DB2 IBM DB2 Records Manager

IBM

Version 4 Release 1 Revision 1

Tutorial Guide

DB2 IBM DB2 Records Manager

IBM

Version 4 Release 1 Revision 1

Tutorial Guide

Notices:

Before using this information, be sure to read the general information under Notices in the back of this book.

First Edition (November 2004)

Before using this information, be sure to read the general information under Notices in the back of this book.

This manual was produced using IBMIDDoc SGML, the Epic editor, and processed for print and online using the ID Workbench.

Contents

Preface

This tutorial guide will help you learn some of the new concepts in IBM DB2® Records Manager.

This chapter is organized as follows:

Who should use this guide

Use this guide if you require guidance for reporting and some new concepts necessary to Administer IBM® DB2 Records Manager.

How to use this guide

This guide uses the following conventions:

bold	Identifies commands, flags, keywords, files, directories, and other items whose names are predefined by the system.
italics	Identifies parameters with actual names or values that you must supply.
`monospace`	Identifies examples of specific data values, examples, of text similar to what you might see displayed, examples of portions of program code similar to what you might write, messages from the system, or information you should actually type.

Ensure that you examine the IBM DB2 Records Manager readme file for additional information. You can find this readme file in the location *install-directory/readme.txt*, where *install-directory* is the directory where you installed IBM DB2 Records Manager.

Product Publications

You can view the following documentation from the IBM DB2 Records Manager Web site at *http://www-3.ibm.com/software/data/cm/cmgr/rm/*:

Document	Part Number
IBM DB2 Records Manager Concepts Guide	SC18-9182-00
IBM DB2 Records Manager Installation Guide	SC18-9185-00
IBM DB2 Records Manager Technical Reference	SC18-9181-00
IBM DB2 Records Administrator's Guide	SC18-9180-00
IBM DB2 Records Manager Import Export Guide	SC18-9183-00

Document	Part Number
IBM DB2 Records Manager New Features Overview	GC18-9184-00

Related Publications

The DB2 Universal Database™ publication Web site contains information related to IBM DB2 Records Manager. The DB2 Web site is located at:

http://www.ibm.com/software/data/db2/library/

Sending your comments

Your feedback helps IBM to provide quality information. Send any comments that you have about this book, or about other Records Manager documentation. You can use either of the following methods to provide comments:

- Send you comments from the Web. Visit the online Readers' Comment Form (RCF) for IBM Data Management page at:

  http://www.ibm.com/software/data/rcf
- Send your comments by e-mail to **comments@vnet.ibm.com**. Ensure that you include the name and part number of the book, if applicable. If you are commenting on specific text, include the location of the text within the documentation set (for example, a chapter and section title, a table number, a page number, or a help topic title).

Contacting IBM software support

The IBM software support Internet site provides you with self-help resources and electronic problem submission. The IBM software support home page can be found at *www.ibm.com/software/support*. The IBM DB2 Records Manager support site can be found at *http://www.ibm.com/software/data/cm/cmgr/rm*.

Voice support is available to all current contract holders via a telephone number in your country (where available). For specific country phone numbers, see to the IBM Software Support Handbook, Appendix B: Contact Information, found at *http://techsupport.services.ibm.com/guides/webhndbk.pdf*.

Chapter 1. New Feature Examples

This chapter contains information about using some of the new features in this release.

This chapter is organized as follows:

Exclusive relationships

You can configure IBM Records Manager to prevent the addition of new descendant components though exclusive relationships. This kind of relationship determines the types of components to allow in a relationship with respect to its existing members. For example, you can configure a file plan component to have any number of files (as children), however, if one of those files contains folders, that file is the end of a branch and it can contain only folders.

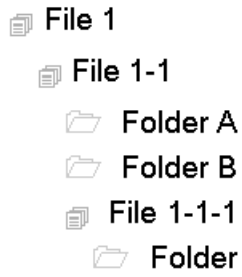
An exclusive relationship is one that is exclusive within its view. The source file plan component of an exclusive relationship (for example, a file in a file to file relationship) cannot participate in relationships other than a file after it is the parent of a file. An exclusive relationship can only exist where there are several relationships that have the same source.

You can define a hierarchical file plan view that has the following structure:

- Root to File
- File to File (Exclusive)
- File to Folder
- File to Document
- Folder to Document

Relationship List

Relationship	Source	Target
File to File	File	File
File to Folder	File	Folder
Folder to Document	Folder	Document
Root to File	Root	File



In this view, you can make a file the parent of another file, a folder, or a document. However, because the **File to File** relationship is exclusive, the following rules apply:

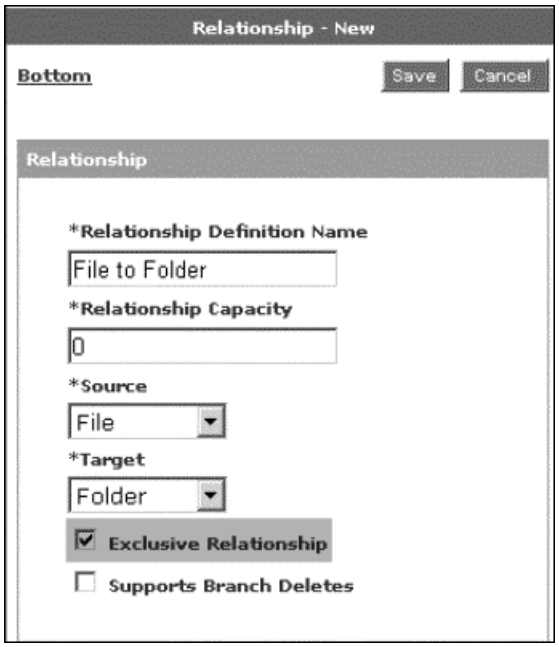
- After a file is the parent of another file, it cannot become the parent of folder or document.
- After a file is the parent of a folder or document, it cannot become the parent of a file.
- A file can be the parent of a file and folder simultaneously.

Example: Exclusive relationships

In this example, we will configure the **File to Folder** relationship to be exclusive. This means that folders cannot exist in files that contain other types of components.

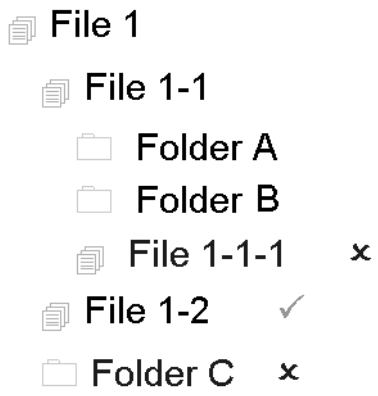
To specify exclusivity, you must set the **Exclusive Relationship** option for the **File to Folder** relationship.

Note: You must set the **Exclusive Relationship** option before creating a relationship definition (before clicking **Save**). You are not able to create an exclusive relationship when existing components break the rules of the proposed exclusive relationship.



Specifying exclusivity for the **File to Folder** relationship implies the following:

- Once a file owns a folder, it must exclusively own only folders
- Once a file owns another file, it cannot contain any folders



Note: When a component type participates as the source in exactly two relationship definitions:

- File -> File
- File -> Folder

Configuring one or the other as exclusive is equivalent to configuring both as exclusive.

Example: Enforcing classification using exclusive relationships

Exclusive relationships enforce the lowest level classification (lowest level filing) because once a Folder is created, the Folder level is fixed for the current branch of the classification scheme. By defining folder - record relationships to be exclusive, you can ensure that each folder contains records of the same type.

In the following example, specifying exclusive relationships for the **Folder to Document** and **Folder to Email** relationships does not permit the mixing of email and documents within a single folder.

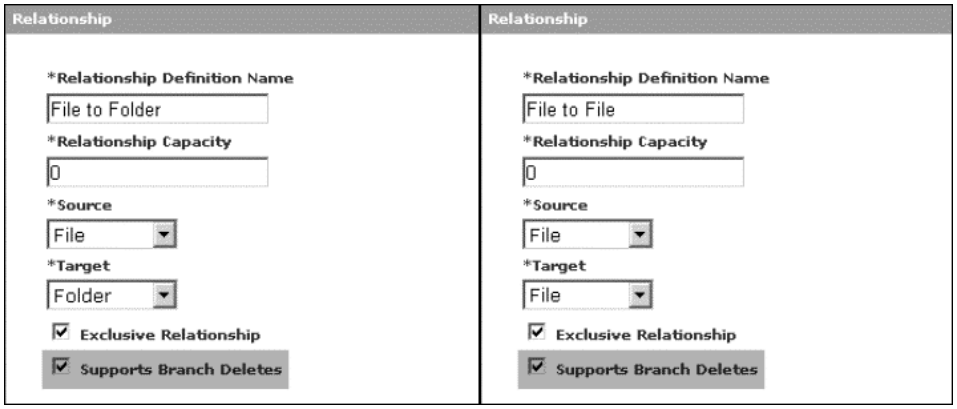
Relationship List

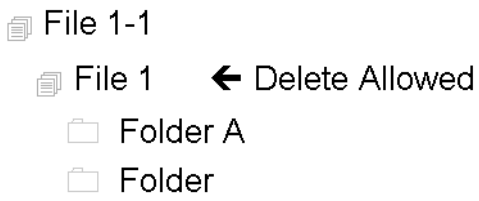
Relationship	Source	Target	Relationship Capacity	Exclusive Relationship	Actions
File to File	File	File	0	No	
File to Folder	File	Folder	0	Yes	
Folder to Document	Folder	Document	0	Yes	
Folder to Email	Folder	Email	0	Yes	
Root to File	Root	File	0	No	

Branch deletes

File plan design has been extended to allow (or disallow) the ability to delete a branch of the file plan, depending on the types of relationships in that branch.

In this example, we will configure the **File to Folder** relationship to support branch deletes. To specify branch deletes, you must set the **Supports Branch Deletes** option for the **File to Folder** relationship.





Note: To delete a component (and its descendents), all the relationships in the branch must be of types that support branch deletes.

The deletion of **File 1** is allowed only when both the **File to File** and **File to Folder** relationship definitions are configured to **Support Branch Deletes**. If the folders contain descendents, those relationships must also support branch deletes.

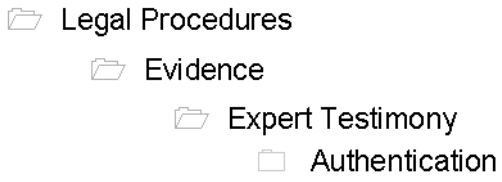
Note: Deleting a file plan component with a large number of descendents can time out. To allow the operation to complete without timing out, IRM can execute the operation offline when the number of descendants exceeds a configurable threshold. You can set this threshold by following these steps:

1. Click **Tools > System Configuration**.
2. In the **Long Operations Items Threshold**, specify a numeric value.
3. Click **Save**.

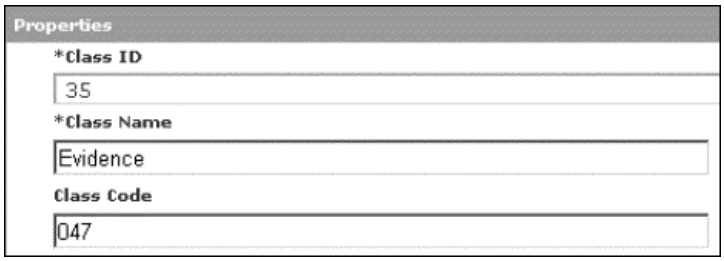
Enhanced file plan naming conventions

In IRM, it is mandatory that every component in the file plan have a name. The names in the hierarchical path cumulate to form the unique Address or Full Path of the component.

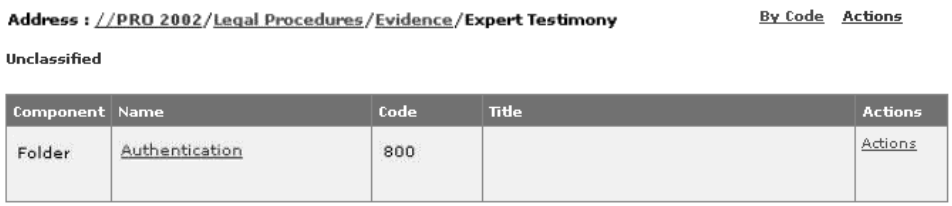
For example, for the following structure:



Where the **Class Name** and **Class Code** fields are:



You would have the following address:

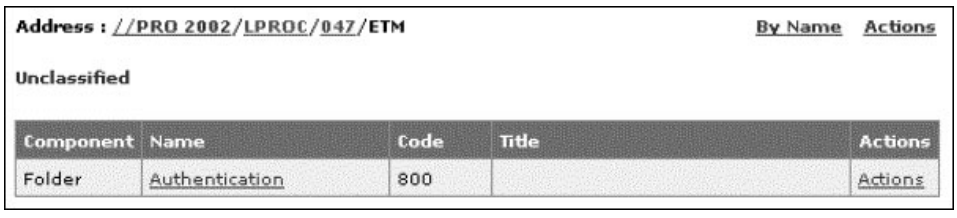


In IRM, you can express file plan component names as textual names, structured alpha-numeric characters, or a combination of both. These file plan component naming conventions let you define more meaningful names and codes for your components, and let you switch between conventions. For example:

- 1000-1-1 (alpha-numeric name)
- Administration - General - Operations (textual name)

Example

Every component can have a secondary name, the code, that also cumulates down the hierarchy to also form a unique path to the component. The **Address by Code** and **Address by Name** are interchangeable. You can easily change between view modes by selecting **By Name**, or its toggle **By Address**.



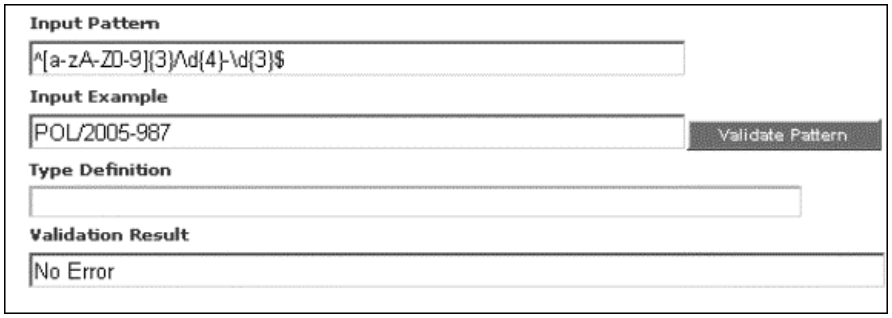
By default, the file plan component code is not mandatory. If any code in the path is missing, **Address By Code** cannot be used to uniquely identify a component.

Instead, IRM displays a placeholder in the path for each missing code. However, you can specify that it is mandatory for the code to be configured separately for each component definition.

Configurable name patterns

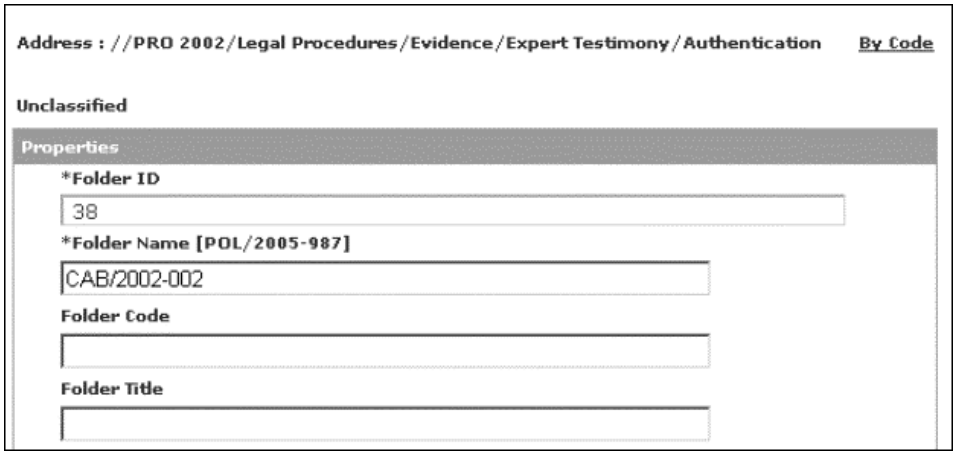
In IBM Records Manager, you can specify expressions for user input, and then validate that input to ensure that it is correct. This means that when you design forms containing text fields, you can control the naming of file plan components using special input patterns. For example, to create a text field that always begins with numeric values followed by a colon, and then alphabetic characters, you can write the expression [0-9]*:[A-Z, a-z]*. To ensure appropriate user input, include an example of this regular expression in the corresponding label for the field.

For example, to configure a naming pattern for the **Folder – Name** attribute definition, you can specify the following input pattern:



You can click **Validate Pattern** to validate your expression. After you verify your expression, ensure that the **Validate Results** field displays "No Errors". For additional information about creating expressions, see *http://www.regular-expressions.info/*.

After you specify an input pattern, the folder data entry form displays the input example in the label of the **Folder Name** text field.

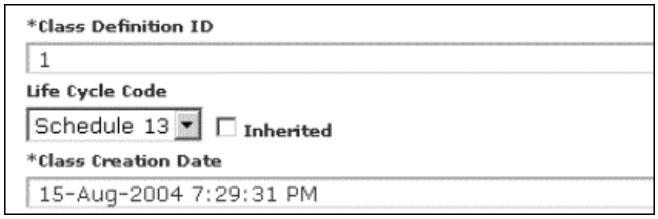


Explicit inheritance of meta data

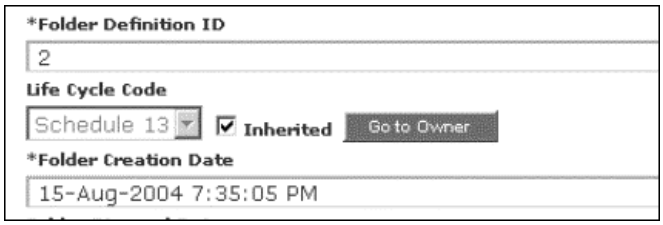
In IRM, meta data inheritance is explicit. This means that by default, when you add a new file plan component, that component automatically inherits a selection of attributes that are also inheritable (from its parent). In addition, if you modify any attributes for a component, those changes are also applied to the inheritable attributes for all of the descendants of that component. Not only are the values inherited, they are physically replicated when they are added or modified. This means that you can see them and report on them. Furthermore, user queries and reports can operate against meta data values, regardless of whether they are inherited or not.

For example, if you want to assign a Life Cycle Code to a top level class, IRM will propagate the value down to the descendants, where applicable.

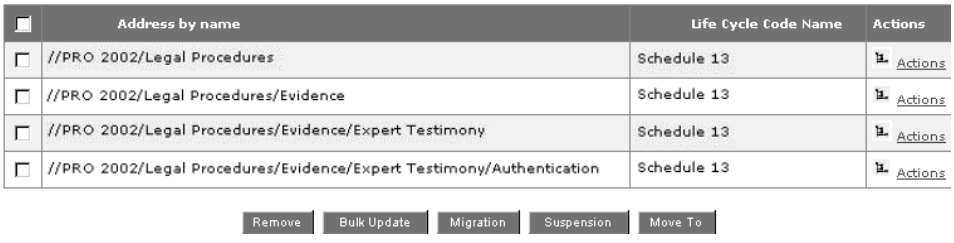
In the following illustration, the top level class is the owner of the assigned value, **Schedule 13**.



In the following illustration, the descendant folder inherits its Life Cycle Code, and the value can be overridden at any level, as required.



In the following illustration, query results correctly show the assignment of **Schedule 13** through the hierarchy.



Partitioned components

Partitions have two dates that determine the time a document spends in a folder. The **Open Date** is the date the folder was opened (this date must be less than or equal to the close date). If an open date is not specified, IBM DB2 Records Manager sets the open date to the creation date. The **Close Date** is the date the folder was closed. If the close date is not specified, IBM DB2 Records Manager sets the close date to the current system date. If you reopen a folder and add records to it, IBM DB2 Records Manager does not automatically change the close date. For example, a folder that you created on August 04 2003 into which you added documents dated June 09 2003, will have an open date of June 09 2003. Otherwise, the time span is incorrect.

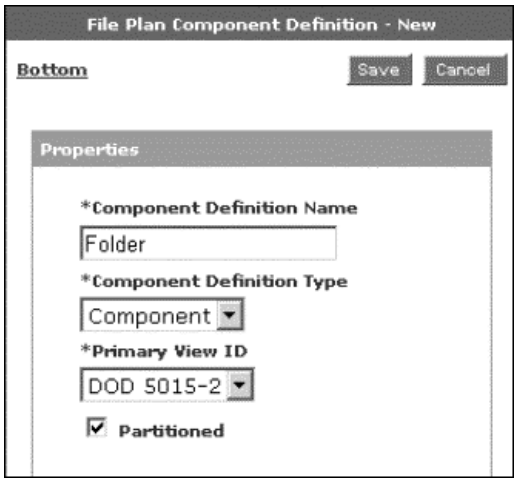
Partitions are a special type of component because they:

- Cannot exist independently from the component they partition (the container)
- Can be disposed of independently from their container
- Have a set of date fields that can be used to trigger their life cycle

Attribute Caption	Attribute Data Type
Partition ID	Integer
Partition Definition ID	Integer
Container ID	Integer
Sequence Number	Integer
Partition Name	String
Partition Code	String
Partition Creation Date	Date-Time
Partition Open Date	Date
Partition Close Date	Date
Closed	Boolean
Last Addition Date	Date
Last Review Date	Date
Last Retrieval Date	Date
Life Cycle Phase ID	Integer
Partition Disposal Date	Date

Example

In this example, we will define a component definition, called **Folder**, and select the **Partitioned** option to make it a partitioned component.



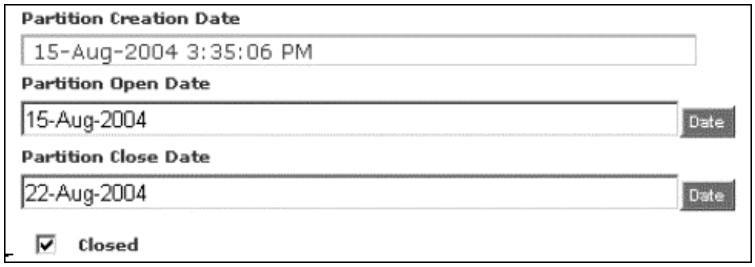
When a partitioned component is created, IRM automatically opens an initial partition and records the open date. The initial part is the current (most recent) part to which all records filed into the folder are assigned.When a new part is created, the previous part is automatically closed and its close date is also recorded. The new part becomes the current part.

The following illustration shows the query results where we can see that **Part 3** for **Folder** is the currently open part.

	Component Definition Name	Address By Name	Open Date	Close Date
☐	Folder	//PRO 2002/Legal Procedures/Evidence/Expert Testimony/Authentication	15-Aug-2004	
☐	Folder Part	//PRO 2002/Legal Procedures/Evidence/Expert Testimony/Authentication.Part 1	15-Aug-2004	22-Aug-2004
☐	Folder Part	//PRO 2002/Legal Procedures/Evidence/Expert Testimony/Authentication.Part 2	22-Aug-2004	29-Aug-2004
☐	Folder Part	//PRO 2002/Legal Procedures/Evidence/Expert Testimony/Authentication.Part 3	29-Aug-2004	

Users can file records into the **Folder** as usual, and IRM will automatically assign the records to Part 3, the current open part.

Note: Components cannot be assigned to closed partitions. Authorized users can re-open a previous partition, file records, and then subsequently close the partition without modifying the opening and closing dates. To reopen a partitioned component, clear the **Closed** option for the component.

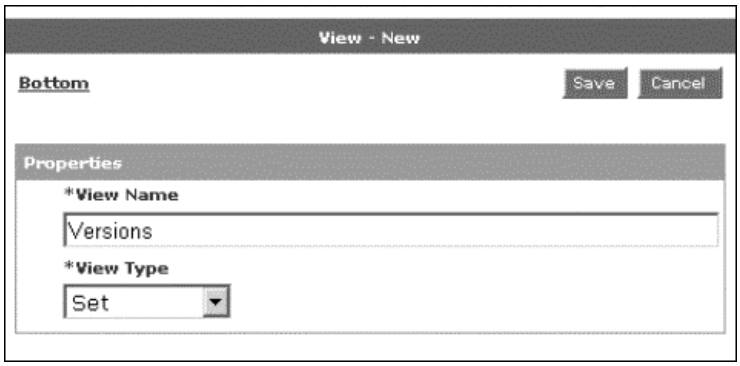


Set views

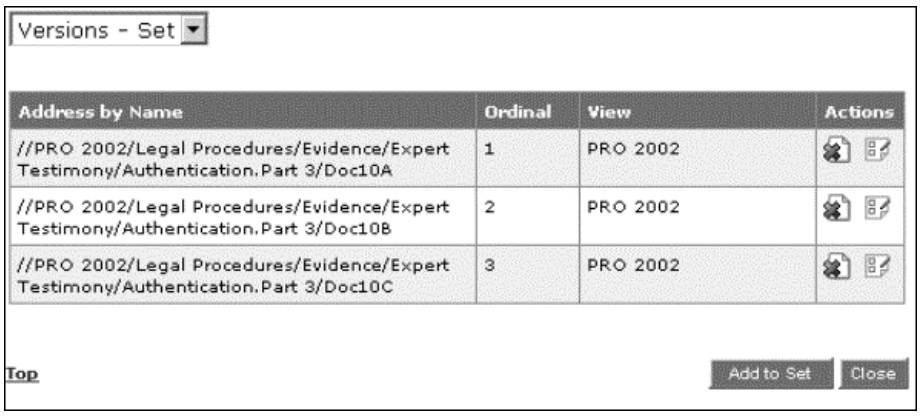
A set is a relationship that contains an ordered number of components that are members of a particular set. You can use a set to link unrelated file plan components, such as components, records, folders, and classes.

A set view is a type of relationship that contains an ordered number of components that are independent of each other. Unlike the hierarchy and link views, a set does not have a direction. Because an ordinal value describes the order of the records, sets can optionally provide you with version management capabilities for your records.

For example, you can create a new view, called Versions, that allows a group of related components to be "linked" together into a set.



You can then add components to the set, thereby creating an association between those file plan components. After you create an association between these file plan components, they are members of the same set, they are linked together, and they have an order within the set.



Chapter 2. Reporting in IRM

This chapter describes how to generate a report in IRM, create a report layout, how to customize it, how to add it to IRM, and how to assign user access to it. Although there are various applications available to help you design your report layout, this tutorial guides you through the creation process using the Style Designer from Altova StyleVision. For additional information about this application, see *http://www.xmlspy.com*.

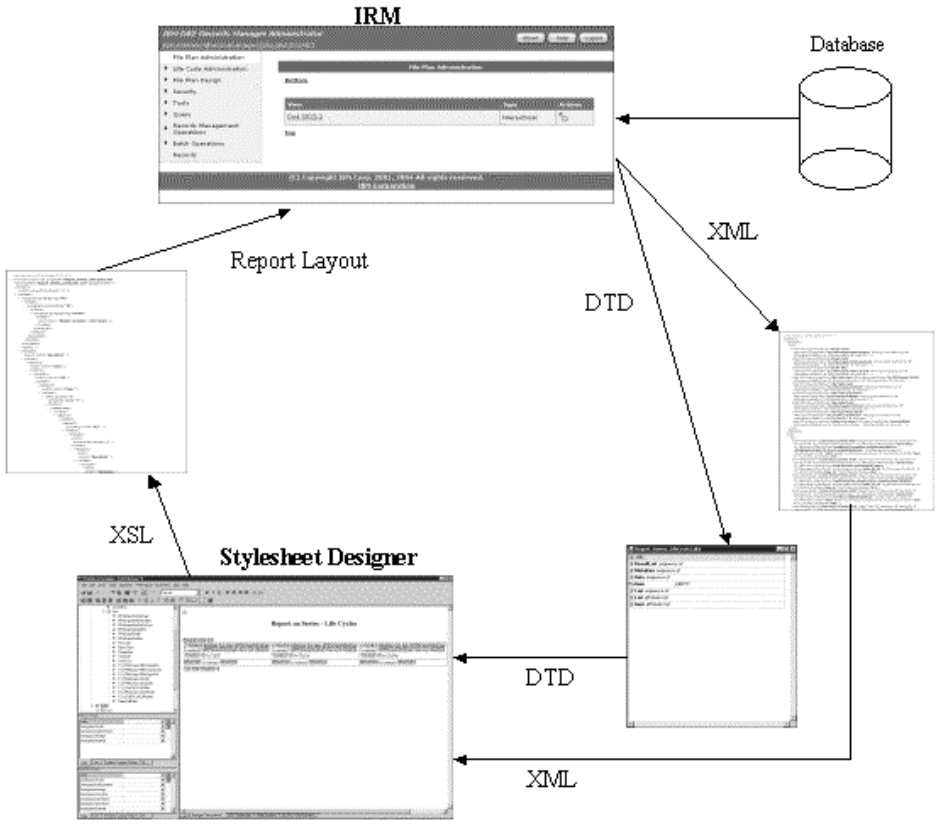
This chapter is organized as follows:

Reporting overview

In this tutorial, we will create a report layout for the **Series** file plan component in the DoD5015.2 sample file plan. In addition, we will use the Altova Stylesheet Designer application to create a report layout (an xsl-fo file whose extension is *xsl*). For additional information about Stylesheet Designer, see*http://www.xmlspy.com*.

The following illustration shows the process involved in generating a report, creating a report layout, and then incorporating that report layout in IRM. In this tutorial, we will use the IRM Engine to generate an XML and corresponding Document Type Definition (DTD) file. We will design our report layout based on the XML using Stylesheet Designer. After we finish designing the report layout, Stylesheet Designer generates the xsl-fo file which we will add to IRM.

Note: A **DTD** file contains the tags and attributes that describe the content in an XML document. It is the XML content model. **DTD** ensures that all the documentation is formatted the same way. You can import the **DTD** into a third-party tool and create custom report layouts.



The DoD 5015.2 sample file plan

To obtain the initial information in our database, we imported the sample DoD 5015.2 file plan into an empty database. In addition, we applied some Life Cycle codes and phases to some **Series** components; however, you are not required to assign any codes or phases.

For information about importing the DoD 5015.2 sample file plan, see the *IBM DB2 Records Manager Import Export Guide*.

Figure 1 shows the **Series** components in the DoD 5015.2 sample file plan.

File Plan Administration

Bottom — Up Level

Address : //Dod 5015.2

Unclassified

Component	Name	Code	Title	Actions
Series	Series 0031		COMBATANT COMMAND COMMANDER/DEPUTY COMMANDER/CHIEF OF STAFF CORRESPONDENCE	Actions
Series	Series 0105		UNIT MANNING DOCUMENTS (UMDs)	Actions
Series	Series 0106		MANPOWER AND PRODUCTIVITY ENHANCEMENT STUDIES	Actions
Series	Series 0205		PAYROLL CORRESPONDENCE	Actions
Series	Series 0216		STANDARDS OF CONDUCT	Actions
Series	Series 0220		LABOR MANAGEMENT RELATIONS RECORDS	Actions
Series	Series 0414		LAW LIBRARIES	Actions
Series	Series 0927		FOIA CONTROL	Actions
Series	Series 0942		SCIENCE ADVISOR RECORDS/ACTIVITIES	Actions

Top — Up Level

Figure 1. DoD 5015.2 sample file plan components

Figure 2 shows the relationships defined in the DoD 5015.2 sample file plan.

Relationship List

Relationship	Source	Target	Relationship Capacity	Exclusive Relationship	Actions
Folder to Doc	Folder	Document	0	No	
Folder to Email	Folder	E-Mail	0	No	
Rec Cat to Folder	Record Category	Folder	0	No	
Root to Series	Root	Series	0	No	
Series to Rec Cat	Series	Record Category	0	No	

Figure 2. DoD 5015.2 sample file plan relationships

Figure 3 and Figure 4 show two of the eleven Life cycle Codes currently defined in the DoD 5015.2 sample file plan.

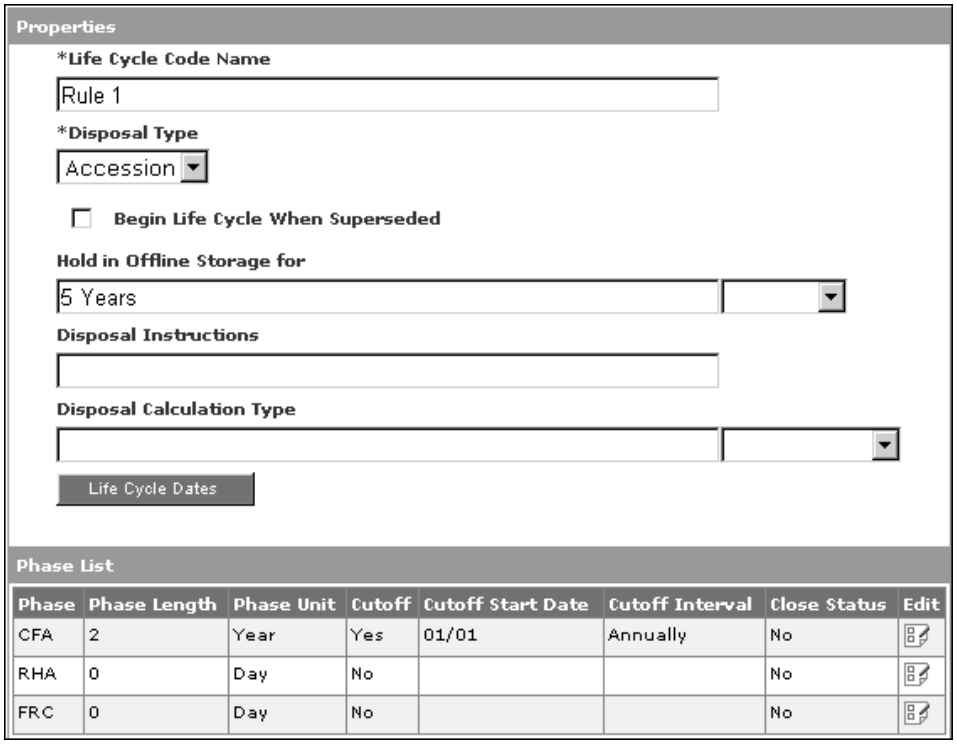


Figure 3. Life Cycle - Rule 1

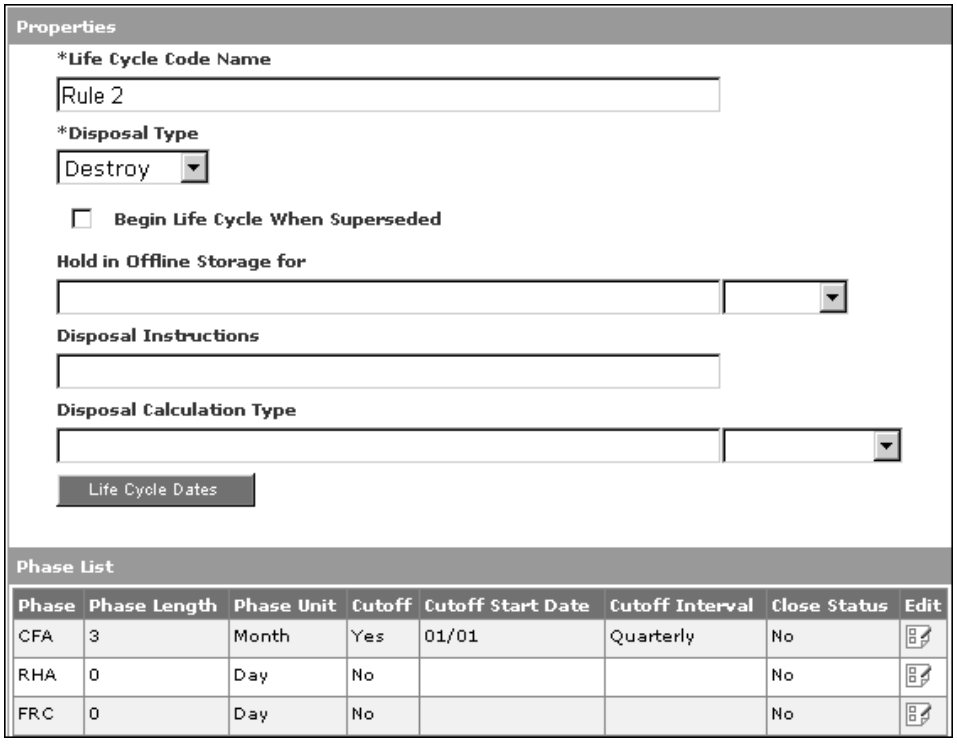


Figure 4. Life Cycle - Rule 2

Creating Report Layouts for IRM

With our sample file plan data, we can create various reports. Based on your requirements, you might want to create a report that generates results that return the number of failed logon attempts, all suspended components, or components that are waiting for disposal. Regardless of the type of report you want to create, it is beneficial for you to create a customized report layout that you can reuse.

To generate a report layout, you can follow these steps:

Prerequisites

Before you can use Stylesheet Designer, you will need to do the following:

- Install Adobe Reader
- Install Altova Stylevsion 2004 (it includes the Stylesheet Designer)
- Install J2SE Runtime 1.4.x from *http://www.java.com*
- Install and configure Apache FOP processor 0.20.5 in Stylesheet Designer

  To install the Apache FOP processor:

  1. Open **Stylesheet Designer**.
  2. Click **Help > Components download**
  3. Click **XSLT & XSL:FO Processors > Apache FOP processor 0.20.5**.
  4. Open and install **ApacheFOP.exe**.
  5. In **Stylesheet Designer**, click **Tools > XSL-FO Options**.
  6. In the **FO Processor** area, click **Browse** to navigate to the location of the file *fop.bat* for Windows, and *fop.sh* for AIX. The default location is *C:\Program Files\Altova\FOP*.
  7. Click **OK**.

Step 1: Generating XML and DTD files from IRM

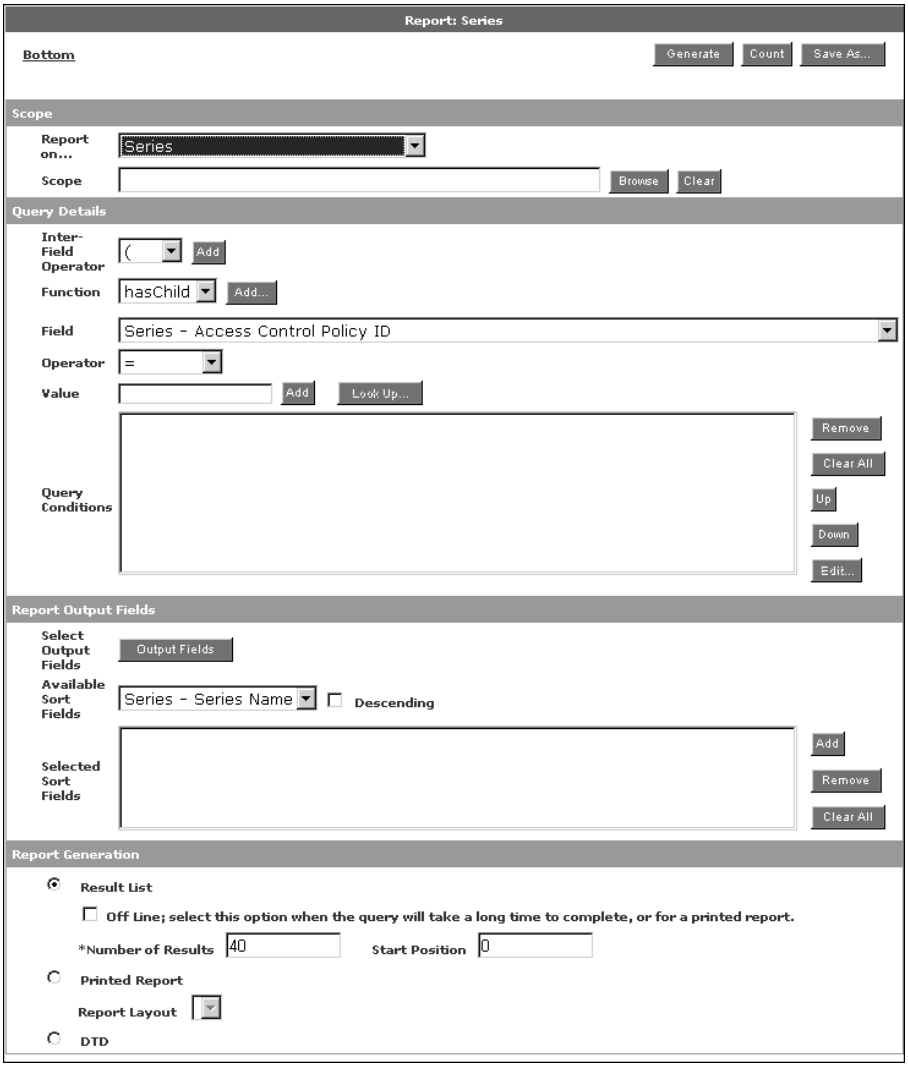
Before we can create a report layout, we will need to create a query to obtain a result list containing all of the fields for our report. For this tutorial, we will generate an advanced query that returns specific information for the Series file plan component, such a life cycle codes and phases. To create the results that we will use later to create a report layout, we begin by generating an XML result list, and a corresponding DTD file within IRM for our query.

To generate an XML file and a corresponding DTD files for the output:

1. Start and log on to IRM. Log on as the Administrator so that you have full access and permissions.
2. Click **Query > Advanced**.

On this window, you will specify what this query will report on, any specific conditions for the query, the output fields for the report, and the output format.

Note: The following illustration shows the advanced query that display when you select the Series component to report on. The fields that display are different depending on what you select in the **Report On** field.



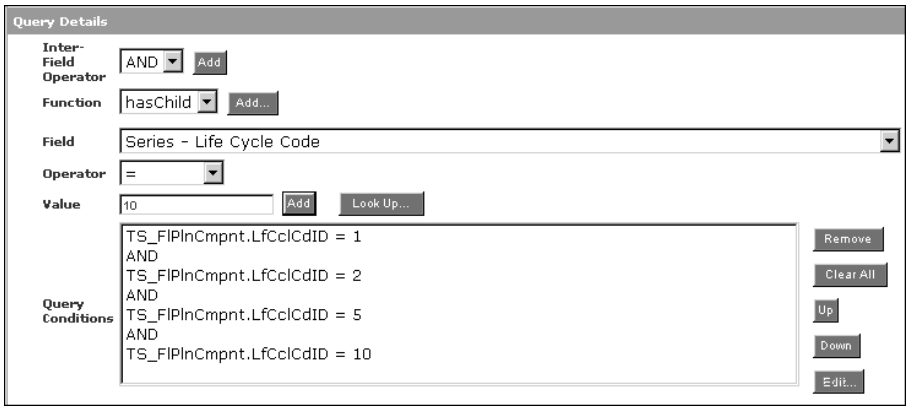
3. In the **Scope** area, select a component from the **Report on** list. The component type you specify determines what you want to report on for this query. For this tutorial, **Series** was selected.

   For larger file plans, you might want to narrow the query results by specifying a scope in the **Scope** box.
4. In the **Query Details** area, specify the query criteria that you want to include in your report. Ensure that you select an operator between each of the conditions.

   Note: For this tutorial, we used the default data from the DoD 5015.2 sample file plan, and then we assigned some Life Cycle codes (rules) and phases to the **Series** components, for illustrative purposes. We will specify that we only want to report on **Series** components whose Life Cycle Codes are Rule 1 (ID

is 1), Rule 2 (IS is 2), Rule 5 (ID is 5), and Rule 10 (ID is 10). However, for simplicity, you can choose to not specify any query conditions to return all of the **Series** components.

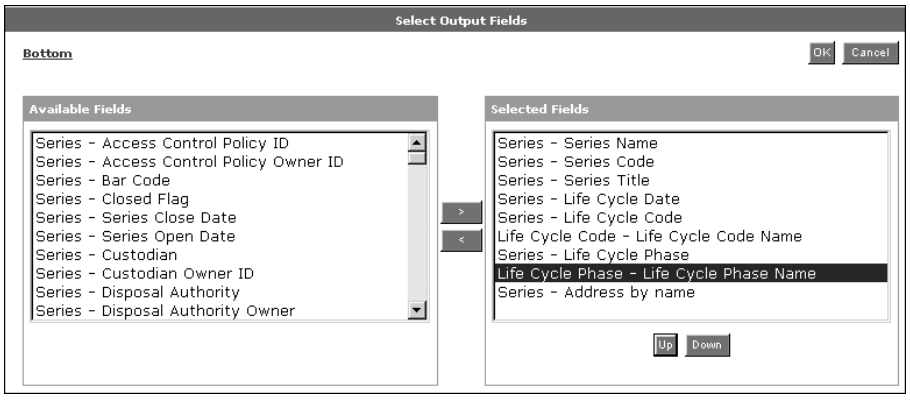
The following illustration shows the resulting expression created for this query.



5. In the **Report Output Fields** area, click the **Output Fields** button opposite the **Select Output Fields** label.
6. In the **Available Fields** box, select the fields that you want to appear in the report and click the ">" button to add them to the **Selected Fields** box.

   Note: Enlarge the **Select Output Fields** window to see the full name of the fields in the list.

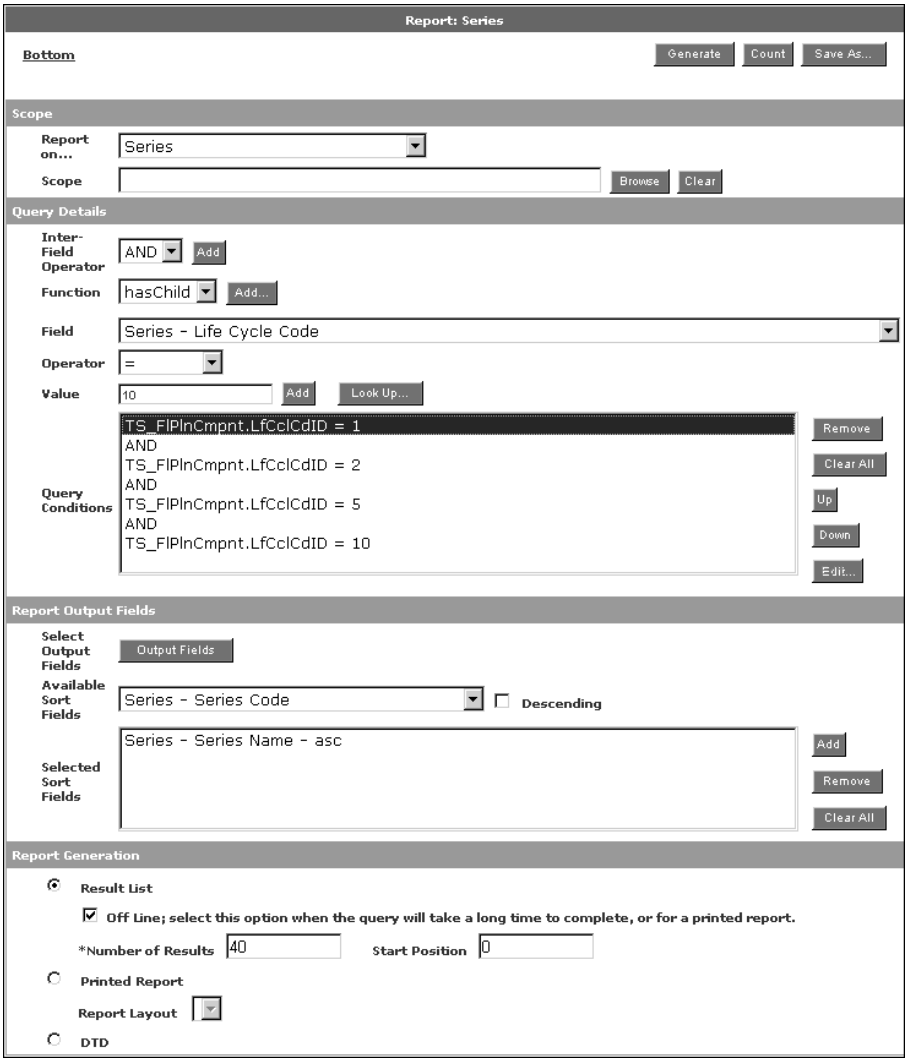
   For this tutorial, the following illustration shows the fields that were selected:



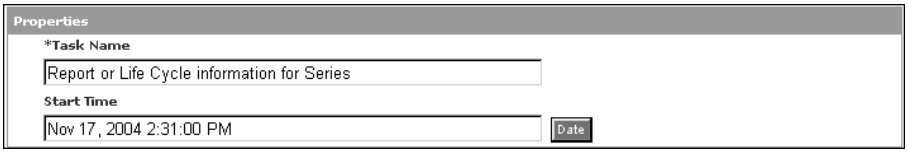
   Note: Use the **Up** and **Down** buttons to change the order of these fields in the report.
7. Click **OK**.
8. In the **Sort Available Sort Fields** list, select the items that you want to sort on. By default, sorting is in ascending order. To sort in descending order select the **Descending** option. to sort on this field. Click **Add** to add that sort item to the **Selected Sort Fields** list. The order of the items in the **Selected Sort Fields List** lists determines the sort order for the query results.

   For this tutorial, we sorted the results on the **Series Name** in ascending order.
9. In the **Report Generation** area, select **Result List**.
10. Select **Off-Line**.

    The following illustration shows the options you selected for the **Series** query.



11. Click **Generate**.
12. In the **Task Name** box, type a name for this query task, and specify the date and time you want IRM to start processing this query task.



13. Click **Start and Open Status Page**.
14. Click **Refresh** opposite the **Task** field until the status of the query task completes.

Task Name	Status	Start Time	% Done	Status Time	Actions
Report or Life Cycle information for Series	Completed	Nov 17, 2004 2:31:00 PM	100	Nov 17, 2004 2:32:22 PM	

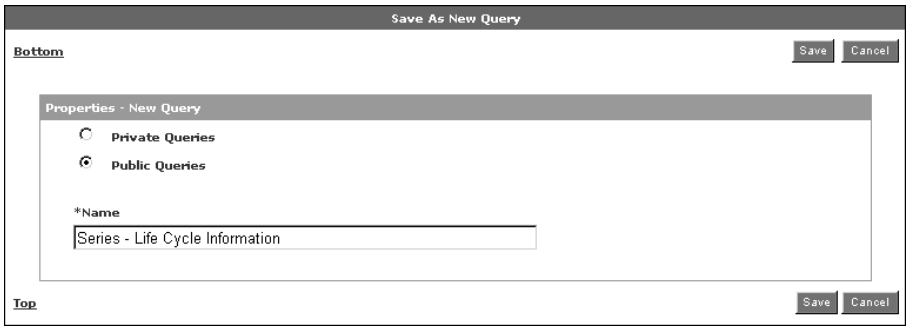
15. After the task completes, you can click the **View Results** icon in the Actions column to see the results of the query.
16. On the **Batch Process** window, click the **Export Result as XML** icon in the **Actions** column for this specific task.

17. In the **Export Result as XML** window, specify a name for the task in the **Task Name** box, and specify a date and time. To start this task immediately, do not specify andy date or time.
18. Click **Start and Open Status Page**.
19. Click **Refresh** until the status of the Export Result as XML task completes.

20. Click the **Download XML** icon in the **Actions** column for this specific task.
21. Click **Save**.
22. Save this results file to a working directory, and rename the file to a more meaningful name. For example, *Report_Series_LifeCycles1.xml*.
23. Click **Open** to see the XML results of the export process, and then close the window.
24. Click **Close**.

    Now, you want to create a corresponding DTD file.
25. On the same **Query Report** window, in the **Report Generation** area, select **DTD**.
26. Click **Generate**.
27. Save this results file to a working directory, and rename the file to a more meaningful name. For example, *Report_Series_LifeCycles1.dtd*.
28. If you want to make this query available for later use, on the **Report Query** window, click **Save As**.

    You can specify a descriptive name for the query, and indicate if the query is privately or publicly available.



29. Select **Public Queries**, specify a descriptive name for this query, and click **Save**.

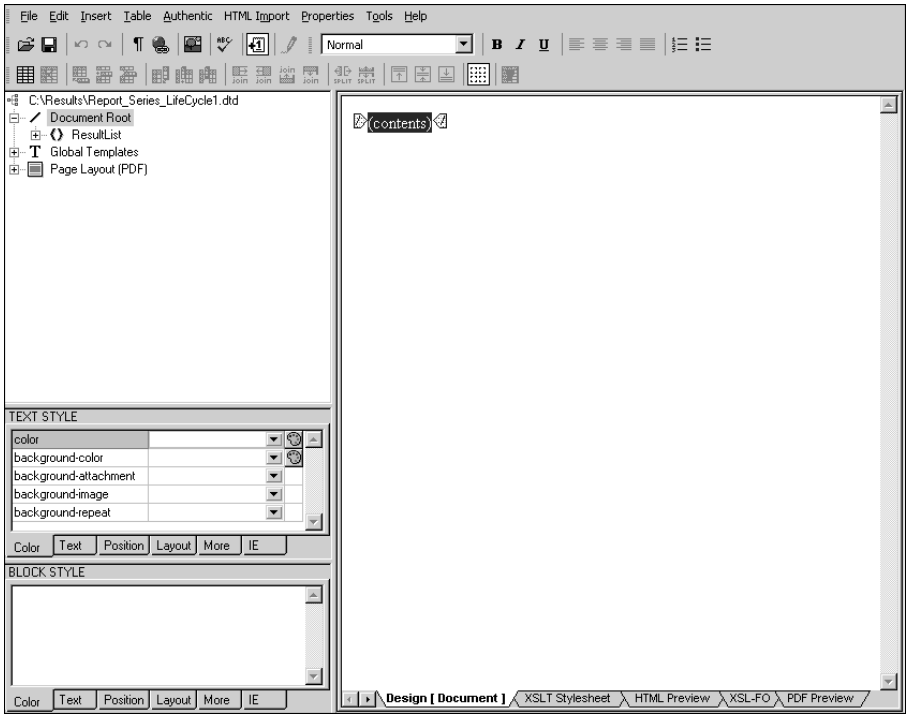
Step 2: Creating a report layout in Stylesheet Designer

Although there are various applications available to help you design your report layout, this tutorial guides you through the creation process using the Style Designer from Altova StyleVision. For additional information about this application, see *http://www.xmlspy.com*. You want to create a report layout from the XML query results output file (*Report_Series_LifeCycles1.xml*) and DTD query results output file (*Report_Series_LifeCycles1.dtd*) using Stylesheet Designer.

To create a report layout:

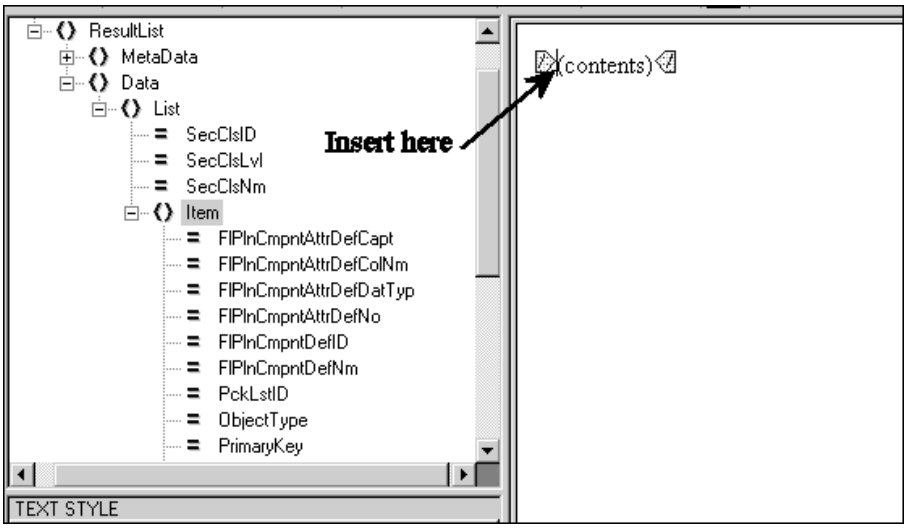
1. Open Stylesheet Designer. For Windows, click **Start > Programs > STYLEVISION 2004 > Stylesheet Designer**.
2. Click **File > Open**.

3. Change **File of type** to **Schema files (*.xsd, *.dtd, *.xdr)**.
4. Navigate to the location where you saved *Report_Series_LifeCycles1.dtd* and click **Open**.

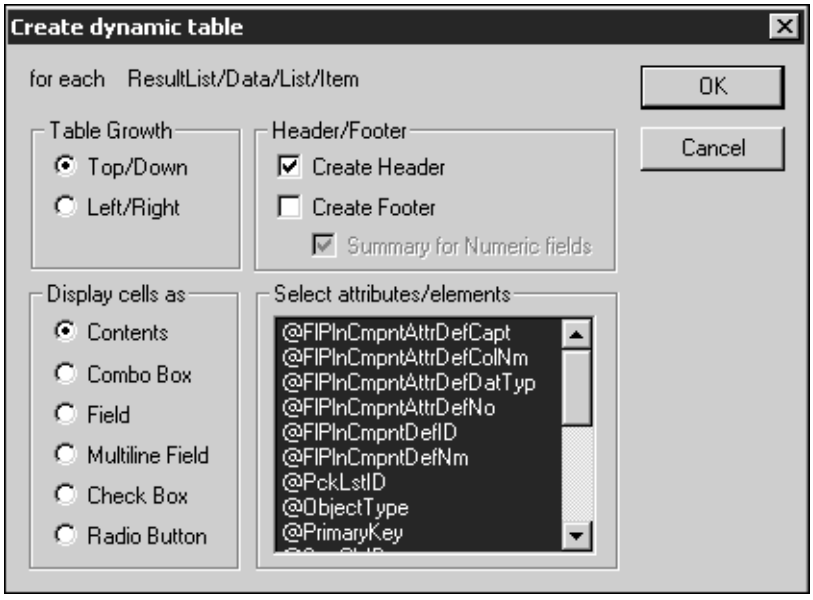


5. Click **File > Assign Working XML File**.
6. Navigate to the location where you saved *Report_Series_LifeCycles1.xml* and click **Open**.

   Next, you will expand the schema tree.
7. Expand **ResultList**.
8. Expand **Data**.
9. Expand **List**.
10. Expand **Item**.
11. In the schema tree, select **Item** and drag the entire node to **(contents)**.



12. Select **Create Table**.



All of the items in the **Select attributes/elements** list are currently selected.

13. In the **Create dynamic table** dialog, from the **Select attributes/elements** area, ensure that the list contains only those items that you want to appear as output fields in the report layout. For this tutorial, ensure that only the following items are selected:
    - @TS_FIPlncmpnt.FIPlnCmpntName
    - @TS_LfCcICd.LfCcICdNm
    - @TS_LfCcICd.LfCcIPhsNm
14. Click **OK**.

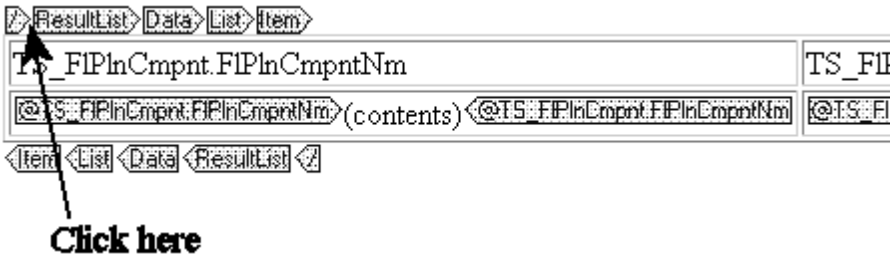
    The following illustration shows a portion of the report layout structure:

15. To preview the results, click the **PDF Preview** tab (bottom). If you receive an FOP error, see the instructions about installing the Apache FOP processor in "Prerequisites" on page 15. Click the **Design [Document]** tab to return to your design view.

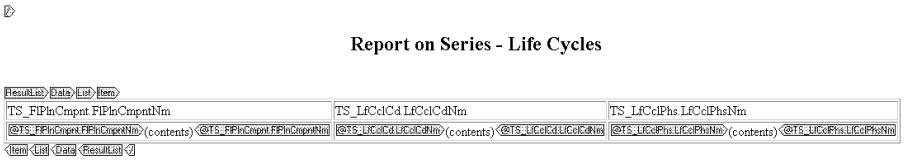
Step 3: Adding and formatting a title for the report layout

To add a title to the report and customize it:

1. Click the **Design [Document]** tab (bottom)
2. Position the cursor between the document start tag and **ResultList** start tag.



3. Type **Report on Series - Life Cycles**.
4. Press ENTER to add a line break.
5. Select the entire title **Report on Series - Life Cycles**.
6. Click **Insert > Format**, and then select **Block (div)**. Selecting this option prevents the generation of multiple titles in the PDF file.
7. Click **Insert > Format**, and then select **Heading 2 (h2)**.
8. Click **Insert > Format**, and then select **Center**.



9. Navigate to the bottom of the window and click **PDF Preview**.

   The following illustration shows the PDF preview of the report. The titles of the columns that display in the report are the default system names for the items. To specify the display name used in IRM, see "Step 4: Adding additional customization to the report layout using Stylesheet Designer (Optional)."

Report on Series - Life Cycles

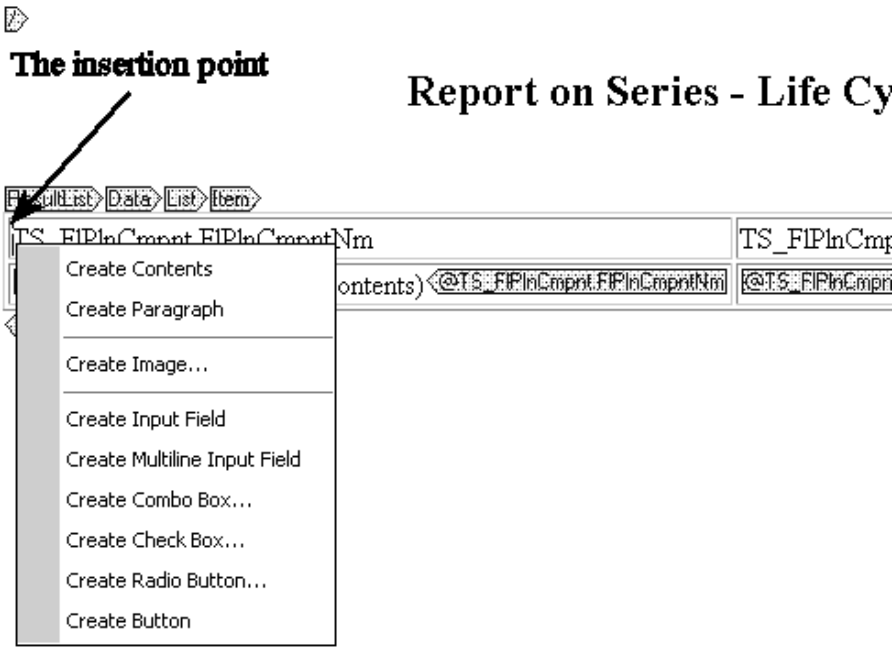
TS_FlPlnCmpnt.FlPlnCmpntNm	TS_LfCclCd.LfCclCdNm	TS_LfCclPhs.LfCclPhsNm
Series 0031	Rule 1	CFA
Series 0105	Rule 10	RHA
Series 0106	Rule 10	CFA
Series 0205	Rule 5	FRC
Series 0216	Rule 1	CFA
Series 0220	Rule 2	FRC
Series 0414	Rule 1	CFA
Series 0927	Rule 1	RHA
Series 0942	Rule 1	CFA

Step 4: Adding additional customization to the report layout using Stylesheet Designer (Optional)

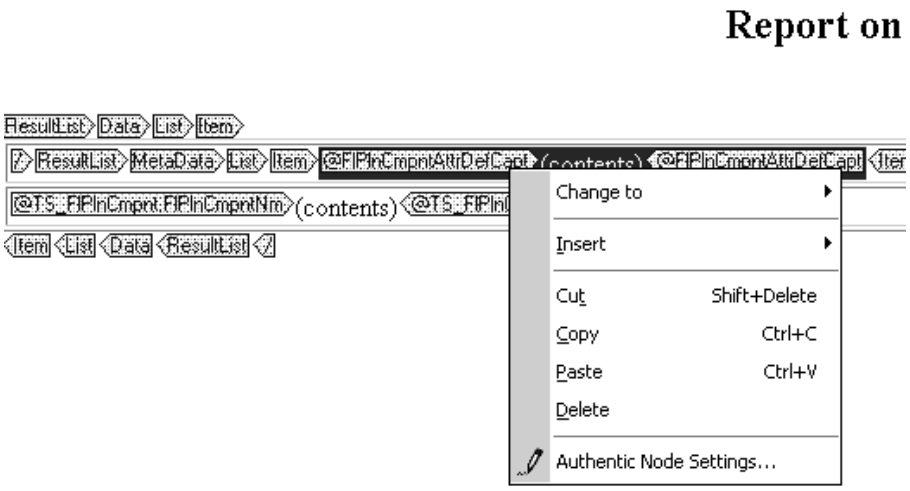
You might want to change the default heading in the table to use the Attribute Caption specified during File Plan Design. For example, by default, the Attribute Caption for column **TS_FlPlnCmpnt.FlPlnCmpntNm** is "Series Name" described in the m eta data element in the output file *Report_Series_LifeCycle1.xml*.

To change the default heading title for columns in the table:

1. Click the **Design [Document]** tab (bottom).
   Next, you will expand the schema tree.
2. Expand **ResultList**.
3. Expand **MetaData**.
4. Expand **List**.
5. Expand **Item**.
6. In the schema tree, select **FlPlnCmpntAttrDefCap** and drag it to the header of the first column, before **TS_FlPlnCmpnt.FlPlnCmpntNm**.



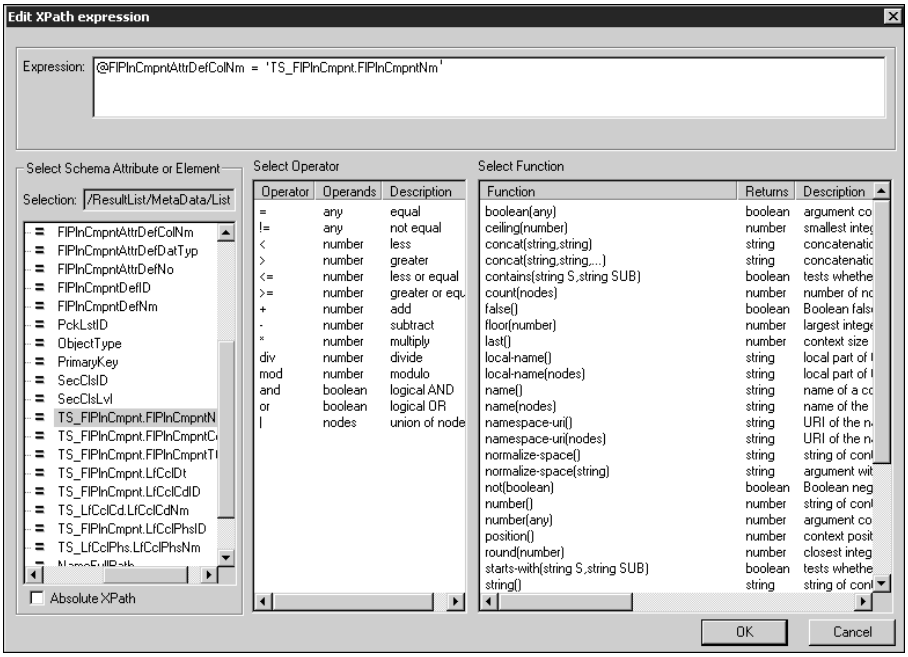
7. Click **Create Contents**.
8. Right Click on **@FlPlnCmpntAttrDefCapt** and select **Insert-> Condition**.



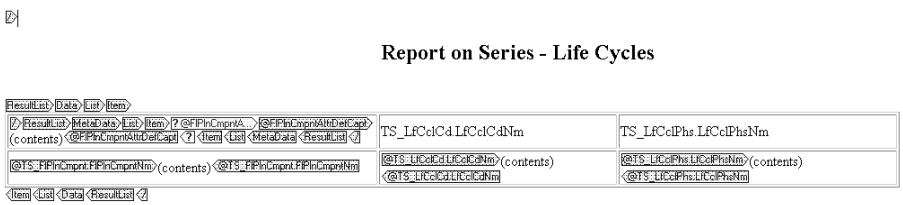
9. In the **Select Schema Attribute or Element** box, double click **FlPlnCmpntAttrDefColNm** to add it to the **Expression** box.
10. In the **Select Operator** box, double click the equal sign to add it to the **Expression** box.

11. In the **Select Schema Attribute or Element** box, double click **@TS_FlPlnCmpnt.FlPlnCmpntNm** to add it to the **Expression** box.
12. Edit the expression to remove the "@" character before **TS_FlPlnCmpnt.FlPlnCmpntNm**, and add single quotation marks around **TS_FlPlnCmpnt.FlPlnCmpntNm**.

    Your expression will look like the following:



13. Click **OK**.
14. Delete text **TS_FlPlnCmpnt.FlPlnCmpntNm** in the first column of the table header.

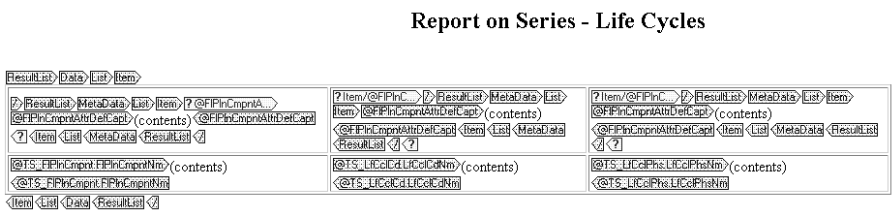


The name of the column also changes to from **TS_FlPlnCmpnt.FlPlnCmpntNm** to **Series Name**.

Report on Se

Series Name
Series 0031
Series 0105
Series 0106
Series 0205
Series 0216
Series 0220
Series 0414
Series 0927
Series 0942

15. Perform the similar operation for the other two columns. The expressions will look like the following:
    - Item/@FlPlnCmpntAttrDefCapt = 'TS_LfCclCd.LfCclCdNm'
    - Item/@FlPlnCmpntAttrDefCapt = 'TS_LfCclPhs.LfCclPhsNm'



16. To see the results, navigate to the bottom of the window and click **PDF Preview**.

At this point, you do other customization for your report, such as adding header and footers. To learn more about the features of Stylesheet Designer, see the information at *http://www.xmlspy.com/features_stylesheet.html*.

Step 5: Saving the report layout

After you define all of the elements in your report layout, such as the header, footer, and other formatting options, you can save your report as an *.xsl* file.

To save the report layout:

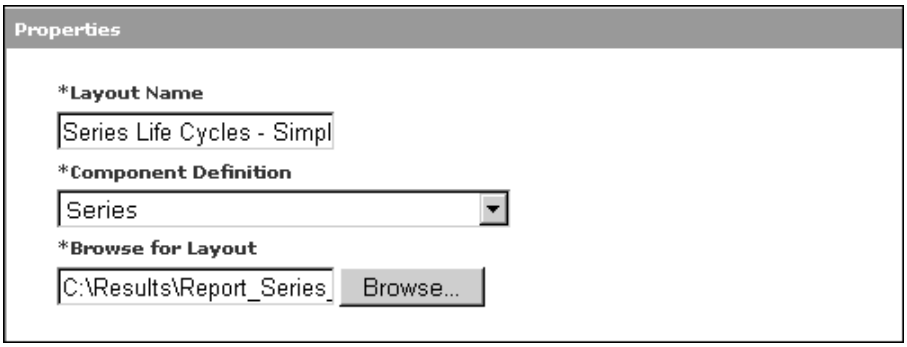
1. Click **File > Save**.
2. Save the files using the name *Report_Series_LifeCycle1.xsl*. Ensure that you change the extension to ***.xsl***.
3. Click **Save**.

Step 6: Adding the report layout to IRM

You can add report layouts to IBM DB2 Records Manger and apply them to reports. By default, IBM DB2 Records Manager includes some sample report layouts. Three of the sample layouts are based on custom file plan designs, and two are generic so you can use them with any file plan design. You can also create new report layouts using a third-party tool, such as the one we created earlier using Stylesheet Designer.

To add a report layout:

1. Start and log on to the IRM Administrator Client.
2. Click **Tools > Manage Report Layouts**.
3. Click **Add** on the **Manage Report Layouts** screen.
4. In the **Layout Name** box, specify a descriptive name for the report layout.
5. Select the **Component Definition** that defines the report. For our tutorial, we will use **Series - Life Cycles**.
6. Click **Browse** to navigate to the location of your report layout file you created earlier using Stylesheet Designer. (*Report_Series_LifeCycle1.xsl*).
7. Click **Open**.



8. Click **Save**.

Note: IBM recommends that you create a folder for your XSL files. Save the existing XSL files before you modify them. You must back up the XSL files for your created reports. IBM DB2 Records Manager stores the source files, but you cannot retrieve the source files.

Step 7: Assigning permissions to the report layout

You can add report layouts to IBM DB2 Records Manger and apply them to reports. IBM DB2 Records Manager includes some sample report layouts. Three sample layouts are based on custom file plan designs. Two are generic so you can use them with any file plan design. You can also create new report layouts using a third-party tool.You can follow the Report Layouts section of IRM on-line help (user guide) to add report layout, assign permission to report layout and generate report.

You must assign report layouts to the users or groups who require them. Use **Manage Report Layouts** to assign report layouts to users and groups.

To assign user access to report layouts:

1. Click **Tools > Manage Report Layouts**.
2. In the **Component Filter** box, select the component definition you want to filter.
3. To assign the layout you want to use, click the **Assign layout to users** icon.
4. In **All Users** list, select the users to which you want to assign the report layout. Select one or more users, and then click the right arrow to place them in the **Selected Users** list.
5. Click **Save**.

To assign group access to report layouts

1. Click **Tools**, and then click **Manage Report Layouts**.
2. In the **Component Filter** box, select the component definition you want to filter.
3. To assign the layout you want to use, click the **Assign Layout to Groups** icon.
4. In **All Groups** list, select the groups to which you want to assign the report layout. Select one or multiple groups and then click the right arrow to place them in the **Selected Groups** list.
5. Click **Save**.

Generating a report using an existing report layout

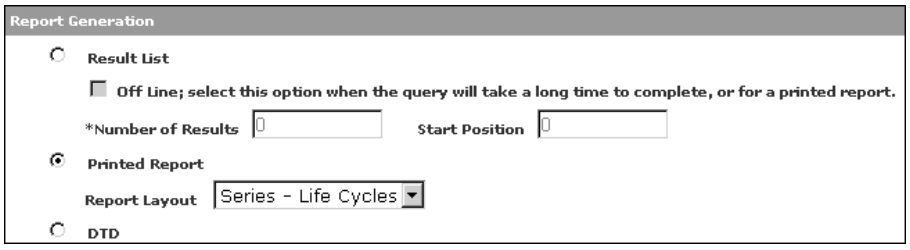
In IRM, the Records Administrator created and saved the query called **Series - Life Cycle Information** ("Step 1: Generating XML and DTD files from IRM" on page 15), added the report layout called **Series - Life Cycles** ("Step 6: Adding the report layout to IRM" on page 25), and assigned users permissions to the report layouts ("Step 7: Assigning permissions to the report layout" on page 26). Now, users are ready to run either the saved query, or create their own query, and assign the report layout to it.

Note: In this tutorial, we will use the saved **Series** query created earlier. If you choose to create a new query, you must ensure that your query contains the output fields specified in the report layout; otherwise, the resulting PDF report might contain columns with no data.

To generate a report using a report layout:

1. In IRM, click **Query > Saved**.
2. In the **Actions** column, select **Open Saved Query**.
3. In the **Report Generation** area, select **Printed Report**.
4. In the **Report Layout** box, select **Series - Life Cycles**.

   Note: If your report layout does not appear in the **Report Layout** list, ensure that you have been assigned permissions to access that layout.



5. Click **Generate**.
6. In the **Task Name** box, type a name for this query task, and specify the date and time you want IRM to start processing this task.
7. Click **Start and Open Status Page**.
8. Click **Refresh** opposite the **Task** field until the status of the query task completes.
9. In the **Actions** column for your task, select the **Download Printed Report** icon. You can choose the **View Results** icon to verify the results before proceeding with saving the PDF file.

   You can either choose to **Open** or **Save** the PDF report.
10. For this tutorial, click **Save**.

11. Navigate to the location to save this file, and change the name of the PDF file to something more meaningful.
12. Click **Save**.

Index

Glossary

A

accession. Involves the permanent transfer of a record and its meta data to another authority that assumes responsibility and ownership of the record.

advanced query. A query that generates a report based on system, custom file plan components, and file plan structure.

API. Application Programming Interface

application programming interface. A software interface that enables applications to communicate with each other. An API is the set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by the underlying licensed program. The IBM Records Manager API provides server components for the host application to access IBM Records Manager. You can embed IBM Records Manager into any line-of-business application. A programmer can also use the API to modify, enhance, customize, or completely re-write the existing IBM Records Manager Administrator user interface.

archive. Persistent storage used for long-term information retention, typically very inexpensive for each stored unit and slow to access, and often in a different geographic location to protect against equipment failures and natural disasters.

attribute. A unit of data that describes a certain characteristic or property of an object.

B

bar code. A code used for identification purposes. In IBM Records Manager, applying pre-printed bar codes to items simplifies the file plan administration process. Users add new file plan components by scanning the items directly into the file plan.

binary large object. A sequence of bytes with a size ranging from 0 bytes to 2 gigabytes. This string does not have an associated code page and character set. Image, audio, and video objects are stored in BLOBs.

C

character large object (CLOB). A sequence of characters (single-byte, multibyte, or both) up to 2 gigabytes. A CLOB can store large text objects. Also called a character large object string. Compare to binary large object (BLOB).

charge-out. Checking out a component from the file plan. A charged-out, component is locked.

classify. A method of assigning retention and disposition rules to records. Depending on the particular implementation, it can be manual or process-driven. You can present users with a list of allowable file codes from a drop-down list (manual classification). Ideally, the desktop process or application can automate classification by triggering a file code selection from a property or characteristic of the process or application.

client application. An application written with the Content Manager APIs to customize a user interface.

CLOB. See character large object.

common query. A set of frequently used queries that you can easily execute from Query-Common. The queries you can execute in Query-Common are Audit, Reservation, Charge Out, User/Group, and Life Cycle Code.

component definition. See file plan component definition.

Cut-off. Breaking or ending files at regular intervals, usually at the close of a fiscal or calendar year, to permit their disposal, or to transfer complete blocks or segments.

D

DAO. Data access objects. They are object created with Visual Basic.

data types. A definition of a set of data. In IBM Records Manager, there are eight data types available for attributes: String, Binary, Boolean, Date, Date-Time, Double, Integer, and Character Large Object (CLOB).

declare. To designate that a particular document is a corporate record. Once declared as a corporate record, edit and delete control of the document is passed from the user to the record keeping process, as administered by the corporate records management professionals. As a record, a document can only be modified or deleted by the records management process, not by the end user. Users must be aware of those documents that are records (declared), versus those that are not yet declared. Declaration can be manual whereby the user decides when to declare, and then sets a property or selects a menu option to declare the document. Alternatively, it can be and automated process whereby specifying a certain property triggers the automatic declaration of the record.

descriptor. See security descriptor.

disposal authority. A code or rule for approving the disposal of certain records.

disposition. The last stage in the record life cycle. Disposing a file plan component (either accession or destruction) also disposes its descendants.

document. A document managed by the host application (any form or format), an email message or attachment, a document created within a desktop application such as MicroSoft Word, regardless of the format. There are two types of document: electronic and non-electronic. An electronic document is stored in electronic format and it can be read. If declared as a record, an electronic document becomes a managed record. A non-electronic physical object can take on many forms (such as maps, paper, VHS video tapes, and CDs). You cannot record the physical object using the same method as electronic documents; however, you can store its descriptive meta data, and then track this information within IBM Records Manager (a profile). If declared as a record, a non-electronic object becomes a managed record.

E

exclusive relationship. A relationship that is exclusive within its view. An exclusive relationship can only exist where there are several relationships that have the same source. This kind of relationship determines the types of components to allow in a relationship with respect to its existing members. For example, a file component can have any number of files (as children), however, if one of those files contains folders, that file is the end of a branch and it can contain only folders.

F

file plan. An organization will have a common classification scheme for the entire organization, called a file plan. The file plan is typically a hierarchical set of subjects or business activities. Each node or subject file is annotated with a unique code called a file code. A given file code refers to a specific subject file within the file plan. Each subject file has an official retention rule (when, why, and how to delete) assigned to it. Each record must be assigned a file code that matches the appropriate subject file with in the file plan.

file plan administration. The design and administration of a corporate file plan. The records manager can design file plan components (classes of file plan objects such as files, records, and folders), and then define the attributes of these classes and their relationships (for example, files can contain files, records, and folders). Various views of the file plan may be defined. For instance, a warehouse view might present a view of the physical folders in the organization, whereas a numeric view might present the sorted numeric structure for maintenance purposes. The records manager can create pick lists enforcing consistency within the file plan, component profiles that define the characteristics of the file plan, and default values to simplify daily file creation tasks. Policies, permissions, and suspensions can be assigned to file plan objects.

file plan component. The classification of file plan objects, such as files, records, and folders.

file plan component definition. A declaration that the specific type of file plan component will exist within the file plan. After you define a file plan component definition, an unlimited number of actual file plan components of that declared type can exist within the file plan. A file plan component definition is a meta-object, used to declare the type (either record or component) of the actual components.

file plan management. The process of designing, building, and maintaining a file plan.

file plan relationship definition. The relationship of components within the file plan. For example, a folder can contain files, records, and other folders.

file plan view. A collection of relationships between components that comprise the file plan. In the same way that a view in a relational database is a collection of joined tables that comprise a schema. File plan views give each component in the file plan a context. No file plan components can exist outside a view. Every file plan component must be in at least one view (Hierarchical, Link, and Set).

FPC. See file plan component.

H

hierarchical view. A hierarchical views represent a tree-like structure in a parent and child relationship. Hierarchical views can also represent containment (for example, a box can contain a folder). You can have as many hierarchical views in a file plan as you require, but you must have at least one hierarchical view in a file plan.

host. An application that uses IBM DB2 Records Manager to provide life cycle retention management. The responsibilities of the host application are to generate electronic information, to provide tools to manipulate that data, and maintain a repository to store the information.

host application. Business software application into which you embed IBM DB2 Records Manager.

host configuration. Host configuration lets an integrator register a business application as the host with IBM DB2 Records Manager. The integrator registers a business application by specifying

information about their business application into which IBM DB2 Records Manager Administrator is integrated.

I

inheritance. The passing of class resources or attributes from a parent class to a child class.

IRM. IBM Records Manager

J

JDBC. Java database connectivity

Java Virtual Machine. Interprets compiled Java binary code for a computer's processor so that it can perform a Java program's instructions.

L

life cycle. A collection of phases through which any file plan component managed as a record must transit before it is disposed. A life cycle can consist of one or more phases; each phase lasting specific duration and denoting records management activity that must be performed at the beginning, or at the end of the phase. These phases comprise the life cycle duration.

life cycle code. A life cycle rule applied to a file plan component.

life cycle event. IBM Records Manager logs the following file plan component events as they occur: a component transitions to another phase, a component is disposed of (destroyed or accessioned), and a component fails to transition.

life cycle inheritance rules. When a file plan component does not have its own life cycle code, it inherits it from its closest ancestor in its primary view.

life cycle management. The records life cycle is the life span of a record from its creation, or to its final disposition. Typically, it is described in three stages: creation, maintenance and use, and final disposition. IBM DB2 Records Manager applies management to all three stages. With e-records, the records manager can create and maintain the official rules that determine when to destroy (or permanently keep) electronic records, as well as record and enforce any conditions that apply to destruction (for example, to destroy two years following contract completion). Finally, the records manager can carry out the physical destruction of electronic records, maintaining a legal audit file.

life cycle phase. A life cycle consists of one or more phases; each phase lasting specific duration and specifying records management activity to perform at the beginning, or at the end of the phase. These phases comprise the duration of a life cycle.

life cycle operations. The process of executing the rules that govern the life cycles of components (the transitioning of file plan components through their life cycles). After file plan components transition through all phases, they are ready for disposition.

life cycle rule. A rule that determine the following items: Time - How much time a component spends in any one phase of its life cycle. Security - Whether a component's security changes as it transitions from one phase to another in its life cycle. Disposal - How a component is disposed of when it completes the last phase of its life cycle. Cut off - Whether a component is cut-off when it enters a phase in its life cycle. Close - Whether a component is closed when it enters a phase in its life cycle. Interim transfer - Whether a component undergoes an interim transfer when it enters a phase in its life cycle. Begin Life Cycle When Superseded - Sets a file plan component to begin its life cycle when it is replaced by a new version. Event-Based Disposition - Sets disposition to be event based instead of time based.

life cycle suspensions. The suspension of a file plan component. If a file plan component does not qualify for transitioning, it will remain in its current life cycle phase until the removal of the suspension.

link view. A collection of unidirectional peer-to -peer type relationships. You can use a link view to establish a one-way relationship between two file plan component types. For example, a cross-reference link between two documents where two file plan components are cross-referenced to each other. There is no hierarchy between these documents. Users are aware of the existence of one document because it is cross-referenced to another.

logic extension class. Classes let you apply business rules to file plan components. You apply the business rules by associating logic extension classes with file plan component definition types and, then executing them.

M

migration. The process by which you move a file plan component from one physical location to another.

O

ORB. Object Request Broker acts as a "broker" between the client request for a service from a component, and the completion of the request.

P

paging block size. A value that restricts the number of query results that display on a page.

partition. A set of access rules that restrict user actions, at the system or component level.

partitioned component. The sub-division of a file plan component into parts, separated by date. When IBM Records Manager creates an instance of a component, it automatically creates an initial partition (part) for that component. When users add records to the component, IBM Records Manager automatically inserts the records into the current part, until a new partition is created for the component. When a new partition is created, IBM Records Manager closes the initial part, and the new part receives any new records (opening a new part automatically closes the preceding part).

permissions. Permissions define what an individual user or group can do to a specific file plan component, such as a filing or a record. Permissions relate who can do what, and where in the file plan. Internally, the relationship between the who, the what, and the where, is called an access control policy.

phase. See life cycle phase.

pick list. Pick lists let you define values a user can select from a pre-defined list of choices. The IBM DB2 Records Manager pick list feature lets you create, edit, and delete pick lists and assign them to both file plan components and non-file plan components, such as users and groups.

primary view. A primary view provides a path for a file plan component to inherit properties that are not specifically set for that file plan component, a unique way to navigate to the file plan component, and a security context for the file plan component. It is a mechanism by which a file plan component inherits its inheritable properties (hierarchical view). Every file plan component gains its context within the file plan by belonging to at least one view.

profile. A data entry form containing attributes that users must specify for a component. Profiles restrict user access to certain fields on the form, as well as limiting the actions on these fields.

put away. The action of returning one or more charged out file plan components.

Q

query. A request for information from the database based on specific conditions.

R

record. Any form of recorded information that is under records management control. Records are either physical or electronic. Records can be any of the following types: Document - A document that was declared as a record. Once declared as a record, the document is under records management control. Folder - A physical folder containing documents. You can declare individual documents within the folder as records (declared as non-electronic documents). Box - A box containing paper documents. Usually contains folders, which are individually managed as records, but may alternatively contain records other than folders, such as loose documents about a specific subject. Non-electronic - A declared physical document of any form (such as maps, paper, VHS video tapes, and CDs). The physical object is not recorded in electronic form; however, you can use IBM Records Manager to store descriptive meta data about the objects, and track that information within the objects profile. If declared as a record, a physical object becomes a managed record. A document (electronic or non-electronic) is not considered a record until it is declared.

records management. The administrative infrastructure represents the tasks that the records manager performs on the entire organization's collection of declared records. End users never see this process; it conducted within the Records Manager Administration Client (a browser-based web application). Records management consists of the following broad activities; file plan administration, records security control, life cycle management, and reporting.

relationship definition. See file plan relationship definition.

repository. A physical storage area for documents and electronic records. Refers to the Host Application's repository.

reservation. A request by a user to borrow a file plan component on a future date. A file plan component could be a document, box, folder, or any other item you can deliver.

retention rules. The set of rules which specify how long to keep (retention) records, and what to do with them at the end of their life cycle (disposition)

retention schedule. See retention rules.

RMA. See records management application.

root view. The root is the top of any hierarchical file plan view. The root file plan component definition is a system object that you cannot edit or delete. You use the root file plan component definition in hierarchical relationships (relationship types that belong to hierarchical views) as the default. Every hierarchical view must have the root file plan definition as its root.

S

security descriptor. A series of words (from a common collection of words) allocated to file plan components

and users, or to file plan components and groups. Security descriptors provide you with enhanced access control.

set view. A type of relationship that contains an ordered number of components that are independent of each other. Unlike the hierarchy and link views, a set does do not have a direction. Because an ordinal value describes the order of the records, sets provide you with version management capabilities for your records.

simple query. A query that generates a report for one or more fields in a custom file plan component. A simple query has three parts: scope, query details, and report output options.

SQL. Structure Query Language, SQL is an American National Standards Institute. standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database.

suspend. A suspended file plan component no longer qualifies for transition in the current phase of its life cycle. When you suspended a file plan component, it remains in its current life cycle phase until the removal of the suspension.

system component definitions. Component definitions that represent business objects used specifically by IBM Records Manager. The system components included with IBM Records Manager are not the file plan components in the file plan hierarchy; they are required for the file plan to function, such as life cycle codes, user accounts, and group accounts.

V

view. See file plan view, hierarchical view, primary view, set view, and link view.

W

WSDL. Web Services Description Language. It is an XML based language used to describe the services you offer. WSDL is derived from SOAP and from the IBM Network Accessible Service Specification Language.

Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku Tokyo 106, Japan

Licensing

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS"WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J74/G4
555 Bailey Avenue
San Jose, CA 95141
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Portions of this product were developed by The Apache Software Foundation (http://www.apache.org/) Copyright © 1999 The Apache Software Foundation. All rights reserved.

THE LOG4J, XERCES, XALAN SOFTWARE ARE PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL IBM OR THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following terms apply to the LOG4J, XERCES, XALAN components:

1. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the above disclaimer in the documentation and/or other materials provided with the distribution.
2. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: ""This product includes software developed by the Apache Software Foundation (http://www.apache.org/)."" Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
3. The names ""log4j, xerces, xalan"" and ""Apache Software Foundation"" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
4. Products derived from this software may not be called ""Apache"", nor may ""Apache"" appear in their name, without prior written permission of the Apache Software Foundation.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries or both:

IBM

DB2

WebSphere

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Printed in USA