprise Records 5.1: File Plan Design

(Course code F181)

Student Notebook

ERC 1.0

Authorized

IBM | Training

Trademarks

IBM® and the IBM logo are registered trademarks of International Business Machines Corporation.

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

FileNet®	InfoSphere™

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.

September 2011 edition

The information contained in this document has not been submitted to any formal IBM test and is distributed on an “as is” basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

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 the names and addresses used by a real business enterprise is entirely coincidental.

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® and the IBM logo are registered trademarks of International Business Machines Corporation.

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

FileNet® InfoSphere™

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 Enterprise Records 5.1: File Plan Design

Duration: 1 day

Overview

This course is for those whose job includes responsibility for designing the file plan for an IBM Enterprise Records system and making decisions regarding record retention, disposition, and security. You use the IBM Enterprise Records web application to create the file plan. You work with a records administrator, an installer, a database administrator, and a programmer. You must be able to organize and communicate records management system requirements to the other roles.

You work with a fully functioning IBM Enterprise Records system to practice the skills required for designing file plans for records management.

Audience

Anyone who is responsible for planning the records management strategy for their organization that uses IBM Enterprise Records.

Prerequisites

F178 - IBM Enterprise Records 5.1: Core Skills

F040 - IBM FileNet P8 Prerequisite Skills 4.5

Skills taught

After completing this course, you should be able to:

- Design a functional and efficient records management file plan.
- Coordinate file plan development and deployment.

Course outline

- Coordinate file plan development
- Core file plan design concepts
- Create a functional classification file plan
- Create a retention model file plan
- Create a case model file plan

Unit 1. IBM Enterprise Records 5.1: File Plan Design

What this unit is about

This course is for those whose job includes responsibility for designing the file plan for an IBM Enterprise Records system and making decisions regarding record retention, disposition, and security. You use the IBM Enterprise Records web application to create the file plan. You work with a records administrator, an installer, a database administrator, and a programmer. You must be able to organize and communicate records management system requirements to the other roles.

You work with a fully functioning IBM Enterprise Records system to practice the skills required for designing file plans for records management.

What you should be able to do

After completing this unit, you should be able to:

- How to coordinate file plan development
- Core file plan design concepts
- How to create a functional classification file plan
- How to create a retention model file plan
- How to create a case model file plan

How you will check your progress

- Successfully complete the lesson exercises.

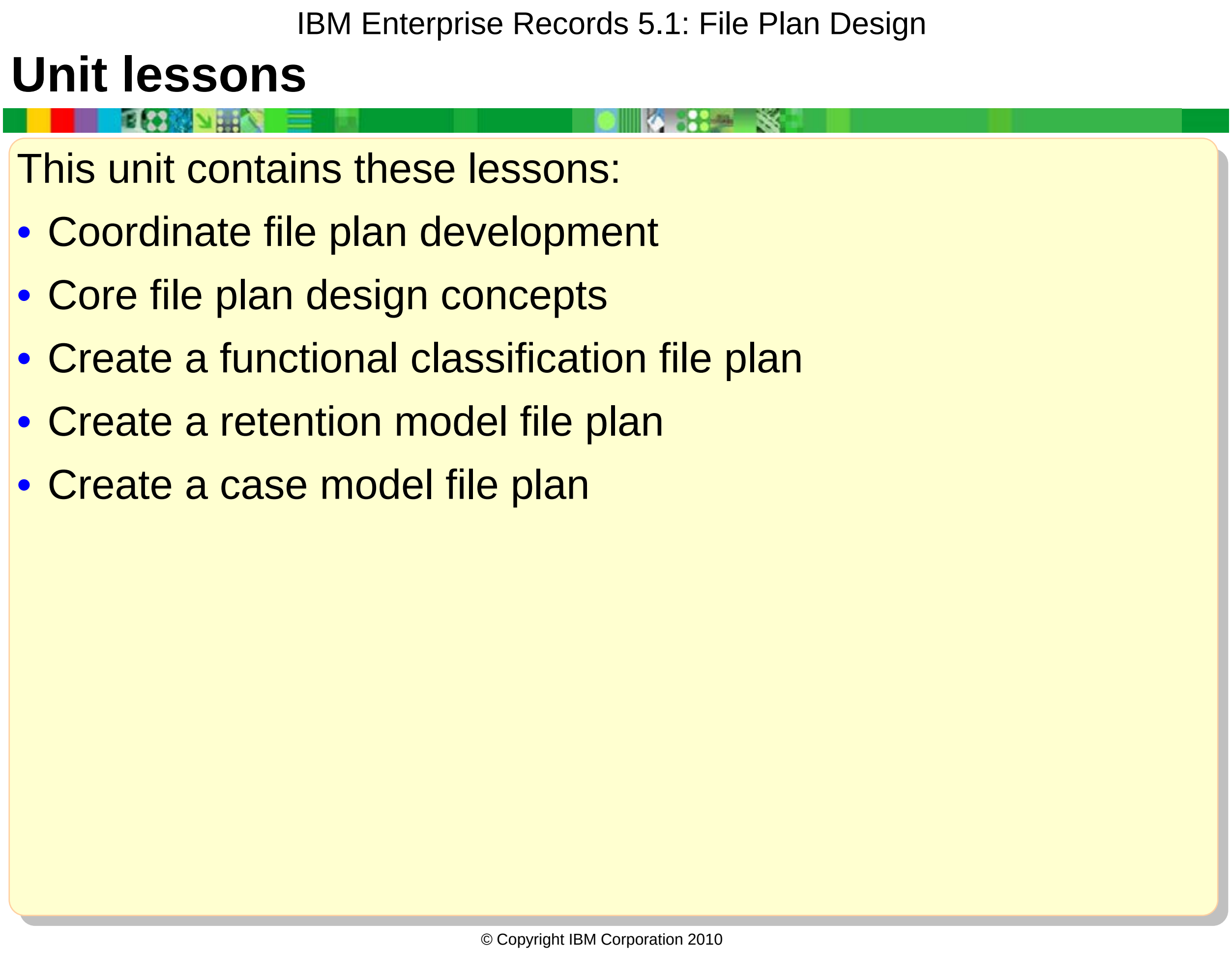
References

IBM FileNet P8 Documentation

http://publib.boulder.ibm.com/infocenter/p8docs/v5r1m0

Best practices to improve performance for IBM InfoSphere Enterprise Records (formerly IBM FileNet Records Manager)

http://www-01.ibm.com/support/docview.wss?rs=3286&context=SSNVVQ&uid=swg21330987



IBM Enterprise Records 5.1: File Plan Design

Unit lessons

This unit contains these lessons:

- Coordinate file plan development
- Core file plan design concepts
- Create a functional classification file plan
- Create a retention model file plan
- Create a case model file plan

Figure 1-1. Unit lessons F1811.0

Notes:

Lessons in this unit

This unit has 5 lessons. After the first lesson, each lesson relies on information and skills taught in the prior lessons. For best results, do these lessons in the sequence presented.

Coordinate file plan development. In this lesson, you learn about coordinating file plan development within the broader context of a records management system.

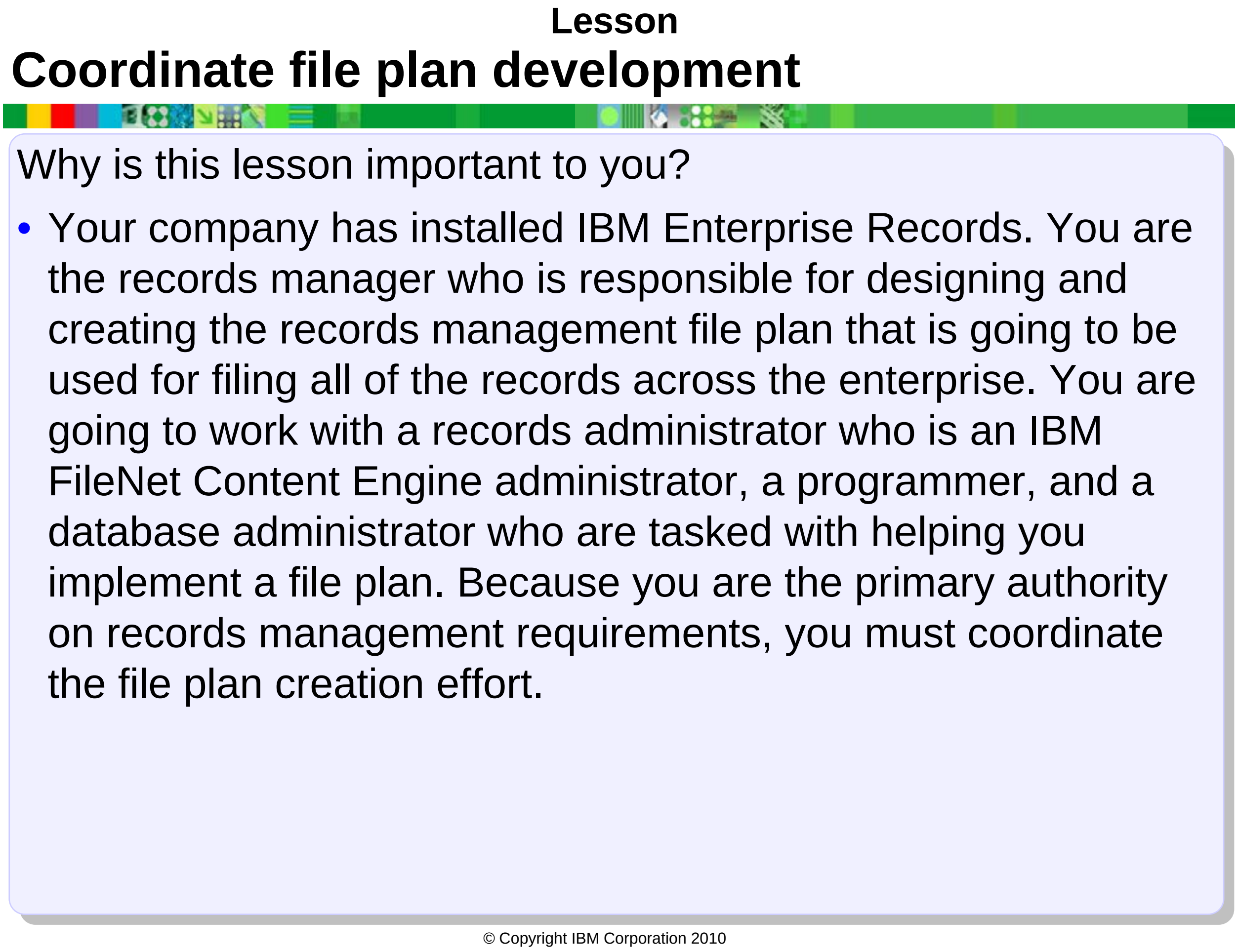
Core file plan design concepts. In this lesson, you learn about key concepts in file plan design that have a large impact on records management system performance.

Create a functional classification file plan. In this lesson, you learn about the classification system endorsed by the International Organization of Standardization and how to structure a file plan that is flexible with regard to changes in regulation, but resilient with respect to organizational structures.

Create a retention model file plan. In this lesson, you learn about designing a file plan in which the primary purpose is retaining documents for a given period of time, and other considerations are nominal.

Create a case model file plan. In this lesson, you learn about designing a file plan in which records are case folders with several documents with different declare times, formats, and originating locations, but which must be destroyed together after the case is closed.

Lesson 1.1. Coordinate file plan development



Lesson

Coordinate file plan development

Why is this lesson important to you?

- Your company has installed IBM Enterprise Records. You are the records manager who is responsible for designing and creating the records management file plan that is going to be used for filing all of the records across the enterprise. You are going to work with a records administrator who is an IBM FileNet Content Engine administrator, a programmer, and a database administrator who are tasked with helping you implement a file plan. Because you are the primary authority on records management requirements, you must coordinate the file plan creation effort.

Figure 1-2. Coordinate file plan development F1811.0

Notes:

Coordinate file plan development

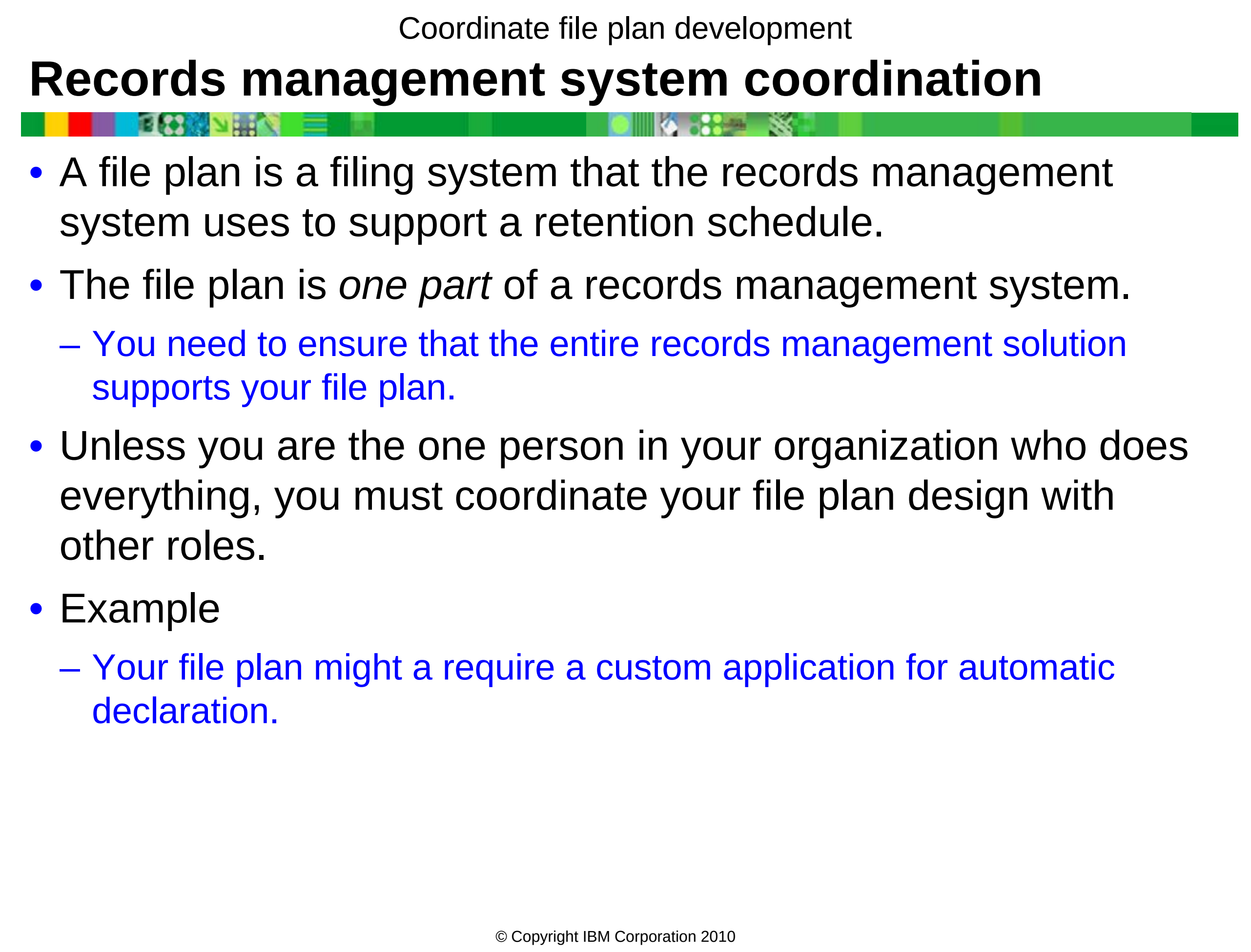
Activities that you need to complete

- Coordinate file plan development.

Figure 1-3. Activities that you need to complete F1811.0

Notes:

These are the activities that you need to complete.



Coordinate file plan development

Records management system coordination

- A file plan is a filing system that the records management system uses to support a retention schedule.
- The file plan is *one part* of a records management system.
  - You need to ensure that the entire records management solution supports your file plan.
- Unless you are the one person in your organization who does everything, you must coordinate your file plan design with other roles.
- Example
  - Your file plan might a require a custom application for automatic declaration.

Figure 1-4. Records management system coordination F1811.0

Notes:

Help path

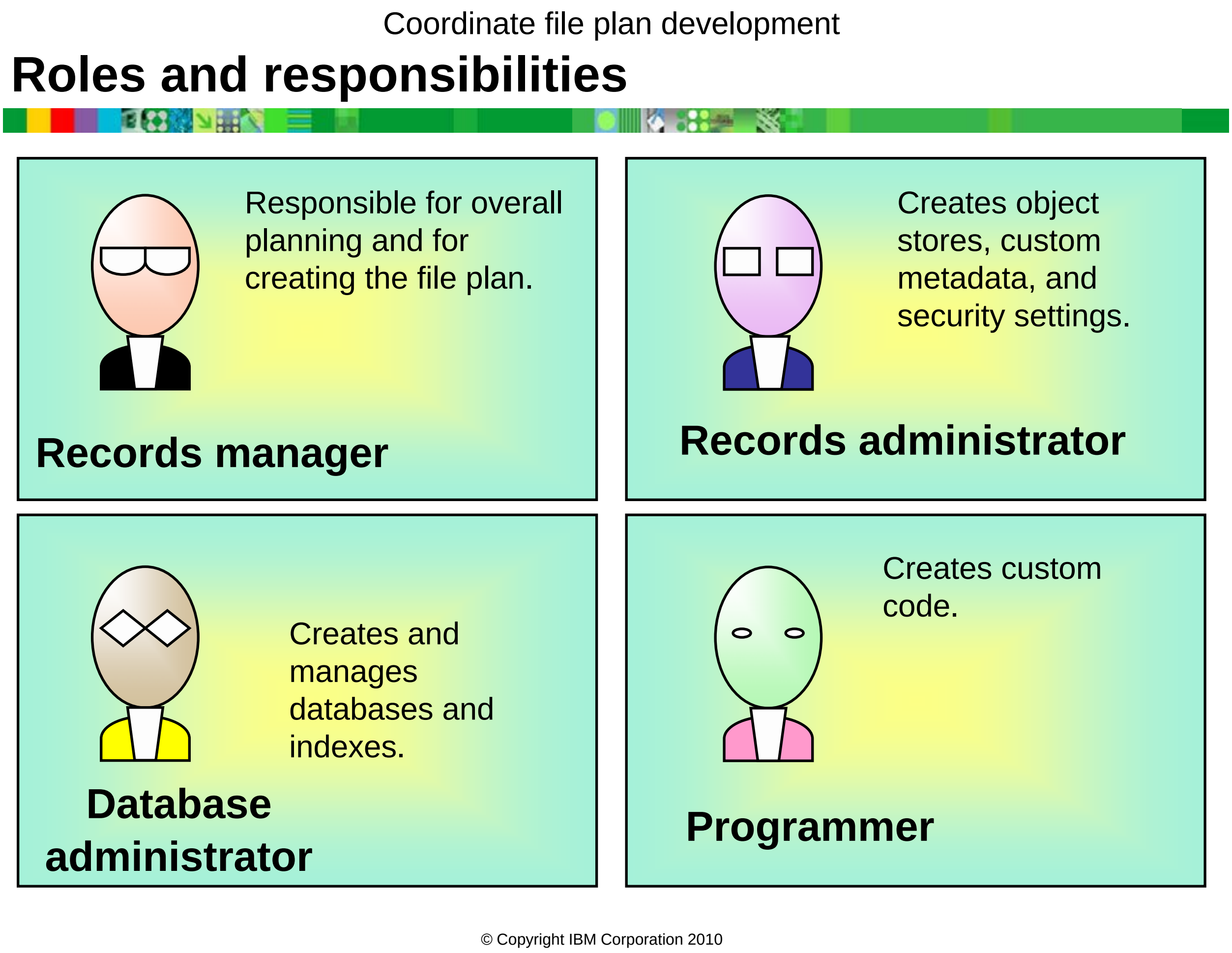
- Search for "file_plans.htm"

This course is about designing file plans. As a records manager, you need to design the file plan, but the file plan is only part of the entire records management system. You also need to create the disposition schedules to apply to the file plan, the actions that are performed at the end of the disposition phases, and the trigger events that are used to start the retention interval.

For the file plan to work efficiently, you must also plan the records management system around it. You need to know how your file plan design fits in with the development of the records management solution. Careful planning and coordination are necessary to ensure that the file plan is fully supported by the metadata, the Disposition Sweep schedules, custom programming, workflows, or any other elements of the total solution.

To summarize: You are the records manager and it is your responsibility to ensure that the records management system handles records according to your decisions. To do this, you need to ensure that the entire records management system supports your file plan.

In most organizations, you need to coordinate your file plan design with other people who are responsible for systems outside of your domain of expertise or authorization. For example, your file plan might be designed to support thousands of record declarations in a day. However, without a mechanism for automatically declaring thousands of records per day, the file plan by itself is not a complete solution. You might need to have a programmer develop a custom program for you.



Coordinate file plan development

Roles and responsibilities

Responsible for overall planning and for creating the file plan.

Records manager

Creates object stores, custom metadata, and security settings.

Records administrator

Creates and manages databases and indexes.

Database administrator

Creates custom code.

Programmer

Figure 1-5. Roles and responsibilities F1811.0

Notes:

The diagram shows four roles that need to be coordinated: records manager, records administrator, database administrator, and programmer.

Unless you are the sole administrator who does a bit of everything, you need to coordinate the efforts of several different people in order to get the records management system functioning.

As the records manager, you are the primary decision maker about how records need to be managed. Because of your extensive background in records management regulations, you are the one who decides how the file plan must be designed and how records must be disposed of. You must be aware of the volume of records that your organization declares each day and how best to design a records management system to handle the workload. You must coordinate the efforts of the other roles in order to ensure that the records management system is compliant with regulations and guidelines. Therefore, your role is central to the planning, development, and deployment of the file plan.

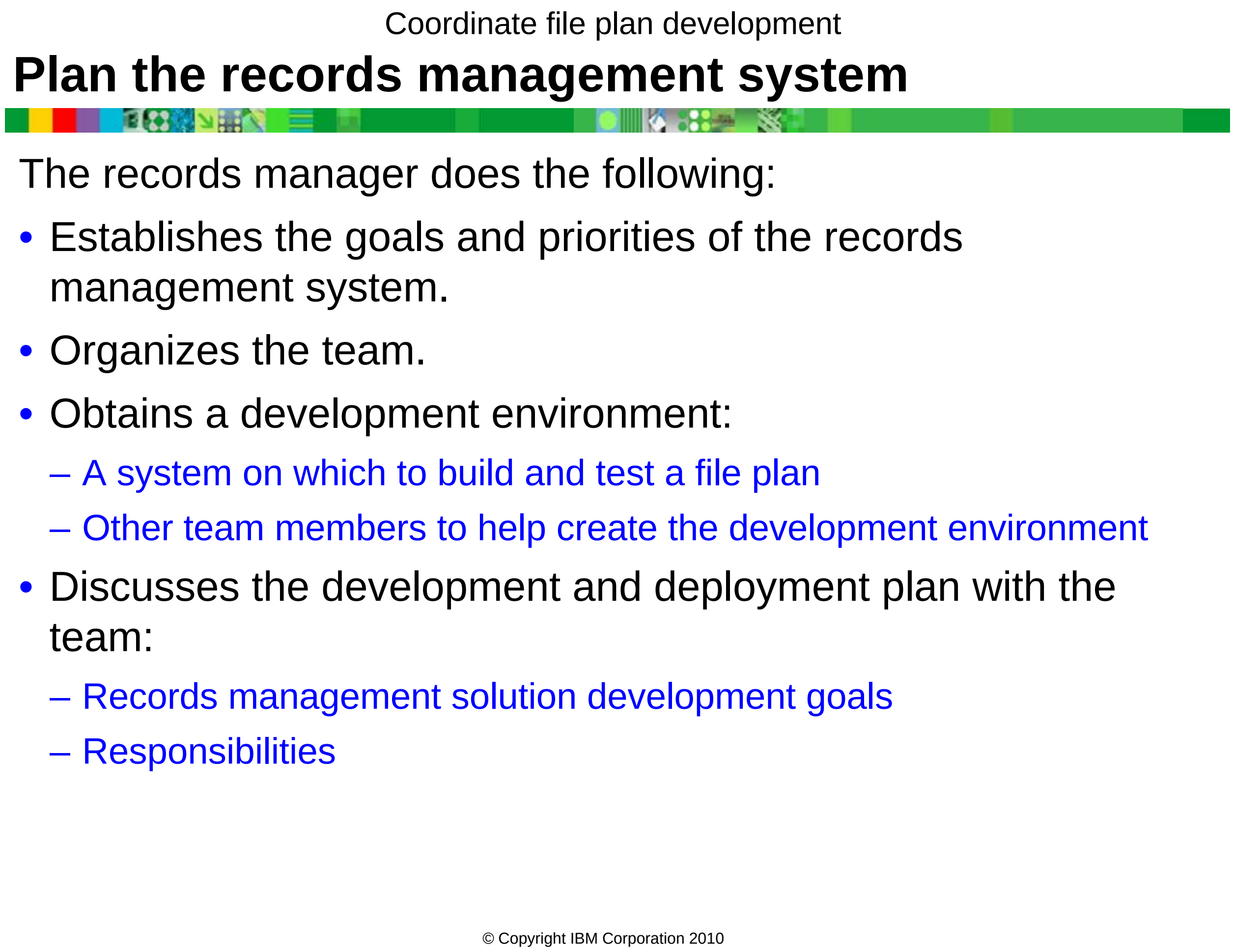
The Records administrator is the person who creates and configures new object stores. This person must create the file plan object store (FPOS) and the record-enabled document object store (RDOS or sometimes ROS). The records administrator also adds custom document and record and container classes. For example, if you need case records to have case record numbers, you must get the records administrator to create this case record number property and assign it to the case record class. If a property is needed to trigger disposition, but that property does not exist, you must ask the records administrator to create this property. For example, an employee record might have a Status property. When the Status of the employee changes from *employed* to *not employed*, then the retention period begins. The Employee record class and the Status property must both be created by a records administrator. The records administrator is also most likely to be the person who configures and runs the IBM Enterprise Records Sweep processes, such as Disposition Sweep and Hold Sweep. For an efficient file plan design, you need to decide how often the Disposition Sweep process must run, and you must communicate this decision and other configuration decisions to the records administrator.

If you are planning to use business processes to declare records, you also need to work with a business process designer and a Process Engine administrator in addition to the records administrator.

The Database administrator is responsible for maintaining the database. The database administrator creates the databases used by the Content Engine and by IBM Enterprise Records. IBM Enterprise Records is tightly integrated with the IBM FileNet P8 Content Engine. Performance of the Enterprise Records system is therefore dependent upon the performance of the Content Engine server, including the database. The database administrator keeps the database efficient by indexing and monitoring the performance of the database. The database administrator can run queries against the database to collect workload statistics, find bottlenecks, and make decisions about which properties to index in order to make the system run faster. The database administrator might also have recommendations for how to design a file plan or searches in order to prevent potential database performance problems.

The programmer is someone who can create code using the Content Engine and IBM Enterprise Records APIs to perform various automated functions. For example, you might want some custom code to automatically declare records. You might also need automation to create record categories or to create new folders or volumes automatically in order to ensure that records are properly filed. Automation can be created using the IBM P8 Java API, the Enterprise Records BDS API, or Content Engine custom events. BDS includes bulk declaration services and bulk disposal services.

Other roles include more specific talents, such as security administrators and IT administrators.



Coordinate file plan development

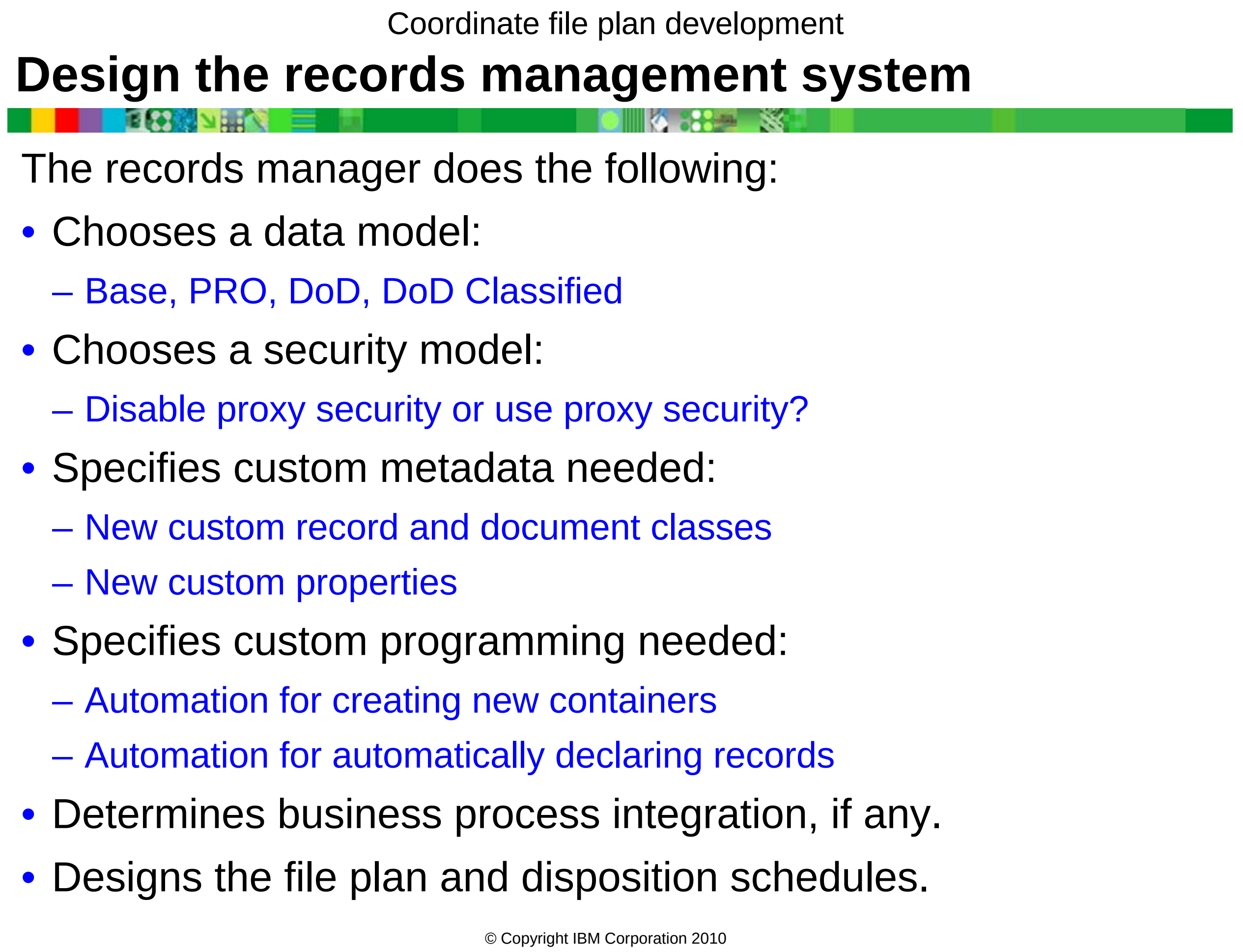
Plan the records management system

The records manager does the following:

- Establishes the goals and priorities of the records management system.
- Organizes the team.
- Obtains a development environment:
  - A system on which to build and test a file plan
  - Other team members to help create the development environment
- Discusses the development and deployment plan with the team:
  - Records management solution development goals
  - Responsibilities

Figure 1-6. Plan the records management system F1811.0

Notes:



Coordinate file plan development

Design the records management system

The records manager does the following:

- Chooses a data model:
  - – Base, PRO, DoD, DoD Classified
- Chooses a security model:
  - – Disable proxy security or use proxy security?
- Specifies custom metadata needed:
  - – New custom record and document classes
  - – New custom properties
- Specifies custom programming needed:
  - – Automation for creating new containers
  - – Automation for automatically declaring records
- Determines business process integration, if any.
- Designs the file plan and disposition schedules.

Figure 1-7. Design the records management system F1811.0

Notes:

You need to choose one of the available data models to use for the FPOS and communicate this choice to either the Records administrator or the installer—whoever is creating and configuring the FPOS.

Next, you must decide on a security model. The security administrator might need to configure users and groups, and then the records administrator must run a security script wizard to set up security for the records management groups when creating the FPOS. You and the records administrator must decide how the details of security need to work. Who needs to be able to have access to the records after they are declared? If you want record security to remain unchanged after declaration, you might need to have the administrator disable the security proxy.

You must also determine whether you need custom classes and properties. In most instances, you are going to use custom classes and properties when you build the system. Very often you use custom properties for triggering cutoff. You need to work out the custom property details with the records administrator before the creation of the FPOS and RDOS.

If your plan requires custom programming, you need to start planning it. If you have a custom program to automatically create containers for your records, for example, you need to determine how those containers are going to be added, closed, and disposed of.

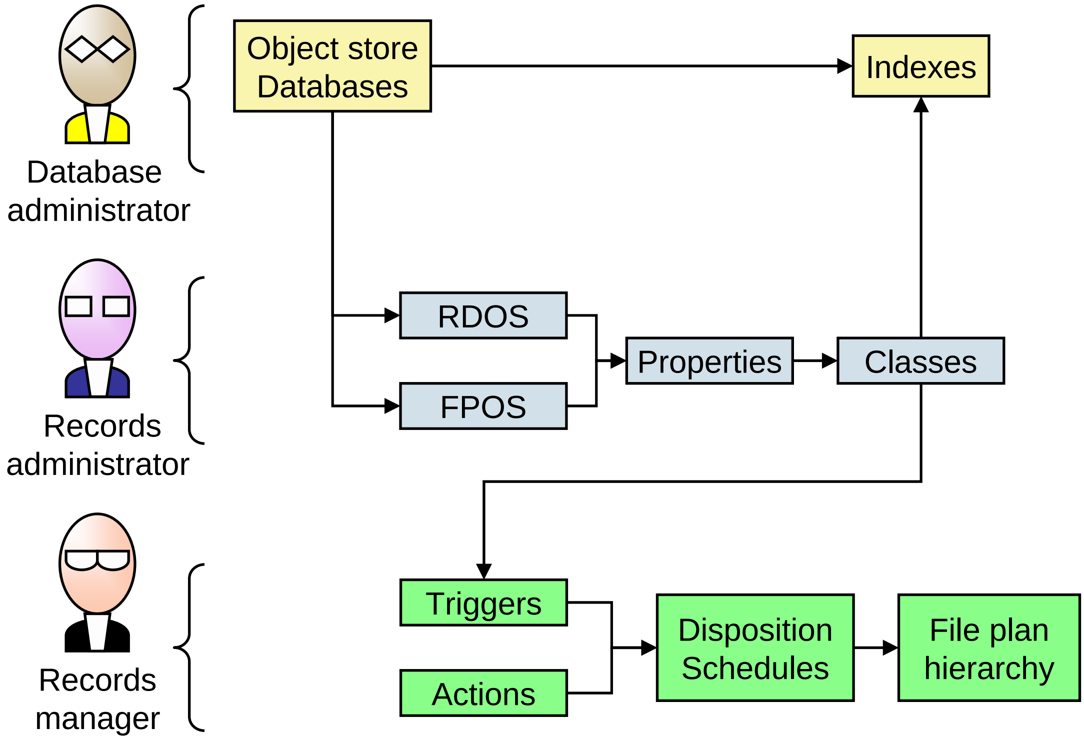
If you are planning to use workflows from the IBM FileNet P8 Process Engine to declare records, you need to discuss how this needs to work with a process designer. The process designer must know about the custom metadata that you are using as well, so that properties from the document can be mapped within the workflow environment.

This course focuses on file plan design. You might be tempted to jump in and start building a file plan without a design, and as long as you are working on a development system, there are no serious consequences for doing this. In fact, you might want to try out different ideas on a development system before you decide how to design your file plan. However, before you create the file plan that you intend to use in production, you need to be aware of how your design affects performance. Building a file plan without paying serious attention to its design is likely to lead to unexpected difficulties.

There is no single method that everyone uses for file plan design. You might use a text outline. You might use a diagram, or you might use a table. The format that you choose for the file plan design is not as important as ensuring that the design clearly shows the necessary features. The design needs to specify where the records are filed, where disposition schedules are applied and inherited, and what the aggregation level and triggers are. In this course, file plan designs are shown using a diagram format.

Coordinate file plan development

Build the system



Database administrator

Records administrator

Records manager

Object store Databases

Indexes

RDOS

FPOS

Properties

Classes

Triggers

Actions

Disposition Schedules

File plan hierarchy

Figure 1-8. Build the system F1811.0

Notes:

The diagram shows the order in which records management assets need to be created and the role that usually creates them. Although object dependencies are shown, exporting and importing are not. In most cases, the records manager builds the file plan on a development system, which is later exported to the production system.

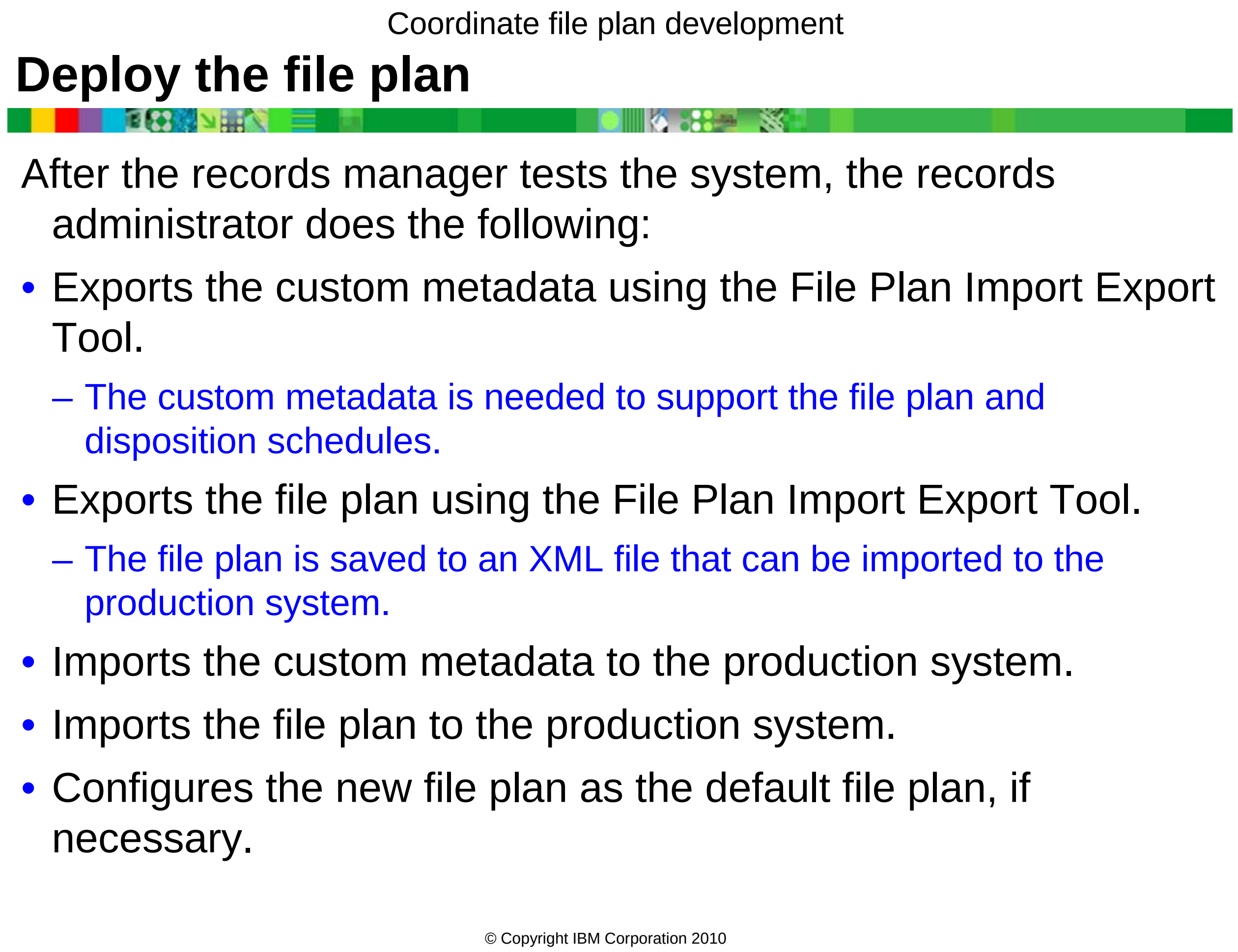
The database administrator must create the object store databases before the object stores can be created. After the properties and classes are created, the database administrator can create indexes after the custom properties and classes are created in order to keep the system efficient. The database administrator can run tests to determine which properties are the best candidates for indexing.

The records administrator then creates the RDOS and FPOS. In some production environments, the object store that is to become the RDOS might already exist with documents on it. In order to use this object store, the records administrator must first enable the object store for record declaration, and then enable the document classes for declaration. If there are existing document instances, they are not affected by the change in the metadata. So, in order to declare already existing documents, the records administrator must perform a bulk update to these documents. After creating the FPOS and the RDOS, the records administrator can create the custom properties and classes.

In many cases, event triggers are based on custom properties. For example, you might need a custom date property on a custom folder class to trigger cutoff. You might use a Binary (True or False) property as a switch that you can use to manually trigger cutoff. If you manage case folders, the case might have an open state and a closed state. You can represent the state with a custom property. When the case is closed, you can change the value of this property in order to trigger cutoff. In these cases, the properties must exist before you can create the event trigger. If your event triggers are not dependent on the custom classes and properties created by the records administrator, you can create them concurrently.

As the records manager, you must create both triggers and disposition actions before you can create the disposition schedule. In many cases, the disposition actions already exist because they are usually created during Enterprise Records installation, so you need to verify that they exist before you create the disposition schedules.

You do not need to create the disposition schedules before you create the file plan hierarchy, but it is a convenient time to do it. When you create a child container, it automatically inherits the disposition schedule of its parent container. Therefore, if you create the parent container and apply the disposition schedule to it before you create the child containers, you do not need to manually set the disposition schedule on the child containers.



Coordinate file plan development

Deploy the file plan

After the records manager tests the system, the records administrator does the following:

- Exports the custom metadata using the File Plan Import Export Tool.
  - The custom metadata is needed to support the file plan and disposition schedules.
- Exports the file plan using the File Plan Import Export Tool.
  - The file plan is saved to an XML file that can be imported to the production system.
- Imports the custom metadata to the production system.
- Imports the file plan to the production system.
- Configures the new file plan as the default file plan, if necessary.

Figure 1-9. Deploy the file plan F1811.0

Notes:

Coordinate file plan development

File plan development and deployment

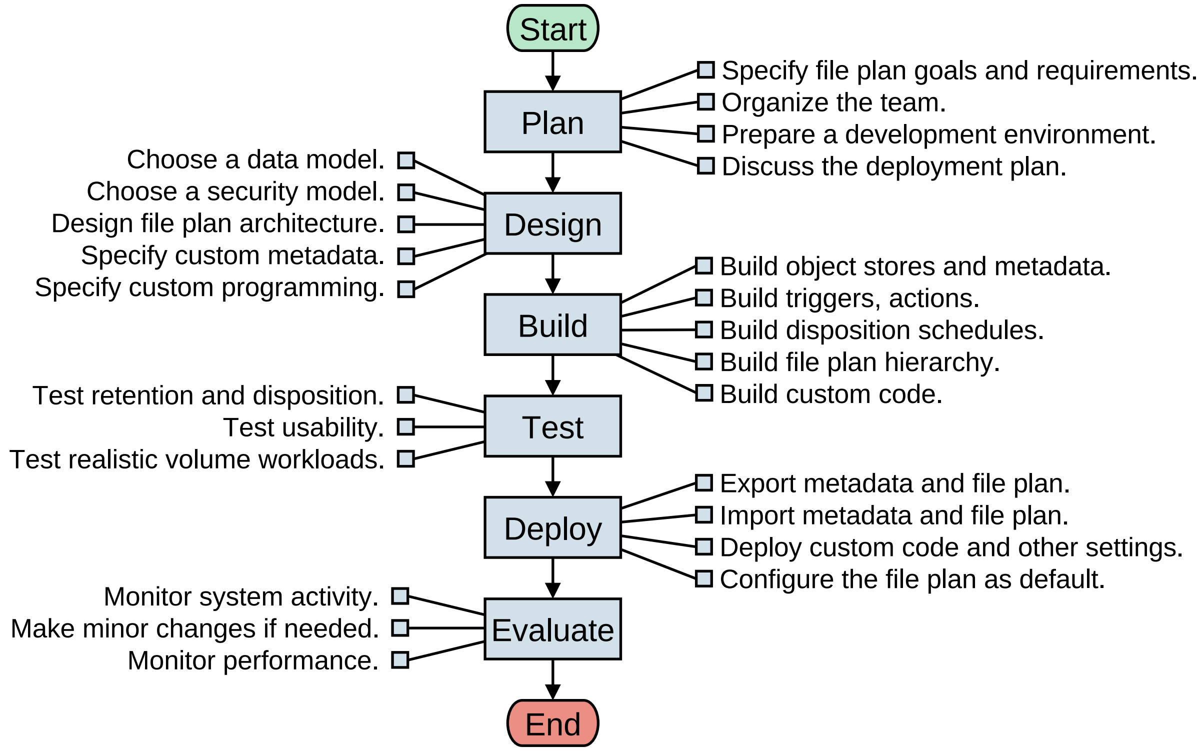

Start
Plan
□ Specify file plan goals and requirements.
□ Organize the team.
□ Prepare a development environment.
□ Discuss the deployment plan.
Design
Choose a data model. □
Choose a security model. □
Design file plan architecture. □
Specify custom metadata. □
Specify custom programming. □
Build
□ Build object stores and metadata.
□ Build triggers, actions.
□ Build disposition schedules.
□ Build file plan hierarchy.
□ Build custom code.
Test
Test retention and disposition. □
Test usability. □
Test realistic volume workloads. □
Deploy
□ Export metadata and file plan.
□ Import metadata and file plan.
□ Deploy custom code and other settings.
□ Configure the file plan as default.
Evaluate
Monitor system activity. □
Make minor changes if needed. □
Monitor performance. □
End

Figure 1-10. File plan development and deployment F1811.0

Notes:

The diagram shows an overview of the file plan development and deployment process.

Plan

Specify file plan goals and requirements.

Organize the team.

Prepare a development environment.

Discuss the deployment plan.

Design

Choose a data model.

Choose a security model.

Design file plan architecture.

Specify custom metadata.

Specify custom programming.

Build

Build object stores and metadata.

Build triggers, actions.

Build disposition schedules.

Build file plan hierarchy.

Build custom code.

Test

Test retention and disposition.

Test usability.

Test realistic volume workloads.

Deploy

Configure security groups and roles.

Export metadata and file plan.

Import metadata and file plan.

Deploy custom code and other settings.

Web application configuration.

Evaluate

Monitor system activity.

Make minor changes if needed.

Monitor performance.

Coordinate file plan development

Activities

In your Student Exercises

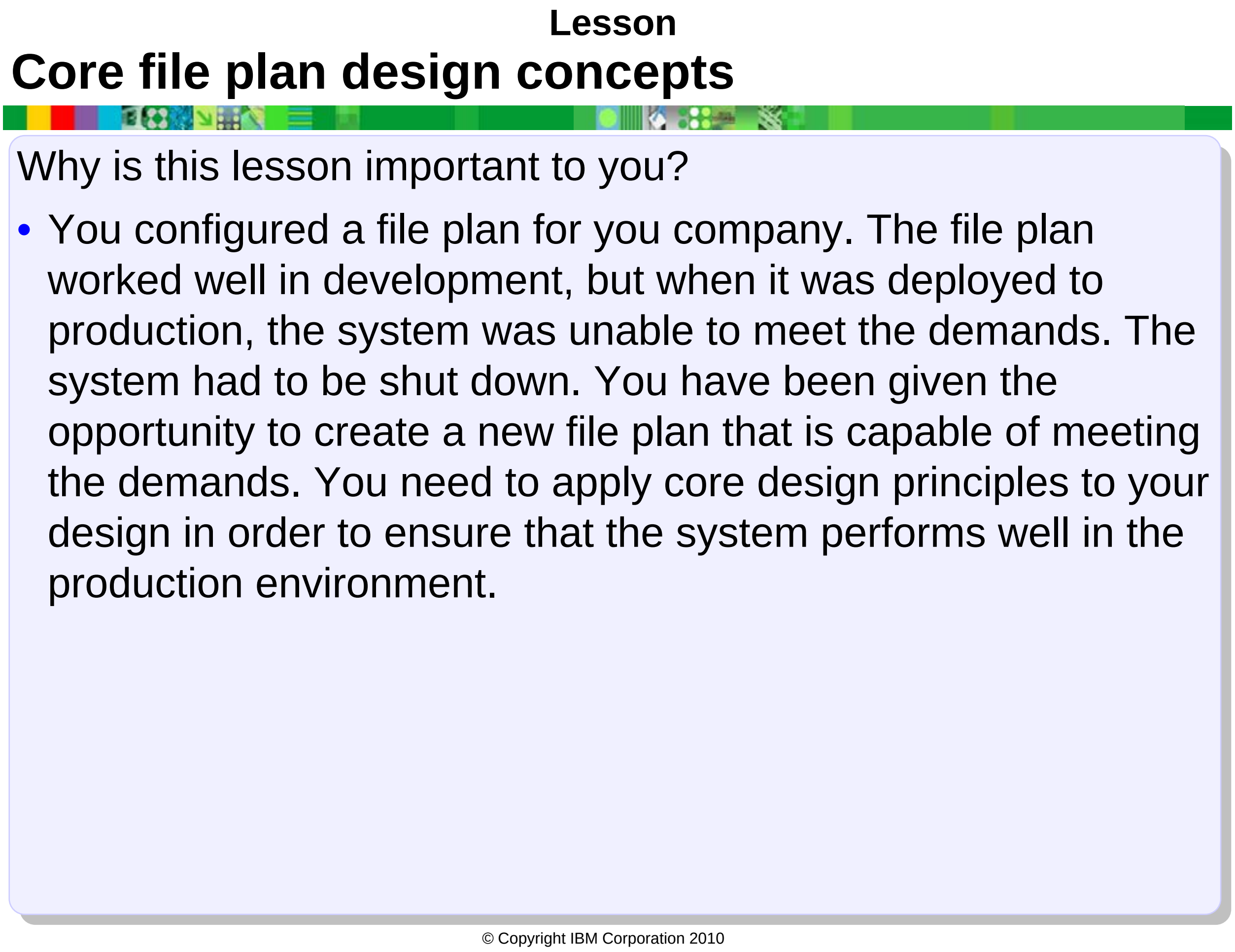
- Unit: Unit name
- Lesson: Lesson name
- Activities:
  - Coordinate file plan development.

Figure 1-11. Activities F1811.0

Notes:

Use your Student Exercises to perform the activities listed.

Lesson 1.2. Core file plan design concepts



Lesson

Core file plan design concepts

Why is this lesson important to you?

- You configured a file plan for you company. The file plan worked well in development, but when it was deployed to production, the system was unable to meet the demands. The system had to be shut down. You have been given the opportunity to create a new file plan that is capable of meeting the demands. You need to apply core design principles to your design in order to ensure that the system performs well in the production environment.

Figure 1-12. Core file plan design concepts F1811.0

Notes:

At low record volumes, the design of your file plan does not impact performance very much. You can design the file plan without needing to understand how your decisions affect performance. However, when the volume increases, the poorly designed file plans begin to fail under the greater strain imposed by poor designs. However, by this time, it is usually too late to redesign the file plan without considerable cost. By learning some simple core design concepts, you prepare yourself to create a file plan that can handle large volumes of records efficiently and effectively.

Core file plan design concepts

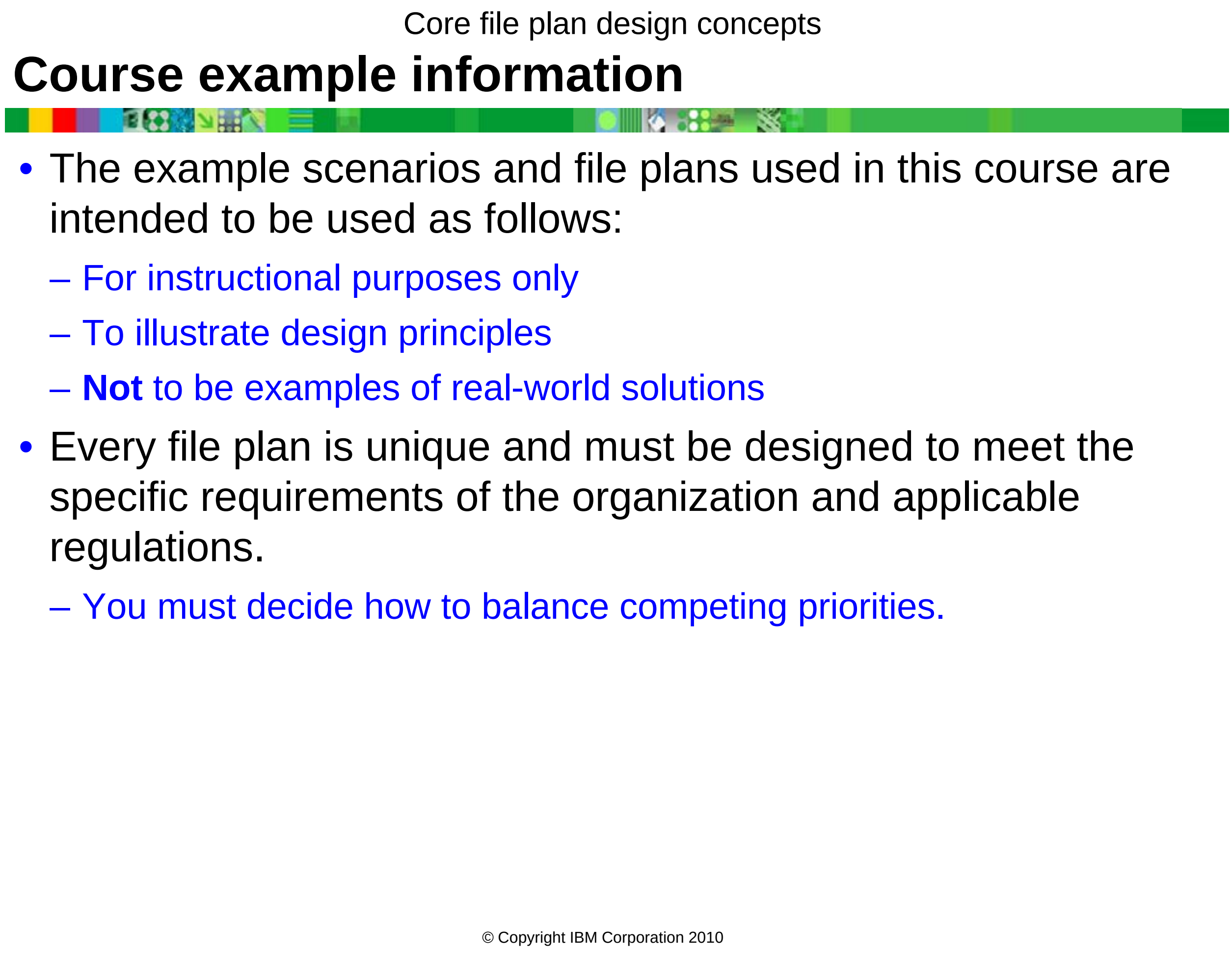
Activities that you need to complete

- Examine core file plan design concepts.

Figure 1-13. Activities that you need to complete F1811.0

Notes:

These are the activities that you need to complete.



Core file plan design concepts

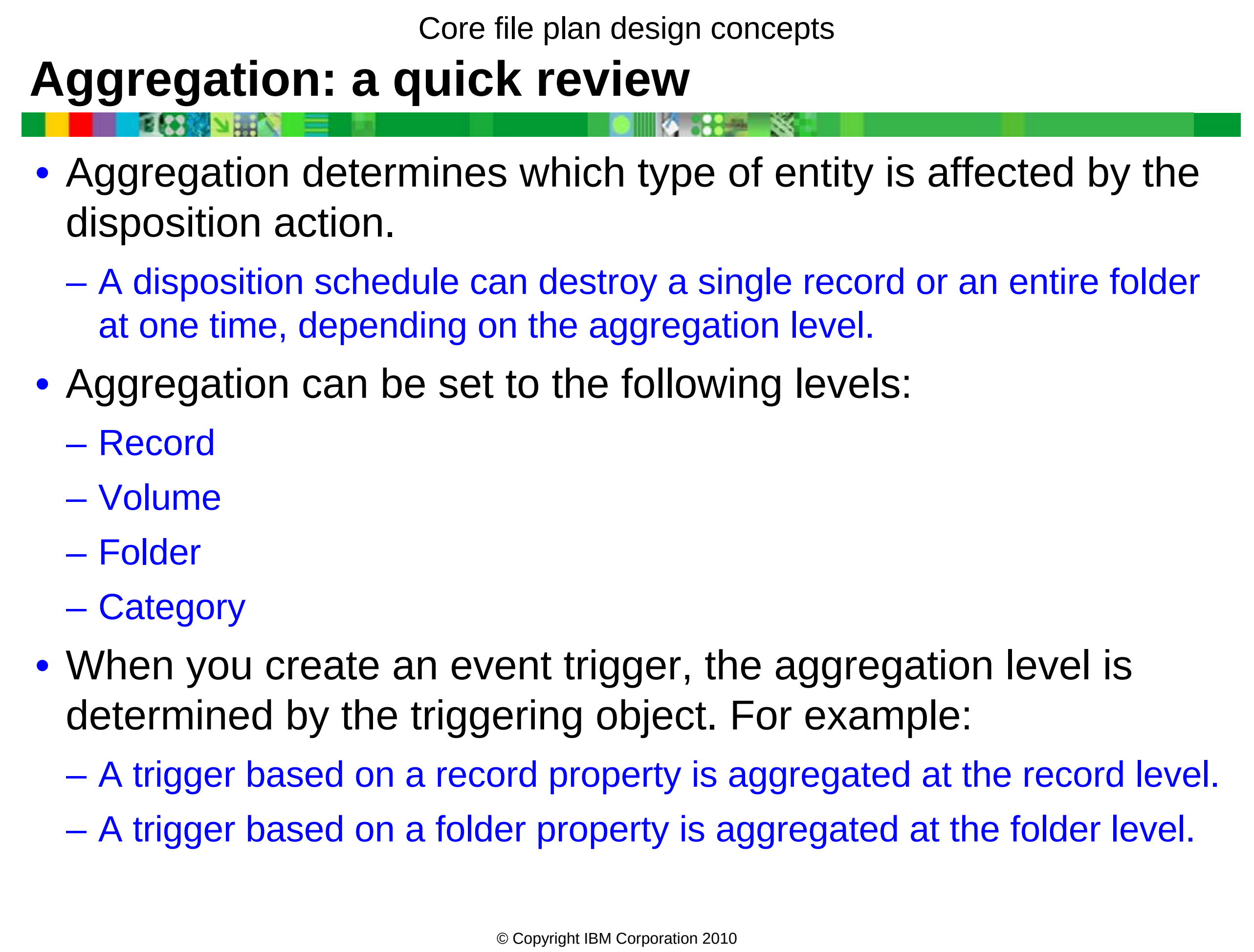
Course example information

- The example scenarios and file plans used in this course are intended to be used as follows:
  - – For instructional purposes only
  - – To illustrate design principles
  - – **Not** to be examples of real-world solutions
- Every file plan is unique and must be designed to meet the specific requirements of the organization and applicable regulations.
  - – You must decide how to balance competing priorities.

Figure 1-14. Course example information F1811.0

Notes:

The guidelines in this course are meant to provide insight into how your decisions can affect file plan performance. You must determine which concepts apply to your own record management requirements. Every records management system has different requirements. There are tradeoffs between fidelity and practicality. Fidelity in this case means faithfulness to a perfect ideal. For example, you might want records to be disposed of the instant that they reach the end of retention, but to achieve this goal, you need to run Disposition Sweep manually at the instant that the retention ends for every single record. Although this implementation is highly faithful to the idea of instant disposal, this scenario is so impractical that it defeats the purpose of automating records management, and it is impossible to achieve on a large scale. You must find the optimal balance where fidelity and practicality are within acceptable limits. Because you are the records manager, these decisions are your responsibility.



Core file plan design concepts

Aggregation: a quick review

- Aggregation determines which type of entity is affected by the disposition action.
  - A disposition schedule can destroy a single record or an entire folder at one time, depending on the aggregation level.
- Aggregation can be set to the following levels:
  - Record
  - Volume
  - Folder
  - Category
- When you create an event trigger, the aggregation level is determined by the triggering object. For example:
  - A trigger based on a record property is aggregated at the record level.
  - A trigger based on a folder property is aggregated at the folder level.

Figure 1-15. Aggregation: a quick review F1811.0

Notes:

Help path

- Search for "manage_events.htm"

When aggregation is at the container level, the action affects the container and all of the entities within that container.

Core file plan design concepts

Case study: 5 years of record-level aggregation

- In this example, records must be destroyed after 5 years.
- Disposition Sweep with Auto Destroy runs every quarter.
  - Although Disposition Sweep has nothing to destroy for the first 5 years, it must update disposition properties each quarter.
- After Q1 of the sixth year, Disposition Sweep checks *5.25 years of records* in order to destroy the records for a *single quarter*.

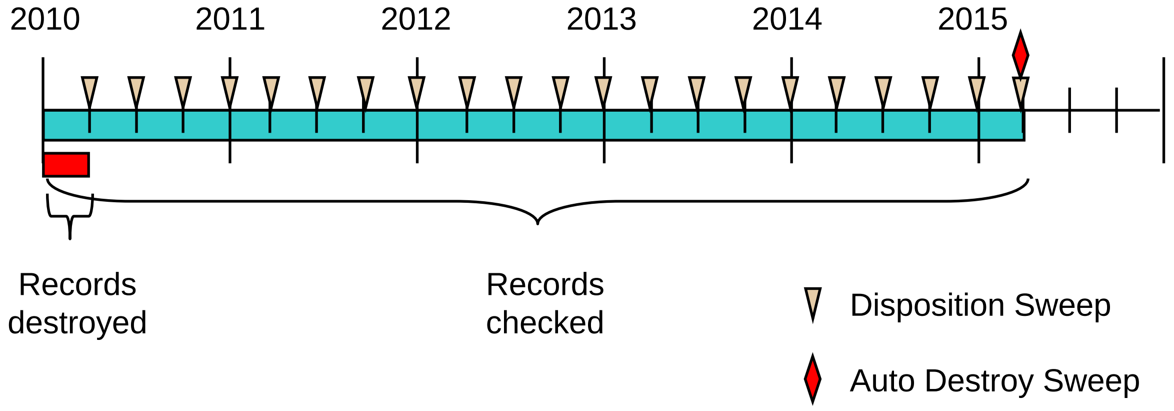


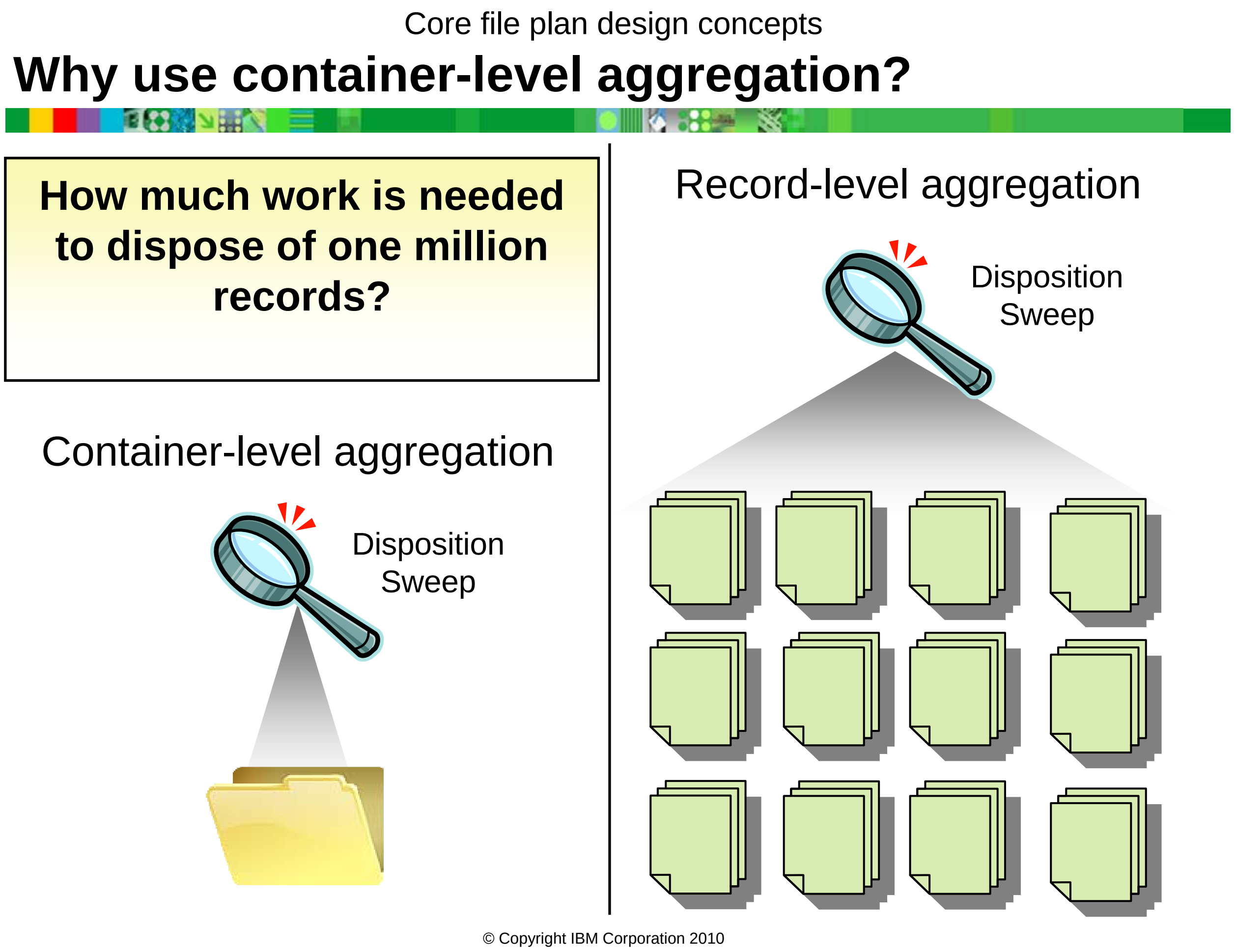
Figure 1-16. Case study: 5 years of record-level aggregation F1811.0

Notes:

The diagram shows how Disposition Sweep must sweep all of the records that have accumulated in 5.25 years in order to destroy the records from the first quarter.

Disposition Sweep runs a short time before Auto Destroy. The Auto Destroy sweep process is a Disposition Sweep process with the autodelete flag activated. For simplicity, it is called Auto Destroy Sweep in this course.

In this example, if you start keeping records at the beginning of 2010 and the records have a 5-year retention schedule, then no destruction occurs until after 2015. After the first quarter of 2015, the Auto Destroy Sweep begins to find records that are ready to be destroyed. However, Disposition Sweep has to check all of the records from the entire span of 5 years in order to destroy the records of the first quarter of 2010.



Core file plan design concepts

Why use container-level aggregation?

How much work is needed to dispose of one million records?

Container-level aggregation

Disposition Sweep

Record-level aggregation

Disposition Sweep

Figure 1-17. Why use container-level aggregation? F1811.0

Notes:

The diagram shows that for container-level aggregation, Disposition Sweep must process only a single entity. For records-level aggregation, however, Disposition Sweep must process every single record. For one million records, the difference in time and processing is substantial.

Core file plan design concepts

Record-level aggregated sweep process

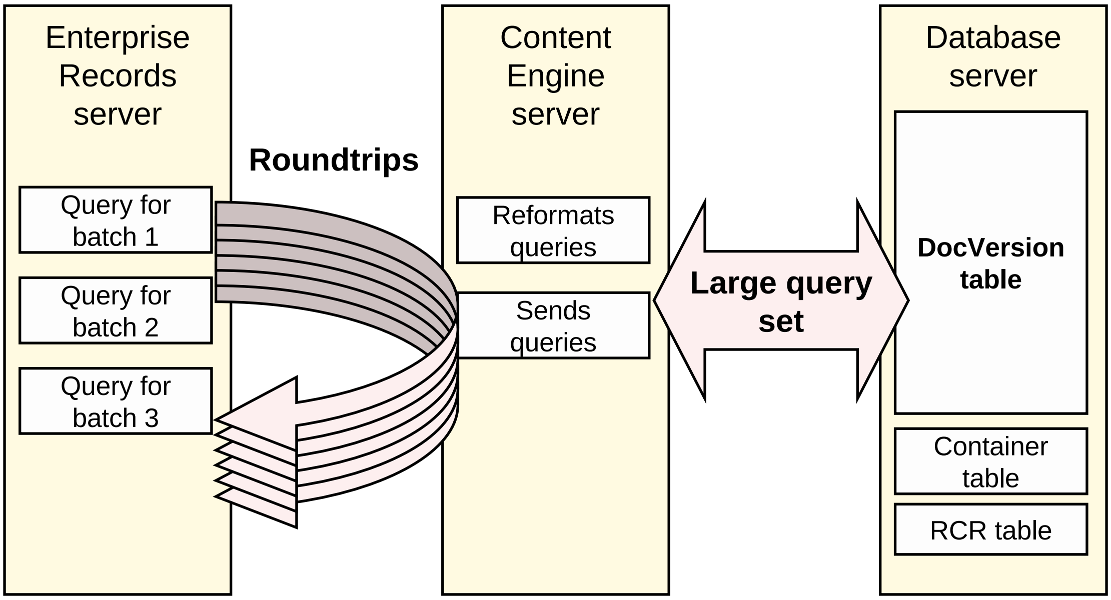


Figure 1-18. Record-level aggregated sweep process F1811.0

Notes:

The diagram shows the roundtrips between the Enterprise Records server and the Content Engine server. It also shows the query sent to the Database server. For record-level aggregation, many roundtrips are made and a large query is sent to the DocVersion table, which causes a large query return to be sent back.

Imagine a scenario in which you must destroy one million records.

From an Enterprise Records perspective, the main performance factor is the number of roundtrips between the Enterprise Records Server and the Content Engine server. The size of the database table, as measured by the number of rows in that table, plays a large part of how long a query takes to execute.

For the purposes of Disposition Sweep, the Content Engine server can query any of three database tables, including the following:

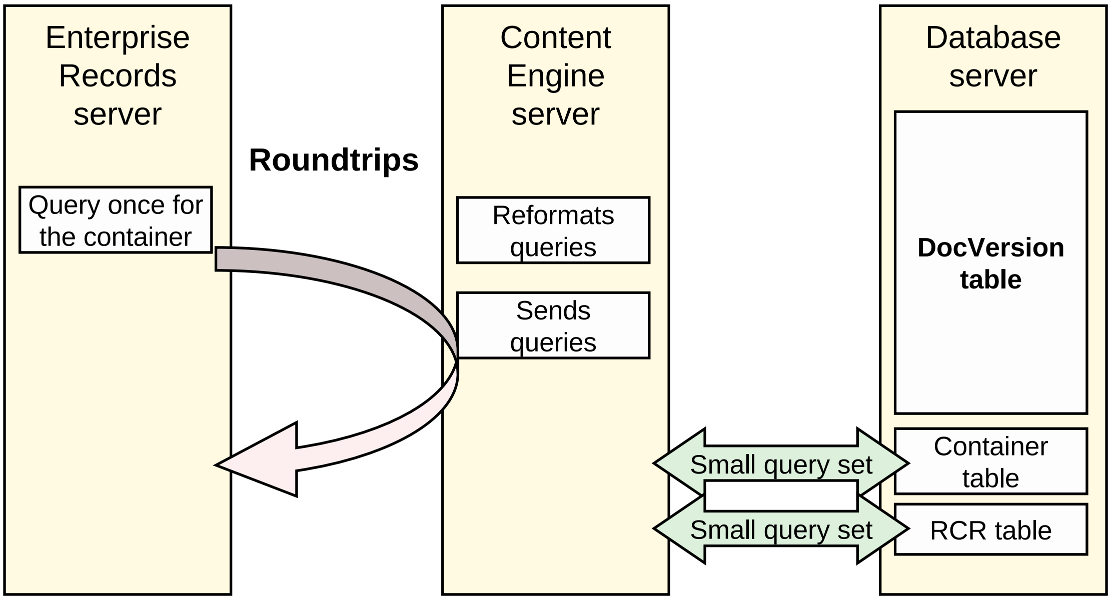
- The DocVersion table, which contains the records and is by far the largest of the three.
- The Container tables, which hold the folders.
- The RCR table, which holds the links between each record and its container.

Enterprise Records does not query the database directly, but sends a query to the Content Engine server, which reformats the query and possibly sends additional queries to the database. Therefore, for best performance, you want to use queries that require the fewest roundtrips and that query the smallest database table.

To destroy one million records, a Disposition Sweep process is run before the Auto Destroy process in order to ensure that all of the disposition-related data are updated first. During the Disposition Sweep, if you use record-level aggregation, the Content Engine must query against the huge DocVersion table and touch every single record. Additionally, because there are so many records, Disposition Sweep cannot retrieve all of them in a single query, which means it must use a paged query (which is slower) and make repeated queries (roundtrips) to the Content Engine in order to process batches of records until it gets through one million.

Core file plan design concepts

Container-level aggregated sweep process



Enterprise Records server

Query once for the container

Roundtrips

Content Engine server

Reformats queries

Sends queries

Small query set

Small query set

Database server

DocVersion table

Container table

RCR table

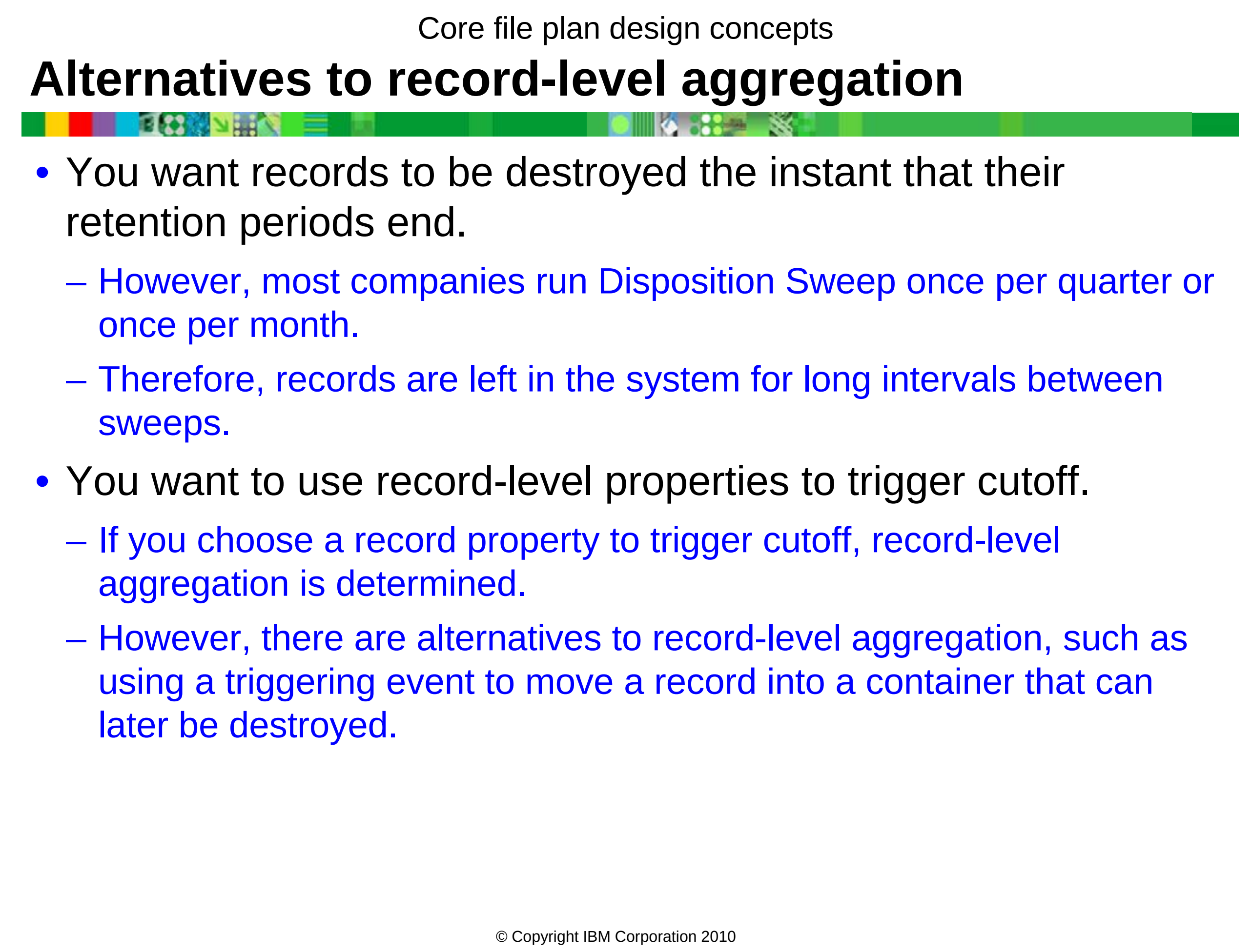
Figure 1-19. Container-level aggregated sweep process F1811.0

Notes:

The diagram shows the roundtrips between the Enterprise Records server and the Content engine server. It also shows the query sent to the Database server. For container-level aggregation, much less roundtrip activity occurs than with record-level aggregation, and small queries are sent to the Container table and the RCR table.

If you use container-level aggregation instead of record-level aggregation, the Content Engine queries the Container table, which is much smaller than the DocVersion table. Then, the Content Engine only needs to process, update, and validate against a single container instead of one million records.

After Disposition Sweep runs, you can run the Auto Destroy sweep. During the Auto Destroy sweep process, the Content Engine must touch each record in order to destroy it, but the query sent to the Content Engine is optimized to avoid a query against the DocVersion table. Because the Record ID is the only information that is required to destroy the record, the query is run against the RCR table to retrieve the Record IDs. The Auto Destroy process queries the RCR table because it is much smaller than the DocVersion table.



Core file plan design concepts

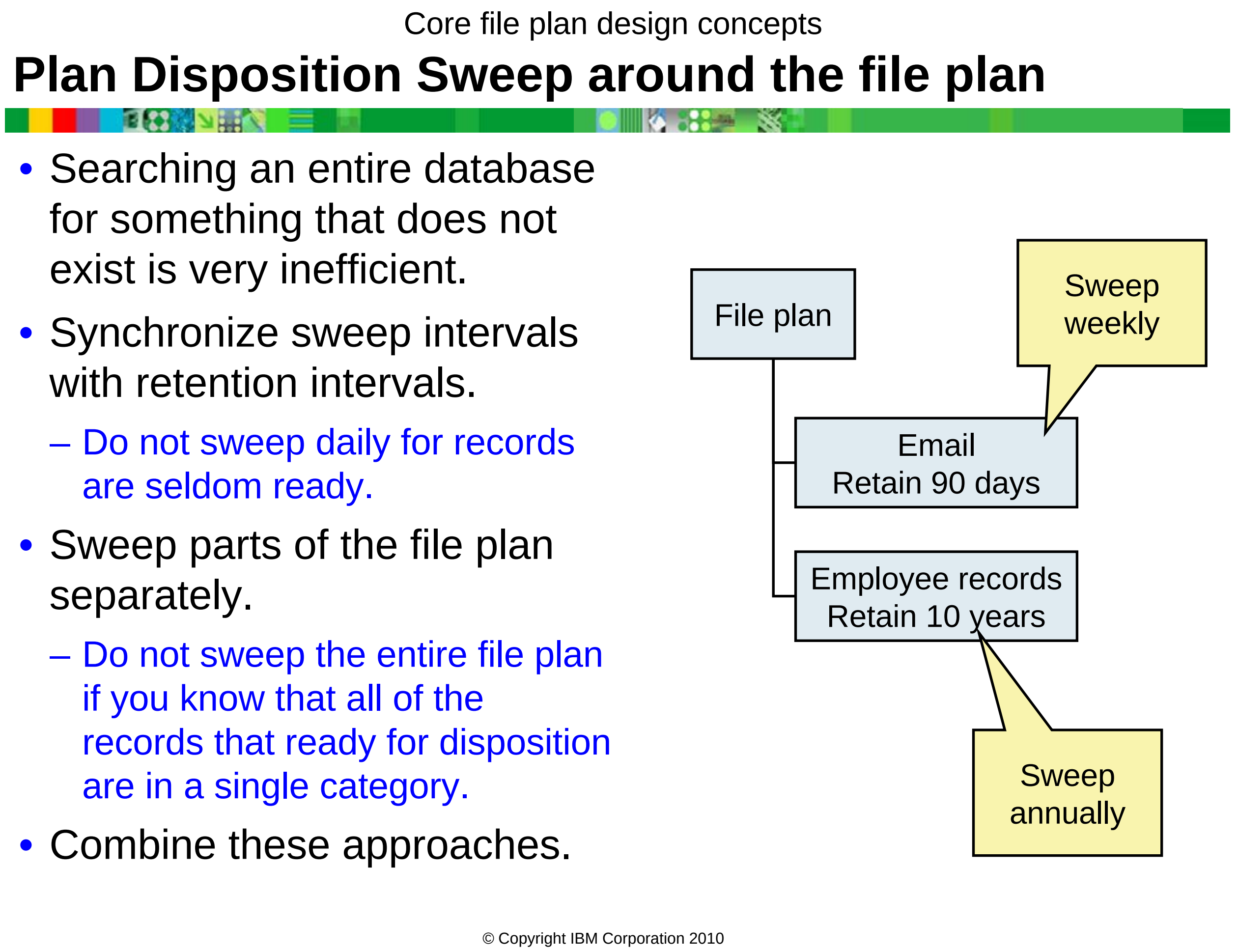
Alternatives to record-level aggregation

- You want records to be destroyed the instant that their retention periods end.
  - – However, most companies run Disposition Sweep once per quarter or once per month.
  - – Therefore, records are left in the system for long intervals between sweeps.
- You want to use record-level properties to trigger cutoff.
  - – If you choose a record property to trigger cutoff, record-level aggregation is determined.
  - – However, there are alternatives to record-level aggregation, such as using a triggering event to move a record into a container that can later be destroyed.

Figure 1-20. Alternatives to record-level aggregation F1811.0

Notes:

Record-level aggregation is an option that you can choose, but it is not always the best approach. You need to be aware of practical alternatives before you commit to record-level aggregation so that you are not choosing an inefficient system based on incomplete information.



Core file plan design concepts

Plan Disposition Sweep around the file plan

- Searching an entire database for something that does not exist is very inefficient.
- Synchronize sweep intervals with retention intervals.
  - – Do not sweep daily for records are seldom ready.
- Sweep parts of the file plan separately.
  - – Do not sweep the entire file plan if you know that all of the records that ready for disposition are in a single category.
- Combine these approaches.

File plan

Email
Retain 90 days

Employee records
Retain 10 years

Sweep weekly

Sweep annually

Figure 1-21. Plan Disposition Sweep around the file plan F1811.0

Notes:

The diagram shows a file plan with two categories. In one category, email is retained for 90 days. In the other category, employee records are retained for 10 years. One approach to effectively managing these records is to make sure that sweep processes are configured to sweep these areas separately. Additionally, the email category is swept on a weekly basis while the employee records category is swept annually.

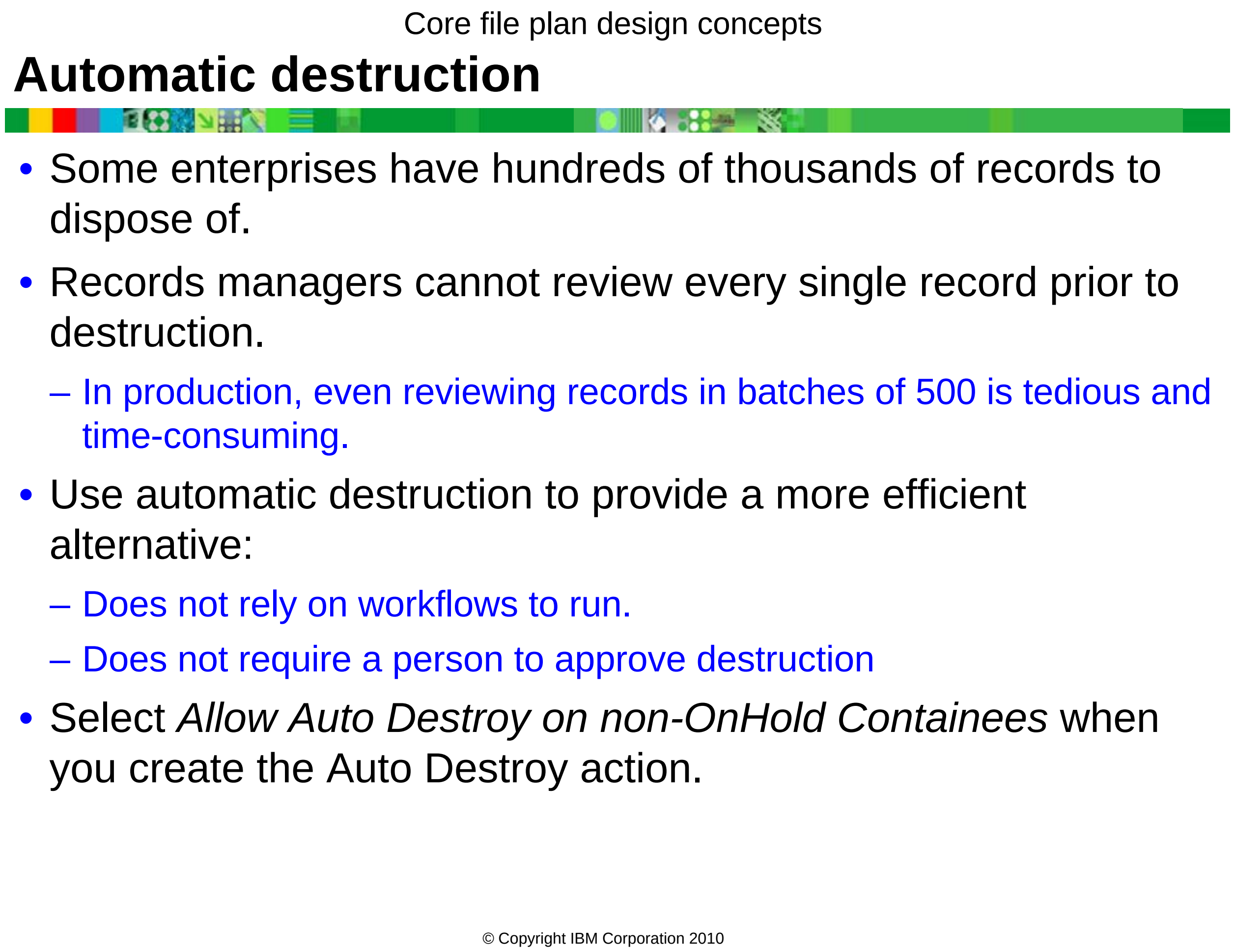
One of the most inefficient processes that a system can perform is to search an entire database for something that does not exist. Imagine having to check a list of millions of items until you reach the end of the list before you can determine that the object you are looking for is not in the set. Disposition Sweep must check the entire database whenever it is sweeping the entire file plan in order to determine whether or not any records are ready for disposition. If no records are ready for disposition when Disposition Sweep runs, then the sweep process takes a very long time to search the database for records that are not ready for disposition.

You can make sweep processes more efficient by synchronizing the sweep intervals with the retention intervals. For example, you might sweep more often if record retention intervals are fairly short and sweep less often if record retention intervals are longer. Sometimes, a single day beyond retention is absolutely unacceptable, in which case you must accept the wasted processing as a consequence of the regulatory requirement. You must decide the point of tradeoff between fidelity and practicality based on your scenario.

You can also sweep parts of the file plan separately. If there is no requirement for you to sweep the entire file plan at the same time, you might consider sweeping different areas of the file plan at different times. For example, you might have a limited timeframe of six hours per week. Disposition Sweep cannot complete during this time, but if you sweep half of the file plan on alternating weekends, you can complete the sweep every two weeks. You might also sweep a specific container in the file plan if that container has all of the records of that type that are ready for disposition, such as a container for all of the 5-year retentions that are 5 years old.

Sweeping different areas of the file plan separately also allows you to maintain separate sweep schedules for different parts of the file plan, so that you can sweep some areas weekly and other areas monthly, quarterly, or annually, depending on retention needs.

The records administrator is usually the person who configures the sweep processes, so you need to collaborate with the records administrator to ensure that the sweep processes are configured to efficiently work with the file plan and disposition schedules.



Core file plan design concepts

Automatic destruction

- Some enterprises have hundreds of thousands of records to dispose of.
- Records managers cannot review every single record prior to destruction.
  - In production, even reviewing records in batches of 500 is tedious and time-consuming.
- Use automatic destruction to provide a more efficient alternative:
  - Does not rely on workflows to run.
  - Does not require a person to approve destruction
- Select *Allow Auto Destroy on non-OnHold Containees* when you create the Auto Destroy action.

Figure 1-22. Automatic destruction F1811.0

Notes:

After you configure a disposition schedule to use an Auto Destroy action, the automatic destruction is executed using Disposition Sweep. First, any entities that are controlled by a disposition schedule with an auto destroy phase must be processed by Disposition Sweep in the usual fashion to compute the disposition date values, allowing the entities to be ready for disposition at the appropriate time. An additional step is then required to execute the automatic destruction. This step is performed as a separate command line option for the Disposition Sweep tool. When this command is invoked, the system immediately and automatically destroys all entities having the Auto Destroy action that are ready for disposition and that are not on hold.

Auto Destroy on non-OnHold Containees

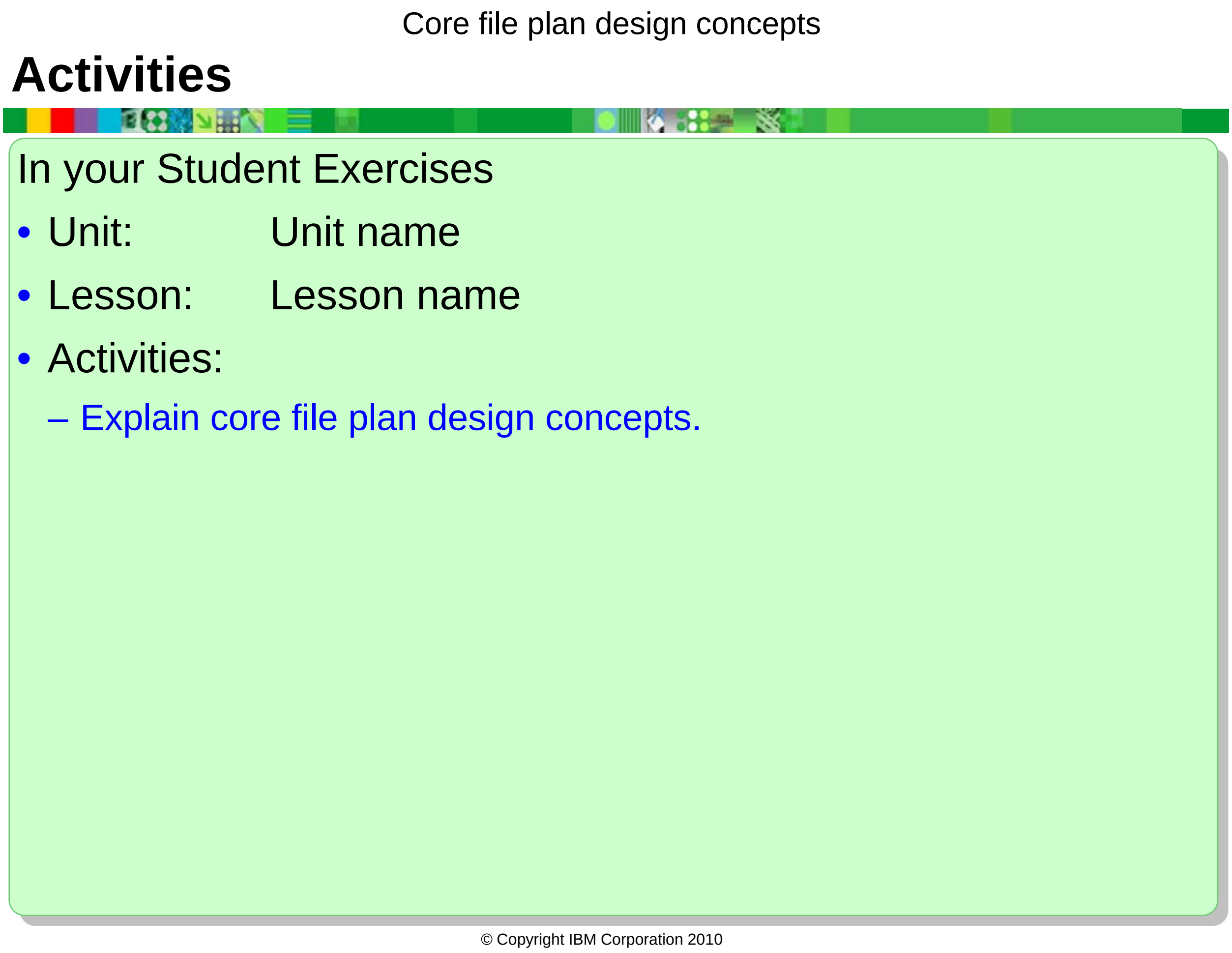
When you create the Auto Destroy action, you can choose whether or not to allow Auto Destroy on non-OnHold Containees.

If this option is **not** selected, then, when Auto Destroy runs and finds a record that **is** on hold, it prevents the entire container and all of its contents from being destroyed.

If this option **is** selected, the Auto Destroy action destroys records in a container that are not on hold, even if it finds some records that are on hold.

Selecting this option is usually more efficient. When the Auto Destroy sweep process runs without the *Allow Auto Destroy on non-OnHold Containees* option, it must sort the records before deleting them so that if a record is found that is on hold, it can roll back the entire process and restore the container to its original condition. All of the records in the container that might otherwise be destroyed must wait until the next time the Auto Delete sweep process runs again.

However, if the *Allow Auto Destroy on non-OnHold Containees* option is selected, the records are destroyed without first ordering them. If a record is found that is on hold, the previously destroyed records are not restored. The destroy process continues until only the records that were on hold remain in the container.



Core file plan design concepts

Activities

In your Student Exercises

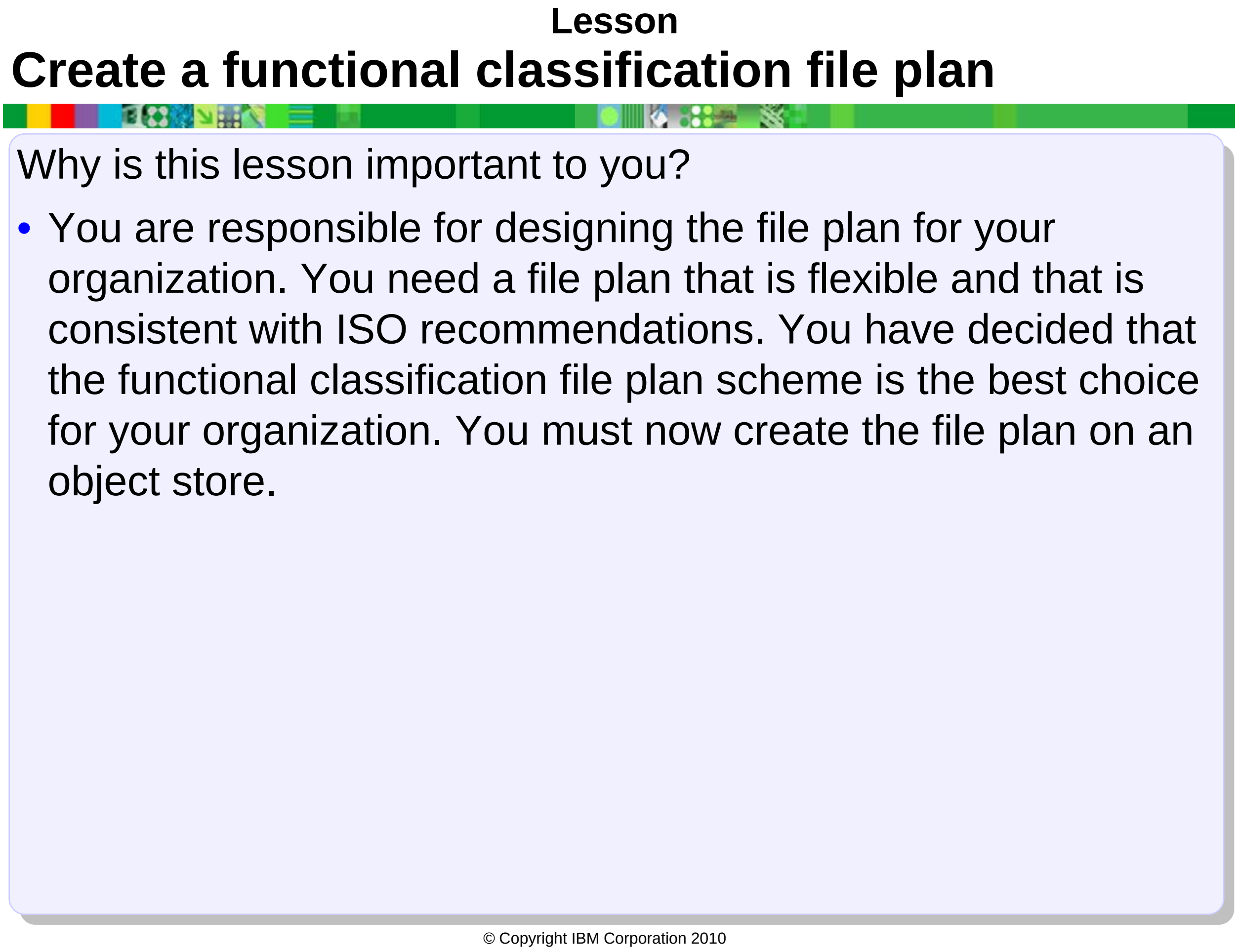
- Unit: Unit name
- Lesson: Lesson name
- Activities:
  – Explain core file plan design concepts.

Figure 1-23. Activities F1811.0

Notes:

Use your Student Exercises to perform the activities listed.

Lesson 1.3. Create a functional classification file plan



Lesson

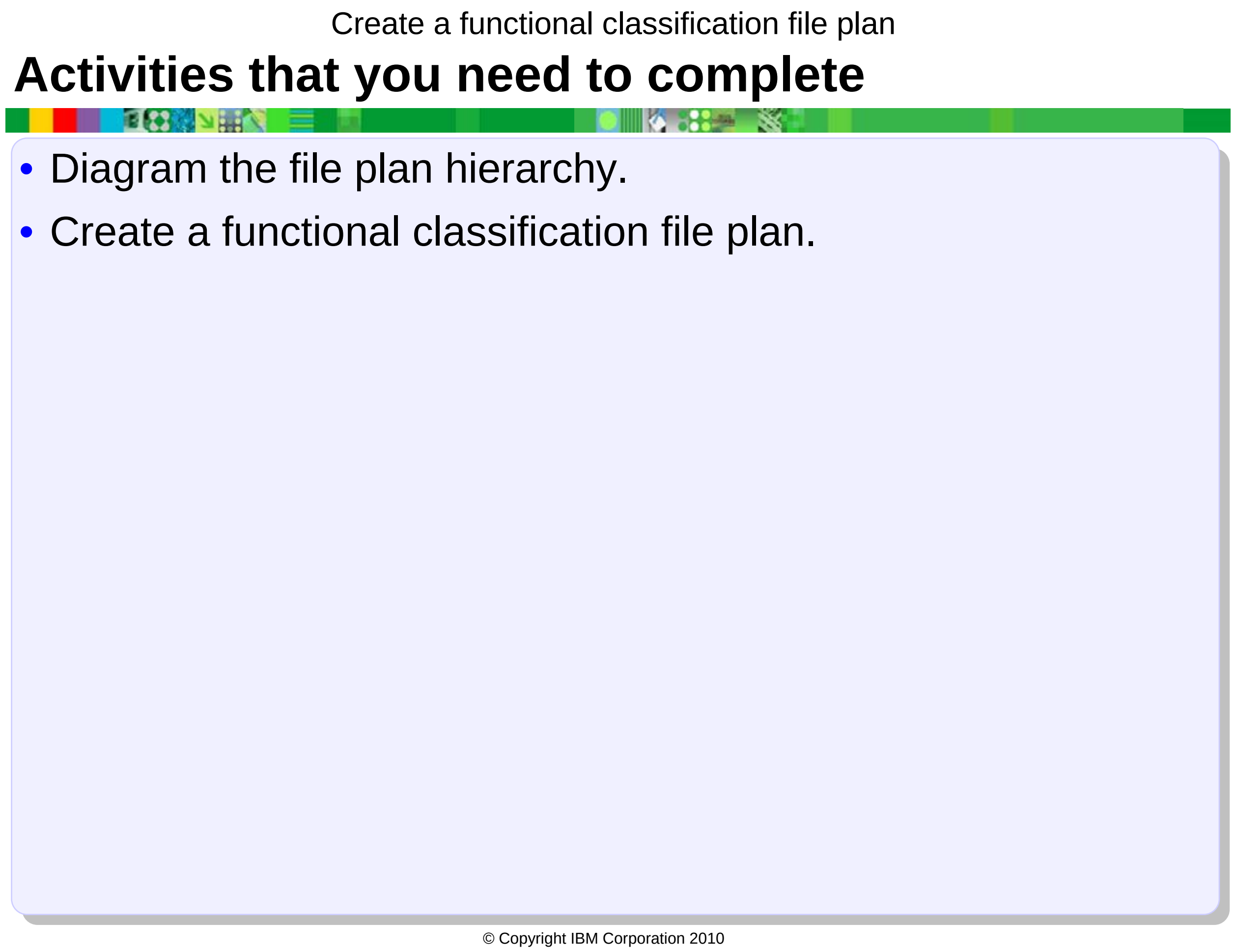
Create a functional classification file plan

Why is this lesson important to you?

- You are responsible for designing the file plan for your organization. You need a file plan that is flexible and that is consistent with ISO recommendations. You have decided that the functional classification file plan scheme is the best choice for your organization. You must now create the file plan on an object store.

Figure 1-24. Create a functional classification file plan F1811.0

Notes:



Create a functional classification file plan

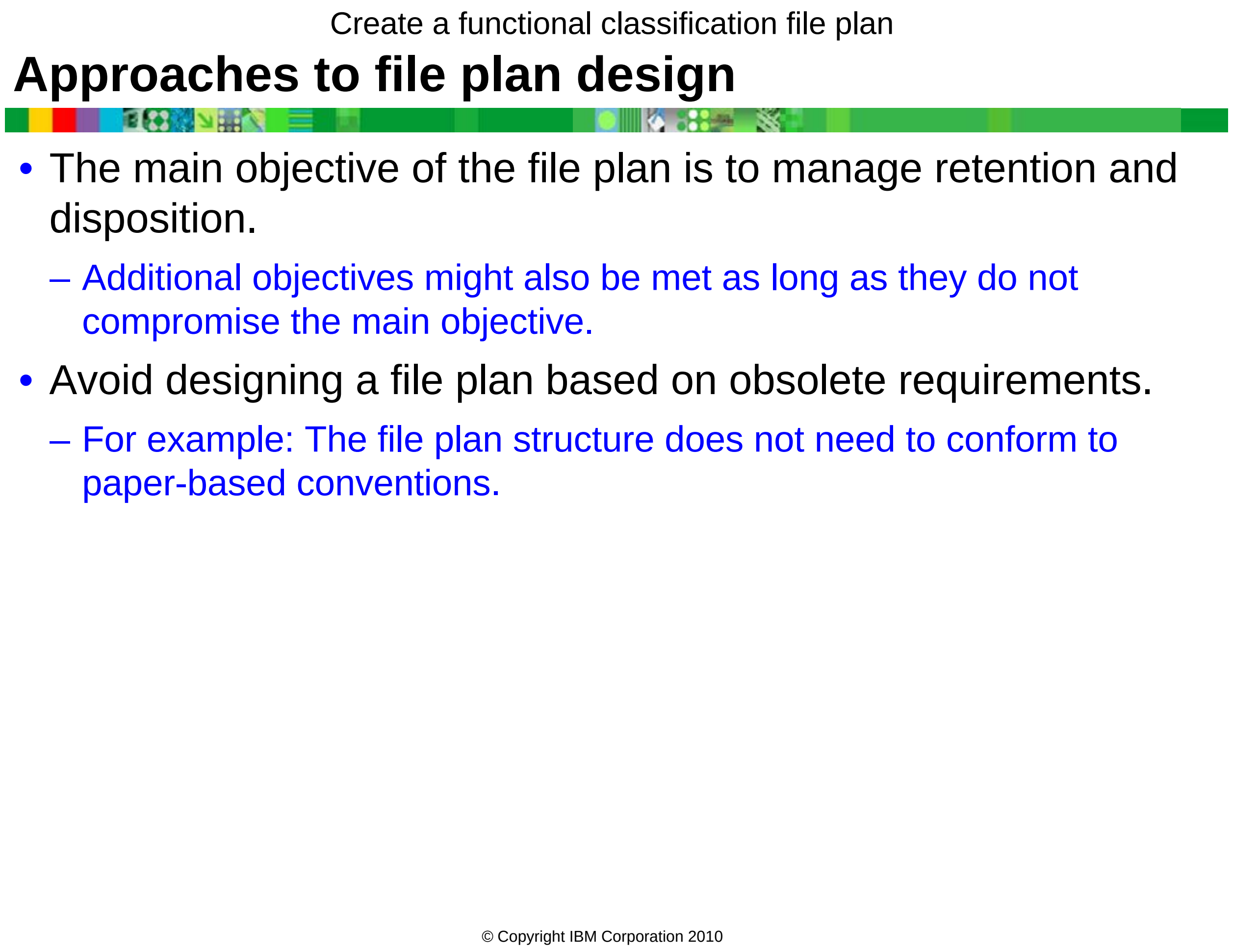
Activities that you need to complete

- Diagram the file plan hierarchy.
- Create a functional classification file plan.

Figure 1-25. Activities that you need to complete F1811.0

Notes:

These are the activities that you need to complete.



Create a functional classification file plan

Approaches to file plan design

- The main objective of the file plan is to manage retention and disposition.
  – Additional objectives might also be met as long as they do not compromise the main objective.
- Avoid designing a file plan based on obsolete requirements.
  – For example: The file plan structure does not need to conform to paper-based conventions.

Figure 1-26. Approaches to file plan design F1811.0

Notes:

Create a functional classification file plan

Functional classification scheme

- A functional classification scheme is recommended by the International Organization of Standardization (ISO).
- Records are classified according to their purpose.
- File plan structure:
  - Level 1: Functions
  - Level 2: Activities
  - Level 3: Transactions
- To create a functional classification scheme:
  - Identify the main functions of an organization, and then sort them into activities and transactions.
  - Create a category hierarchy that represents the functions, activities, and transactions of the organization.

Figure 1-27. Functional classification scheme F1811.0

Notes:

The functional classification scheme is a way of organizing the file plan using a hierarchical system by classifying records first by function, then by activity, and then by transactions. The International Organization of Standardization (ISO) advocates using this method. Records are classified according to why they exist, rather than their subject matter. This method of organization provides clarity in the relationship between an organizations practices and its records.

Functions

Functions represent the major responsibilities that are managed by the organization to fulfill its goals. Functions are high-level aggregates of the organization’s activities. Functions are normally not aligned with organizational structures, because they are more stable than administrative units, which are often consolidated or further divided during organizational restructures. Common functions can be and often are dispersed across the structural components of an organization.

Activities

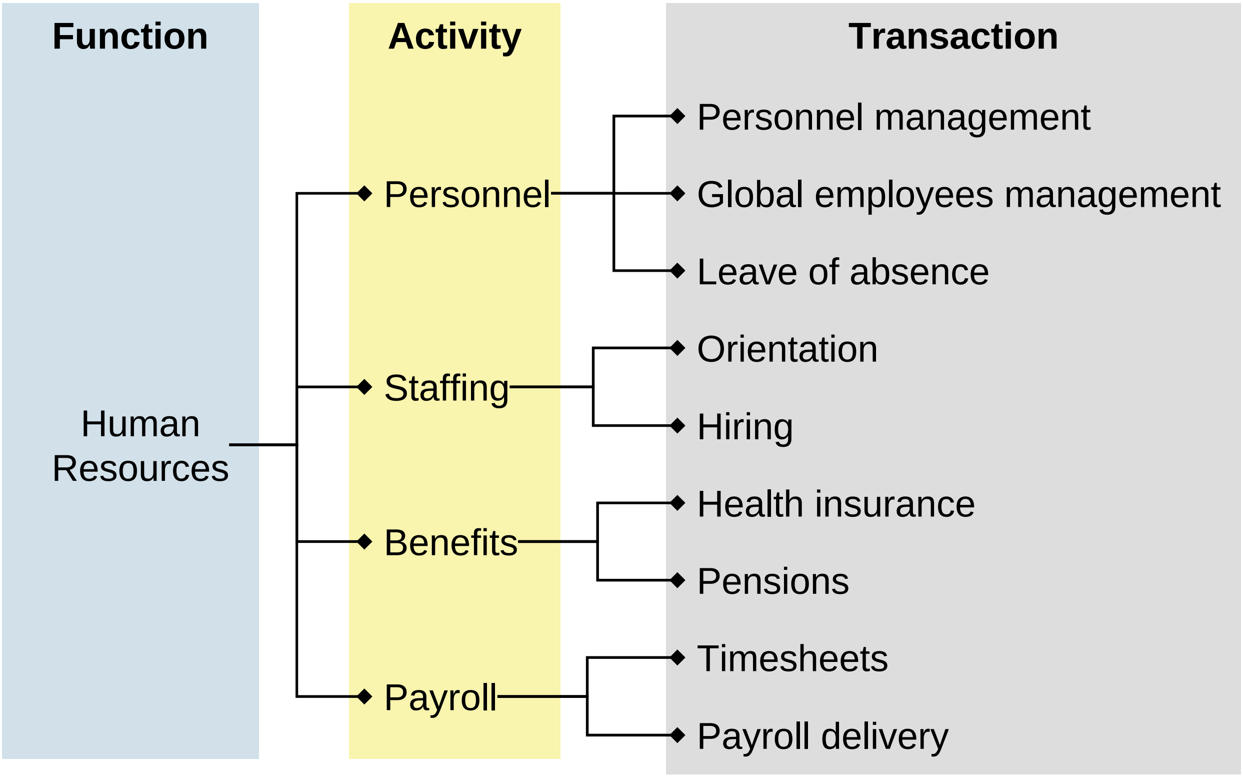
Activities represent the major tasks performed by the organization to accomplish each of its functions. Multiple activities can be associated with each function. An activity must be based on an interrelated grouping of transactions producing a particular outcome.

Transactions

Transactions represent the smallest units of business activity. Transactions help define the scope or boundaries of activities and provide the basis for identifying the records that are required to meet the business needs of the organization. The identification of transactions also helps in the formulation of the record description as part of the retention schedule of an organization.

Create a functional classification file plan

Example of a functional classification scheme



Function

Activity

Transaction

Human Resources

Personnel

Staffing

Benefits

Payroll

Personnel management

Global employees management

Leave of absence

Orientation

Hiring

Health insurance

Pensions

Timesheets

Payroll delivery

Figure 1-28. Example of a functional classification scheme F1811.0

Notes:

The diagram shows an example of a functional classification scheme. It shows a single function of an organization, Human Resources, and lists some of the main activities: Personnel, staffing, benefits, and payroll. This list is not complete, but is a representative sample. Each of these activities is further divided into transactions:

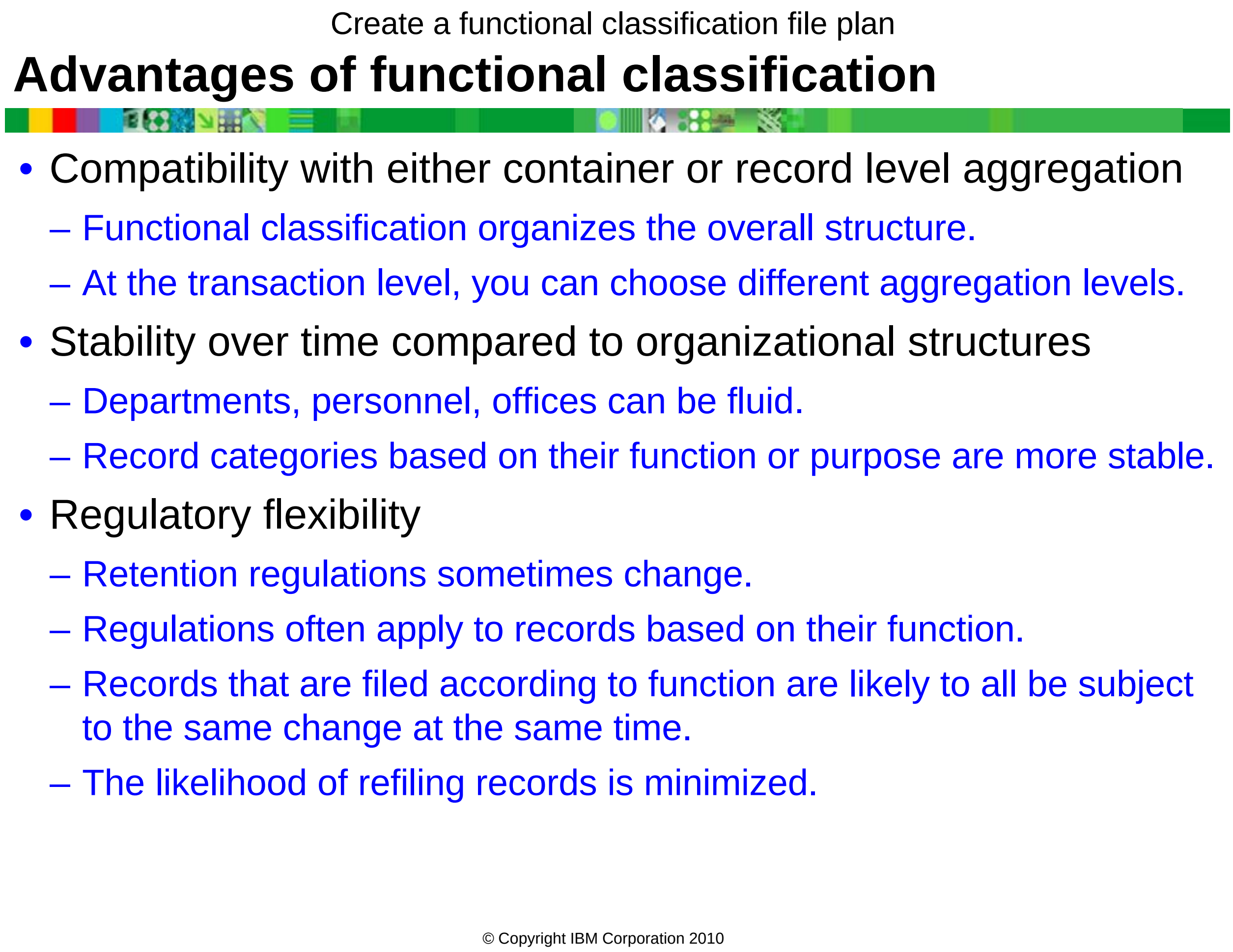
Payroll includes personnel management, global employees management, and leave of absence.

Staffing includes orientation and hiring.

Benefits includes health insurance and pensions.

Payroll includes timesheets and payroll delivery.

The functional classification scheme can have more than three levels. For example, there might be subsets of activities. Usually, there are between three and five levels. Adding more levels makes the system more complex. Beware of unnecessary complexity.



Create a functional classification file plan

Advantages of functional classification

- Compatibility with either container or record level aggregation
  - Functional classification organizes the overall structure.
  - At the transaction level, you can choose different aggregation levels.
- Stability over time compared to organizational structures
  - Departments, personnel, offices can be fluid.
  - Record categories based on their function or purpose are more stable.
- Regulatory flexibility
  - Retention regulations sometimes change.
  - Regulations often apply to records based on their function.
  - Records that are filed according to function are likely to all be subject to the same change at the same time.
  - The likelihood of refiling records is minimized.

Figure 1-29. Advantages of functional classification F1811.0

Notes:

Compatibility

The functional classification scheme is compatible with either container-level aggregation or record-level aggregation. How you organize your functional structure does not determine how you aggregate.

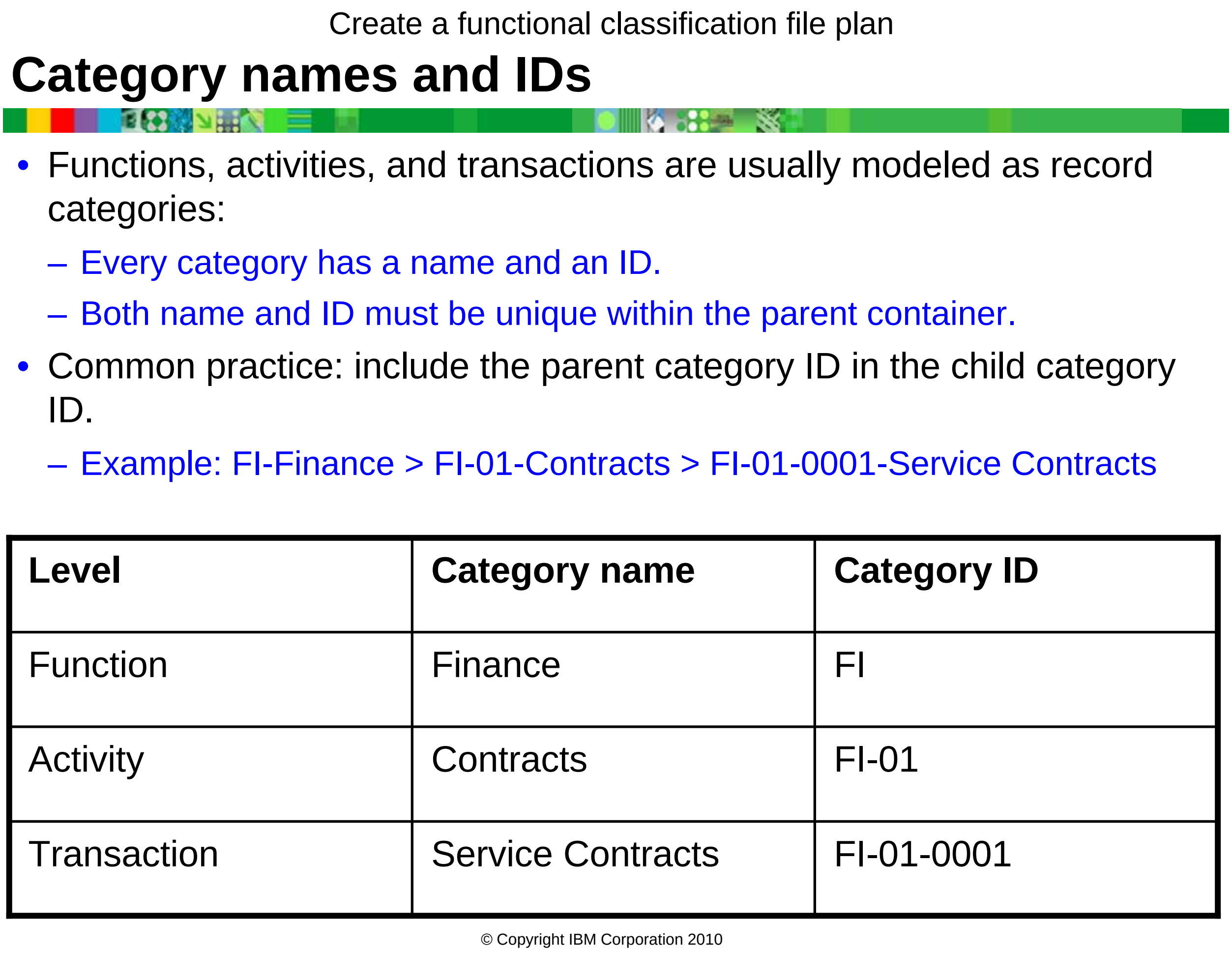
Stability

Record categories that are organized according to the purpose of the record are more stable than records that are organized according to more changeable structures, such as organizational structures. Departments can shift from one to another, as well as duties or responsibilities, so if the record categories are based on departmental structures, the records themselves need to be refiled whenever this happens. Records that are based on functional classification, however, do not move, because the purpose of the record does not change, regardless of changing departments.

Flexibility

Regulations for records occasionally change. When a regulation changes, the retention or disposition rules for records might be impacted. If records are not organized properly, someone must go through the entire file plan and find the records to which the new regulation applies. In many cases, the regulations affect records based on their purpose. So, having records organized by purpose means that in the event of a regulatory change, the records that are affected are all together, so they can be updated as a group.

For example, you might have many records that are kept for 3 years, but some are kept for 5 years. You might want to have two categories based on years, one for 3 years and one for 5 years. However, a new regulation stating that bills payable must be kept for 7 years has been established. Previously, these records were kept for 5 years. You must go through and find all of the bills payable and refile them. But they have been filed in the same category as all of the other records in the 5-year category, so the process is long and tedious. However, if you have filed all of the bills payable in the same category, based on the function classification scheme, then you do not need to refile the records: you only need to change the disposition schedule on the bills payable category.



Create a functional classification file plan

Category names and IDs

- Functions, activities, and transactions are usually modeled as record categories:
  - Every category has a name and an ID.
  - Both name and ID must be unique within the parent container.
- Common practice: include the parent category ID in the child category ID.
  - Example: FI-Finance > FI-01-Contracts > FI-01-0001-Service Contracts

Level	Category name	Category ID
Function	Finance	FI
Activity	Contracts	FI-01
Transaction	Service Contracts	FI-01-0001

Figure 1-30. Category names and IDs F1811.0

Notes:

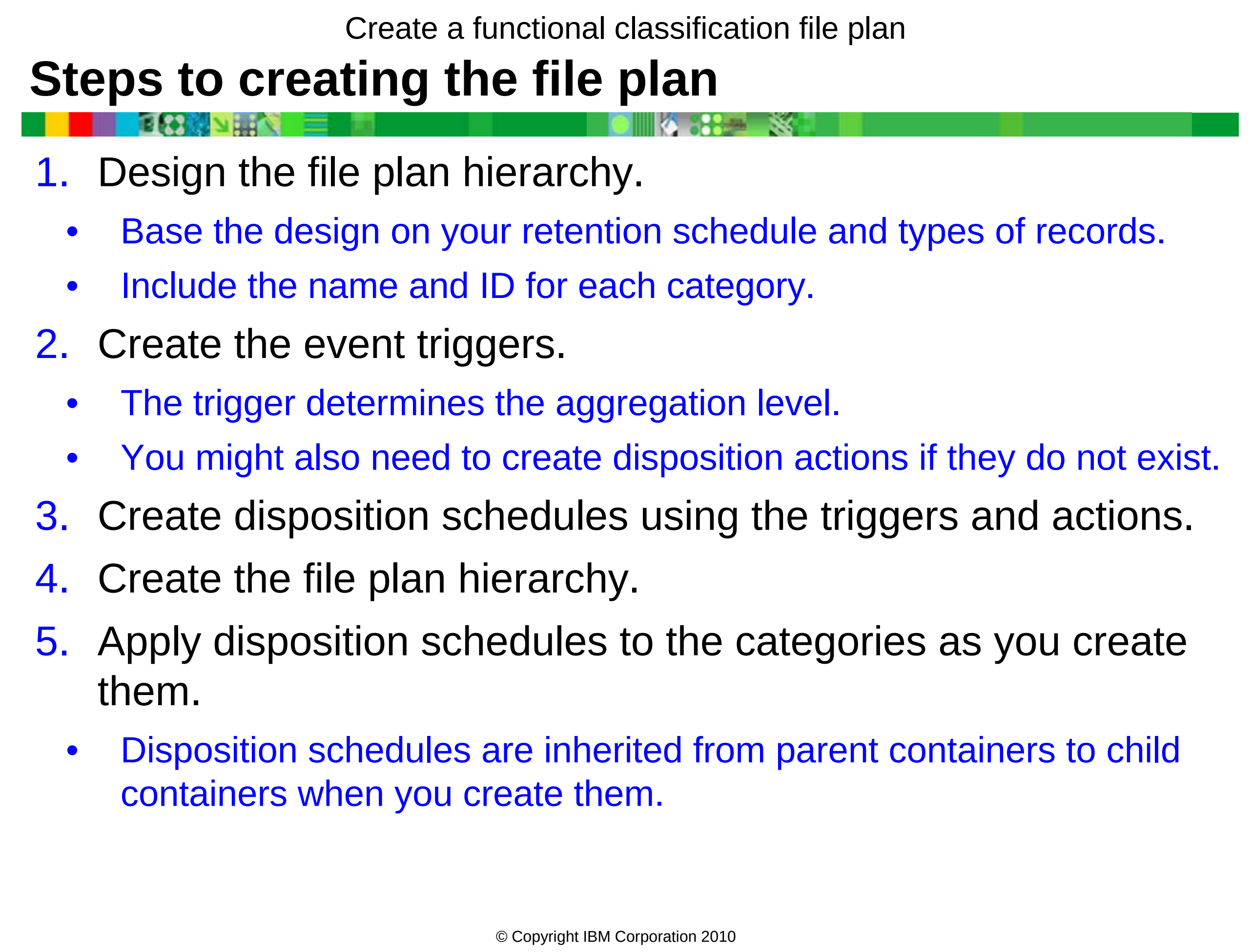
Table contents:

Level,Category name,Category ID

Function,Finance,FI,

Activity,Contracts,FI-01,

Transaction,Service Contracts,FI-01-0001



Create a functional classification file plan

Steps to creating the file plan

1. Design the file plan hierarchy.
   - Base the design on your retention schedule and types of records.
   - Include the name and ID for each category.
2. Create the event triggers.
   - The trigger determines the aggregation level.
   - You might also need to create disposition actions if they do not exist.
3. Create disposition schedules using the triggers and actions.
4. Create the file plan hierarchy.
5. Apply disposition schedules to the categories as you create them.
   - Disposition schedules are inherited from parent containers to child containers when you create them.

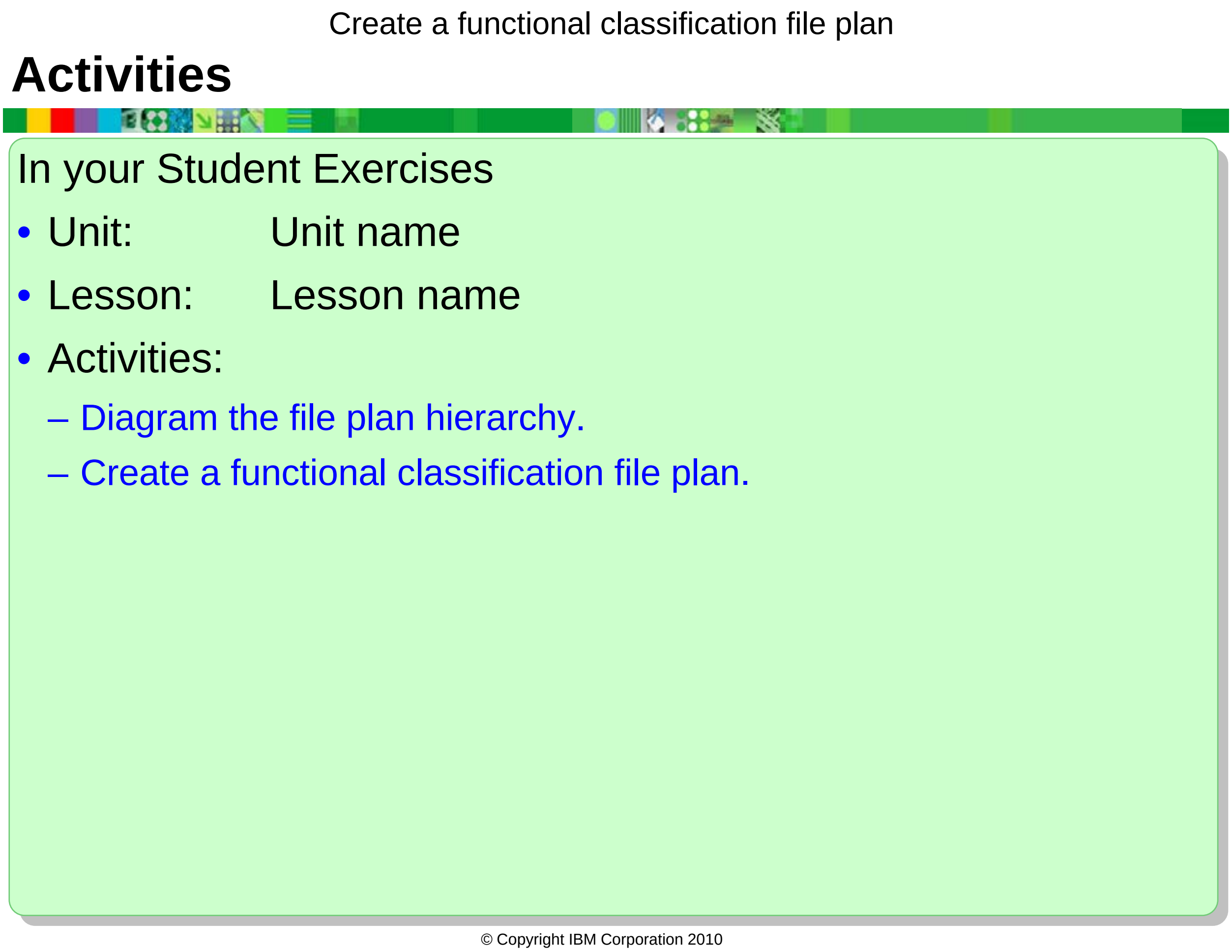
Figure 1-31. Steps to creating the file plan F1811.0

Notes:

Before you can create the event triggers, you must be certain that the record classes and properties exist. Because record classes and properties are created by the Content Engine or records administrator, you need to make sure that these tasks have been done before you begin.

You create event triggers on the Configuration page in Enterprise Records. When you create internal event triggers, you determine the aggregation level that applies to the disposition schedule to which the trigger is added. Therefore, you must decide on the aggregation before you create the trigger. Disposition actions are usually created during the installation and initial configuration of Enterprise Records. However, if they do not exist, you must create them before you create the disposition schedule.

You might decide to create the file plan hierarchy before you create the disposition schedules. This approach can be used for the top levels of the hierarchy that do not use disposition schedule inheritance.



Create a functional classification file plan

Activities

In your Student Exercises

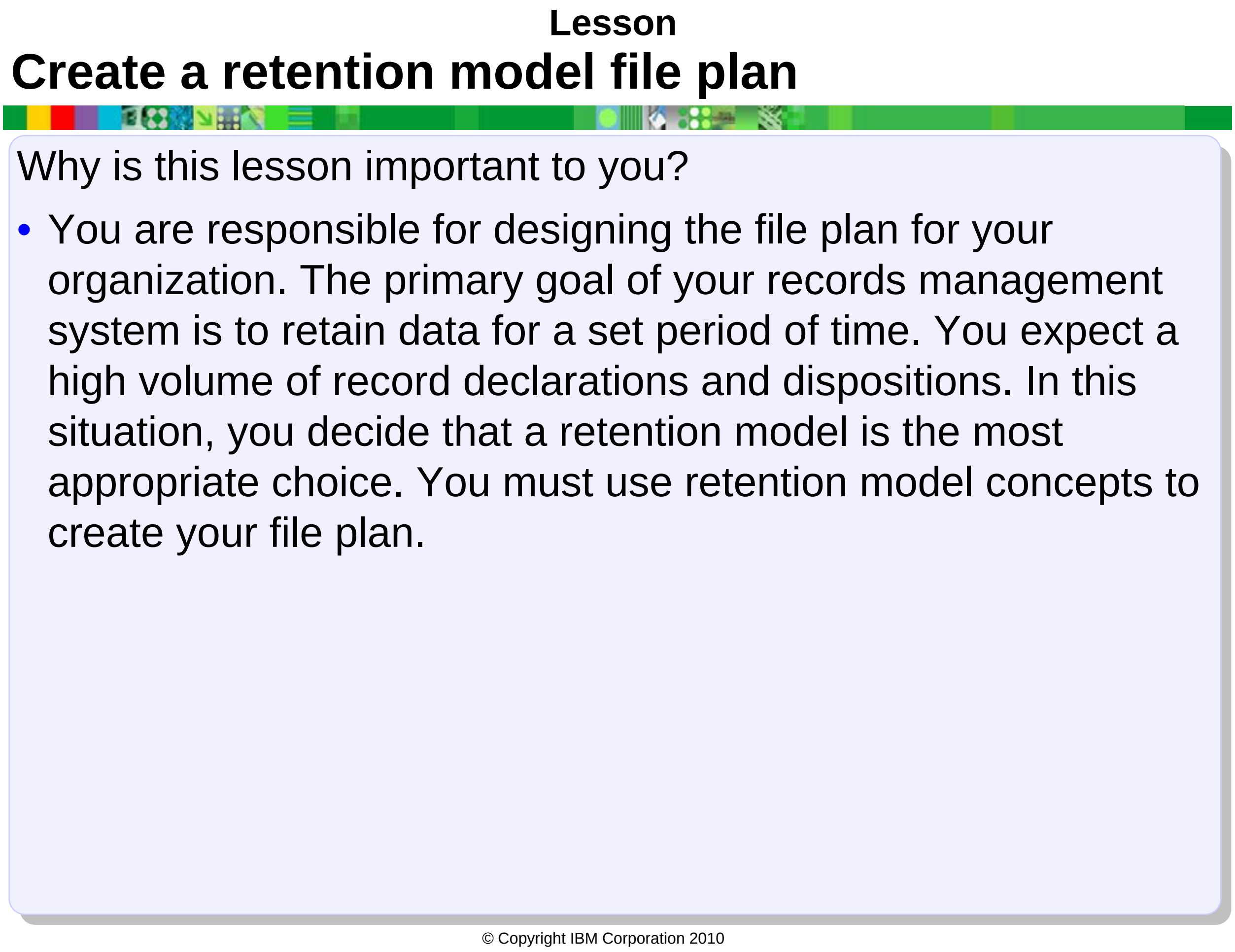
- Unit: Unit name
- Lesson: Lesson name
- Activities:
  - – Diagram the file plan hierarchy.
  - – Create a functional classification file plan.

Figure 1-32. Activities F1811.0

Notes:

Use your Student Exercises to perform the activities listed.

Lesson 1.4. Create a retention model file plan



Lesson

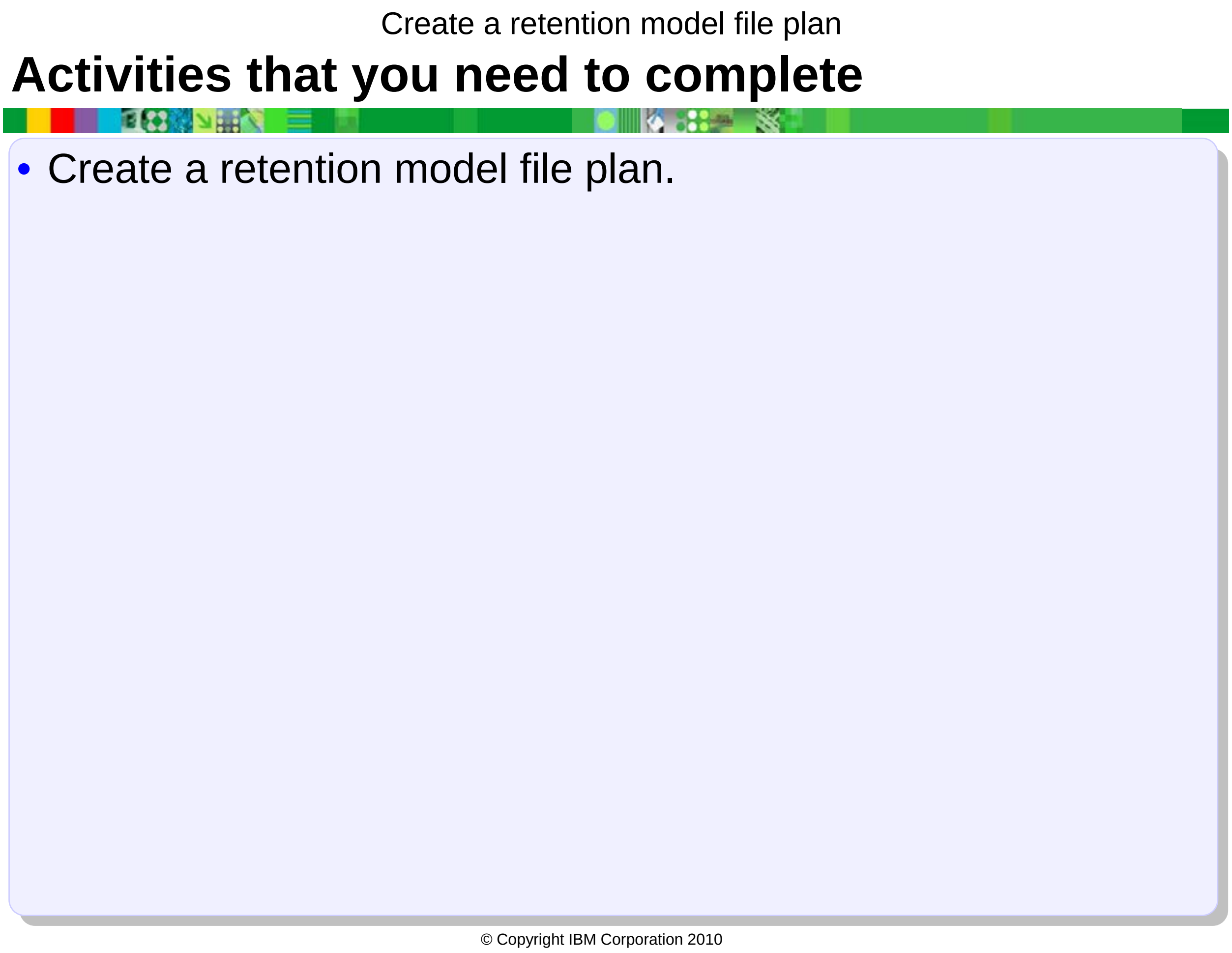
Create a retention model file plan

Why is this lesson important to you?

- You are responsible for designing the file plan for your organization. The primary goal of your records management system is to retain data for a set period of time. You expect a high volume of record declarations and dispositions. In this situation, you decide that a retention model is the most appropriate choice. You must use retention model concepts to create your file plan.

Figure 1-33. Create a retention model file plan F1811.0

Notes:



Create a retention model file plan

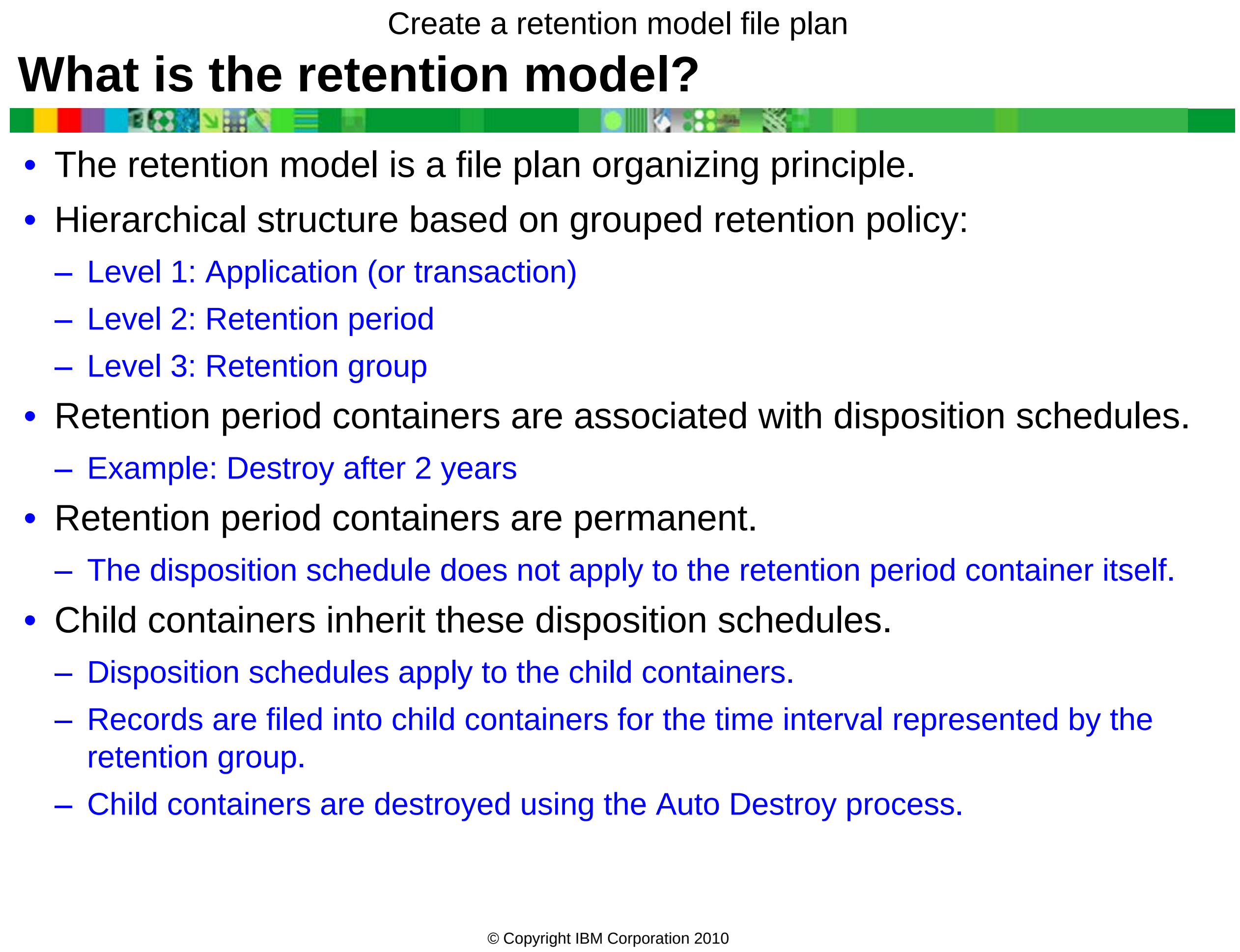
Activities that you need to complete

- Create a retention model file plan.

Figure 1-34. Activities that you need to complete F1811.0

Notes:

These are the activities that you need to complete.



Create a retention model file plan

What is the retention model?

- The retention model is a file plan organizing principle.
- Hierarchical structure based on grouped retention policy:
  - Level 1: Application (or transaction)
  - Level 2: Retention period
  - Level 3: Retention group
- Retention period containers are associated with disposition schedules.
  - Example: Destroy after 2 years
- Retention period containers are permanent.
  - The disposition schedule does not apply to the retention period container itself.
- Child containers inherit these disposition schedules.
  - Disposition schedules apply to the child containers.
  - Records are filed into child containers for the time interval represented by the retention group.
  - Child containers are destroyed using the Auto Destroy process.

Figure 1-35. What is the retention model? F1811.0

Notes:

The hierarchical structure of the retention model has three main levels.

Application

The application level is used to group all of the containers for a particular application of the retention model. For example, if email originating from the Human Resources department needs to be handled separately from the email originating from the Billing department, you need to create separate record category hierarchies for each. In this course, the application level of the retention hierarchy corresponds to the transaction level of the functional classification scheme.

Retention period

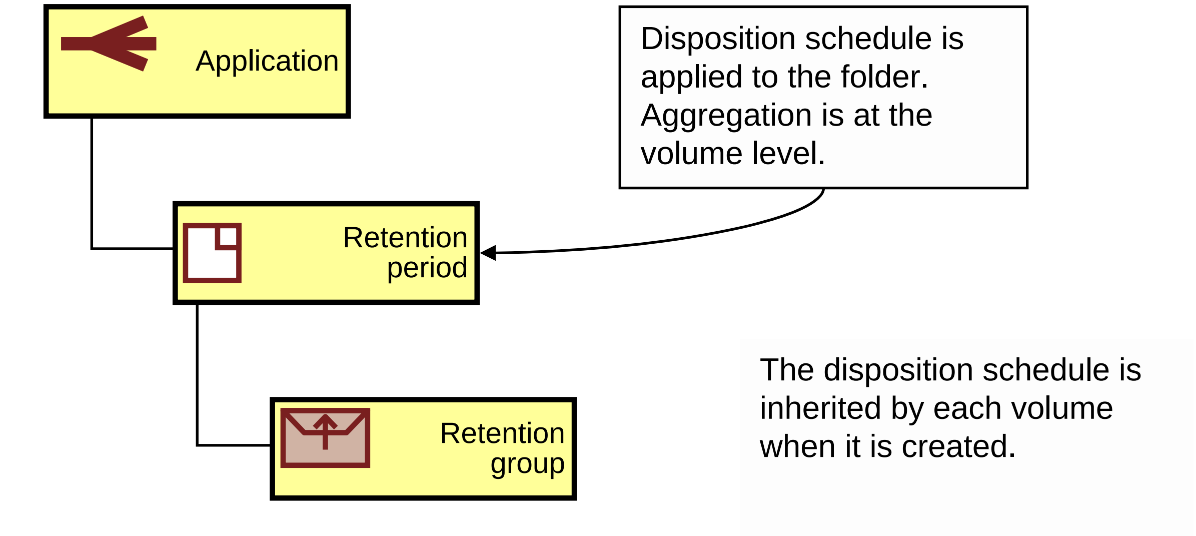
The retention period level is used to group all of the containers that have a similar retention period. For example, if some records need to be kept for 30 days, some for 60 days, and some for 90 days, you need to create three retention period containers at this level of the hierarchy.

Retention group

The third and lowest level in the hierarchy is used to group all records that need to be destroyed on the same day. Use a naming convention that includes a date for the containers at this level of the hierarchy. For example, you might use a container for every day of the year so that all records declared on a specific day are filed into the same container. The name of this container is based on the date that the records were declared.

Create a retention model file plan

Retention model structure



Disposition schedule is applied to the folder. Aggregation is at the volume level.

The disposition schedule is inherited by each volume when it is created.

Figure 1-36. Retention model structure F1811.0

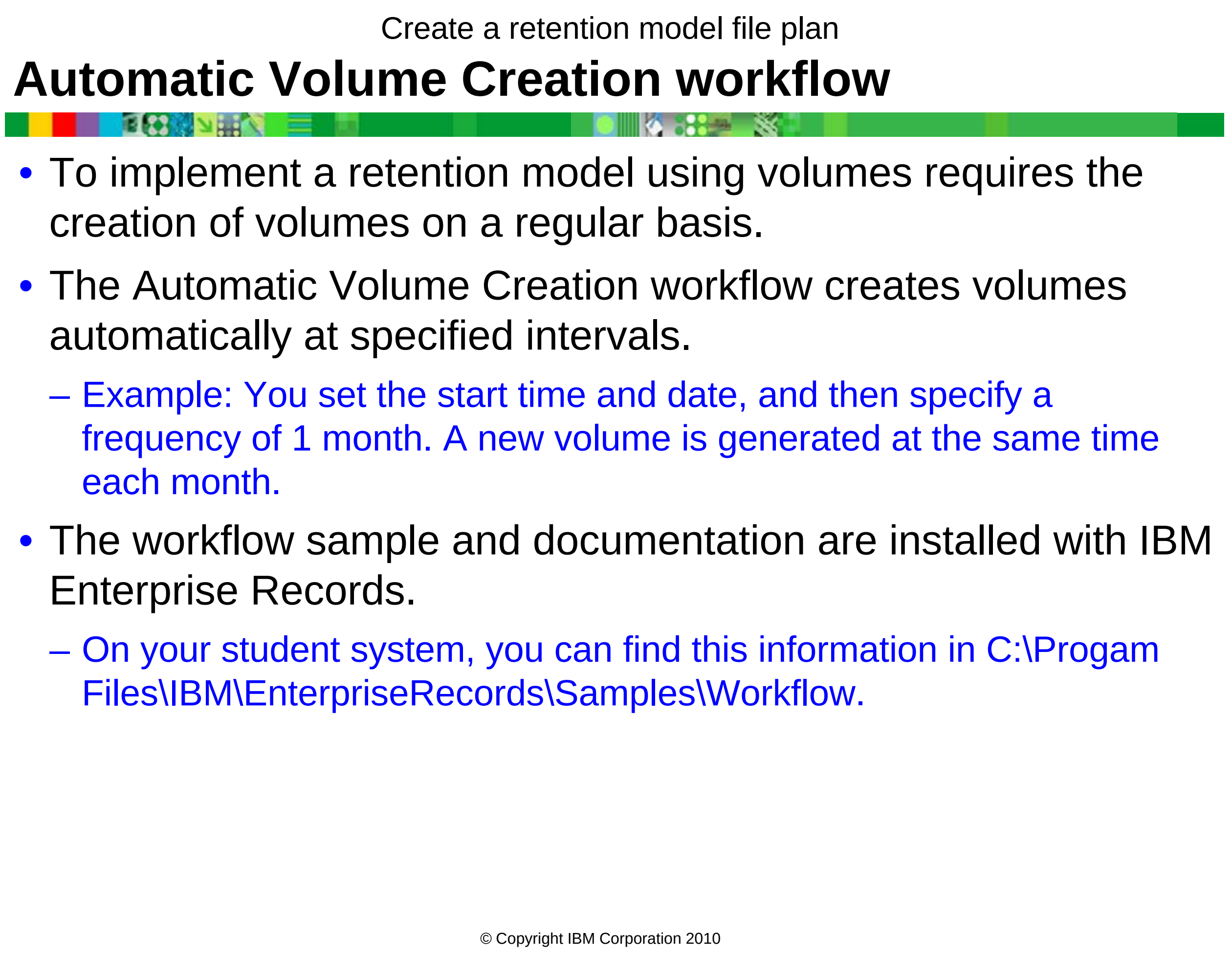
Notes:

The diagram shows the general structure of the retention model hierarchical file plan structure. It shows an application as a record category, a retention period folder, and a retention group volume. The disposition schedule is based on a time frame, for example 2 years. The disposition schedule is inherited by child volumes. The disposition schedule is aggregated at the volume level, so each volume is destroyed at the end of the retention period.

Container types

You can use categories for all levels in this model, but the category class must be include properties on which to base the event trigger. If you can determine that processing categories is faster than processing volumes, then you might choose to use all categories. But you must manage them correctly because the aggregation level of the event trigger does not distinguish between categories that are used for retention period containers and categories used for retention group containers if they are all categories.

For example, if you create a category for records to be kept for 1 year and attach a disposition schedule that destroys categories after 1 year, then the retention period category is destroyed at the same time as the first retention group category! To avoid this problem, you must create disposition schedules that are triggered by events that are never true for the retention period-level containers.



Create a retention model file plan

Automatic Volume Creation workflow

- To implement a retention model using volumes requires the creation of volumes on a regular basis.
- The Automatic Volume Creation workflow creates volumes automatically at specified intervals.
  - Example: You set the start time and date, and then specify a frequency of 1 month. A new volume is generated at the same time each month.
- The workflow sample and documentation are installed with IBM Enterprise Records.
  - On your student system, you can find this information in C:\Progam Files\IBM\EnterpriseRecords\Samples\Workflow.

Figure 1-37. Automatic Volume Creation workflow F1811.0

Notes:

Help path

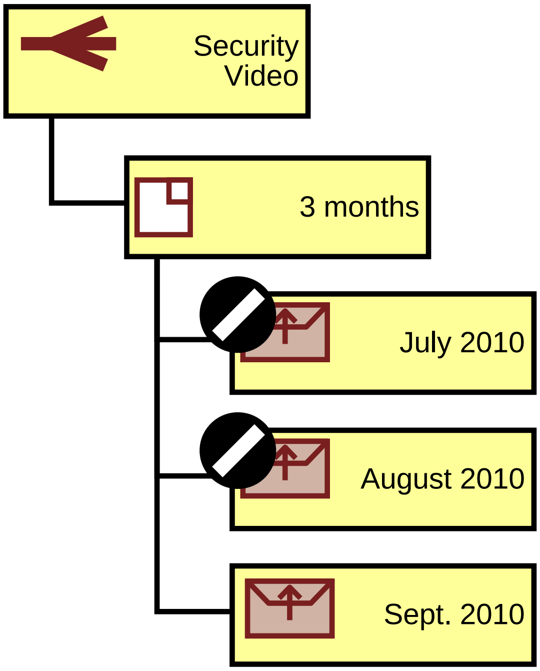
IBM FileNet P8 Documentation > Installing additional IBM FileNet P8 products > IBM Enterprise Records Installation and Upgrade > (Optional) Configuring IBM Enterprise Records after installation > Installing the Auto Volume Creation workflow

Auto Volume Creation workflow

The Auto Volume Creation workflow is a sample workflow in the IBM Enterprise Records installation package. After you launch the workflow, volumes are created automatically at the frequency of the days, weeks, or months that you set. The Auto Volume Creation workflow is useful for a retention model file plan.

Create a retention model file plan

Retention model example



At the end of each month, a workflow automatically creates a new volume. The previously open volume closes, triggering cutoff.

Retention is 3 months after cutoff.

The volume for the entire month is destroyed using the Auto Destroy sweep process.

Figure 1-38. Retention model example F1811.0

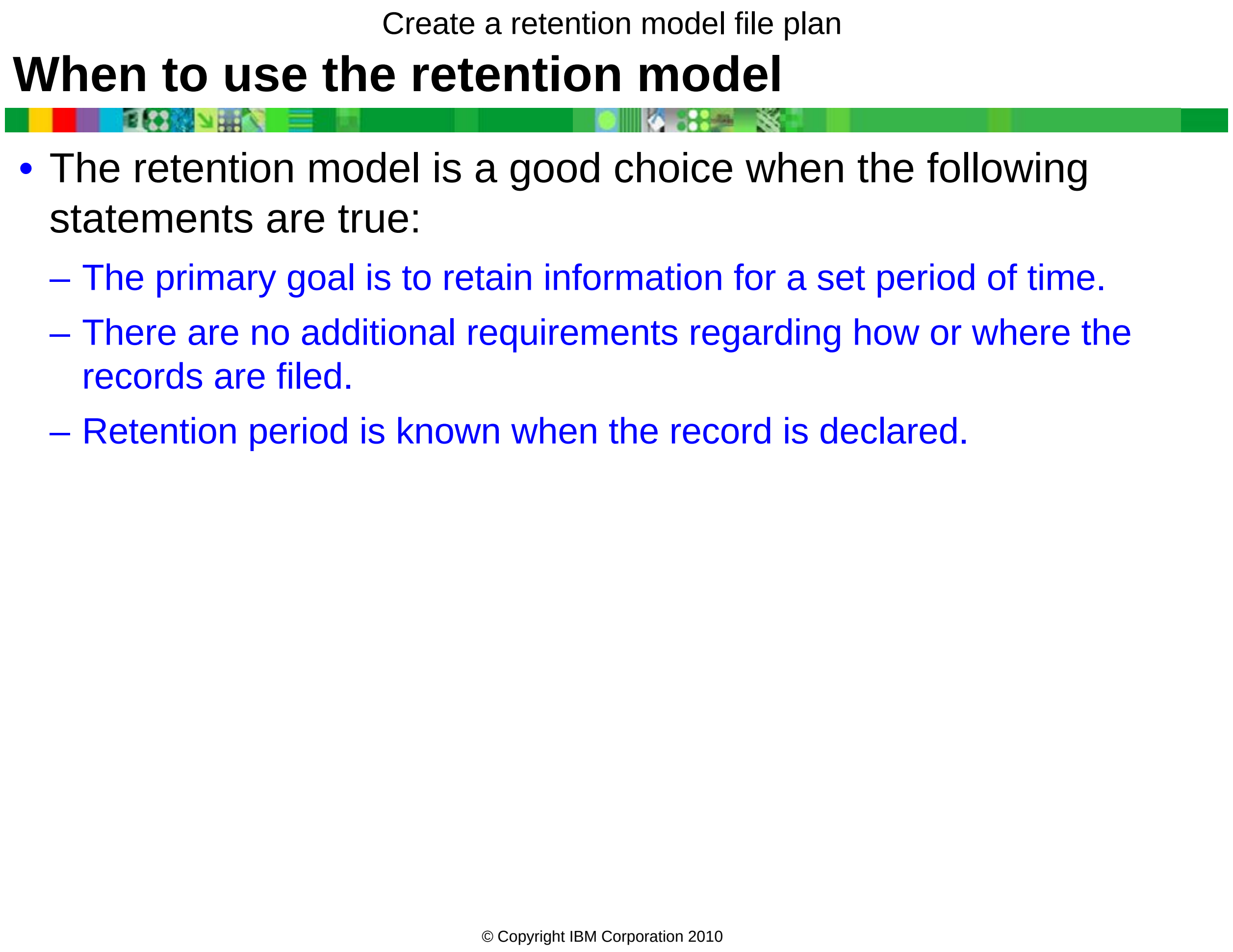
Notes:

The diagram shows an example of a retention model. A category named Security Video is the activity. The next level is the retention period. The retention period is a folder. Within the retention period folder are several retention group volumes. Each volume represents 1 month. It is reasonable to configure Disposition Sweep and Auto Destroy to run on a monthly schedule.

At the end of each month, the workflow automatically creates a new volume. When a new volume is opened, the previously open volume closes automatically, triggering cutoff.

Retention is 3 months after cutoff.

The volume for the entire month is destroyed using the Auto Destroy sweep process.



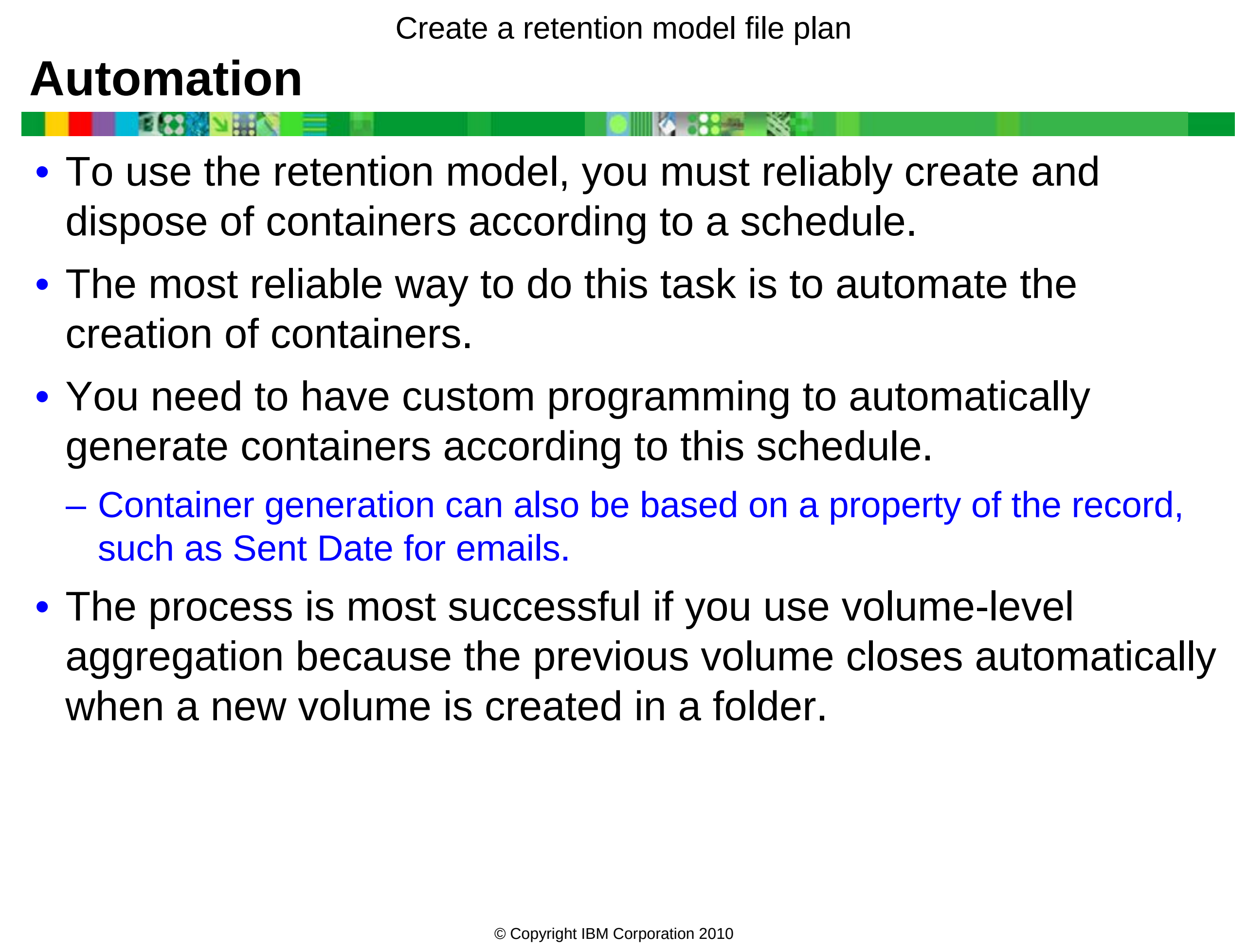
Create a retention model file plan

When to use the retention model

- The retention model is a good choice when the following statements are true:
  - The primary goal is to retain information for a set period of time.
  - There are no additional requirements regarding how or where the records are filed.
  - Retention period is known when the record is declared.

Figure 1-39. When to use the retention model F1811.0

Notes:



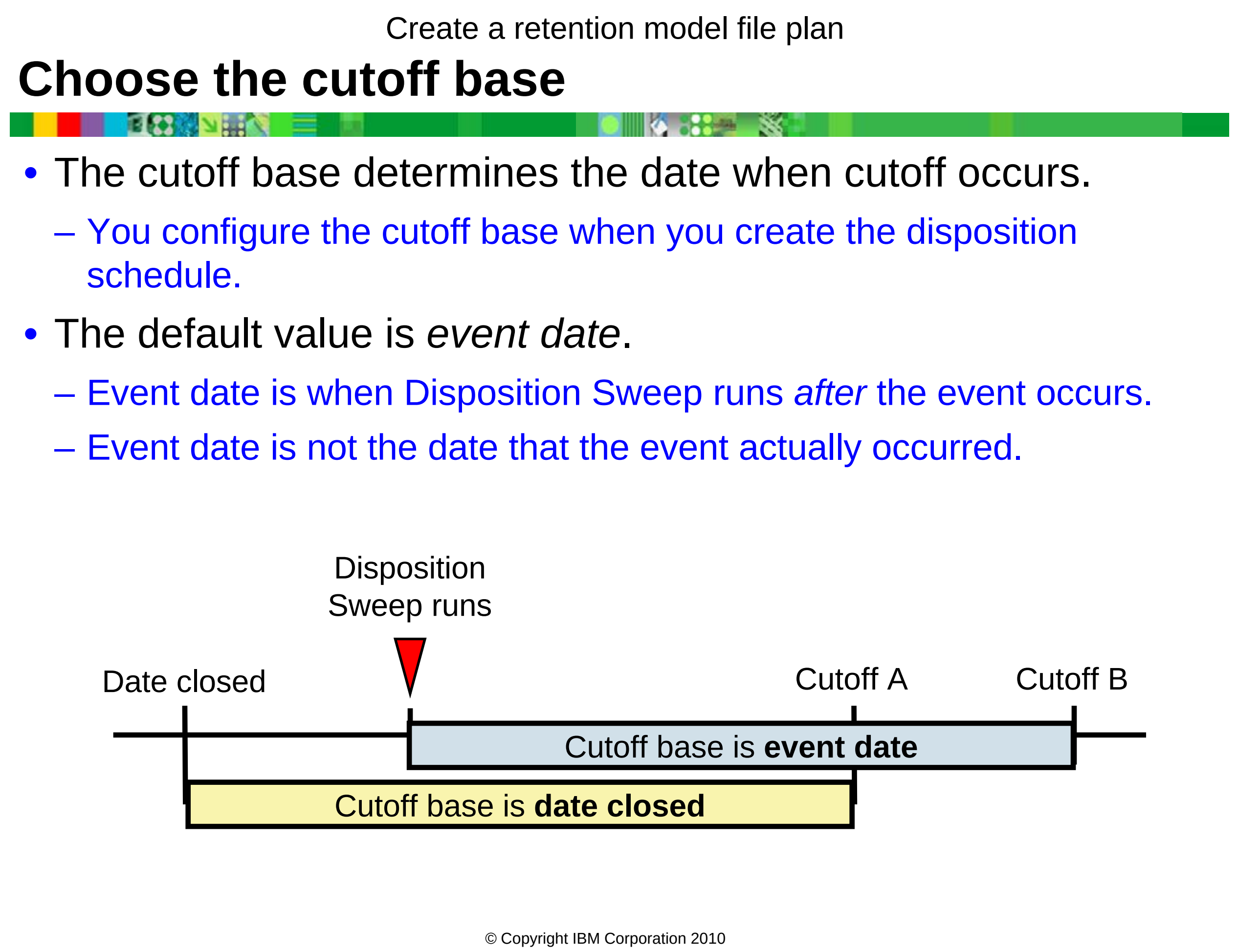
Create a retention model file plan

Automation

- To use the retention model, you must reliably create and dispose of containers according to a schedule.
- The most reliable way to do this task is to automate the creation of containers.
- You need to have custom programming to automatically generate containers according to this schedule.
  - Container generation can also be based on a property of the record, such as Sent Date for emails.
- The process is most successful if you use volume-level aggregation because the previous volume closes automatically when a new volume is created in a folder.

Figure 1-40. Automation F1811.0

Notes:



Create a retention model file plan

Choose the cutoff base

- The cutoff base determines the date when cutoff occurs.
  - – You configure the cutoff base when you create the disposition schedule.
- The default value is *event date*.
  - – Event date is when Disposition Sweep runs *after* the event occurs.
  - – Event date is not the date that the event actually occurred.

Figure 1-41. Choose the cutoff base F1811.0

Notes:

The diagram shows how the cutoff date is determined by the cutoff base. The cutoff trigger is the closing of a container. If the date the container that is closed is used as the cutoff base, then cutoff occurs 1 year after the container is closed (Cutoff A), regardless of when Disposition Sweep runs within that interval. If the event date is used for the cutoff base, then cutoff occurs 1 year after Disposition Sweep registers that the container as closed (Cutoff B).

Using the cutoff base

Sometimes, you might want to set the Date Closed property as the cutoff base. The cutoff base determines the cutoff date. If the cutoff base is based on the Date Closed property, then cutoff is determined by the date when the container is closed, no matter when Disposition Sweep runs. If the cutoff base is the event date, then cutoff is determined by the date when Disposition Sweep runs after the event trigger has occurred. If there is a long delay between when the event trigger occurs and the time when Disposition Sweep runs, the cutoff date is still determined by when Disposition Sweep runs.

Example

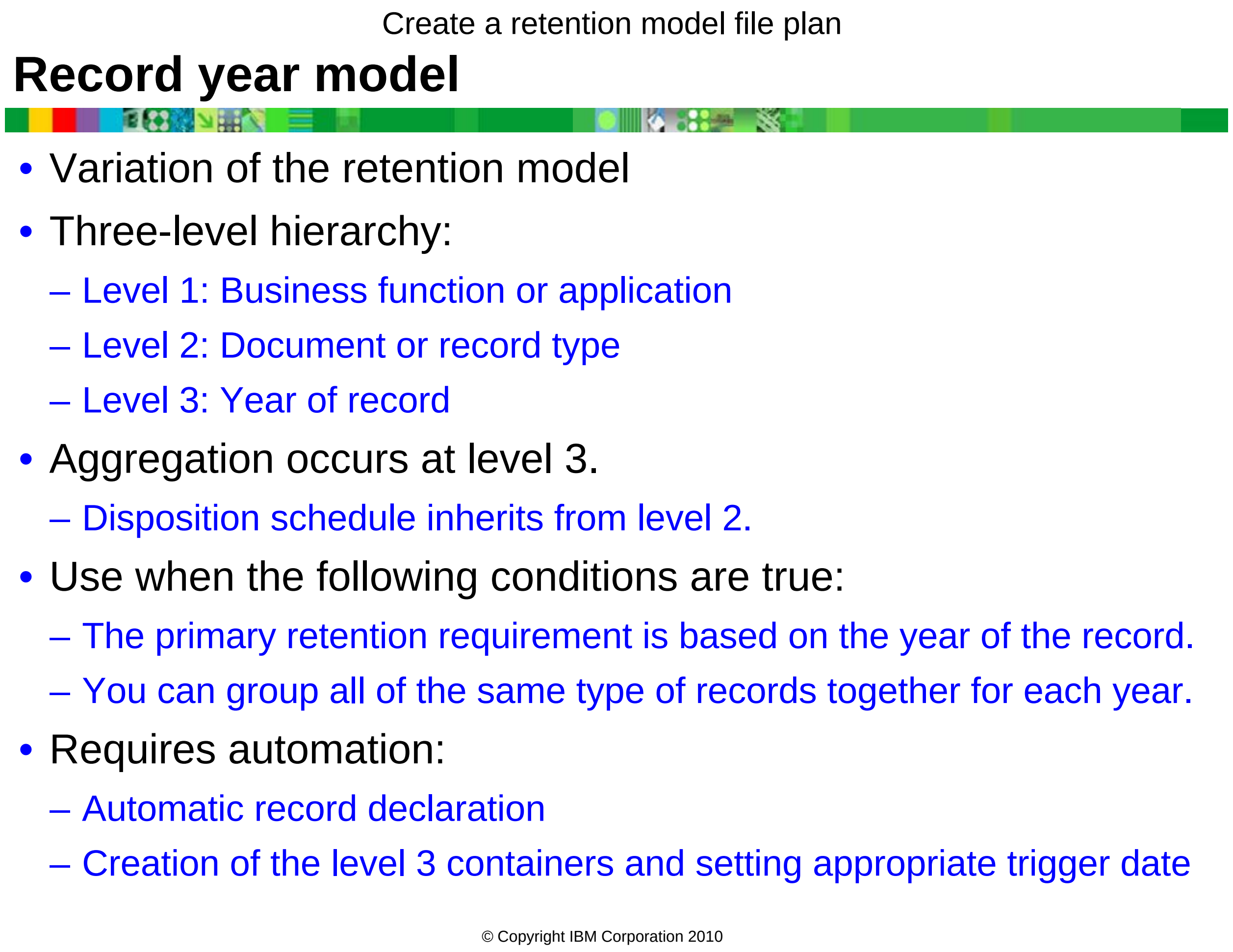
Fred is terminated on 6 July 2010. Records are supposed to be destroyed 7 years after termination.

Disposition Sweep runs on 1 January 2011. During the sweep, Fred's employment status is flagged as terminated.

If the *event date* is the cutoff base, then Fred's records are destroyed on 1 January 2018.

If the *Date Terminated* property is the cutoff base, then Fred's records are destroyed on 6 July 2017.

The cutoff *trigger value* is the same in both cases: Date Terminated IS NOT NULL. Only the cutoff *base* has changed. If you are running Disposition Sweep often enough, the difference is negligible. However, there might arise a case in which you need to update the cutoff base. If the cutoff base has a date value, then you can change the date to a new date. When Disposition Sweep runs, the Current Phase Execution Date is updated according to the new value. However, the event date is a system property that cannot be changed.



Create a retention model file plan

Record year model

- Variation of the retention model
- Three-level hierarchy:
  - – Level 1: Business function or application
  - – Level 2: Document or record type
  - – Level 3: Year of record
- Aggregation occurs at level 3.
  - – Disposition schedule inherits from level 2.
- Use when the following conditions are true:
  - – The primary retention requirement is based on the year of the record.
  - – You can group all of the same type of records together for each year.
- Requires automation:
  - – Automatic record declaration
  - – Creation of the level 3 containers and setting appropriate trigger date

Figure 1-42. Record year model F1811.0

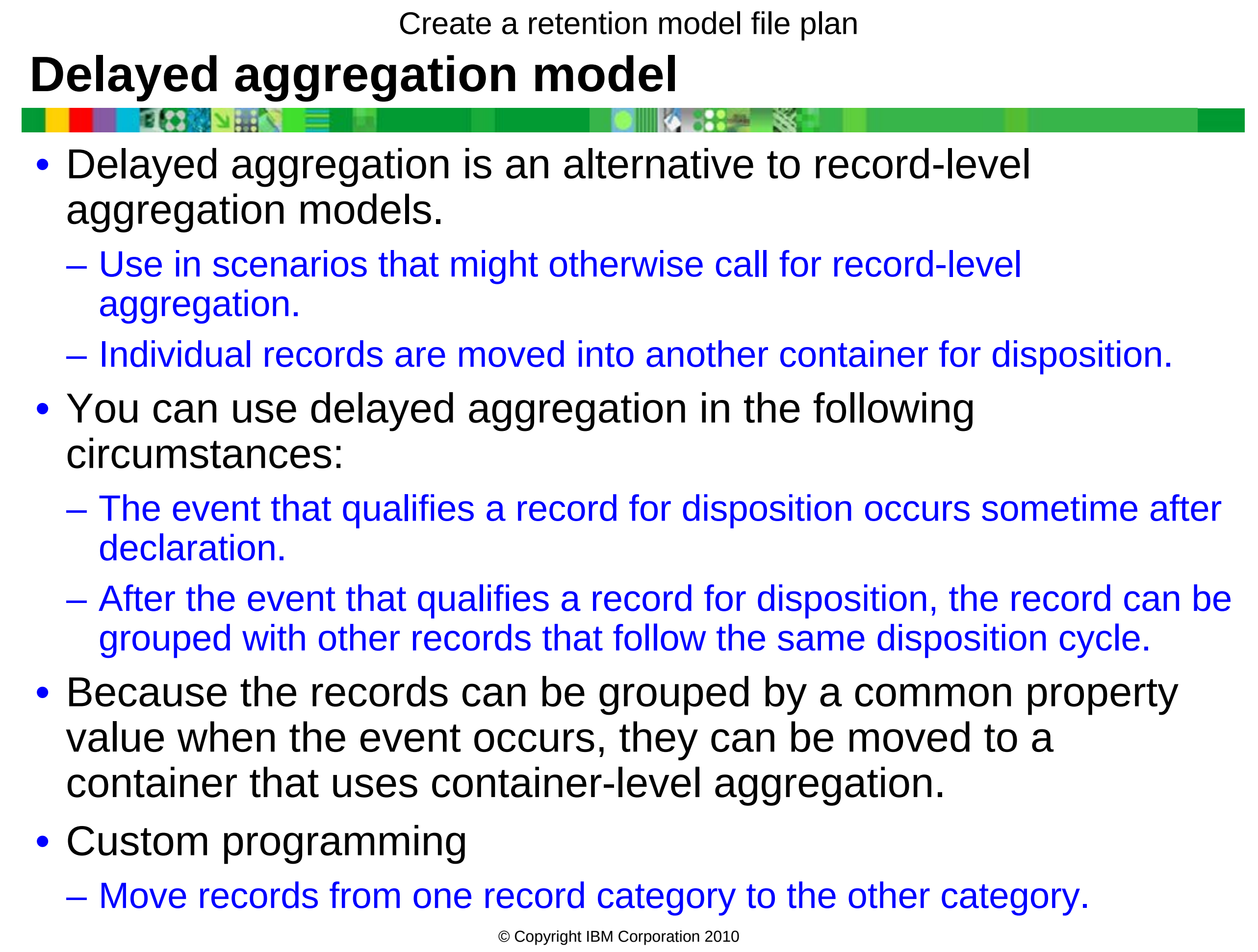
Notes:

The record year model is a variant of the retention model. It can be used in situations in which the primary retention requirement is for records of a given type to be destroyed based on the year of record.

The records are declared automatically into the appropriate year based on the level 1 and level 2 attributes of the record plus the year of record as determined by some date value or other attribute of the record at the time of declaration.

For example, correspondence or finance might be declared into the appropriate year of record based on either the date scanned (for paper) or the date received (for email). For some applications, you can use the Date Created property on the RDOS.

The trigger date value is assigned only to the record year containers (level 3). Level 1 and level 2 categories must not have a trigger date value. The trigger date property is a custom property that must be added to the container class to be used for the level 3 containers.



Create a retention model file plan

Delayed aggregation model

- Delayed aggregation is an alternative to record-level aggregation models.
  - – Use in scenarios that might otherwise call for record-level aggregation.
  - – Individual records are moved into another container for disposition.
- You can use delayed aggregation in the following circumstances:
  - – The event that qualifies a record for disposition occurs sometime after declaration.
  - – After the event that qualifies a record for disposition, the record can be grouped with other records that follow the same disposition cycle.
- Because the records can be grouped by a common property value when the event occurs, they can be moved to a container that uses container-level aggregation.
- Custom programming
  - – Move records from one record category to the other category.

Figure 1-43. Delayed aggregation model F1811.0

Notes:

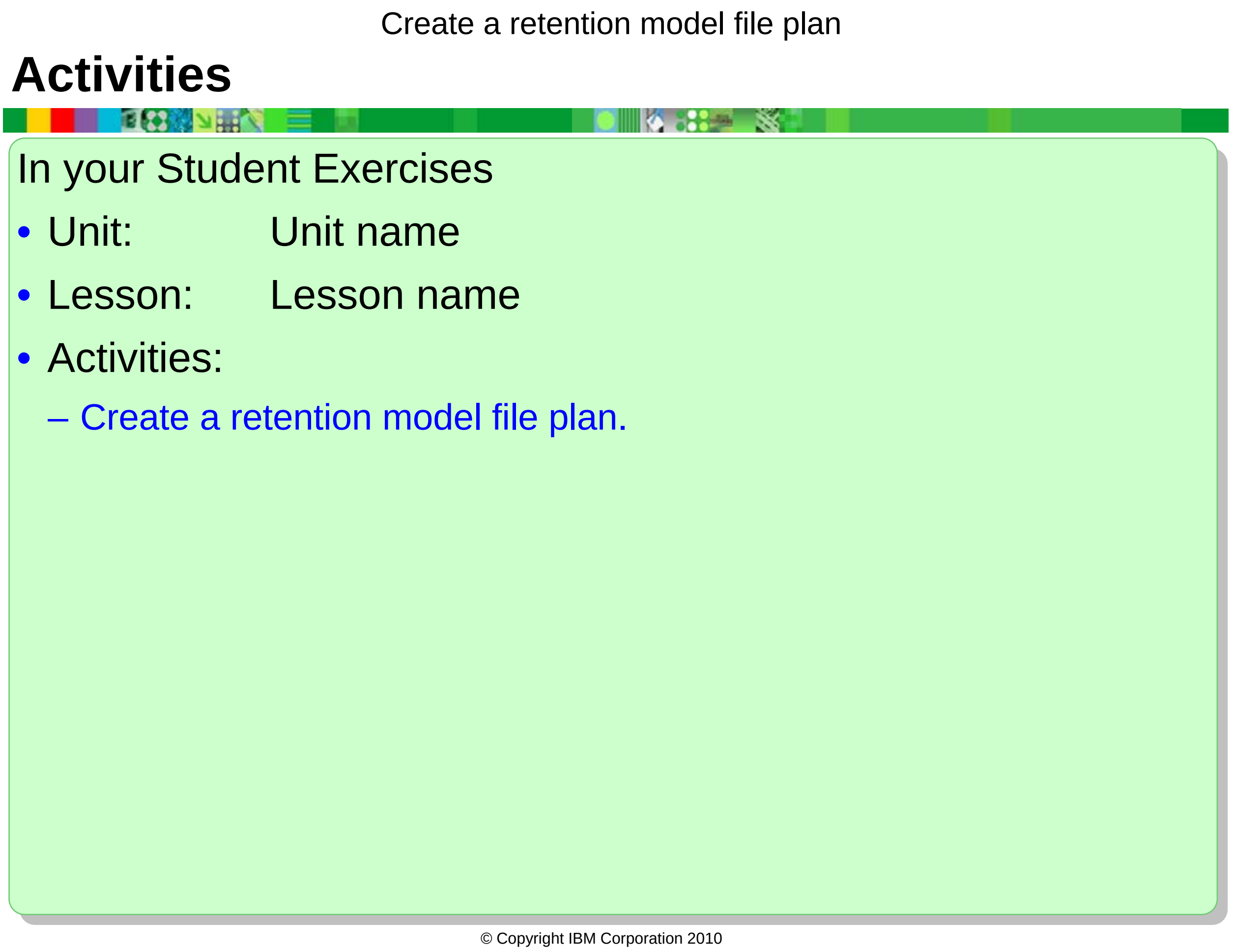
Record-level aggregation is often used in cases in which some property of the document or record itself is the trigger for the beginning of disposition. However, record-level aggregation is inefficient. One way to circumvent the inefficiency of record-level aggregation in these cases is to use a delayed aggregation model.

In the delayed aggregation model, records are initially filed into a container that does not have a disposition schedule. These records are in the active state until the event occurs that triggers the beginning of the retention period. In this case, though, the triggering event is not used as the cutoff trigger. Instead, the event is used as a trigger to move the record into a different container. The new container has a disposition schedule associated with it that is aggregated at the container level.

For example, a record has a trigger-date value that has a null value and is declared into an Active Records container. At some point afterward, the record is no longer active, so the trigger-date value is set to a non-null value. A custom program then moves the record into a container that is scheduled to be destroyed on a specific date. All of the records that are moved into this container can then be destroyed at the same time.

In some circumstances, this model is actually less efficient than record-level aggregation because the time needed to move records into a different container might exceed the time required to process the records individually without moving them.

Consider the cost of creating containers and moving the records against the cost of using record level aggregation. For example, if it is necessary to create several containers, which will contain only a few records, then using this model can result in higher processing costs than using the record-level aggregation.



Create a retention model file plan

Activities

In your Student Exercises

- Unit: Unit name
- Lesson: Lesson name
- Activities:
  - – Create a retention model file plan.

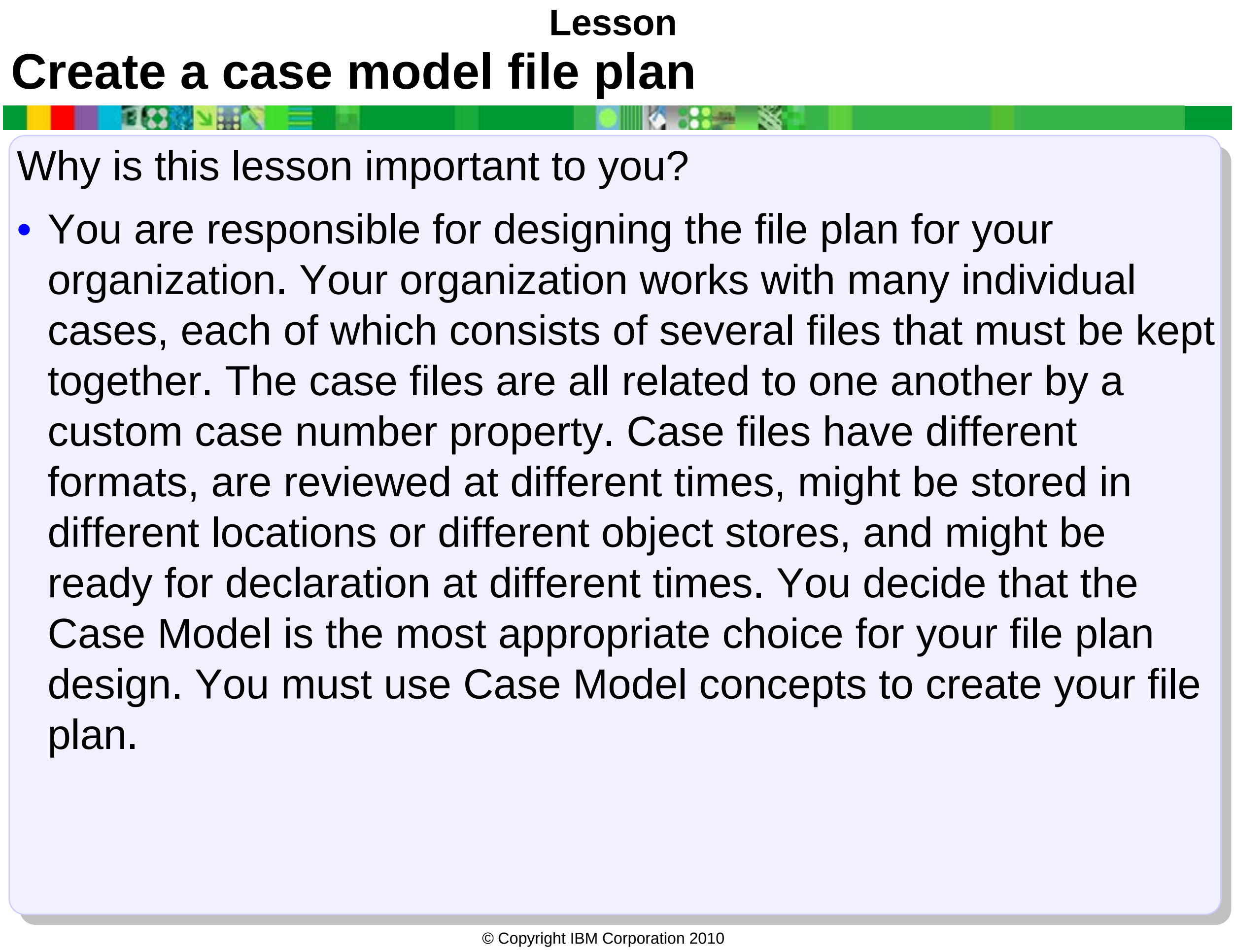
© Copyright IBM Corporation 2010

Figure 1-44. Activities F1811.0

Notes:

Use your Student Exercises to perform the activities listed.

Lesson 1.5. Create a case model file plan



Lesson

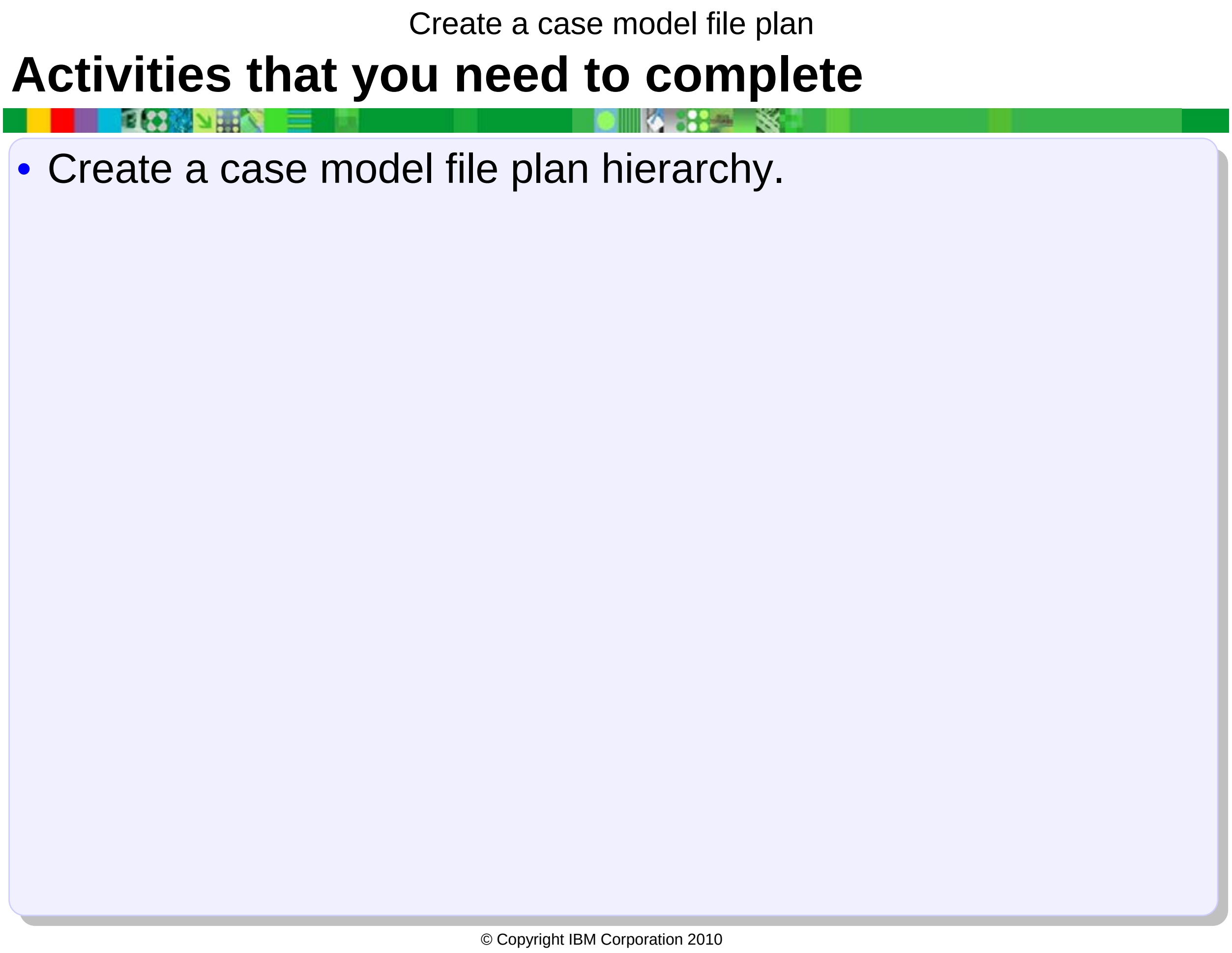
Create a case model file plan

Why is this lesson important to you?

- You are responsible for designing the file plan for your organization. Your organization works with many individual cases, each of which consists of several files that must be kept together. The case files are all related to one another by a custom case number property. Case files have different formats, are reviewed at different times, might be stored in different locations or different object stores, and might be ready for declaration at different times. You decide that the Case Model is the most appropriate choice for your file plan design. You must use Case Model concepts to create your file plan.

Figure 1-45. Create a case model file plan F1811.0

Notes:



Create a case model file plan

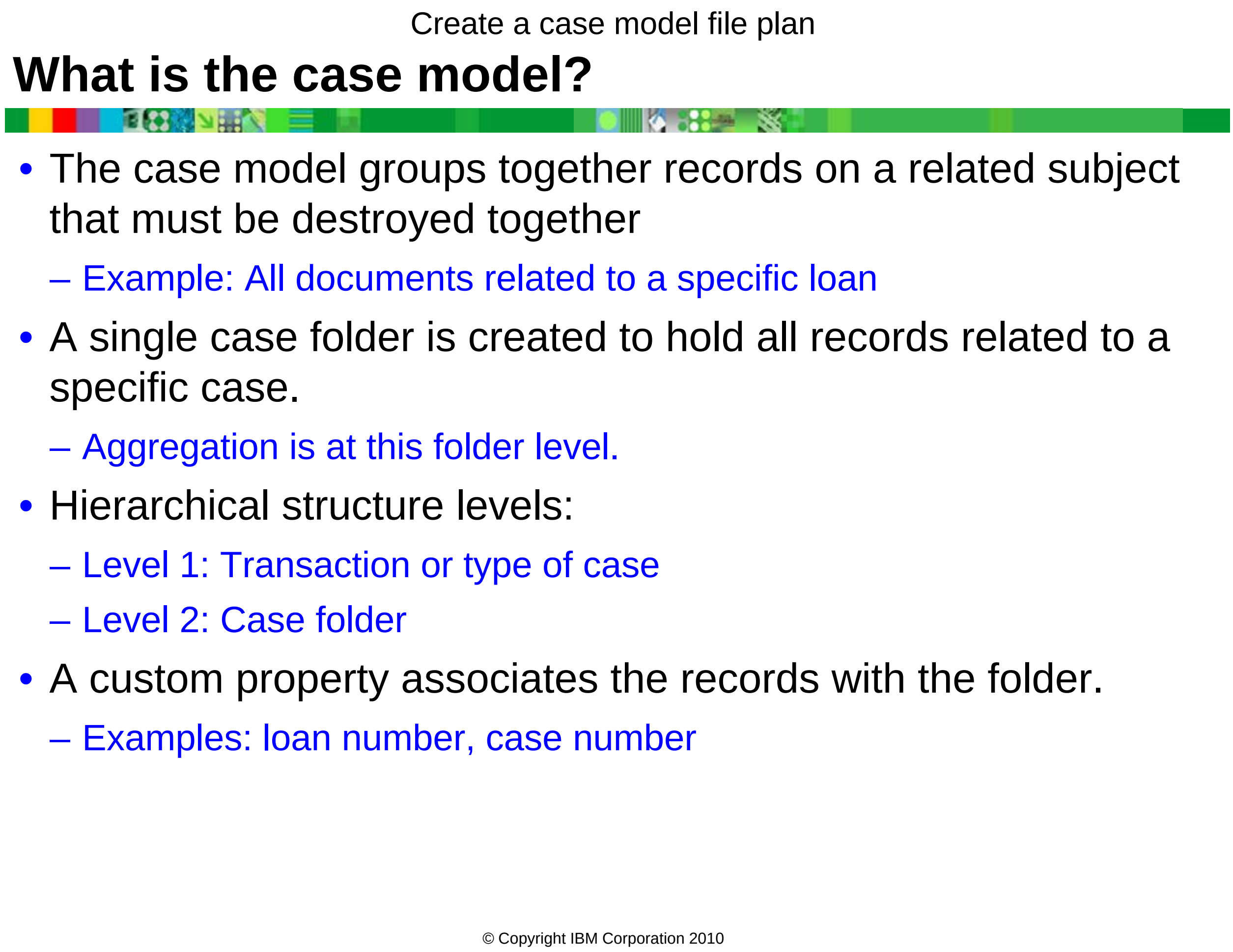
Activities that you need to complete

- Create a case model file plan hierarchy.

Figure 1-46. Activities that you need to complete F1811.0

Notes:

These are the activities that you need to complete.



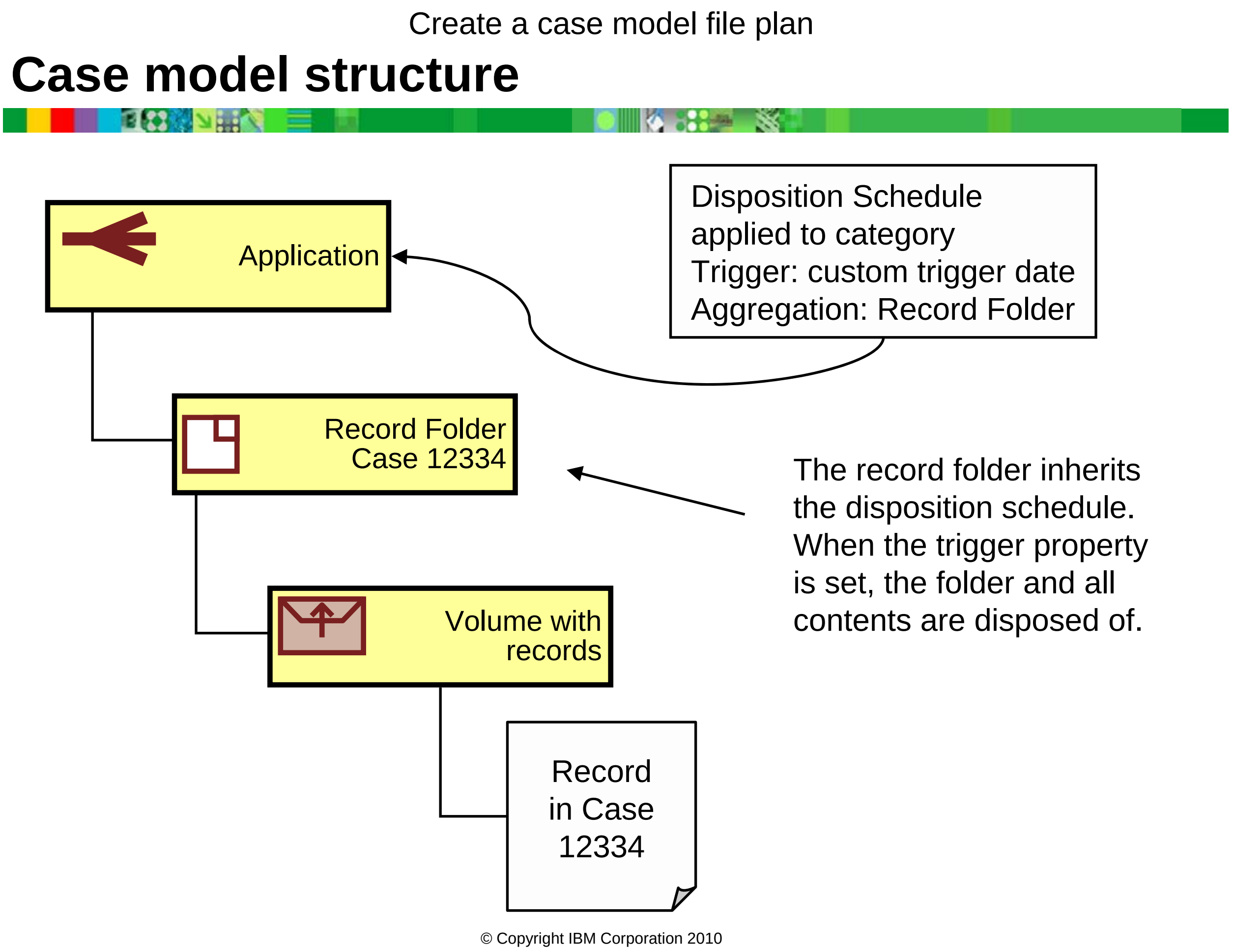
Create a case model file plan

What is the case model?

- The case model groups together records on a related subject that must be destroyed together
  - – Example: All documents related to a specific loan
- A single case folder is created to hold all records related to a specific case.
  - – Aggregation is at this folder level.
- Hierarchical structure levels:
  - – Level 1: Transaction or type of case
  - – Level 2: Case folder
- A custom property associates the records with the folder.
  - – Examples: loan number, case number

Figure 1-47. What is the case model? F1811.0

Notes:



Create a case model file plan

Case model structure

Disposition Schedule applied to category
Trigger: custom trigger date
Aggregation: Record Folder

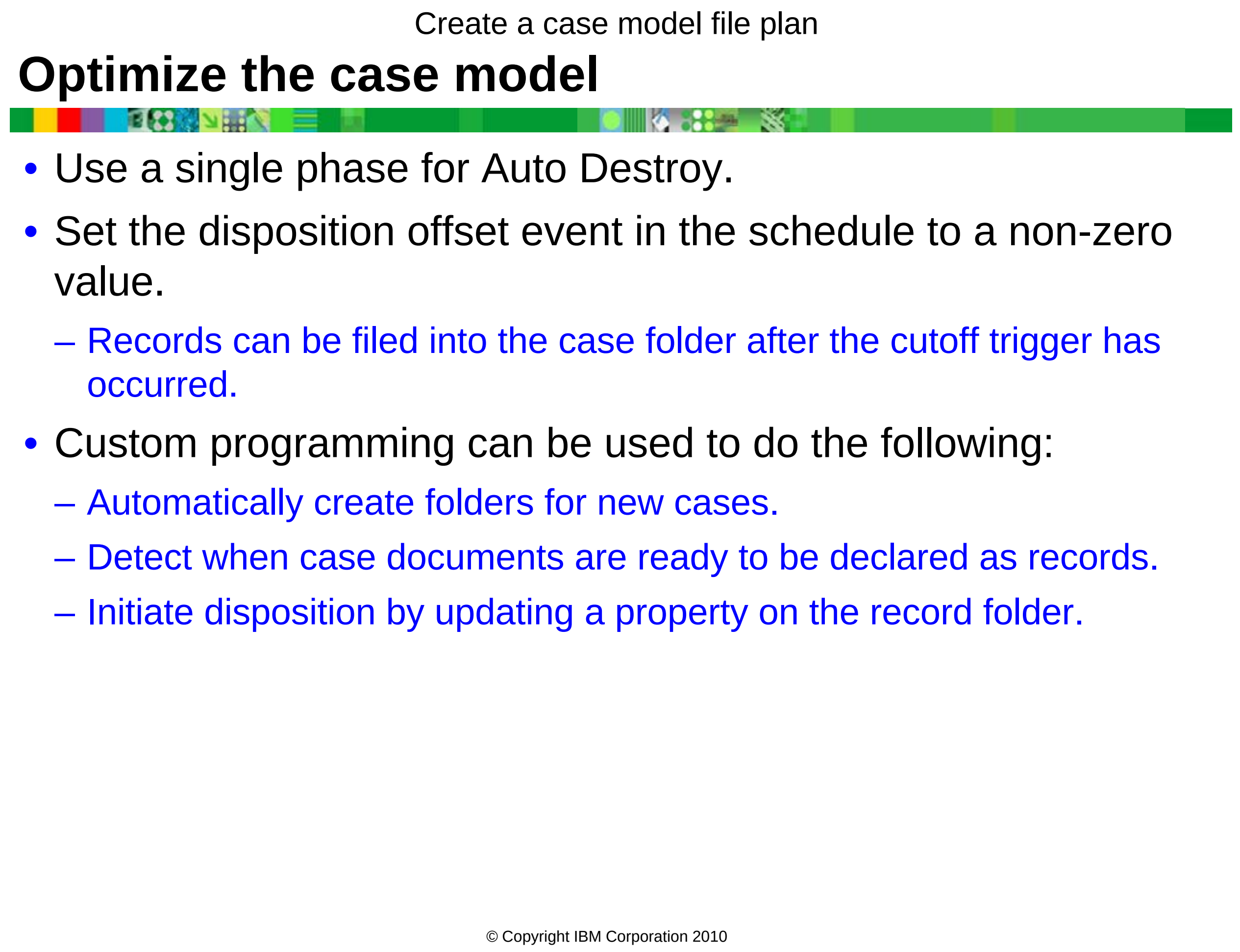
The record folder inherits the disposition schedule. When the trigger property is set, the folder and all contents are disposed of.

Figure 1-48. Case model structure F1811.0

Notes:

The diagram shows the structure of the case model file plan design. An application category is associated with a disposition schedule. The category contains a record folder, which inherits the disposition schedule. The schedule is aggregated at the folder level. The folder has a volume that has one or more records in it. Each record has a custom property value, such as a case number, that matches the identifier of the case folder.

Records that are declared in a case model typically have a custom property that is used to identify which folder to file the record into.



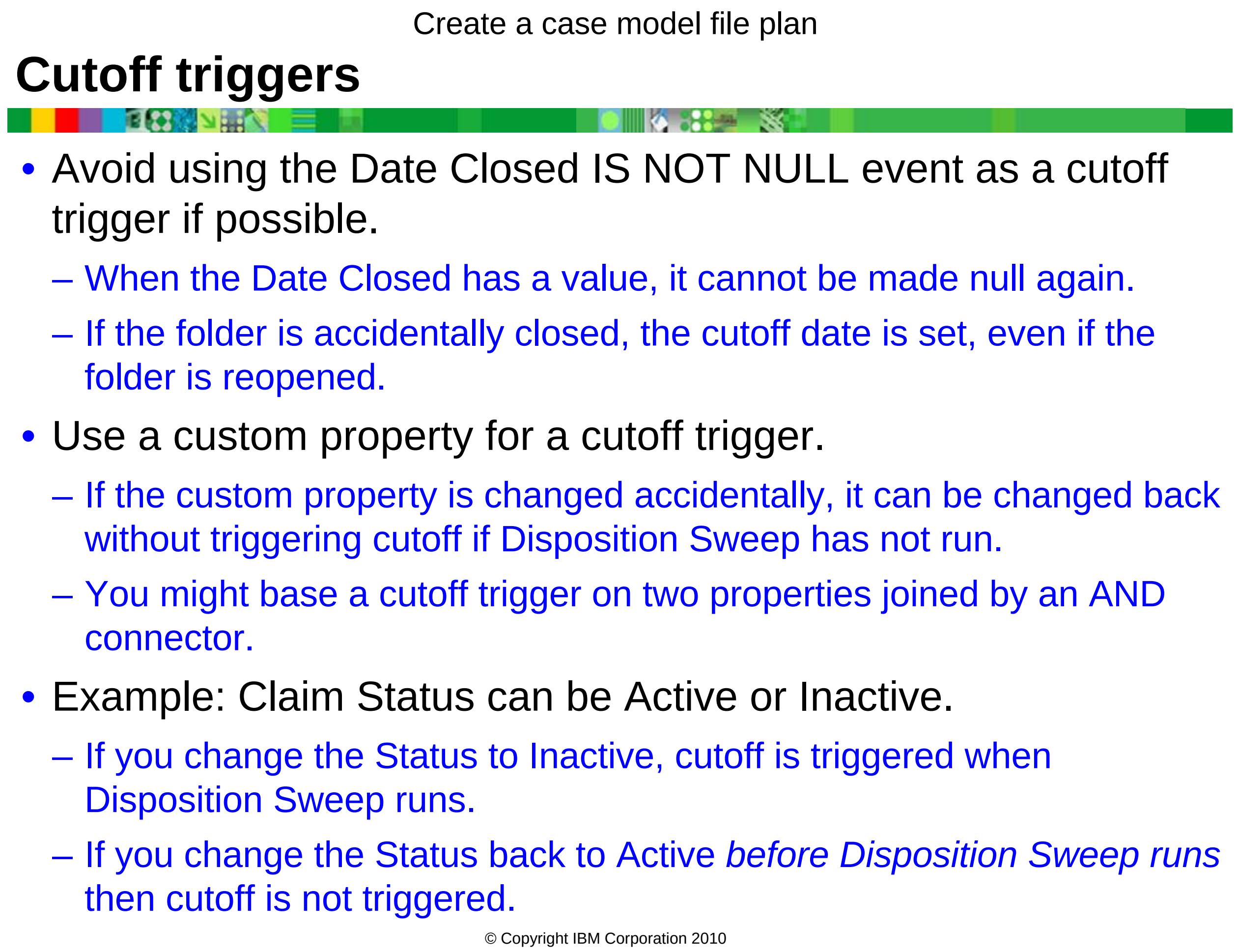
Create a case model file plan

Optimize the case model

- Use a single phase for Auto Destroy.
- Set the disposition offset event in the schedule to a non-zero value.
  - Records can be filed into the case folder after the cutoff trigger has occurred.
- Custom programming can be used to do the following:
  - Automatically create folders for new cases.
  - Detect when case documents are ready to be declared as records.
  - Initiate disposition by updating a property on the record folder.

Figure 1-49. Optimize the case model F1811.0

Notes:



Create a case model file plan

Cutoff triggers

- Avoid using the Date Closed IS NOT NULL event as a cutoff trigger if possible.
  - – When the Date Closed has a value, it cannot be made null again.
  - – If the folder is accidentally closed, the cutoff date is set, even if the folder is reopened.
- Use a custom property for a cutoff trigger.
  - – If the custom property is changed accidentally, it can be changed back without triggering cutoff if Disposition Sweep has not run.
  - – You might base a cutoff trigger on two properties joined by an AND connector.
- Example: Claim Status can be Active or Inactive.
  - – If you change the Status to Inactive, cutoff is triggered when Disposition Sweep runs.
  - – If you change the Status back to Active *before Disposition Sweep runs* then cutoff is not triggered.

Figure 1-50. Cutoff triggers F1811.0

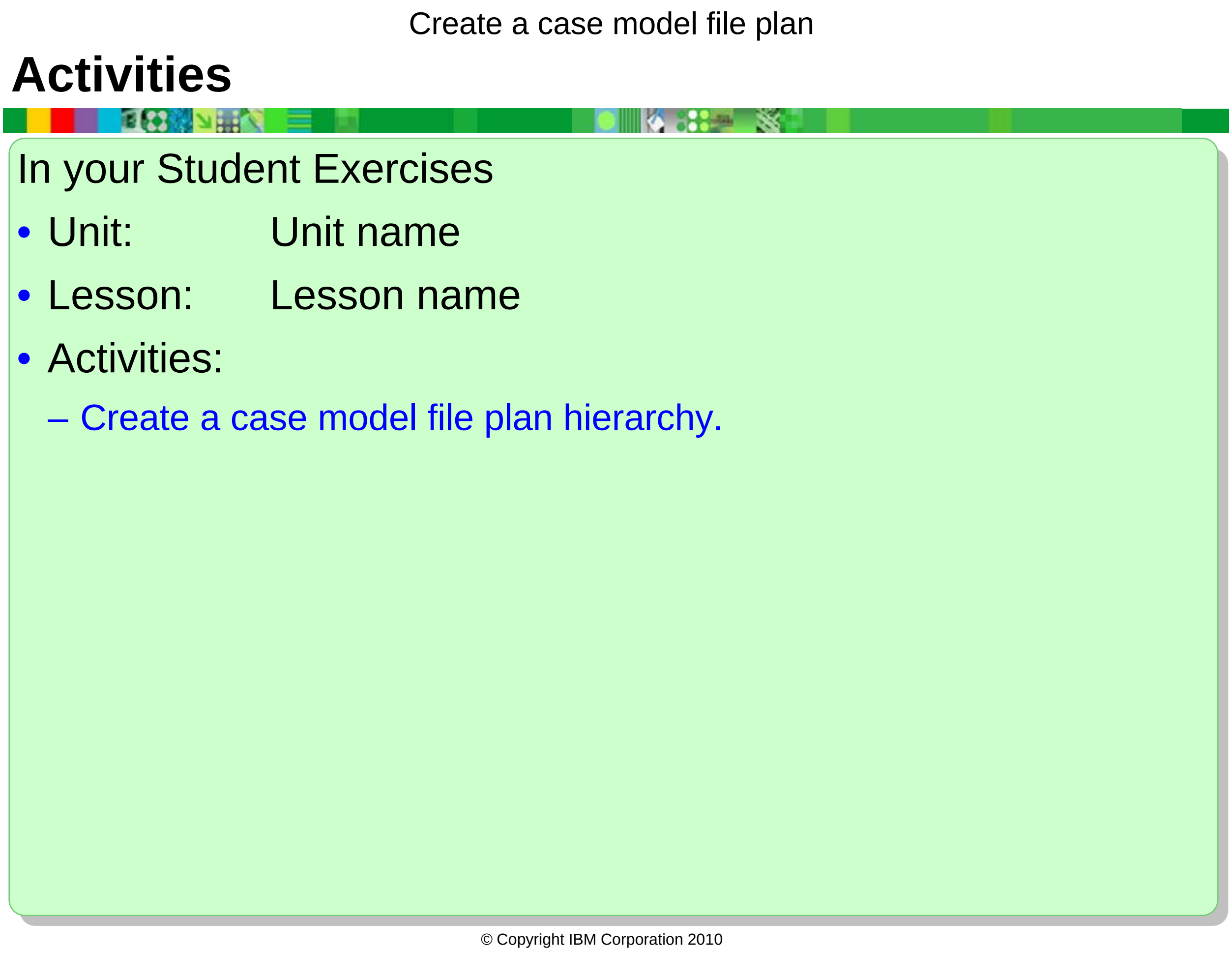
Notes:

When using the case model, use caution when planning the cutoff trigger. If you use the Date Closed IS NOT NULL event as the trigger, then cutoff occurs when the folder is closed. The drawback to this approach is that someone might mistakenly close the folder. The folder can be reopened, but the Date Closed property is still not null, so cutoff is triggered regardless of whether the folder is reopened.

If you use a custom property, such as a binary property, as a cutoff trigger, then if the value changes and then changes back before Disposition Sweep runs, the cutoff is not triggered.

Cutoff base

You need to carefully consider the cutoff base as well as the cutoff trigger. Instead of the event date, you might select a custom trigger date. The trigger date might be updated after cutoff, which can affect the Current Phase Execution Date.



Create a case model file plan

Activities

In your Student Exercises

- Unit: Unit name
- Lesson: Lesson name
- Activities:
  - Create a case model file plan hierarchy.

Figure 1-51. Activities F1811.0

Notes:

Use your Student Exercises to perform the activities listed.

Glossary

A

action
See disposition action.

aggregation
Part of an internal event trigger that determines which type of IBM Enterprise Records entity is affected by the disposition action. For example, depending on the aggregation level, a disposition schedule can destroy a single record or an entire folder at one time. When the aggregation level is a container, the action affects all of the entities at that level or below.

alternate retention
An alternate retention period applied to entities that meet specified conditions. In IBM Enterprise Records, multiple alternate retentions can be defined in the same disposition phase. For example, if records are kept in multiple countries, each country might have different laws regarding retention. Records can be retained in each country using a retention interval based on a country property.
See also disposition schedule and disposition phase.

auto destroy
Permanently deletes or destroys records without the use of a workflow. The record removal is immediate when it has reached the end of the retention schedule.

B

box
A container that provides a mechanism to model physical entities that contain other physical entities. Derives from the PhysicalContainer class.
See PhysicalContainer.

C

catalog
When declaring a record, the step in which the record class and file plan location are specified.

charge-out
In physical records management, the checking out of a physical record from its home location. This action is handled by the Physical Record Management (PRM) workflow.

charge-in
In physical records management, the checking in of a physical record to its home location.
See also charge-out.

classification guide
Security classification guides (SCG) are available only in a DoD Classified data model. Persons with Original Classification Authority can delegate the authority to classify information by creating guidelines to be used by authorized derivative classifiers. Only users assigned to the Classification Guide Administrator security role can create or modify security classification guides.

classified
When using the DoD Classified data model, a record can be defined as a classified record upon declaration. Classified records have special access restrictions in addition to normal record security.

compliance
Acting in accordance with certain accepted standards, laws, and guidelines.

conditional hold
See dynamic hold.

container
An IBM FileNet P8 folder. In IBM Enterprise Records, a container can be a folder, category, box, volume, or hybrid folder. All of these containers are subclasses of the RM Folder class, which is a subclass of Folder.
See folder.

cutoff
The event that signifies the end of the active period of an entity and the start of disposition.

Cut Off workflow
A workflow that is launched by the cutoff event. The purpose of the Cut Off workflow is to ensure that the records manager reviews the entity after the cutoff trigger and approves the cutoff date. The different phases of the disposition schedule start only after approval of the cutoff date.

D

data model
A template for a file plan object store, to be compliant with certain records management standards. The data model can include metadata and security features. When a new file plan object store is created, a data model must be chosen. Four data models are available:
Base: Satisfies the requirements of most corporations.
Department of Defense (DoD): Includes the properties required by version 2 of the DoD standard (DoD 5015.2)
Department of Defense Classified (DoD Classified): Includes the properties required by version 2 of the DoD Classified standard (DoD 5015.2) for managing classified records
Public Records Office (PRO): Includes the properties required by the PRO 2002 standard.

declare
The act of creating a record object. Declaration and cataloging happen simultaneously. Declaration can be manual or automatic.

declassification review sweep
See sweep processes.

default retention
The phase retention period that applies if either no alternate retentions are specified or if the entity does not meet any alternate retention conditions.

destruction
The removal of the record and the object of the record from the system. For electronic documents, both the record object and the document object are deleted. For physical objects, the record object is deleted. Optionally, the metadata of destroyed records can be retained after the record itself is destroyed, providing a record of the destruction of the record.

discovery
In law, the pretrial phase in a lawsuit in which each party can request documents and other evidence from other parties or compel the production of documents and other evidence using the legal system.

disposal phase
A part of a disposition schedule that controls the retention of entities in a particular state for a specified time period and the disposition action that is performed at the end of the retention period. Also called a phase or a disposition phase. Each phase has a phase retention period and a phase action.

disposition phase
See disposal phase.

disposition
Actions performed on a record after cutoff. Disposition is applied through disposition schedules that are created in IBM Enterprise Records and associated with containers. Disposition includes one or more disposal phases. Each phase has a phase retention period and a disposition action that occurs at the end of that retention period.

disposition action
An action performed on entities after the cutoff is reached or when their retention period in a disposal phase is over. For vital records, it is a periodic review. Disposition actions are created in IBM Enterprise Records. Each action is associated with a workflow. Some examples of actions include Destroy, Review, Export, Transfer, and Vital Review. Actions need to be initiated manually when the retention period of the phase is over. Each phase has an associated disposition action. Each disposition action (except auto destroy) is associated with a disposition workflow. Also called phase action.

disposition hold
A temporary suspension of disposition processing. A hold can be created and then applied to an entity or group of entities. Each hold is for a specific use and can be applied to several entities at one time. In addition, an entity can be placed on several holds at the same time.

disposition schedule
Disposition instructions that specify how long to keep the entity and how to dispose of it. In IBM Enterprise Records, a disposition schedule has one or more disposition phases. Disposition schedules are created in IBM Enterprise Records and associated with containers. The disposition schedule is inherited by all contained elements within the container, but applies only to the entity type specified by the aggregation.

disposition sweep
See sweep processes.

disposition workflow
A workflow that is associated with a disposition action that automates that part of the disposition process. IBM Enterprise Records comes with several workflows. Examples of disposition workflows include Destroy, Export, and Interim Transfer.
See also disposition action.

document
An object saved in an object store that has properties and security and can additionally have content, versions, lifecycles, and subscriptions. Documents are instances of the Document class or one of its subclasses.

dynamic hold
Refers to the ability to specify conditions for entities to be placed on hold. A scheduled Hold Sweep process determines if any entities meet the conditions of the holds. If so, the hold is applied automatically. Also called Conditional hold.

E

electronic record folder
A folder used for declaring records having electronic data.

entity
A generic term that can apply to a record object or an IBM Enterprise Records container.

event
In IBM FileNet Content Engine, a change in the metadata that, when specified in an event subscription, initiates an event action. For example, an event can be the addition of a document to a folder. The event action might be to declare that document as a record. In IBM Enterprise Records, an event is used to trigger the start of the disposition process or, in the case of vital record review, to trigger the vital review action.
See also event action, event subscription, and event trigger.

event action
In IBM FileNet Content Engine, a script or workflow that the Content Engine runs, as defined in a subscription. Event actions can be used to launch workflows and to declare records.

event subscription
In IBM FileNet Content Engine, a definition of conditions required to initiate an event action. An event subscription specifies the class to which the subscription applies, the event that must occur (such as adding a document or changing a property value), and the event action that is triggered.
See also event action.

event trigger
In IBM Enterprise Records, an event that triggers the start of the disposition process. Each event trigger has a condition. When an event occurs that meets the condition, Disposition Sweep marks the entity as being ready for disposition. Several types of event triggers can be configured in IBM Enterprise Records: internal events, external events, recurring events, and predefined date events. In addition, a calendar date in the disposition schedule can be defined to be the cutoff trigger. Also called a trigger, cutoff trigger, or disposal trigger.

external event
An event that occurs outside the system, but that can directly impact the cutoff and disposition of entities. For example, a change in administration might delay disposing of unnecessary or old records. External event triggers are similar to predefined date events, except that the date field is not a required property, which means that the trigger can be created without knowing the future date of the event.

F

file plan
In IBM Enterprise Records, a container hierarchy that defines the organization of records. The file plan also determines the security and disposition of contained entities. Entities can inherit security and disposition from the parent container in the file plan.

file plan object store (FPOS)
An object store that hosts a file plan. The administrator must create an FPOS by importing the appropriate data models and performing other configurations. After the FPOS is configured, the records manager can create the file plan on it.

FPOS
See file plan object store.

folder
In IBM FileNet Content Engine, an object that can contain other objects. In IBM Enterprise Records, a container that contains record volumes. *See also* volume.

H

hold
See disposition hold.

hold sweep
See sweep processes.

I

IBM Enterprise Records
An add-on product to the FileNet P8 system that has special record management capabilities. A records management application (RMA) as defined in the DoD standard 5015.2.

interim transfer
Temporarily transfers records to some other location. The original record remains in the IBM Enterprise Records system until final disposition occurs.

interim transfer workflow
A workflow that ensures that the home location of a physical entity and location of an electronic entity are changed to the specified location at the end of the retention period of a phase. The records manager must approve the interim transfer of each entity. Before approving the interim transfer of a physical entity, the records manager must ensure that the physical entity has been manually transferred to the new location.

internal event
An event trigger that refers to a change in the metadata of an entity. These events are triggered automatically when the specified condition is fulfilled. For example, the system can track when a volume closes and trigger cut off and disposition at that time. An internal event acts on the type of entity specified in the aggregation field.
See also event trigger.

N

naming pattern
Specifies rules used to automatically generate names when new containers are added to a file plan. For example, a container naming pattern can be used to automatically ensure that each new container has a unique category ID. Naming patterns consist of one or more pattern levels that apply to an entire level in the file plan hierarchy (for example, the tree diagram of the file plan).
See also record pattern.

O

offset
An optional time gap between the event trigger and cutoff.

P

permanent record
A record that has been identified as having sufficient historical or other value to warrant continued preservation by the organization beyond the time that it is normally required for administrative, legal, or fiscal purposes.

phase
See disposal phase.

PhysicalContainer
A container used for declaring records for physical items.

physical record
Metadata describing a physical object like paper, tapes, compact disks, and so on.

physical record folder
A container used for declaring records for physical items, such as paper records. A physical folder is a virtual entry for a paper folder.

predefined date event trigger
In IBM Enterprise Records, an external event trigger with a required date field.

R

RDOS
See record-enabled document object store.

record
A file that references and contains information about another electronic file (document) or a physical object. A record is created to place the document or physical object under corporate or governmental control. The record specifies how the document or object is to be stored, accessed, and, optionally, disposed of. A record is metadata.

record-enabled document object store (RDOS)
An object store that has been configured to allow record declaration. Electronic documents on an object store that is not configured as an RDOS cannot be declared as records.
Note: Do not confuse the RDOS and the FPOS. In ecm_help and in the *IBM Enterprise Records Installation and Upgrade* guide, RDOS is called ROS. For the IBM Enterprise Records courseware, the word *document* was added to emphasize the distinction between the RDOS, in which documents are stored, and the FPOS, in which record objects are stored.

record pattern
Used to constrain the names of new records to a pattern that is associated with the container. It is similar to a naming pattern except that it does not generate names, only constrains them. Users must be careful when adding records to a container with a record pattern because the pattern does not allow declaration if the record name is not compliant with the pattern. Care must be exercised when using record patterns with automated declaration.
See also naming pattern.

records manager
An IBM Enterprise Records security role, the duties of which include setting up the file plan, triggers, and disposition schedules. Sometimes referred to as a records management professional, or records officer.

records management system
Any system for managing records. In the IBM Enterprise Records courses, a records manager system includes the file plan, disposition schedules, naming patterns, record classes and properties, locations, workflows, and anything else that can be created for records management.

records administrator
An IBM Enterprise Records security role, the duties of which include setting up security, object stores, document and record classes, and metadata.

records reviewer
An IBM Enterprise Records security role (in the PRO data model), the duties of which include reviewing entities that are ready for disposition, declaring records, and performing basic record-related operations, such as filing or copying records. In the DoD and Base data models, this person is called a Privileged User.

records user
A IBM Enterprise Records security role, the duties of which include declaring and viewing records.

retention period
At a high level, how long to keep a record. In IBM Enterprise Records, a part of a disposition phase that specifies the length of time between cutoff and the phase action. A disposition schedule can have several phases of retention, each with its own retention period. Total retention time is equal to the retention period of the final phase of disposition. The retention period is always relative to cutoff, not to the end of a prior phase. For example, if a review phase is set for one year after cutoff and the second phase is set for a year after the review, then the phase retention period for the second phase is two years (after cutoff).

retention schedule
See disposition schedule.

record types
A categorization of records that has a unique disposition schedule. Record types are used when a group of records existing in a record container needs to have a disposition schedule that is different from the one currently associated with the container. Usually, record types are used when some records must be destroyed before the rest of the records in the container. If a record type has a longer retention than other records in the container, the container is placed on hold until all the records are ready for disposition.

recurring event
Events that recur automatically after a specified time interval. They are used to trigger periodic reviews of vital records. For example, a recurring event called Monthly review with a specified frequency of one month can be created to cause a monthly review of the associated entity.
See also Vital records.

ROS
See record-enabled document object store.

S

screening workflow
A workflow that prompts a reviewer to decide if the disposition of an entity should proceed before executing workflows associated with its disposition phase. Screening is optional and is specified when a disposition phase is created.

spoliation
The willful or accidental destruction of a record prior to its scheduled destruction.

sweep processes
Daemon processes that are scheduled to run at appropriate times in the business day. Sweeps carry out automatic operations, depending on their configurations.
Disposition Sweep calculates disposition-related properties, launches the Vital Review workflow,

and launches the Cut Off workflow where applicable. Disposition Sweep can optionally be configured to perform the auto destroy action.

Hold Sweep finds entities that satisfy the conditions for dynamic holds and applies the hold to those entities.

Declassification review sweep applies only to classified records for which the Declassify On Date or Declassify On Event values are not specified. IBM Enterprise Records uses the Default Declassification Timeframe to declassify these records.

sweep profile

A customized configuration for a sweep process that is saved as a separate file. Multiple sweep profiles provide a way to run sweep processes using different configuration settings without having to reconfigure the sweep process each time.

T

transfer

The act or process of moving records from one location to another, especially from the location the record is used to offsite storage facilities or NARA (National Archives and Records Administration).

transfer mapping file

An XML file that maps IBM FileNet Content Engine property names to XML property names. IBM Enterprise Records Transfer tool includes this file when importing or exporting IBM Enterprise Records entities. When you transfer records and record folders while they are still active, the transfer mapping capability tracks the entities by the organizations receiving and originating the entities.

trigger

See event trigger.

V

vital records

Records that are deemed by an organization as important enough to require periodic review. Whenever a recurring review event occurs, the vital records review workflow associated with the event is launched.

volume

A volume (also record volume) serves as a logical subdivision of a record folder. A folder can contain one or any number of volumes. A volume has no existence independent of the folder. A volume cannot contain a subfolder or another volume.

W

workflow

A business process to accomplish a task. In IBM FileNet BPM (Business Process Management), workflows are automated managed by the IBM FileNet Process Engine. IBM Enterprise Records includes several workflow definitions for performing records management tasks, including the following: screening, cutoff, and disposition actions.

workflow definition

An electronic representation of the activities and resources required to accomplish a business process. The workflow definition acts as a processing template that the IBM FileNet Process Engine uses each time the workflow runs, routing the work to the specified participants, along with data, attachments, and other information needed to complete the activities.

Z

ZeroClick

Describes the ability to automatically declare records without user involvement. Example: a document is declared as a record automatically when it is added to an IBM FileNet Content Engine folder. A record can also be declared as part of a workflow. IBM Content Collector can direct IBM Enterprise Records to declare e-mail messages as records automatically.

IBM®