

IBM StoredIQ Data Workbench
Version 7.6.0.7

User Guide



IBM StoredIQ Data Workbench
Version 7.6.0.7

User Guide



Note

Before using this information and the product it supports, read the information in Notices.

This edition applies to Version 7.6.0.6 of product number 5724M86 and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2001, 2016.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of contents

Tables	v
-------------------------	----------

About this publication	vii
IBM StoredIQ Platform product library	vii
Contacting IBM StoredIQ customer support	vii
Contacting IBM	vii
How to send your comments	viii

Chapter 1. Overview of IBM StoredIQ

Data Workbench	1
User personas of IBM StoredIQ Platform	1
Key concepts of IBM StoredIQ Data Workbench	2
IBM StoredIQ Platform index	2
Information set	2
Filter	3
Overlay	3
Scope operation	3
Action	4
Report	4

Chapter 2. Getting started with IBM StoredIQ Data Workbench 7

Chapter 3. Infoset management 9

Creating an infoset	9
Interrupting infoset creation	10
Searching for a specific infoset	11
Editing an infoset	11
Deleting an infoset	12
View of the infoset details	12
Viewing infoset ancestry	13
Saving infoset ancestry as a workflow	13
Execution log	13
Viewing action log of infosets	13
Generating an audit report	14
Viewing data objects	14
Retraining a classification model	16

Chapter 4. Filter administration 17

Library of available filters	17
Searching for an existing filter	17
Deleting a filter	17
Cloning a filter	17
Editing a filter	18
Viewing the filter code	18
Building a filter	18
Previewing filter results	24
Configuring columns when previewing filter results	24
Set Operations	25
Scope operations	29
Duplicate Operations	30

Chapter 5. Refinement of infosets and data 33

Data maps	33
Refining an infoset	34
Grouping an infoset by data source type	34
Viewing infoset with different views	35
Selecting data overlays	36
Downloading CSV data	36

Chapter 6. Enhancements to infoset 37

Running an enhancement	37
Searching for a specific enhancement	37

Chapter 7. Actions on data 39

Searching for a specific action	40
Scheduling and running an action	40
Canceling a scheduled action	40

Chapter 8. Data reports 41

IBM StoredIQ Data Workbench report types	41
User reports	44
Format of CSV reports	44
Searching for a report	44
Generating a system report	45
Generating a term hit report	45
Generating a user report	46

Chapter 9. Request management 47

My Requests	47
User notification	48

Chapter 10. Exceptions. 49

Searching for a specific exception object	50
Viewing exception object details	50
Exporting the exception objects list	50

Appendix A. Filter condition reference and expression language. 51

Full-text syntax	51
Full-text macros	54
Extended ASCII characters	55
Attribute condition reference	57
Specification of attribute conditions	57
Conditions for string attributes	58
Conditions for enumeration attributes	58
Conditions for date attributes	59
Conditions for number attributes	60
Conditions for sparse attributes	60
Conditions for compound attributes	61
Conditions for multivalued attributes	61
Access mode attribute	62
Type of object to match	62
Filter condition reference and expression language	63
Multiple matching values for operators	63

At-least expressions	64
Parent and child conditions	64
No-inheritance conditions	65
Analytics filter expressions	66

Appendix B. IBM StoredIQ discovery export 67

Concordance DAT load file	67
Concordance DAT (Light)	67
EDRM XML	68
Result directory and EDRM XML load file	69
File system directory layout	69
EDRM XML considerations	73
EDRM XML metadata tags	74

Appendix C. Metadata by attribute type 87

System metadata attributes	87
Documentum server metadata attributes.	90
Centera server metadata attributes.	93
Hitachi HCAP Server metadata attributes	95
Discovery Accelerator metadata attributes	96
FileNet metadata attributes	97

Email metadata attributes	99
Data object property attributes.	110
SharePoint attributes	148
Calendar metadata attributes	186
Contact metadata attributes	186
Digital signatures metadata attributes	188
Processing metadata attributes.	188
Task metadata attributes.	191
Attribute value types.	193
IBM Lotus Notes metadata attributes	194
Enterprise vault metadata attributes.	196
Jive metadata attributes	197
Chatter metadata attributes.	198
Box attributes	199
Content Manager attributes.	203

Notices 205

Privacy policy considerations	207
Terms and conditions for product documentation	208
Trademarks	208

Index 211

Tables

1. Infoset heading names and descriptions	9	22. Column headings in attribute table.	87
2. Infoset Ancestry Column Definitions	12	23. System metadata attributes	87
3. Execution Log Column Definitions.	13	24. Metadata for Centera server attributes	93
4. Data objects default column definitions	15	25. Metadata for Hitachi HCAP	95
5. Set Operations	25	26. Metadata for Discovery Accelerator	96
6. Available operation infoset options and descriptions	26	27. Metadata for FileNet	97
7. View By selection options.	35	28. Metadata for email	99
8. Enhance options and descriptions	37	29. Data object property attributes	110
9. Events column definitions.	49	30. Metadata for SharePoint	148
10. Types column definitions	49	31. Metadata for calendar.	186
11. Exception objects column definitions	49	32. Metadata for contact	186
12. Boolean operator syntax	52	33. Metadata for digital signatures.	188
13. Terms	52	34. Metadata for processing data	189
14. Wildcard	53	35. Metadata for task items	191
15. Proximity syntax	54	36. Attribute value types	193
16. Extended ASCII Character syntax	56	37. Metadata for IBM Lotus Notes.	194
17. Attribute type prefix	57	38. Metadata for enterprise vault	196
18. Available object types	63	39. Metadata for Jive	197
19. Keyword examples	63	40. Metadata for Chatter	198
20. Valid data types for EDRM XML Tags.	74	41. Box attributes	199
21. EDRM data tags	75	42. Content Manager attributes	203

About this publication

IBM® StoredIQ Data Workbench User Guide describes how to use the data management tool to learn about the data in your system and manage it to achieve your business goal.

IBM StoredIQ Platform product library

The following documents are available in the IBM StoredIQ Platform product library.

- *IBM StoredIQ Platform Deployment and Configuration Guide*, SC27-6386
- *IBM StoredIQ Platform Overview Guide*, GC27-6398
- *IBM StoredIQ Platform Data Server Administration Guide*, SC27-5692
- *IBM StoredIQ Administrator Administration Guide*, SC27-5688
- *IBM StoredIQ Data Workbench User Guide*, SC27-5691
- *IBM StoredIQ eDiscovery User Guide*, SC27-5693
- *IBM StoredIQ Policy Manager User Guide*, SC27-5694

Contacting IBM StoredIQ customer support

For IBM StoredIQ technical support or to learn about available service options, contact IBM StoredIQ customer support at this phone number:

Support and assistance

- 1-866-227-2068

To e-mail IBM StoredIQ customer support, use this email address:

- storediqsupport@us.ibm.com

For information about how to contact IBM, see the Contact IBM web site at <http://www.ibm.com/contact/us/>

IBM Knowledge Center

The IBM StoredIQ publications can be found from IBM Knowledge Center.

PDF publications

The IBM Publication Center site offers customized search functions to help you find all the IBM publications you need.

Contacting IBM

For general inquiries, call 800-IBM-4YOU (800-426-4968). To contact IBM customer service in the United States or Canada, call 1-800-IBM-SERV (1-800-426-7378).

For more information about how to contact IBM, including TTY service, see the Contact IBM website at <http://www.ibm.com/contact/us/>.

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

Send your comments by using the online reader comment form at https://www14.software.ibm.com/webapp/iwm/web/signup.do?lang=en_US&source=swg-rcf.

Chapter 1. Overview of IBM StoredIQ Data Workbench

IBM StoredIQ Data Workbench is a data visualization and management tool that helps you actively manage your company's data.

IBM StoredIQ Data Workbench helps you find out how much data you have, where it is, who owns it, and when it was last used. When you have a clear understanding of your company's data landscape, IBM StoredIQ Data Workbench helps you take control of it and make informed decisions about your data and act on that knowledge.

Big data is a pervasive problem for the business. IBM StoredIQ Data Workbench is developed to help you learn about your data, make educated decisions about that data, and turn a dangerous risk into a valuable asset.

IBM StoredIQ Data Workbench can be used in the following ways:

- Find all emails from a specific sender in any time frame that you specify.
- Identify unused data that stays in the system for a long time.
- Search for specific format of files on the network for a defined time period. For example, search image files that were created in 2007, and see how much space these files take on the network.
- Provide a visual overview of how data is retained. Break down the data source so that forbidden information such as credit-card numbers or Social Security numbers can be identified.

User personas of IBM StoredIQ Platform

The IBM StoredIQ Platform provides applications and interfaces for three user types.

System administrator

This person is responsible for IBM StoredIQ Platform installation setup, system and network configuration and maintenance, and administration activities. These activities are required to be done before other users of the IBM StoredIQ applications can start their work. The administrative activities mainly include:

- Getting the data servers and data centers ready for use.
- Adding volumes and ensuring security of data and data source.
- Harvesting volumes and generating system info sets.
- Managing users, actions, target sets and concepts.
- Creating reports and using auto-classification models.
- Creating Mule scripts.

Data expert

A data expert understands both business processes and technical implementation. This person is responsible for responding to requests from the business user in a timely fashion, managing data in the IBM StoredIQ applications, such as IBM StoredIQ Data Workbench.

Legal user

The legal user is an end user who is looking to complete some business processes. This person is responsible for managing matters, such as

filtering, adding, deleting matters, configuring matters, or creating and managing boxes. The legal user's work is dependent upon the system administrator and data expert. The system administrator sets up the IBM StoredIQ Platform environment and authorizes the right and scale of work that the legal user can do with the IBM StoredIQ eDiscovery application. The data expert controls and manages the data in the legal system.

Policy user

The policy user is responsible for creating and running a set of pre-determined steps to automatically execute against data that is being managed by the IBM StoredIQ solution. The policy user uses the IBM StoredIQ Policy Manager application to create new policy runs. Each policy run is based off pre-configured scripts that were deployed to IBM StoredIQ Policy Manager.

Key concepts of IBM StoredIQ Data Workbench

Key concepts of IBM StoredIQ Data Workbench, such as information set, filters, action, and indexes are explained. These concepts are key to understanding IBM StoredIQ Data Workbench as a whole.

IBM StoredIQ Platform index

Based on the volumes that are added in the data servers, indexes are generated through IBM StoredIQ Administrator or IBM StoredIQ Data Server to examine, classify, and map data in the network. They are used to search for data and find out what and how much data you have in your system.

There are two types of indexes: metadata index and full-text index.

Metadata index contains all the information about the data at a specific location on a network. It includes descriptive information or attributes about the data such as a file name, file size, created date, and owner.

Full-text index is a more detailed index on the contents of the data itself. By reading the contents of the data, the words or characters that are contained within the data can be referred to and searched against.

Information set

An information set, abbreviated as an info set, is the core concept in using the IBM StoredIQ applications. It is created and used to collect specific data to manage the business system.

There are two types of info sets: system info sets and user info sets.

System info sets are generated after volumes are harvested from IBM StoredIQ Administrator or IBM StoredIQ Data Server. They can be viewed from the IBM StoredIQ Data Workbench user interface but users cannot edit or delete them. They can also be manually created by the administrator to target certain volumes in the IBM StoredIQ Administrator application. These system info sets can be edited or deleted by the system administrator. System info sets must be generated or created before any user info sets can be created.

A user info set is created out of an existing system info set by a user. It is defined to contain specific data that a user needs to operate upon the system. For example,

you can create a user infoset that contains a person's emails in Year 2000. Then, you can act on the data within this infoset: to move, copy, or delete it from your system.

Data Map

Data Map is the visualization of an infoset. It provides a visual layout of the data and in-depth information about data source types, data categories, size or amounts, the number of data objects and details. For more information, see "Data maps" on page 33

Filter

A filter is created upon the available information that was populated by the index. It is used to classify or refine the existing infoset to create a new infoset.

A filter can have multiple attributes. You can apply several attributes of a filter to one infoset to create a new infoset that consists of the data that you need.

Example of filtering an infoset

A system infoset contains all files, emails, and documents of all company employees. You need to retrieve some specific data about Josh Smith: Josh's emails with a subject of stock option and Josh's files that are larger than 1 GB. To get these two sets of data, you can use the filter to refine the system infoset into two user infosets.

To create the first infoset about Josh's emails with the subject of stock options, you need to take the following steps:

- Apply a name filter attribute to find Josh Smith.
- Apply a file filter attribute to find Josh's emails with the .MSG extension.
- Apply an email filter attribute to find Josh's emails with a subject of stock options.

To create the second infoset to get Josh's files of larger than 1 GB, you need to take the following steps:

- Apply a name filter attribute to find Josh Smith.
- Apply a filter attribute of size larger than 1 GB to all of Josh Smith's files that are larger than 1 GB.

Overlay

Overlays are configurable filters that display hits or matches in a selected infoset.

Within the data map, color intensifies for data objects that match the overlay change. The greater the overlay matches, the more red that tile appears within the data map.

Scope operation

Contained data, such as data within .ZIP, .TAR, or .PST files, is hierarchical and can have different relationship and connections with other data. Depending on how that data is viewed, that data can give a different perception than what is represented.

The **Scope Operations** tab helps you understand more about what data is represented by data sets. Within scope operations, you can conduct expansion or collapse operations.

- In an expand operation, all files within an infoset are expanded, so creation of an infoset can be more accurate.
- In a collapse operation, all opened or expanded files within an infoset are collapsed, so an infoset that is created can be small.

Note: If the files within the infoset are not container files, then the **Expand** operator or **Collapse** operator has no effect.

Action

An action is an activity that is created by an administrator and conducted on an infoset. Available IBM StoredIQ Data Workbench actions include copy, copy to retention, delete, discovery export, modify attribute, move, Watson Curation, Step-up Snippet, and Step-up Full-text.

Actions do not alter infosets. An infoset is a grouping of data, not the data itself, and actions are applied to the actual objects, not the infoset. When you copy, you copy the actual file, not the infoset. The same is true for copying to retention or discovery export. Actions can be scheduled to run immediately or at a predetermined time and date.

Report

IBM StoredIQ Data Workbench reporting function provides external views of infosets and validates IBM StoredIQ processes. Reports can also be customized with the BIRT Report Designer.

You can share the information that is contained within infosets with the reporting component, which allows infosets to be transferred to other media types for review and analysis. These reports do not affect existing infosets, but provide you with more usable formats in which to understand the files and data that is captured by an infoset.

Reporting is a key step within the data-management process as it validates that processes were completed correctly within IBM StoredIQ. You can customize reports in any of these scenarios:

- Modify reports to carry your organization's custom styles and logos, aligning them with other organization-based artifacts and documentation.
- Alter the format of the content reported in existing reports. For example, you can add more columns, switch axes in a graph, or change the units for some values.
- Design reports to contain information that is not found in other, existing reports

IBM StoredIQ Data Workbench contains these types of reports:

- Compliance Report
- Content Collector Manifest File CSV Export
- CSV All Audited Objects Export
- CSV Attribute Summary
- CSV Exception List Export
- CSV Infoset Data Object Export
- Data Assessment Report

- Data Topology Report
- Duplication Summary Report
- Overlay Hit Report
- Term Hit Report

Chapter 2. Getting started with IBM StoredIQ Data Workbench

Before you can use IBM StoredIQ Data Workbench, the administrator must set up the IBM StoredIQ Platform environment.

About this task

The administrator must ensure that these tasks are completed before you can use IBM StoredIQ Data Workbench:

- Install and configure IBM StoredIQ Platform.
- Ensure that the data servers are up and running.
- Add volumes and create indexes through IBM StoredIQ Administrator or IBM StoredIQ Data Server.
- Harvest volumes and generate system info sets.
- Create and define actions for the user of IBM StoredIQ Data Workbench to manage data objects that are contained within info sets.

To log in to the IBM StoredIQ Data Workbench user interface, follow these steps:

Procedure

1. Open the IBM StoredIQ Platform user interface and enter your email and password.
2. If you have several IBM StoredIQ applications that are installed, click the down triangle arrow to open the list of the available applications.
3. Select to open IBM StoredIQ Data Workbench.

Chapter 3. Infoset management

The infoset window provides the following high-level information for a selected infoset.

Table 1. Infoset heading names and descriptions

Infoset Column Name	Description
Name	This list reflects the name of the selected information set. An infoset must be selected before you can view detailed information, conduct analyses, or act on it.
Total objects	The number of data objects that are contained within the infoset. If you place the mouse over the number of total objects, a window appears. It shows the number of top-level and child-level objects that are contained within the total number of objects. Note: If the infoset is still being created, only the Details window is active. Infosets that are still being created list 0 in this column.
Infoset size	The size of the infoset.
Composition	The composition of the data that is contained within an infoset, either Mixed or Top Level .
Created	The date and time at which the infoset was created. Note: System infosets are listed as N/A in this field, and user infosets list the creation date, even if their creation status shows pending.
Type	The type of infoset, either User or System .
Access	This option lists whether the infoset is Public or Private or the name of person or number of people who have access to the infoset. If you place your mouse over the number of people, a tooltip appears with the names of those users who have access. If you click the link, a window with the users' names and roles appears.
Creator	The person that originally created the infoset. For older infosets, the creator is listed as N/A.
Description	A brief description of the infoset, which was provided during infoset creation.

Creating an infoset

Infosets can be created to include the data map and its function, or they can exclude data maps, resulting in a faster creation time. Additionally, you can interrupt infosets during their creation.

About this task

Infosets are created from other existing system infosets.

To create an infoset from an existing infoset:

Procedure

1. Select the infoset from which you want to create a new infoset. It can be either a system or user infoset. The name of the selected infoset appears in the infoset window.
2. Click **Create**, and then select a filter from the **Name** column in the Library tab.

Note: If this creation is your first time, you must create a custom filter. See Building a filter.

3. Click **Create New Infoset**. The Create New Infoset dialog box appears.
4. In the **New Infoset Name** text box, enter the name of this infoset.
5. In the **Description** text box, enter the description of this infoset.
6. Select or clear the **Include Data Map for the Refine tab** check box.
 - If you select this check box, the data map is generated for this infoset. The infoset takes longer to generate, but you are able to use the data map and its function for this infoset.
 - If you clear this check box, the infoset generates more quickly, but it does not include the data map.

Note: If you opt to create an infoset without a data map, some of the reports that can be run against that infoset are affected. If the Compliance, Data Assessment, Data Topology, Overlay, and Customized Reports are run against an infoset without a data map, they contain no data.

7. To send a notification email that the infoset was created, select **Send email when the process is complete** check box and enter the email address at which the individual must be notified.
8. Click **Select overlays** to determine which overlays must be applied to this infoset. A Create New Infoset window opens, providing available overlays to add to this infoset.

Note: When you are exploring this infoset, these selected overlays are available to be applied within the data map. If the data map is not generated, then overlays cannot be selected.

9. Select overlays in the Available overlays panel and click **Add** to add the overlays. You can add no more than seven overlays.
10. Click **Save and View** to go to the **Infoset Ancestry** tab, or click **Save** to return to the **Library** tab.

Interrupting infoset creation

Infoset creation can sometimes exceed the time that you have available. Infoset interruption makes it possible for you to opt out of the creation process.

About this task

Infoset creation contains different forms. In some instances, you might want to interrupt the infoset's creation, preserve what was found, and then log the unfinished work as an exception. In other instances, you might want to halt infoset creation and then delete that infoset altogether and deleting any artifacts from the creation process.

- After an interrupted and partially created infoset was moved from **Created > Interrupted > Completed**, you can use that infoset. However, you must keep in mind that the completion state depends on when the infoset is stopped.

- When you are deleting an infoset partway through its creation, click **Delete**. When an infoset is deleted, it is no longer available within the user interface.
- If an infoset is processing, a **Stop Processing** link is available within the dashboard. During this processing state, this link can be used to interrupt its creation and access the Stop Processing confirmation window.

To stop infoset creation, follow these steps:

Procedure

1. During infoset creation, in the Create New Infoset dialog box, click **Save and View** or **Save**.

Note: In the **Total objects** and **Infoset size** columns, the infoset is listed as **Pending**.

2. To stop or interrupt infoset creation, you can do either of the following steps:
 - Select the pending infoset, and then click **Stop**.
 - If the pending infoset is active, click **Stop Processing**.
3. Click **OK** to verify your action. The infoset's creation is now changed to **Interrupting**. It then moves to a **Completed** state.

Note: If you opt to use this interrupted infoset, it is only a partial infoset as its creation was interrupted.

4. Explore or delete your infoset.
 - Click **Delete**, deleting the infoset from the user interface and the infoset library as a whole.
 - Select the infoset and explore its content just as you do with any other infoset. You can refine its content, apply actions, and view exceptions on interrupted infosets.

Searching for a specific infoset

When you have multiple infosets available to choose, you can search for a specific infoset instead of using the slider to move through available options.

Procedure

1. In the infoset window, click the down arrow. A complete listing of all available infosets appears.
2. In the **Enter key terms to search....** text box, enter the name of the infoset for which you want to search, and then press **Search**. You can enter either the full infoset name or a portion of it. To remove the search term, click the **X** to the left of the **Enter key terms to search....** text box. Infosets that match the entered search term are returned.
3. Click the infoset that you want to select, and then click **Select Infoset**. The selected infoset appears at the top of the infoset window.

Editing an infoset

When you are editing an existing infoset, you can modify only the infoset's name and description.

Procedure

1. In the Infoset window, use the down arrow to select the infoset that you want to edit.

2. Click **Edit**, which is in the upper right corner of the infoset window.

Note: Because system infosets cannot be edited, this **Edit** link is disabled if a system infoset is selected within the infoset window.
The Edit Infoset dialog box appears.

3. In the Edit Infoset dialog box, you can edit only the infoset's name and description. Click within either the **Name** or **Description** text boxes to edit the infoset.
4. Click **Save** to save your modifications.

Deleting an infoset

Infosets can be deleted in both the **Complete** and **Pending** states.

About this task

An infoset cannot be deleted under these circumstances:

- If an infoset was created with another infoset as its source, then the child infoset must be deleted before you delete the parent infoset.
- If an action was run or is running against the infoset, then the infoset cannot be deleted.

Procedure

1. In the Infoset window, use the down arrow to select the infoset that you want to delete.
2. Click **Delete**, and then click **OK**.

Note: Deleting an infoset permanently removes its history.

View of the infoset details

You can use the **Details** tab to view detailed information about a selected infoset and any updates that are made to the data sources used by the infoset.

There are three tabs within the **Details** area: **Infoset Ancestry**, **Execution Log**, and **Data Objects**.

The **Infoset Ancestry** tab shows the history and genealogy of the selected infoset. For example, it displays the system infoset from which the user infoset was created and which filter was used.

Additionally, the **Infoset Details** section provides the following information about a selected infoset.

Table 2. Infoset Ancestry Column Definitions

Infoset Ancestry Column Heading	Definition
Name	The name of the item selected.
Created	The date and time the selected item was created.
Type	The item's type, either system or user infoset, filter.
Description	The description of the selected item.

Viewing infoset ancestry

View infoset ancestry from the Details window.

Procedure

1. Verify that you selected the infoset for which you want to view its ancestry.
2. In the Details window, click the **Infoset Ancestry** tab. This tab is selected by default.
3. In the **Name** column, click the name of the item for which you want to view ancestry information. The **Ancestor Infoset Details** area populates with the name, description, total objects, top-level objects, child objects, infoset size, composition, created date, type, and creator. The **Ancestor Filter Details** area displays name, description, a link to view the filter code, created date, and type.

Saving infoset ancestry as a workflow

This functionality is removed from IBM StoredIQ Data Workbench as IBM StoredIQ Data Script is disabled. Workflows can be created with IBM StoredIQ Mule scripts.

For information about creating a workflow with IBM StoredIQ Mule scripts, see IBM StoredIQ Mule script and Creating a Mule Script policy.

Contact IBM support for assistance if you need to re-enable IBM StoredIQ Data Script.

Execution log

When the **Execution Log** tab is selected, you can see the name, status, the run start time, and the length of time of an action ran.

The Execution Log column is defined as follows.

Table 3. Execution Log Column Definitions

Execution Log Column Heading	Definition
Name	The name of the action on the selected infoset.
Status	The status of the action on the selected infoset.
Type	The type of action being run on the selected infoset.
Run started	The time at which the action was started. By clicking the arrow to the right of the column name, you can list the column in ascending or descending order.
Run duration	The length of run time of the selected action.

Viewing action log of infosets

By selecting an action in the **Name** column, the **Action Details** section provides information about that action, including name, description, status, and which infosets it was last used on.

Procedure

1. Verify that you selected the infoset for which you want to view its actions log.
2. In the Details window, click the **Execution Log** tab.

3. In the **Name** column, click the name of the action for which you want to view log information. The **Action Details** area populates, displaying name, description, status, the info set on which the action was last used, any exceptions, and the **Generate Report** link. If you click the exceptions percentage, the **Exceptions** tab appears, providing you with greater context and detail.
 - When the **Generate Report** link is clicked and an audit report is generated, a **Source Action** detail becomes available, showing the report's name, the date and time it was created, and the length of time necessary to generate the report. If the report is being generated, its status is shown as **Running**.
 - When the report has finished generating, it can be viewed by clicking the **Download Report** link within **Status**.

Generating an audit report

Audit reports can be generated for actions selected within the Execution Log.

About this task

The Action Audit Report provides detailed information in these areas:

- Execution Summary
- Data Object Result Count Summary
- Info set Summary
- Participating Data Server and Volume Summary
- Action Parameters

Procedure

1. Verify that you selected the info set for which you want to generate an audit report, and then click **Details > Execution Log**.
2. In the **Action Details** area, click **Generate Report**. The Generate Audit Report dialog box appears.
3. In the **Report Name** text box, the report's name is prefilled, based on the selected action in the **Execution Log**.
4. In the **Send To:** text box, enter the email addresses to whom this report is sent. Separate multiple email addresses with a comma.
5. Click **Generate**, and the audit report generates. When the audit report is generating, its status changes to **Running** in both the **Action Details** area and the **Execution Log**. When the report is generated, it is sent to the email addresses entered in the **Send To:** text box, and it can be opened from that email. The report can also be downloaded within the Execution Log.

Viewing data objects

When the **Data Objects** tab is selected, the data object list returns objects in the info set. You can also spot-check data objects within an info set with Data Object Viewer.

About this task

By clicking a data object, you can view the object itself using Data Object Viewer.

Table 4. Data objects default column definitions

Data Objects Column Heading	Definition
File name	The name of the file within the elected infoSet.
File size	The size of the file.
File path	The directory path of the file.
Created	The creation date and time of the file.
Last modified	The date and time at which the file was last modified.

Procedure

1. In the Details window, click the **Data Objects** tab. Data objects within the infoSet are returned; you can see the data objects that are contained within the infoSet.
2. Click an object's file name to view and verify that object in Data Object Viewer.
3. Click **Close** to close Data Object Viewer.
4. Configure columns when you are reviewing data objects:
 - a. In the Details window, click the **Data Objects** tab.
 - b. Click the blue gear icon, and the Configure Columns: Data Objects window appears.
 - c. In the Configure Column: Data Objects window, click the down arrow to the right of options 2-6, selecting the name of the column. Choose from these options: None, File size, File path, File type, Container path, Last accessed, Created, Last modified, Owner name, or Group name. The File name option cannot be changed. By default, it is the name of the first column when you are viewing data objects. Additionally, as you rename columns, more columns are automatically added to the Configure Column: Data Objects window.
 - d. To reset the columns to the default selections, click **Reset to default column selections**. The column configuration options return to their default settings.
 - e. Click **Save**. The layout of the **Data Objects** area reflects the chosen options.

Note: Only the first two pages of a data object can be viewed.

Feedback for an auto-classification model

When viewing data objects that utilize an auto-classification model, you can provide feedback for the classification model to improve its validity and accuracy through the Data Object Viewer.

Each object is scored as how it pertains to the classification model. You can improve the classification model by reviewing objects and assigning pertinent categories as needed. This feedback is then used to retrain the classification model. More feedback increases the model's validity.

For information about how to provide feedback for the classification model through the Data Object Viewer, see Retraining a classification model.

Retraining a classification model

Use feedback to retrain a classification model.

Before you begin

For more information about retraining a classification model, see Retraining a classification model.

The administrator must create a classification model with IBM StoredIQ Administrator. For information about creating a model, see Building an auto-classification model.

Procedure

1. From IBM StoredIQ Data Workbench, create a user info set.
2. Select the user info set and run the Step-up Snippet in the **Act** tab.
3. Select an enhancement in the **Enhance** tab to apply to the user info set and then click **Run Enhancement**.
4. Create a filter and apply the filter:
 - a. Select the **Auto-classify** option.
 - b. In the list below the **Auto-classify** option, select the classification model for which results are filtered.
 - c. Select the categories. By default, the **Category** option is selected, as are all of the available categories.
 - d. Determine how the results are displayed, selecting either **And the highest score in the selected category** or **And where the score <is/Greater than/Less than/Not equal/Greater than or equal/Less than or equal> <1>**.
5. Click **Preview Filter Results**. The objects appear in the **Filter Results** area.
6. Select one of the returned objects to view in the Data Object Viewer.
7. Click the button to the right of **Auto-class Scores**. A list of categories and score of the object for each category appear. The category with the highest score is listed at the top.
8. Select the categories to which the selected data object belongs. Data objects with a value closer to one (1) are more closely associated with that category. Data objects with a value closer to zero (0) are not associated with that category.
9. Click **Submit**. This feedback helps improve the accuracy and validity of the classification model. Repeat this step for as many different data objects as you like. The number of feedback submissions can be seen within the <model name> Details panel in IBM StoredIQ Administrator.
10. At this step, the administrator needs to select the classification model from IBM StoredIQ Administrator. Click **Retrain** to launch the retrain process for this model. If the model was uploaded without a learning archive (SARC file) or if it received no feedback, then the **Retrain** button is disabled.
11. From IBM StoredIQ Data Workbench, run the enhancement against the user info set again to see the new scores within the Data Object Viewer. The improved scores indicate greater validity and accuracy.

Chapter 4. Filter administration

Filters are objects that can be applied to infosets. They are used to refine the infosets. You can apply one filter with multiple filter attributes to an infoset, continuously refining the infoset until it represents responsive data that you need.

Note: Attributes within a filter category are ORed. For example, when you are creating a file filter that is defined by file name and extension contains .DOC, the filter syntax is "file name OR file extension contains doc". Adding another file attribute category, the categories are ANDed. Therefore, the file category filter attribute "extension OR file name contains doc" and adding the name category with "owner name John Smith," the syntax is "file name" OR "extension contains doc" AND "owner names contains John Smith."

There are five tabs on the **Create** page: **Library**, **Build Filter**, **Set Operations**, **Scope Operations**, and **Duplicate Operations**.

Library of available filters

The **Library** tab lists all filters that are created and available for use with infosets.

Within the **Library** tab, you can view the names and descriptions of all saved filters. Additionally, you can edit and delete those filters.

Searching for an existing filter

Search for an existing filter within the **Library** tab.

Procedure

1. Click **Create**, and then click the **Library** tab.
2. In the **Enter key terms here....** text box, enter the name of the filter for which you want to search, and then press **Search**. You can enter either the full filter name or a portion of it. Filters that match the entered search term are returned. The **Filter Details** area displays the filter's name, description, a link to view the filter's code, and the name of the infoset in which the filter was last used.
3. To remove the search term, click the **X** to the left of the **Enter key terms here....** text box.

Deleting a filter

Filters can be deleted when they are no longer used by an infoset.

Procedure

1. Click **Create > Library**, and then select the name of the filter that you want to delete.
2. Click **Delete**. The Filter Delete Confirmation dialog appears.
3. Click **OK** to delete the filter, or click **Cancel** to cancel the deletion. If the filter is deleted, it is removed from the list of available filters within the **Library** tab.

Cloning a filter

Clone an existing filter to create other or identical filters.

Procedure

1. Click **Create > Library** and then select the name of the filter that you want to clone.
2. Click **Clone**. The Edit Filter window opens.
3. Enter the name and description of the new filter.
4. Enter the text in the search box to find the code that you want to edit.
5. After you edit the code, click **Validate Code**.
6. When the editing is complete, click **Save As**. The newly cloned filter appears in the **Library** tab.

Editing a filter

After filters are created and saved, they can be edited.

Procedure

1. Click **Create > Library**, and then select the name of the filter that you want to edit.
2. Click **Edit**. The Edit Filter dialog box appears.
3. Edit the filter as needed, determining the filter's scope and validating the code.
4. Click **Save** to save your changes.

Viewing the filter code

To open and view the filter code, follow these steps.

Procedure

1. Within the **Filter Details** area, click **View Filter Code**. The Filter Code window appears.
2. Click **Close** to close the Filter Code window.

Building a filter

Create a filter to customize your search for specific data.

Procedure

1. Click **Create** and then click the **Build Filter** tab.
2. Select the check box or boxes of the type of filter attribute that you want to create. Depending on the type of filter you are creating, select from these options: **File**, **Date**, **Size**, **Name**, **Email**, **Full Text**, and **Auto-Classification**.

Option	Description
File	File filters are based on common file fields such as file name, file path, container path, and extension.
Date	Date filters are based on common date information such as accessed, modified, or created dates.
Size	Size filters are based on common data-object file size field such as bytes, KB, MB, and GB.
Name	Name filters are based on name parameters such as the owner name or SharePoint owner.

Option	Description
Email	Email filters are based on common email fields such as To, From, CC.
Full Text	Full-text filters are based on a simple search string against all data objects. This option returns results only for full-text harvests.
Auto-classify	Auto-classify filters are based on the step-up snippet action. They can be run on an infoset, extracting and storing text for any data object to which it is applied, and the auto-classification enhancement.

- In the **Filter Details** area, enter a name and description for the newly created filter, and then click **Save Filter**. The newly created filter can be seen within the **Library** tab.

Creating a file filter

File filters are a type of customized filters. File filters are based on common file fields such as name, extension, and path.

Procedure

- Select the infoset and then click **Select Infoset**.
- Click **Create** and then click the **Build Filter** tab.
- Select the **File** check box and then complete the following steps:
 - In the **Contains** list, select either **Contains**, **Is**, or **Begins with**.
 - In the text box, enter the text about the file.
 - Select the check box for what the particular filter uses, choosing from **File Name**, **Extension**, **File Path**, or **Container Path**. You can select any or all of the file type filter options.
- In the **Filter Details** area, complete the following steps.
 - In the **Filter Name** text box, enter the name of this new filter.
 - In the **Description** text box, enter a brief description of this filter.
 - Click **Save Filter**.

Results

The custom filter is created and saved, and it can now be accessed from the **Library** tab.

Creating a date filter

Date filters are based on common date information such as accessed, modified, or created dates.

Procedure

- Click **Create**, and then click the **Build Filter** tab.
- Select the **Date** check box, and then complete following steps:
 - In the list of operators, select **After**, **Before**, **On**, or **Between**.
 - In the **Date** text box, enter the date for this custom filter.
 - Select the check box as to what fields to which the date filter applies, choosing from **Accessed**, **Modified**, or **Created**. You can select any or all of the date-type filter options.

- d. In the Filter Details window, enter the name of this new filter.
- e. In the **Description** text box, enter a brief description of this filter.
- f. Click **Save Filter**.

Results

The custom filter is created and saved, and it can now be accessed from the **Library** tab.

Creating a size filter

Size filters are based on common data-object file size field such as bytes, KB, MB, and GB.

Procedure

1. Click **Create**, and then click the **Build Filter** tab.
2. Select the **Size** check box, and then complete following steps:
 - a. In the list of operators, select **More Than, Less Than, Is, or Between**.
 - b. In the empty text box between the lists, enter the number that applies to this custom size filter.
 - c. In the size value list, select **bytes, KB, MB, or GB**.
 - d. In the Filter Details window, enter the name of this new filter.
 - e. In the **Description** text box, enter a brief description of this filter.
 - f. Click **Save Filter**.

Results

The custom filter is created and saved, and it can now be accessed from the **Library** tab.

Creating a name filter

Name filters are based on name parameters such as the owner name or SharePoint owner.

Procedure

1. Click **Create**, and then click the **Build Filter** tab.
2. Select the **Name** check box, and then complete following steps:
 - a. In the blank text box, enter the owner or SharePoint owner name.

Note: If you are entering more than one name, separate the names with commas. For example, you would enter multiple names as John Smith, Mary Jones, Ellen Sideways.
 - b. Select the check box as to what fields to which this name filter applies, choosing from **Owner** or **SharePoint Owner**. You can select any or both of the name-type filter options.
 - c. In the Filter Details window, enter the name of this new filter.
 - d. In the **Description** text box, enter a brief description of this filter.
 - e. Click **Save Filter**.

Results

The custom filter is created and saved, and it can now be accessed from the **Library** tab.

Creating an email filter

Email filters are based on common email fields such as To, From, CC, BCC, and email subject.

Procedure

1. Click **Create**, and then click the **Build Filter** tab.
2. Select the **Email** check box, and then complete these options.
 - a. In the blank text box, enter the names or email addresses that is used by this filter. Separate multiple names or email address with commas. You might enter John Smith, Mary Jones or jsmith@company.com, mjones@company.com.
 - b. Select the check box or boxes to which this email filter applies, choosing from **To**, **From**, **CC**, **BCC**, or **Email Subject**.

Note: If you select the **Email Subject** check box, you must also select an operator: **Contains**, **Is**, or **Begins With**, and then enter some text about the email's subject into the empty text box.

3. In the **Filter Details** area, complete the following steps.
 - a. In the **Name** text box, enter the name of this new filter.
 - b. In the **Description** text box, enter a brief description of this filter.
 - c. Click **Save Filter**.

Results

The custom filter is created and saved, and it can now be accessed from the **Library** tab.

Creating a full-text filter

Full-text filters are based on a simple search string for full-text indexed objects. Only objects that were full-text indexed return results.

Procedure

1. Click **Create**, and then click the **Build Filter** tab.
2. Select the **Full Text** check box.
3. In the blank text box, enter the search string that is run against data objects in a full-text index. You might enter stock options AND CEO or Apple NOT IBM.
4. In the **Filter Details** area, complete the following steps.
 - a. In the **Name** text box, enter the name of this new filter.
 - b. In the **Description** text box, enter a brief description of this filter.
 - c. Click **Save Filter**.

Results

The custom filter is created and saved, and it can now be accessed from the **Library** tab.

Creating an auto-classification filter

Auto-classification filters are based on the auto-classification model that is created in IBM StoredIQ Administrator.

Procedure

1. Click **Create**, and then click the **Build Filter** tab.
2. Select the **Auto-classification** check box.
3. What appears in this area is dependent upon the model that is created in IBM StoredIQ Administrator.
 - a. Select the check box that you want: **And the highest score in the selected category** or **And where score is**. Then, use the lists to select **Greater than**, **Less than**, or **Equal to** and the appropriate decimal amount.
4. In the **Filter Details** area, complete the following steps:
 - a. In the **Name** text box, enter the name of this new filter.
 - b. In the **Description** text box, enter a brief description of this filter.
 - c. Click **Save Filter**. The Save Created Filter confirmation appears, verifying that you want to save this filter. Click **OK**.

Creating customized searches with the code view

Use the code view to create customized searches with search syntax.

Procedure

1. Click **Create**, and then click the **Build Filter** tab.
2. Click **Code View**.

Note: To return to the filter's form view on the **Build Filter** tab, click **Filter Form View**. Then, you can create and save filters with the prebuilt forms.

3. In the large text box, create and enter a customized search with correct filter syntax. For more information about appropriate filter syntax, including tips and example, see Syntax Primer.
4. Click **Validate Code**. If the entered expression is valid, a message that states **The filter expression passed validation** is returned. If it is invalid, a message that states **The filter expression is not valid** is returned.
5. In the **Filter Details** area, complete the following steps.
 - a. In the **Name** text box, enter the name of this new filter.
 - b. In the **Description** text box, enter a brief description of this filter.
 - c. Click **Save Filter**.

Syntax primer:

This primer offers syntax examples and tips for using Boolean operators, multiple terms or term phrases, wildcard, and proximity filters for the full-text filter or full-text source code filter.

Note: Saved filters are available as overlays when you are creating new infosets. For more information, see Creating an infoset.

Boolean operator syntax

AND	The AND operator ensures that both terms appear in the data object. For example, the query chairman AND stock returns data objects where chairman and stock appear. && functions the same as AND.
OR	The OR operator returns data objects where either term appears. The query chairman OR stock returns data objects that have chairman, stock, or both in the data object. functions the same as OR.

NOT	The NOT operator ensures that a term does not appear in any data objects returned. The query Apple NOT IBM returns data objects where word Apple appears, but not IBM. ! functions the same as NOT.
-----	---

Multiple terms or term phrase syntax

" " and ' '	Quotation marks around a term phrase return data objects that contain that phrase in that order. For example, "September 20 1998" returns only data objects with the exact phrase of September 20 1998. Double quotation marks allow for wildcards in terms whereas single quotation marks do not.
Multi-item, no Boolean operators	A search for multiple words without Boolean operators or quotation marks is an implied OR search. For example, September 20 1998 returns data objects that have September OR 20 OR 1998. Stop words are removed.
Parentheses	<ul style="list-style-type: none"> Grouping with Boolean operators and parentheses controls how operators are evaluated. For example, (\$10 OR billion) AND options returns data objects where \$10 or billion might be in the data object, but options must also exist. Boolean operators (AND, OR, NOT) work in a grouping or in a proximity group. So ("Sahara Desert" AND "Southeast Asia") OR ("Pacific Ocean" AND "Bavaria") returns data objects that have Sahara Desert and Southeast Asia or Pacific Ocean and Bavaria.

Wildcard syntax

*(Asterisk)	<ul style="list-style-type: none"> A single asterisk finds multiple characters or letters at beginning of a term. A filter for *olol returns results for betaxolol, carteolol, levobetaxolol, levobunolol, metipranolol, and timolol. An asterisk matches any number of characters in the middle of a term. te*t returns results for text, test, tent, and tempest. An asterisk that is placed after a partial word finds terms that begin with those letters at the end of a term. A filter for ap* returns results for Apple, April, applied, and application. An asterisk alone finds all objects.
?	The question mark matches a single character. For example, compl? returns results for comply but not complies. To filter for more than one character, use a question mark for each letter, for example, ap??e.

Proximity syntax

w/x	<ul style="list-style-type: none"> Filtering uses the w/x syntax (x represents the number of terms between the two terms) to use the relationship between search terms. w/0 matches terms that are next to each other in any order, and w/1 matches terms with one word between them. The terms can appear in any order. Grouping with proximity allows for even more targeted results. For example, ("stock" OR "option") w/3 (2007) finds either stock or option within 3 terms of 2007. Terms can have wildcards or phrases.
-----	--

pre/x	<ul style="list-style-type: none"> • The pre/x syntax (x represents the number of words between the two terms) enables users to target proximity filters by dictating which term comes before another term. For example, chairman pre/15 options returns data objects where chairman comes up to 15 terms before options. • pre/0 matches terms that are next to each other and in the same order; pre/1 matches terms with at most one word between them and in the same order. • Terms can have wildcards or phrases.
-------	--

Notes regarding syntax

When you are examining text, IBM StoredIQ evaluates the requested operators and gives precedence to some over others.

- When parentheses are used to group operators, the expression in parentheses is given the highest precedence.
- NOT: "OctoberNOT November OR December" means "(October AND (NOT November)) OR December".
- Proximity operators ("w/" and "pre/"): "October w/3 November pre/5 December" means "(October w/3 November) pre/5 December". ("October" and "November" must be within three words of each other, but both must be before "December" by no more than five words.) "October w/3 November and December" means "(October w/3 November) and December".
- The AND operator: "October AND November OR December" means "(October AND November) OR December".
- The OR operator or no operator: "October November OR December" means the same as "(October OR November)OR December".

Previewing filter results

You can preview sample, returned data before you save that infoset.

Procedure

1. Select the infoset from which you want to create a new infoset.
2. Apply a filter to this infoset by either using an existing filter or creating a new filter.
3. Click **Preview Filter Results**, and data that is contained within that infoset appears within the **Filter Results**. It includes the name, file size, file path, its creation date and time, and the date and time it was last modified.

Configuring columns when previewing filter results

When you preview data objects that are contained within an infoset, you can select what type of information is displayed, ensuring that the returned data is meaningful to you.

Procedure

1. Select the infoset from which you want to create a new infoset.
2. Apply a filter to this infoset by either using an existing filter or creating a new filter.
3. Click **Preview Filter Results**, and a list of data objects that are contained within that infoset appears.
4. Click the blue gear icon. The Configure Column - Preview Filter Results dialog box appears.

- In the Configure Column - Preview Filter Results dialog box, click the down arrow to the right of options 2-6, selecting the name of the column. Choose from these options: None, File size, File path, File type, Container path, Last accessed, Created, Last modified, Owner name, or Group name.

Note: The File name option cannot be changed. By default, it is the name of the first column when you are previewing data. Additionally, as you rename columns, more columns are automatically added to the Configure Column - Preview Filter Results dialog box.

- To reset the columns to the default selections, click **Reset to default column selections**. The column configuration options return to their default settings.
- Click **Save**. The layout of the **Filter Results** area reflects the chosen options.

Set Operations

Set Operations allows infosets to be combined in different ways to produce another infoset.

You can highlight specific information within the infoset, and apply these types of set operations to infosets: union, intersection, symmetric difference, and subtraction.

Table 5. Set Operations

Operation	Description	Example
Union	A union infoset is the combination of two or more infosets: the primary infoset and the check boxes of whatever available infoset you select.	For example, you have three infosets, two of which contain PSTs by owner and a third that contains all Exchange email. You want to combine all three infosets to make it easier to run actions on all of the data objects therein. All three infosets combine in their entirety, resulting in a new, larger infoset. The combination of the affected areas is turned gray within the interface.
Intersection	An intersected infoset is the result of two or more infosets that contain some of the same objects. This overlap, which is where identical objects are in all infosets, is called the intersection. It is this intersection that results in the new infoset.	For example, you have two custodians, each with a significant amount of data, some of which is shared. You want to ascertain which of the shared data objects are identical. In this case, you must create an intersection infoset as you want to find identical, common data objects between the two. Only the area of similarity, the gray intersection between the two white circles, is found in the new infoset.

Table 5. Set Operations (continued)

Operation	Description	Example
Symmetric Difference	A symmetric difference infoset combines a primary and selected infoset and removes identical objects from them. The remaining data objects from both filtered infosets then combine to create a new infoset.	For example, you again have two custodians, each with a significant amount of shared data. You want to ascertain which of the shared data objects are not identical. In this case, you must create a symmetric difference infoset as you want to find data objects that are different between the two. Only the area of difference between the two infosets is the white intersection. This area of difference results in a new infoset of everything but the data that is shared between the two circles.
Subtraction	A subtraction infoset combines a primary and selected infoset, removes identical objects, and then subtracts the remaining objects from the primary infoset. The remainder data objects result in the subtraction infoset.	For example, you have two infosets, both of which contain similar PSTs by owner. You want to combine both infosets and retain only the unique data objects to make a new infoset. The two infosets combine, identical data objects are removed, and only the remaining objects, that area in gray, create the new infoset.

This table describes the options available for set operations for each infoset.

Table 6. Available operation infoset options and descriptions

Available Operation Infoset Option	Description
Name	The name of the infoset against which a set operation can be conducted.
Total Objects	The total number of data objects that are contained within the infoset.
Infoset Size	The size of the infoset.
Composition	The composition of the infoset, top level, or mixed level.
Created	The date and time at which the infoset was created.
Type	The type of infoset. Remember, set operations can be conducted only against user infosets. System infosets are not available in the Available Operation Infosets pane.
Search	Search for a particular infoset with the Enter terms... text box and Search .
Details	Click Details to view the Infoset Details window. Information regarding the infoset's name, description, total number of objects, number of parent and child objects, size, composition, created date and time, type, and percentage of exceptions appears.

Note: For union and intersection set operations, check boxes appear to the left of the infoset names, but for symmetric difference and subtraction operations, radio buttons appear.

Creating a union infoset

Union infosets are the combination or joining in their entirety of two or more infosets.

Procedure

1. Select the infoset to which you want to combine one or more infosets. This infoset is the primary infoset. The name of the primary infoset appears in the infoset window.

Note: Set operations can be conducted only on user infosets, not system infosets.

2. Click **Create** and then click **Set Operations**.
3. Click **Union** and then select the check boxes of the available operation infosets to which you want to join to the primary infoset. To select all of the available operation infosets, select the check box to the left of **Name**.
4. Click **Create New Infoset**. The Create New Infoset dialog box appears.
5. In the **New Infoset Name** text box, enter the name of this infoset.
6. In the **Description** text box, enter the description of this infoset.
7. Select or clear the **Include Data Map for the Refine tab** check box.
 - If you select this check box, the data map is generated for this infoset. The infoset takes longer to generate, but you are able to use the data map and its function for this infoset.
 - If you clear this check box, the infoset generates more quickly, but it does not include the data map.

Note: If you opt to create an infoset without a data map, some of the reports that can be run against that infoset are affected. If the Compliance, Data Assessment, Data Topology, Overlay, and Customized Reports are run against an infoset without a data map, they contain no data.

8. To send a notification email that the infoset was created, select **Send email when the process is complete** check box and enter the email address at which the individual must be notified.
9. Click **Select overlays** to determine which overlays must be applied to this infoset. A Create New Infoset window opens, providing available overlays to add to this infoset.

Note: When you are exploring this infoset, these selected overlays are available to be applied within the data map. If the data map is not generated, then overlays cannot be selected.

10. Select overlays in the Available overlays panel and click **Add** to add the overlays. You can add no more than seven overlays.
11. Click **Save and View** to go to the **Infoset Ancestry** tab, or click **Save** to return to the **Library** tab.

Creating an intersection infoset

Intersected infosets are the result of the combination two or more infosets where identical data objects are the new infoset.

Procedure

1. Select the infoset to which you want to combine one or more infosets. This infoset is the primary infoset. The name of the primary infoset appears in the infoset window.

Note: Set operations can be conducted only on user infosets, not system infosets.

2. Click **Create** and then click **Set Operations**.
3. Click **Intersection** and then select the check boxes of the available operation infosets in which you want to find identical data objects. Data objects that are the same in each of the selected infosets are then used to form a new, intersection infoset. To select all of the available operation infosets, select the check box to the left of **Name**.
4. Click **Create New Infoset**. The Create New Infoset dialog box appears.
5. Follow Step 5 to Step 11 in “Creating a union infoset” on page 27 to finish creating an intersection infoset.

Creating an infoset with a symmetric difference

Symmetric difference infosets remove identical objects from two filtered infosets. The remaining data objects from both filtered infosets then combine to create a new infoset.

Procedure

1. Select the infoset from which you want to remove the symmetric difference of another infoset. This infoset is the primary infoset.

Note: Set Operations can be conducted only on user infosets, not system infosets.

The name of the primary infoset appears within the infoset window.

2. Click **Create** and then click **Set Operations**.
3. Click **Symmetric Difference** and then select the infoset in which you want to find identical data objects. Data objects that are the same in the primary and selected infoset are removed, and the remaining data objects are then combined to form the new infoset.
4. Click **Create New Infoset**. The Create New Infoset dialog box appears.
5. Follow Step 5 to Step 11 in “Creating a union infoset” on page 27 to finish creating an infoset with a symmetric difference.

Creating a subtraction infoset

Subtraction infosets remove objects that are contained within the secondary infoset from the primary infoset. The remaining objects then create a subtraction infoset.

Procedure

1. Select an infoset. This infoset is the primary infoset.

Note: Set Operations can be conducted only on user infosets, not system infosets.

The name of the primary infoset appears within the infoset window.

2. Click **Create** and then click **Set Operations**.
3. Click **Subtraction** and then select the infoset that you want to subtract from the primary infoset. Data objects that are the same in the primary and selected infoset are removed, and the remaining data objects form the new infoset.
4. Click **Create New Infoset**. The Create New Infoset dialog box appears.
5. Follow Step 5 to Step 11 in “Creating a union infoset” on page 27 to finish creating a subtraction infoset.

Scope operations

Contained data, such as data within .ZIP, .TAR, or .PST files, is hierarchical and can have many different relationship and connections. Depending on how that data is viewed, that data can give a different perception than what is represented. The **Scope Operations** tab helps users to understand more fully what data is represented by data sets.

There are two actions available within scope operations:

Expand

This operator expands all of the files within an info set. Any nested hierarchies or contained files are opened, expanding into a new info set that is a potentially more accurate representation of the contained data.

Note: If the files within the info set are not container files, then the **Expand** operator has no effect.

Collapse

This operator collapses or retracts all of the opened or expanded files within an info set. It results in a new, smaller info set that is a potentially more accurate representation of the data.

Note: If the files within the info set are not container files, then the **Collapse** operator has no effect.

Expanding all contained objects

When you are using scope operations to expand contained objects, such as nested message threads, email attachments, or compressed archive contents, the resulting info set can contain all objects that are included with the container and container itself.

Procedure

1. Select the info set in which you want to expand hierarchies. This info set is the primary info set. The name of the primary info set appears within the info set window.
2. Click **Create**.
3. Click **Scope Operations**.
4. Click **Expand**. Nested hierarchies and associations are expanded, and data object counts likely either remain the same or increase.
5. Click **Create New Info set**. The Create New Info set dialog box appears.
6. Follow Step 5 to Step 11 in “Creating a union info set” on page 27.
7. Review and validate the objects within the new info set as needed.

Collapsing objects into containers

When you are using scope operations to collapse an info set, nested and compressed objects such as message threads, email attachments, or archive contents are no longer part of the info set. The resulting info set data object counts can be lower than the original info set. The resulting info set can better represent how the data objects appear according to the file system.

Procedure

1. Select the info set in which you want to collapse hierarchies. This info set is the primary info set. The name of the primary info set appears within the info set window.

2. Click **Create**.
3. Click **Scope Operations**.
4. Click **Collapse**. Nested hierarchies and associations are collapsed, and data object counts likely either remain the same or decrease.
5. Click **Create New Infoset**. The Create New Infoset dialog box appears.
6. Follow Step 5 to Step 11 in “Creating a union infoset” on page 27.
7. Review and validate the objects within the new infoset as needed.

Duplicate Operations

Duplicate Operations provides a way for you to identify a wide variety of duplicate data in your system. By generating a new duplicate identification infoset, you can apply filters, operations, reports, or actions to start your data deduplication process.

Duplicate Operations compares objects of two infosets that are based on each other's hash value. If an object's hash value matches, the system can flag that object as a duplicate object. To conduct this operation, you must specify a primary infoset to apply against the infoset that you want to identify its duplicate objects. The operation then creates a new duplicate identification infoset that contains all duplicate objects except any objects from the primary infoset.

Creating a duplicate identification infoset

Use the **Duplicate Operations** tab to identify duplicate objects. A new duplicate identification infoset is created to include the duplicate objects.

Procedure

1. From the **Infoset** list, select an infoset as a source infoset that you want to identify its duplicate objects.
2. Click the **Create** tab.
3. Click the **Duplicate Operations** sub tab.
4. In the Available Operation Infosets pane, select an infoset as a primary infoset to compare to the source infoset.
5. Click **Create New Infoset**. The Create New Infoset dialog window opens.
6. Enter the name of the new infoset in both **New Infoset Name** text box and **Description** box.
7. Select or clear the **Include Data Map for the Refine tab** check box.
 - If you select this check box, the data map is generated for this infoset. The infoset takes longer to generate, but you are able to use the data map and its function for this infoset.
 - If you clear this check box, the infoset generates more quickly, but it does not include the data map.

Note: If you opt to create an infoset without a data map, some of the reports that can be run against that infoset are affected. If the Compliance, Data Assessment, Data Topology, Overlay, and Customized Reports are run against an infoset without a data map, they contain no data.

8. Select **Send email when the process is complete** and enter the email address where you want to be notified after the infoset is created.
9. Click **Select overlays** to determine which overlays must be applied to this infoset. A Create New Infoset window opens, providing available overlays to add to this infoset.

Note: When you are exploring this infoset, these selected overlays are available to be applied within the data map. If the data map is not generated, then overlays cannot be selected.

10. Select overlays in the Available overlays pane and click **Add** to add the overlays. You can add no more than seven overlays.
11. Click **Save and View** to view the infoset that is being created in the **Infoset Ancestry** tab. Or, click **Save** to return to the **Duplicate Operations** tab.

Results

A new duplicate identification infoset is created to consist of all duplicate objects except any objects from the primary infoset. It is ready for you to start the data deduplication process.

Chapter 5. Refinement of infosets and data

The **Refine** tab provides you with a visualization of an infoset. You can see how much data you have, data source types, and breakdown of the data into predefined groups such as file type category or created date.

You can use the visualization and its controls to refine or cull down the data objects that are contained in the selected infoset. The visualization rules out the non-essential data that you do not need.

By saving this visual refinement, you can create new infosets that contain a much further refined set.

Data maps

The data map displays information about data source and allows you to select data source type, view options, data overlay, and details about data size and list of files.

The data map contains these areas:

Data Map Option	Description
Group By	<p>The Group By area contains these options:</p> <ul style="list-style-type: none">• Data Source Type: This view displays the different data source types that make up the selected infoset. By selecting or clearing the data source types, the data map automatically reconfigures, providing you with a visual overview of your data.• None: This option removes data source types from the data map.
View By	<p>The View By area contains these options:</p> <ul style="list-style-type: none">• Category View: View data by category, including container files and archives, images, spreadsheets, and email.• Size View: View data by size, from less than one MB to over 100 GB. The Size View is only active for infosets that have a top-level composition.• Created View: View data by created date from within the past 30 days to more than five years ago.• Last Modified View: View data by its last modified date, from within the past 30 days to more than five years ago.• Last Accessed View: View data by its last accessed date, from within the past 30 days to more than five years ago. <p>By selecting or clearing the viewing options, the data map automatically reconfigures, providing you with a visual overview of your data.</p>
Select Data Overlay	<p>The Select Data Overlay list lets you select an overlay to be applied to your data, helping you to analyze your data differently.</p>

Data Map Option	Description
Data Map	<p>Under the Data Map, you can see the total number of objects being viewed as Viewing x of x total objects. This total is affected by the selection of the Object Count and Object Size tabs.</p> <ul style="list-style-type: none"> • If Object Count is selected, this figure indicates the number of data objects that are being viewed compared to the total number of data objects. • If Object Size is selected, this figure indicates the data objects' size in bytes compared to the total byte size of the data objects. <p>These same tabs, Object Count and Object Size, provide you with different viewing options within the Data Map itself.</p> <ul style="list-style-type: none"> • If Object Count is selected, the data map displays the number of objects and their percentage of that data source when you place the pointer over a category. • If Object Size is selected, the data map displays the size and percentage of the objects for that data source when you place the pointer over a category. Object Size is only active for infosets that have a top-level composition.
Data Map Details	<p>This area displays details about a tile that is selected in the data map. By clicking an individual map tile, details about that specific tile are displayed. You can also see the total number of data objects referred to by the map tile.</p> <p>Note: Selected tiles have a green border. Tiles over which the pointer was placed have a black border.</p> <p>The Data Map Details also shows you the Total Size of the selected tile. The bar graph automatically populates, providing a visual representation of how the size of the selected tile relates to that of the size of the infoset as a whole.</p> <p>By clicking View Full List, the Data Map Details dialog appears. From here, you can download information about the selected tile to a CSV file.</p>

Refining an infoset

Use the **Refine** tab to perform the refining tasks.

Procedure

1. On the home page of IBM StoredIQ Data Workbench, from the **Infoset** dropdown list, select the infoset for which you want to refine its data and click **Select Infoset**. The name of the selected infoset appears within the infoset dropdown list window.
2. Click the **Refine** tab. The Data Map opens, providing a visual overview of the data within the selected infoset.
3. Proceed to the following refining procedures to accomplish your tasks.

Grouping an infoset by data source type

To view the selected infoset and different data sources it uses, you can select **Data Source** under **Group By**. To view only the infoset without its types of data source, select **None**.

Procedure

1. Select or clear the check boxes to the left of the data source types. Data sources that are cleared are removed from the data map, and the view is automatically reconfigured to reflect these changes.
2. Use the data map to better understand your info set and the data sources that are used to make up that info set.

Viewing info set with different views

Objects within info sets can be viewed by different data views, ranging from category or size to its last accessed or modified date.

Procedure

1. To view the selected info set, select one of these categories from **View By: Category, Size, Created, Last Modified, or Last Accessed**. The info set is displayed within the Data Map, and the map tiles correspond to the selected view type.

Table 7. View By selection options

View By Option	Available Category Options
Category	These options are available when Category is selected: <ul style="list-style-type: none">• Container files and archives• Images• Spreadsheets• Miscellaneous• System files• Multimedia• Computer programming• Word processing• Presentation• Email• Collaboration
Size	These options are available when Size is selected: <ul style="list-style-type: none">• Less than 1 MB• 1 MB to 10 MB• 10 MB to 100 MB• 100 MB to 1 GB• 1 GB to 10 GB• 10 GB to 100 GB• Over 100 GB <p>Note: This option is only active when you are viewing a top-level info set.</p>
Created	These options are available when Created is selected: <ul style="list-style-type: none">• Last 30 days• 1 - 6 months ago• 6 - 12 months ago• 1 - 3 years ago• 3 - 5 years ago• Over 5 years ago

Table 7. View By selection options (continued)

View By Option	Available Category Options
Last Modified	<p>These options are available when Last Modified is selected:</p> <ul style="list-style-type: none"> • Last 30 days • 1 - 6 months ago • 6 - 12 months ago • 1 - 3 years ago • 3 - 5 years ago • Over 5 years ago
Last Accessed	<p>These options are available when Last Accessed is selected:</p> <ul style="list-style-type: none"> • Last 30 days • 1 - 6 months ago • 6 - 12 months ago • 1 - 3 years ago • 3 - 5 years ago • Over 5 years ago

2. Select or clear the check boxes to the left of the viewing options. The viewing options that are cleared are removed from the Data Map, and the view is automatically reconfigured to reflect the changes.

Selecting data overlays

Data overlays, which are configured during infoSet creation, can be applied to an infoSet.

Procedure

1. To apply an infoSet overlay, you must first create and save a filter. You can also use any available infoSet. For more information, see Building a filter.
2. Select an infoSet from the **Select Data Overlay** dropdown list.
3. Select **Show all** or **Show only matches**, depending on what you want to view.

Downloading CSV data

Data for a selected tile of an infoSet can be downloaded as comma-separated-value files.

Procedure

1. Select the infoSet for which you want to download CSV data.
2. Click **Refine**, and then select a tile in the Data Map.
3. In Data Map Details, click **View Full List**. The Data Map Details dialog box appears.
4. Click **Download to CSV**, and then open or save the CSV file as needed.

Chapter 6. Enhancements to infoset

The **Enhance** tab provides you with a visual layout of available enhancements. Enhancements are created as models within IBM StoredIQ Administrator. When an enhancement is applied to an infoset, it updates that infoset's index.

Table 8. Enhance options and descriptions

Enhance Option	Description
Name	The name of the enhancement that can be applied to an infoset.
Type	The enhancement's type. Currently, the only option is Auto-class .
Description	An optional description of the enhancement.

In the **Enhancements Details** area, a summary of the selected enhancement is displayed, including the enhancement's name, description, and its last usage date.

Running an enhancement

Enhancements are another way to refine or distill an existing infoset. Within an enhancement, existing patterns and rules are used to define the criteria for which the enhancement is comprised. By applying an enhancement to an infoset, the same criteria that make up the enhancement is applied to the selected infoset, creating a new, refined infoset.

About this task

Enhancements such as an auto-classification model must be created and defined within IBM StoredIQ Administrator by an administrator. If an enhancement is not available within IBM StoredIQ Administrator, contact the administrator.

Procedure

1. Select the infoset that you want to enhance.
2. Click **Enhance**.
3. Select the enhancement that you want to apply to the selected infoset.
4. Click **Run Enhancement**. The Run Enhancement Confirmation window appears. Click **Run and View** or **Run** to run the enhancement.

Searching for a specific enhancement

When you have multiple enhancements from which to choose, you can search for a specific enhancement instead of using the slider to move through available options.

Procedure

1. Select the infoset that you want to enhance. The selected infoset appears at the top of the infoset window.
2. Click **Enhance**.
3. In the **Enter key terms...** text box, enter the name of the enhancements for which you want to search, and then press **Search**. You can enter either the full

enhancement name or a portion of it. To remove the search term, click the **X** to the left of the **Enter key terms....** text box. Enhancements that match the entered search term are returned.

4. Select the enhancement, and then click **Run Enhancement**.

Chapter 7. Actions on data

Actions are preconfigured options that help you to manage data objects that are contained within infosets.

Note: Actions must be created and defined within IBM StoredIQ Administrator by the administrator. They can be scheduled to run immediately or later. If an action is not available within IBM StoredIQ Data Workbench, contact your administrator.

Within IBM StoredIQ Data Workbench, you can perform these actions on data objects within infosets:

Basic Move

A move action moves data objects from one location to another. When you are moving an object, the designated location must be specified.

Basic Copy

A copy action copies data objects from a source volume to a destination volume.

Retention Copy

This action copies data objects from a source to a retention location for a specified retention period.

Discovery Export

This action copies data objects and generates files for loading into third-party legal review tools.

Delete This action removes data objects from the source volumes.

Watson Curator

A copy action that is used in the workflow to automate the management of the data change. Generates manifest and CSV reports to help the users of Watson Curator analyze the data.

Within the Modify Attribute policy, the following options are available within the policy:

Set read-only

Sets the read-only bit on files in an infoset for CIFS and NFS data sources.

Set read/write

Sets the read/write bit on files in an infoset for CIFS and NFS data sources.

Step-up Snippet

The Step-up Snippet is applied against a specific infoset. For each object within the infoset, a short capture of the contents of that object is created. It does not affect the membership of the infoset or its metadata in any way. Step-up Snippet is audited in the harvest audits on the data server. Snippets are commonly used by the auto-classification feature.

Step-up Full-text

The Step-up Full-text index is applied against a specified infoset. For each object within the infoset, a full-text index is created unless the corresponding file on the data source has metadata that no longer matches

the object's metadata. It does not affect the membership of the info set or its metadata in any way. Step-up Full-text is audited in the harvest audits on the data server.

Searching for a specific action

When you have multiple actions available to choose, you can search for a specific action instead of using the slider to move through available options.

Procedure

1. Select the info set on which you want to act. The selected info set appears at the top of the info set window.
2. Click **Act**.
3. In the **Enter key terms...** text box, enter the name of the action for which you want to search, and then press **Search**. You can enter either the full action name or a portion of it. To remove the search term, click the **X** to the left of the **Enter key terms....** text box. Actions that match the entered search term are returned.
4. Click the action that you want to select, and then click **Run**.

Scheduling and running an action

From the **Actions** tab, actions can be scheduled to run immediately or at a predetermined schedule, providing you with greater flexibility and control over your data.

Procedure

1. Within the **Available Actions** area, select the action that you want to schedule, and then click **Run Action**. The Action Scheduling dialog box appears.
2. Within the Action Scheduling dialog box, you must determine an action schedule. Select from these options:
 - **Immediate**: If you select this option, the action is queued to run immediately.
 - **Scheduled**: If you select this option, the action is queued to run according to the schedule you set. In the **On** text box, enter the date, and in the **at** text box, enter the time that this action is scheduled to run.
3. After the action schedule is set, click **Run and View** or **Run**. The action can be seen in the **Details > Execution Log** tab. Actions that were scheduled to run immediately show as **Running** in the **Type** column. Actions that are scheduled to run later are listed as **Queued**.

Canceling a scheduled action

Scheduled or queued actions' executions can be canceled before they run.

Procedure

1. Within the **Details > Execution Log** tab, select the action that you want to cancel.

Note: In order for an action to be canceled, it must be scheduled or queued to run. You cannot cancel an action that is running.

2. In the **Action Details** area, click **Cancel Queued Action**. The Cancel confirmation window appears.
3. Click **OK**.

Chapter 8. Data reports

Data reports help evaluate and verify processes that were run on IBM StoredIQ Data Workbench.

IBM StoredIQ Data Workbench contains these types of system reports:

Compliance Report

Inspects data for compliance with governance policies.

Content Collector Manifest File CSV Export

Exports data and metadata in a user infoset to a modified CSV report for IBM Content Collector to use.

CSV All Audited Objects Export

Lists all audited data objects after the policy is run.

CSV Attribute Summary

Lists a summary of attribute counts for in infoset.

CSV Exception List Export

Lists the found exceptions.

CSV Infoset Data Object Export

Lists the data objects within an infoset.

Data Assessment Report

Assesses data for IT policies.

Data Topology Report

Describes how data is distributed across your organization.

Note: Only the top 100 owners of infoset objects across volumes can appear in the data topology report.

Duplication Summary Report

Lists the duplicate data objects that are contained within an infoset.

Overlay Hit Report

Compares data's responsiveness for various overlays.

Term Hit Report

Shows the number of responsive objects in an infoset as they pertain to a set of full-text terms.

The generated reports are sent to designated users through email or are available as a download link in the **Execution Log** tab for an infoset.

IBM StoredIQ Data Workbench report types

IBM StoredIQ Data Workbench provides the following preconfigured report types.

Compliance Report

The Compliance report indicates whether data is in compliance with established guidelines, specifications, or governance policies. The generated report provides detailed information about data-object compliance in these areas:

- Compliance Overview
- Distribution by Category

- Distribution by Data Source
- Distribution by Size of Data Object
- Distribution by Data Owner
- Distribution by Last Modified Date

Content Collector Manifest File CSV Export

Content Collector Manifest File CSV Export exports path, data attributes, and classification data objects that are contained in an Infoset to a modified CSV report. This CSV report is delivered to the user by email or can be viewed by clicking the **View Report** link in the execution log of the infoset. It is then saved in a location that IBM Content Collector can retrieve and use.

CSV All Audited Objects Export

The CSV All Audited Objects Export report lists all audited data objects after the policy is run. The report includes the following information:

- Name
- Path
- Container path
- Volume name
- Object type
- Owner
- Size
- Ctime
- Mtime
- Atime
- Digest
- Object ID
- Version

CSV Attribute Summary

The CSV Attribute Summary lists the attribute counts for an infoset. The report includes the following information:

- Attribute
- Name
- Count
- Path
- Description

CSV Exception List Export

The CSV Exception List Export report lists the exceptions that are encountered during the execution of an action. The report includes the following information:

- ID
- Source Object Path
- Source Object Container Path
- Source Object Name
- Source Object Version
- Source Object Created Date
- Source Object Last Modified Date
- Destination Object Path

- Destination Object Container Path
- Destination Object Name
- Destination Object Version
- Destination Object Created Date
- Destination Object Last Modified Date
- Category
- Token

CSV Infoset Data Object Export

The CSV Infoset Data Objects Export report lists the data objects within an infoset. The report includes the following information:

- Name
- Path
- Container Path
- Volume Name
- Object Type
- Owner
- Size
- Ctime
- Hash
- Mtime
- Atime

Data Assessment Report

The Data Assessment report assesses data in terms how it interacts with IT policies. The generated report provides detailed information in these areas:

- Distribution by Data Source
- Distribution by Size of Data Object
- Distribution by Data Owner
- Distribution by Last Accessed Date
- Distribution by Last Modified Date
- Distribution by Category

Data Topology Report

The Data Topology Report describes how data is distributed across your organization. The generated report provides detailed information in these areas:

- Overall Summary
- Distribution by Data Source
- Distribution by Category
- Distribution by Data Owner
- Distribution by Last Accessed Date
- Distribution by Last Modified Date

Note: Only the top 100 owners of infoset objects across volumes can appear in the data topology report.

Duplication Summary Report

Lists the duplicate data objects that are contained within an infoset. The generated report provides detailed information in these areas:

- Duplication Overview
- Duplicate Objects

Overlay Hit Report

The Overlay Hit report compares data's responsiveness for various overlays. The generated report provides a detailed analysis about overlay information in these areas:

- Overlay Summary
- Overlay Results

Term Hit Report

Given a series of search terms and a target infoset, the report provides a perspective of how many hits in terms of matching objects each term generates. The goal is for you to quickly access several query options and narrow to the terms that you might be interested to pursue. This report provides the following information:

- Infoset Summary
- Term Hit Summary
- Term Hit by Object Count
- Term Hit by Total Size of Matching Objects

User reports

Use the **User reports** interface to create Custom CSV Export reports. You can select and specify the exact metadata information in the report.

Select a metadata category such as **System metadata** from the list. Then specify and add metadata fields that are to be shown in the report.

Format of CSV reports

The format of the .csv files and manifest files in the reports are described as follows:

- Values are not quoted.
- Values are separated by , (comma).
- There is no whitespace after the comma.
- Lines end with carriage return and line feed characters.

Searching for a report

When you have multiple reports available to choose, you can search for a specific report instead of using the slider to move through available options.

Procedure

1. Select the infoset on which you want to generate a report. The selected infoset appears at the top of the infoset window.
2. Click **Report**.
3. In the **Enter key terms...** text box, enter the name of the report, and then press **Search**. You can enter either the full report name or a portion of it. To remove the search term, click the **X** to the left of the **Enter key terms....** text box. Reports that match the entered search term are returned.

Generating a system report

Provides procedural information for generating a system report.

About this task

If the info set against which you are generating a report was created without a data map, some report types are affected. The Compliance, Data Assessment, Data Topology, Overlay, and Customized Reports contain no data if run against an info set without a data map.

Procedure

1. Select the info set on which you want to generate a system report.
2. Click **Report**.
3. Select the type of report and then click **Generate Report**. The Generate Report dialog box appears.
4. In the Generate Report dialog box, complete these fields:
 - a. In the **Report Name** text box, enter the name of this report. By default, the report is named by the type of report, and the name of the info set.
 - b. In the **Send To** text box, enter the email address to which the generated report is sent.

Note: If you enter more than one email address in the **Send To** text box, separate the email addresses with commas.

5. Click **Generate**. The generated report is sent to the email address listed in the **Send To** text box, or the report can be downloaded from the Execution Log in Details.

Generating a term hit report

Follow these steps to generate a term hit report.

Procedure

1. Select the info set on which you want to generate a report.
2. Click **Report**.
3. Select **Term Hit Report** and then click **Generate Report**. The **Generate Report** dialog box appears.
4. In the Generate Report dialog box, complete these fields:
 - a. In the **Report Name** text box, enter the name of this report. By default, the report is named by its report type and the name of the info set.
 - b. In the **Send To** text box, enter the email address to which the generated report is sent.

Note: If you enter more than one email address, separate the email addresses with commas.

5. Click **Next** to proceed to the Terms page.
6. In the **Term** field, enter a single term and then click **Add Term**.
7. From the **Filter From Library** list, select a filter and then click **Add Filter**.

Note: You can add multiple terms and filters but you can add only a single one each time.

8. **List of Terms and Filters to include in Report** shows all terms and filters that you added. You have options to delete your selections by clicking **Delete** or **Delete All**.
9. Click **Generate** to run the report.

Generating a user report

To create Custom CSV Exports reports, select the **User Reports** tab under **Report**.

Before you begin

Select a user info set on which you want to generate a report.

Procedure

1. Click **Report** to open the Available Reports window.
2. Click the **User Reports** tab.
3. Click **Create**. The Create Report window opens.
4. Name the report.
5. Optional: Enter a description about this report.
6. Click **Next**.
7. Select a metadata category from the list, such as **System metadata**.

Note: Metadata categories that are specific to certain source type appear only as options if the info set has objects on that source type. For example, the SharePoint metadata category displays only if the info set includes SharePoint objects.
8. Select the attribute or attributes from **Available Fields**. Select multiple attributes by combining the mouse click with the Shift key or Alt key.
9. Click **Add** to move the selected attribute or attributes to the **Report Fields** pane.
10. Click **Remove** to delete any report fields that you do not need.
11. Use **Move Up** or **Move Down** to rearrange the sequence of the report fields.
12. Click **Save and Exit**.
13. Select the report on the **User Reports** list and click **Generate Report**.
14. Optional: Enter the email address in the **Send To** field. The report can be downloaded from the **Execution Log** tab in the **Details** window.
15. Click **Generate**.

What to do next

User reports that are never used to generate reports can be deleted from the **User Reports** tab by selecting a report and clicking **Delete**. After a user report is used to generate a report, it cannot be deleted.

For ease of creating a similar user report, you can clone an existing user report by selecting a report and clicking **Clone**. Rename the report and further customize its metadata category and report fields if necessary. Save it as a new user report.

Chapter 9. Request management

The **My Requests** link on the IBM StoredIQ Data Workbench dashboard provides an interface to show the list of requests that you received or responded to so that you can track, audit, or work on them.

The legal user of IBM StoredIQ eDiscovery and IBM StoredIQ for Legal creates boxes to request data from the data expert of IBM StoredIQ Data Workbench. For more information about creating boxes, see *Creating a box* and *Requesting for matter data* under *Managing legal matters* from IBM StoredIQ for Legal documentation on IBM Knowledge Center.

When a box is created, it is placed in a **Pending** state. An email is sent to you, the data expert, requesting that you create infosets to match the box request.

My Requests allows you to manage the requests from IBM StoredIQ eDiscovery and IBM StoredIQ for Legal. When you address the requests by creating the requested infosets and completing the request, the request is set to **Closed**.

My Requests

My Requests displays all requests specifically for the data expert who must address them.

My Requests provides these functions:

- It displays all requests, including both currently received requests and processed requests.
- Through the Requests window, it provides two tabs: **Active** and **Closed**. You can sort the requests by their names and their sent or completed time.

The basic steps to work with **My Requests** are described in the following sequence:

1. Click **My Requests** to open a Requests window.
 - The **Active** tab lists the names of the requests, their status, people who issued the requests, and when they were received.
 - The **Closed** tab lists the names of the requests, their status, people who issued the requests, and when they were acted on. The **View** list provides the options of viewing **All**, **Completed**, **Rejected**, or **Canceled** requests from the **Closed** tab.
2. Click a request item; the window loads the name of the request, the legal user, and the content of the request.
3. If you cannot find a matching infoset from the existing infosets, create an infoset to respond to the requests. For information about how to create an infoset, see “Creating an infoset” on page 9.

All infosets that are listed under **Primary Infosets** are shown as links. Clicking a link opens the infoset's context in the IBM StoredIQ Data Workbench window.

The **Reject** button and **Accept** button are enabled for all **Open** requests.

Clicking **Reject** closes the request item window and sets view to the **Closed** tab. Clicking **Accept** updates the request item window with the **Accepted** status and time stamp.

In the **Fulfillment Notes** field, you must enter notes for the **Reject** action. The request item window does not close if you do not enter notes. Notes for the **Complete** action are optional. Notes are not sent and stored for the **Accept** action.

4. Select the new info set that you created from **Primary Info set** and click **Complete**. The request is back to the default **Request** list window and the **Closed** tab shows the request as **Completed**.
5. You can open the requests from the Closed tab.

If you click a completed request, you see all information about this request: its original request information, its status, and information about the info set that was created for this request. The info set is shown as a link so that you can click it to see its context.

If you open a rejected request, you see only the rejection note and the original request.

Canceled requests store the original requests and time stamps.

User notification

The user notification icon is located at the top of the IBM StoredIQ Data Workbench interface. It becomes active when a new notification arrives.

The user notification icon turns blue when these notifications occur:

- A box request comes from the legal user of IBM StoredIQ eDiscovery.
- A new user info set is complete or fails to be created.
- An action, enhancement, or report execution is complete.

A number on the notification icon displays the count of all unread notifications. If there are more than 99 notifications, the icon displays the notification count as 99+. The items are listed by time that they come in; the latest comes at the top.

If you click a request notification from the list, a window opens to display the details of that item. If you click an info set notification, it opens that specific info set. Clicking the report execution notification takes you to the execution log that has details of the completed execution. Clicking outside the list closes this notification list.

All unread notifications display in dark gray. After you click an unread notification, it changes to the *read* state in light color. Clicking **Mark All Read** marks all notifications that are loaded in the list as *read*.

Chapter 10. Exceptions

Exception lists contain various data objects such as files, emails, which encountered an error when an action was conducted on the infoset. This list helps you to know exactly what data needs attention due to some error or exception. It also serves as an informational trace of what went wrong during the process.

The **Exceptions** tab is divided into three areas: **Event**, **Types**, and **Exception Objects**. When you select an exception within either the **Event** or **Types** areas, contextual details for the selected exception appear within the **Exception Objects** area.

Events

Table 9. Events column definitions

Events Column Heading	Definition
Name	The name of the exception.
Started	The date and time at which the exception started.
Type	The action type.

Types

Table 10. Types column definitions

Types Column Heading	Definition
Name	The name of the exception that was applied to the data objects. Options include Export Error , File Access Error , File Copy Error , Network Error , Other , Permissions Error , Server Error , System Error , or Unsupported Action .
Severity	The severity level of the exception.
% of Infoset	The percent of the infoset with errors.
Count	The number of data objects with an error.
Size	The size of data objects with an error.

Exception objects

Table 11. Exception objects column definitions

Exception Objects Column Heading	Definition
Name	The name of the exception object.
Timestamp	The date and time of the exception.
Event	The run action that resulted in the exception.
Exception	The type of exception encountered.
Message	Based on the exception type, this message provides additional information about the exception itself using information that is pulled from the audit log.

Table 11. Exception objects column definitions (continued)

Exception Objects Column Heading	Definition
File size	The exception object's file size.
File path	The file path to the exception object's location.

Regarding Step-up Snippet and Step-up Full-text: The **Exceptions** tab does not report exceptions for a Step-up Snippet or Step-up Full-text. To see those exceptions, check the harvest audits on individual data servers.

Searching for a specific exception object

When you have multiple exception objects to choose, you can search for a specific exception object instead of using the slider to move through available options.

Procedure

1. Click **Exceptions**.
2. In the **Enter key terms here...** text box of the **Exceptions Objects**, enter the name of the exception object and then press **Search**. You can enter either the full exception object name or a portion of it. To remove the search term, click the **X** to the left of the **Enter key terms....** text box. Exception objects that match the entered search term are returned.
3. Click the exception object that you want to select.

Viewing exception object details

The Exception Object Details dialog box displays exception details that result from different data objects.

Procedure

1. Select the exception objects for which you want to view details.
2. Click **View Details**, and the Exception Object Details dialog box appears. This dialog box displays details about the data object that caused the exception.
3. Click **Next** or **Previous** to view different data objects that caused the exceptions.
4. After you finish viewing exception object details, click **Close**.

Exporting the exception objects list

Use this procedure to export a CSV list of the exception objects that are listed in the **Exception Objects** area.

Procedure

1. Select the event, type, and exception objects that you want to export.
2. Click **Export Exception Objects List**. The Export Exceptions Object List dialog box appears.
3. To send an extra copy of the CSV file to another person, select the **Also send a copy to the following address check box** and enter the email address in the text box. The exceptions object list is exported as a CSV file, and a link to the location is emailed.
4. Click **OK**.

Appendix A. Filter condition reference and expression language

This appendix provides in-depth information about full-text and attribute conditions and how to use the expression language.

Full-text syntax

The full-text editor supports full-text syntax.

The full-text feature supports a wide range of full-text syntax to broaden or narrow filters. A one-word term like `September` returns results from all harvested data objects that include the term in their content. Word matching is case-insensitive. A list of words match any of those words. For example, `September OR October` match all objects that contain either `September` or `October`.

Term expansion

When lemmas are turned off (by default), a simple term such as `went` returns only those documents that contain the exact same character sequence. For more information, see [Configuring full-text index settings](#).

When lemmas are turned on, the simple term `went` is automatically expanded to return documents that contain words with the same meaning as `went`, such as `go`, `goes`, `going`, and `gone`.

Automatic term expansion applies to any terms. Term expansion can be disabled by placing the term within single quotation marks. Term expansion does not occur when a wildcard is present.

Note: To use term expansion for the anticipated results, lemmas must be turned on before the documents are indexed.

Grouping

When you are examining text, IBM StoredIQ Data Workbench evaluates the requested operators and gives precedence to some over others. These rules are described in the following list.

Note: Right-associative and left-associative means that when parentheses are used to group operators, the expression in parentheses is given the highest precedence. If no parentheses are used in the expression, associativity rules are used instead. Right-associative means that the operator is grouped initially with the operand to its right. Left-associative means that the operators/operands to the left are evaluated before the operators/operands to the right.

1. NOT, right-associative. `"October NOT November OR December"` means `"(October AND (NOT November)) OR December"`.
2. Proximity operators (`"w/"` and `"pre/"`), left-associative. `"October w/3 November pre/5 December"` means `"(October w/3 November) pre/5 December"`. (`"October"` and `"November"` must be within three words of each other, but both must be before `"December"` by no more than five words.) `"October w/3 November and December"` means `"(October w/3 November) and December"`.

3. The AND operator, left-associative. "October AND November OR December" means "(October AND November) OR December".
4. The OR operator or no operator, left-associative. "October November OR December" means the same as "(October OR November) OR December".

Boolean Operators

This table lists Boolean operator syntax and descriptions.

Table 12. Boolean operator syntax

Item	Description
AND &&	<ul style="list-style-type: none"> • When used between two terms, the AND operator ensures that both terms appear in the data object. For example, the filter chairman AND stock returns only data objects where chairman and stock both appear. • The double ampersand (&&) can be used instead of the AND operator. AND can be uppercase or lowercase.
OR	<ul style="list-style-type: none"> • When used between two terms, the OR operator returns data objects where either term appears. For example, the filter chairman OR stock returns data objects that have chairman, stock, or both in the data object. • The double pipe () can be used instead of the OR operator. OR can be uppercase or lowercase.
NOT !	<ul style="list-style-type: none"> • When used before a term, the NOT operator ensures that the following term does not appear in any data objects returned. For example, Apple NOT IBM returns data objects where word Apple appears, but not IBM. You can think of the NOT item as "and not" or "but not". • The exclamation point (!) can be used instead of the NOT operator. NOT can be uppercase or lowercase.

Multiple Terms or Term Phrase

This table lists multiple terms or term phrase syntax items and descriptions.

Table 13. Terms

Item	Description
" " and *	<ul style="list-style-type: none"> • Double quotation marks around a term phrase return data objects that have that phrase in order in a data object. For example, "September 20 1998" returns only data objects with the exact phrase of September 20 1998. Double quotation marks allow for wildcards in terms whereas single quotation marks do not. • Stop words are removed from a double-quoted term phrase. For example, stock and option returns documents with stock option. • Wildcards are treated as wildcards. For instance, stop option* returns both stock option and stock options. • Automatic term expansion is performed if lemmas are turned on. For instance, fox jumps returns any of these phrases: fox jumps, fox jumped, foxes jump, and so on.

Table 13. Terms (continued)

Item	Description
' ' and *	<ul style="list-style-type: none"> • Single quotation marks around a term phrase return data objects that have the exact same character sequence in a data object. For example, 'february 4 2013' returns exactly february 4 2013. • Wildcards are treated as regular characters. For instance 'ism' returns exactly *ism. • Stop words are not removed. For instance, 'to be or not to be' returns exactly to be or not to be. • Term expansion is not performed even if lemmas are turned on. For instance, 'went' returns exactly went.
Multi-term, no Boolean operators	A filter for a multiple word without Boolean operators or quotation marks is an implied OR filter. For example, September 20 1998 returns data objects that have September OR 20 OR 1998.
Grouping	<ul style="list-style-type: none"> • Grouping with Boolean operators and parentheses allows users to narrow results by using attributes. For example, (\$10 OR billion) AND options returns data objects where \$10 or billion might be in the data object, but options must exist. • Boolean operators (AND, OR, NOT) work in a grouping or in a proximity group. So ("Sahara Desert" AND "Southeast Asia") OR ("Pacific Ocean" AND "Bavaria") returns data objects that have Sahara Desert and Southeast Asia or Pacific Ocean and Bavaria.

Wildcard

This table lists wildcard syntax items and descriptions.

Table 14. Wildcard

Item	Description
* (Asterisk)	<ul style="list-style-type: none"> • A single asterisk finds multiple characters or letters at beginning of a term. An asterisk that is placed before a partial word finds terms that end with those letters. For example, a filter for *olol returns results for betaxolol, carteolol, levobetaxolol, levobunolol, metipranolol, and timolol. • An asterisk matches any number of characters in the middle of a term. An asterisk that is placed in the middle of a term finds terms that have those characters on either side of the asterisk. For example, te*t returns results for text, test, tent, and tempest. • An asterisk that is placed after a partial word finds terms that begin with those letters at the end of a term. For example, a filter for ap* returns results for Apple, April, applied, and application. • An asterisk alone finds all objects. Note: An asterisk within single quotation marks is treated as a regular character.
?	<p>The question mark matches a single character. For example, compl? returns results for comply but not complies or compliance. A filter for ap? finds ape, but not apple. To filter for more than one character, use a question mark for each letter, for example, ap??e.</p> <p>Note: A question mark within single quotation marks is treated as a regular character.</p>

Proximity

This table lists proximity syntax items and descriptions.

Table 15. Proximity syntax

Item	Description
w/x	<ul style="list-style-type: none">Filtering uses the w/x syntax (x represents the number of terms between the two terms) to use the relationship between terms to get more targeted results. A two-term filter with one term that appears at the beginning of a data object and the other at the end might not be as relevant as when the terms appear in the same paragraph or sentence. For example, in an SEC data object, chairman w/15 options is a more relevant filter than chairman AND options appearing in the same data object that might be several pages long.w/0 matches terms that are next to each other in any order.w/1 matches for terms with one word between them. The terms can appear in any order in the document.Grouping with proximity allows for even more targeted results. For example, ("Sahara Desert" "Southeast Asia") w/5 ("Pacific Ocean" Bavaria) finds either Sahara Desert or Southeast Asia within five terms of Pacific Ocean or Bavaria.Terms can have wildcards or phrases.
pre/x	<ul style="list-style-type: none">The pre/x syntax (x represents the number of terms between the two terms) enables users to further target proximity filters by dictating which term comes before another term in the data object and how far before it is. For example, chairman pre/15 options returns data objects where chairman comes up to 15 terms before options.pre/0 matches terms that are next to each other and in the same order.pre/1 matches terms with at most one word between them and in the same order.Terms can have wildcards or phrases.
w/p	The w/p syntax enables users to find terms within the same paragraph. The terms can appear within the paragraph in any order.
pre/p	The pre/p syntax enables users to find terms within the same paragraph. The terms must appear in the specified order.
w/s	The w/s syntax enables users to find terms within the same sentence. The terms can appear within the sentence in any order.
pre/s	The pre/s syntax enables users to find terms within the same sentence. The terms must appear in the specified order.

Full-text macros

The IBM StoredIQ Platform full-text filter supports a set of predefined macros.

A macro is an abbreviated set of instructions that are used in common filter patterns. For example, the filter {SSN} finds any document that contains Social Security numbers.

Full-text macros have the following characteristics:

- A macro is an alphanumeric name that is enclosed in curly braces, for example, {SSN}.
- A macro's name is case insensitive. For example, the macro {ssn} is the same as the macro {SSN}.

- The macro's value is replaced in the macro's full-text query. Then, the filter is parsed and run.
- Any unrecognized macro results in a syntax error.
- Regular expressions, that is named character classes, do not match macros. That is, a regular expression like `\p{Lower}` (matches any lowercase ASCII character) is not recognized as a macro.

The following macros are defined within the full-text feature of IBM StoredIQ Platform. Each of these macros is a high-confidence macro. It means that the filter `{SSN}` finds any document that contains Social Security numbers with high confidence that the matches really are Social Security numbers. To find anything that might be a Social Security number, use the low-confidence macro `{SSN-1c}`.

Macros can be combined with any other full-text expression, such as `{SSN} w/5 "John Doe"`.

- `{SSN}`: Social Security numbers, which can include dashes and numbers
- `{USPhone}`: US phone numbers, which can include dashes, dots, parentheses, numbers, and letters. This macro can filter for mnemonics like 1-800-flowers.
- `{InternetAddress}`: Legal IP addresses, either as numbers or as names, which can include dots, numbers, and letters.
- `{EmailAddress}`: Legal email addresses, which can include @, dots, numbers, and letters.
- `{CCAmex}`: American Express credit-card numbers, which can include dashes and numbers.
- `{CCDiners}`: Diners Club credit-card numbers (including enRoute credit-card numbers), which can include dashes and numbers.
- `{CCDiscover}`: Discover credit-card numbers, which can include dashes and numbers.
- `{CCJCB}`: JCB credit-card numbers, which can include dashes and numbers.
- `{CCMC}`: MasterCard credit-card numbers, which can include dashes and numbers.
- `{CCVisa}`: Visa credit-card numbers, which can include dashes and numbers.
- `{CreditCard}`: Any credit-card number from the listed providers.

Extended ASCII characters

Extended ASCII characters are handled several different ways in the full-text filter. Some are special characters while others are treated as spaces. Some are treated as letters that depend on the spaces around the characters.

Special characters or punctuation are not indexed and cannot be searched for in filters. The following table describes what behavior applies to these characters in the filter.

Table 16. Extended ASCII Character syntax

Extended ASCII Character Syntax	Description
Special characters	<p>Some special characters generate syntax errors because they cannot be processed. Either remove the character from the filter or surround the filter with single quotation marks. For example, a filter for hello! without quotations generates an error. However, if the filter is 'hello!' (with single quotation marks), the query can be processed.</p> <p>In general, special characters are treated as a space in a filter and they are not indexed. For example, Hello! is indexed as Hello. Searching for 'Hello!' returns zero object. To search for 'hello', data object that has the word hello! or hello is returned. However, there are some exceptions, and they are discussed later in this table</p> <p>Most special characters can also be searched for with quotation marks. For example, "12/24/2014" finds 12/24/2014. The difference between quotation marks and single quotation marks is that quotation marks also have another function: to find words that are next to each other. Because of this, special characters that are used elsewhere in the syntax, such as * and ?, are treated as the full-text operators for wildcards instead of as the characters * and ?.</p> <p>These special characters generate errors unless surrounded by quotation marks: @ / & : ! () { } [] ^ ~ , # =</p>
Period (.)	<p>A period with no space on either side is treated like a letter. For example, a filter for 172.16.0.46 finds the whole number of 172.16.0.46, not partial numbers of 172, 16, 0, 46.</p> <p>A period with a space on either side is treated as a space and ignored. For example, a filter for "1234 Mahana St." ignores the period and just finds 1234 Mahana St only without a period.</p>
At (@)	<p>An @ symbol is indexed like a letter when there are no spaces on either side. For example, a filter for jdoe@yourcompany.com finds the entire email address. An @ symbol with a space on either side or one side is treated as a space and ignored. For example, @home finds home in the data object.</p>
Hyphen (-)	<p>A hyphen is indexed like a letter when there are no spaces on either side. For example, a filter for 012-44-5668 finds the entire number, not 012, 44, 5668 as separate numbers. A hyphen with a space on either side is treated as a space and ignored.</p>
Slash (/)	<p>A slash is indexed like a letter when there are no spaces on either side. For example, a filter for 10/30/1940 finds the entire expression 10/30/1940 (and not 10 30 1940) as separate numbers.</p>
Ampersand (&)	<p>An ampersand is indexed like a letter when there are no spaces on either side. For example, a filter for S&P finds S&P in the data objects. However, a filter for Proctor & Gamble without quotation marks finds either Proctor or Gamble. A filter with quotation marks "Proctor & Gamble" finds the phrase Proctor Gamble only.</p>
Colon (:)	<p>A colon is considered a special character. It is treated as a space in filter and is not indexed. For example, 21:35:15 is indexed as three consecutive words, that is, 21, 35, and 15. Searching for '21:35:15' returns zero object. To search for '21 35 15', data object that has the word 21:35:15 or that has three consecutive words as 21, 35, 15 is returned. Specifying terms with colons without the single quotation marks results in an invalid syntax error.</p>

Table 16. Extended ASCII Character syntax (continued)

Extended ASCII Character Syntax	Description
Apostrophe (')	The apostrophe in the term is indexed as a whole word. For example, a filter for patient's returns results for patient's but not for patient. A filter for the name O'Hara returns results for O'Hara. An apostrophe is treated as a single quotation mark, so it must always be quoted with double quotation marks.
Comma (,)	Commas are left in when surrounded by numerals, and those numerals make a legitimate number. For example, "1,000" is indexed as 1,000. To search this term, specify 1,000 in single quotation marks or quotation marks. Otherwise, it results in an invalid syntax error. For the term "10,00", it is indexed as 10 and 00. To search for this term, specify "10 00".
Number sign (#), dollar sign (\$), percent (%), equal sign (=), underscore (_)	These characters are treated as white space in the filter. For example, a search for \$100 returns results for the number 100 whether 100 refers to dollars, a percentage, or a whole number. The term 'JAVA_HOME' is indexed as 'JAVA HOME'. To search for this term, specify JAVA HOME in quotation marks.

Attribute condition reference

The following section explains how to type in a reference to an attribute condition.

If you always select an attribute from the custom filters or one of the attribute groups, xxx takes care of the attribute reference for you. If you do not select an attribute from the custom filters or one of the attribute groups, then you need to review and use this section.

A typed attribute reference is composed of several parts: attribute-type path/attribute-name. The attribute-type gives the specific type of attribute to which the path and attribute-name refer. This prefix is a short prefix of the attribute type. Attributes can have duplicate names, even within one folder.

Table 17. Attribute type prefix

Attribute-type Prefix	Description
"att:" or "@"	This prefix is a system-defined attribute or user-defined entity.

The "path" describes the folder path to the attribute. Generally, it is optional, but it is required if the attribute name alone is not unique. You do not need to type the full folder path to an attribute.

If an attribute name or folder path name has spaces or punctuation in it, then you must enclose the name in double or single quotation marks. However, you do not need to quote the forward slash (/) between folders or before the attribute name. It is not legal to have part of an attribute name quoted; all of it must be quoted. You must never quote the attribute type prefix.

Specification of attribute conditions

Attributes, entity, tag, or query can be divided up in several ways.

- What type of values does the attribute have? Attributes can have values of number, string, enumeration, boolean, or date. An enumeration attribute is an attribute that has a limited set of possible values, which are shown in a list. When an attribute is a compound attribute, each of its subattributes also has a value type: string, enumeration, boolean, or date. The subattributes can also be either sparse or dense. Currently, all subattributes are single-valued. Compound attributes can also be sparse or dense. (Currently all compound attributes are sparse.)
- Do all objects have a value for this attribute? If the objects all have a value, it is known as a dense attribute. If the objects do not have a value, it is known as a sparse attribute. All objects have an Object name value, so Object name is a dense attribute. Only some objects have a CC attribute value, so CC is a sparse attribute.
- Can an attribute have multiple values within one object? Each object has one name, so Object name is a single-valued attribute. There might be multiple To values on an email, so To is a multivalued attribute. Compound attributes can be single-valued or multivalued. A compound attribute never exists where all of its subattribute values are missing.

These condition operators are applicable to each attribute category.

Note: For illustration purposes, some operators are shown in uppercase, and some are shown in lowercase. The parser recognizes operators in either uppercase or lowercase.

Conditions for string attributes

Tags are considered single-valued, sparse string attributes. They are not considered enumeration attributes, even though they have a limited set of possible values.

```
attribute = value or attribute is value
attribute != value or attribute is not value
attribute contains value
attribute does not contain value
attribute starts with value
attribute does not start with value
attribute ends with value
attribute does not end with value
attribute matches value
attribute does not match value
```

The value can be a single word, a double-quoted string, or a single-quoted string. For matches, the value is a regular expression within quotation marks.

Conditions for enumeration attributes

Some attributes have a limited set of predefined values that they can match, which are called enumeration attributes.

In this case, only '=' and '!=' operators are supported.

```
attribute = value or attribute is value
attribute != value or attribute is not value
```

A list of enumeration attributes is as follows:

- Container type

- Is-part-of
- Box reason
- Encryption type
- System processing state
- Content processing state
- Binary processing state
- System error code
- Content error code
- Binary error code
- Message type
- Latest review status
- Object category

Conditions for date attributes

The conditions for date attributes are listed as follows.

`attribute = value` or `attribute is value`

`attribute != value` or `attribute is not value`

`attribute < value`

`attribute <= value`

`attribute > value`

`attribute >= value`

`attribute between value1 and value2`

`attribute not between value1 and value2`

The value can have one of two forms: an explicit date or a relative date that is offset from today.

An explicit date is in the form of YYYY-MM-DD, for example, 2009-03-08.

- Other date formats are supported.
- Time is not supported.
- The dates are all interpreted as midnight of the listed date.

A relative date offset is specified as a positive number, followed by a required date units, and the ago or ahead keyword. The ago keyword is the default and means a date in the past. The ahead keyword means a date in the future. For example, 3 months ago, 3 days ahead, or 3 weeks ahead are legal relative dates. The allowed date units are as follows:

- days
- weeks
- months
- years

This function is also supported:

- -3 days = 3 days ago
- +3 weeks = 3 weeks ahead

Note: Months are considered as 30 days; years are considered as 365 days.

- Normally, relative date offsets are in the past, but some dates can be in the future, such as a Retention expiration date. For these references, use the ahead keyword instead of ago.
- Relative date offsets can also be given as a positive or negative number, such as -3 days or +3 months without the ago or ahead keyword. In this case, negative numbers are considered in the past and positive numbers are considered in the future.

Conditions for number attributes

Conditions for number attributes are listed as follows.

```
attribute = value or attribute is value
attribute != value or attribute is not value
attribute < value
attribute <= value
attribute > value
attribute >= value
attribute between value1 and value2
attribute not between value1 and value2
```

The value is a positive or negative integer. Size attributes (Size and Size (on disk)) can take an optional size unit after the number. These amounts are the legal size units.

- b or bytes for single bytes
- k or kb for kilobytes
- m or mb for megabytes
- g or gb for gigabytes
- k or kb for kilobytes

Note: The default size unit is bytes. Size units are based on multiples of 1024, not 1000.

Event duration can have time units. These time amounts are the legal size units:

- m or minutes for minutes
- h or hours for hours
- d or days for days

Conditions for sparse attributes

The conditions for multivalued and single valued sparse attributes are listed as follows.

```
attribute APPEARS
attribute NOT APPEARS or attribute DOES NOT APPEAR
```

These conditions specify whether the attribute has any values for matching objects. In this case, it means that one of the subattributes has a value on this object. Tags are considered single-valued, sparse string attributes.

It is not an error to specify these conditions on dense attributes. However, if you do, the parser warns you that the condition is always true or false. These operators can also apply to compound attributes. In this case, it means that one of the subattributes has a value on this object.

Note: As a backward compatible measure, if `att: attribute` appears is given on a dense, boolean attribute, it is converted into `att: attribute = true`. If not appears is given, it is converted into `att: attribute = false`.

Conditions for compound attributes

Conditions for compound attributes are listed as follows.

```
attribute WHERE <sub-attribute-condition>
attribute WHERE (<sub-attribute expression>)
```

A `<sub-attribute-condition>` is just like any of the attribute conditions that are described except it is on a subattribute of the compound attribute. Additionally, a `<sub-attribute-condition>` can be just the subattribute name. It is equivalent to the subattribute **APPEARS**.

A `<sub-attribute-expression>` is any combination of `<sub-attribute-condition>`'s combined with **AND**, **OR**, or **NOT**. This expression must be enclosed in parentheses. The `<sub-attribute-expression>` can also be empty. In this case, it is the same as attribute **APPEARS**.

This example uses **WHERE** expressions:

```
att: "Site owner (SharePoint)" WHERE name = "John Smith"
att: "Site owner (SharePoint)" WHERE (name = "John Smith" AND url APPEARS )
att: "Site owner (SharePoint)" WHERE (name = "John Smith" OR url NOT APPEARS)
att: "Site owner (SharePoint)" WHERE ( )
att: "Collaborator (Box)" WHERE "Collaborator Role" = "editor"
```

Note: This expression is the same as `att: "Site owner (SharePoint)" APPEARS`.

```
att: "Site owner (SharePoint)" WHERE (name)
```

Note: This expression is the same as `att: "Site owner (SharePoint)" WHERE name APPEARS`.

Conditions for multivalued attributes

Conditions for multivalued sparse or dense attributes are listed as follows.

```
attribute APPEARS operator number
attribute APPEARS UNIQUE operator number
```

These operators can also apply to compound attributes, and the operator can be one of the following operators:

- = >
- >=
- <
- <=

The operator number specifies a condition on the number of values present for the attribute on any matching object.

- If operator number is not given, then > 0 is assumed.
- If **UNIQUE** is given, it specifies the number of unique values present for the attribute on any matching object.

It is not an error to specify these conditions on single-valued attributes. However, if you do, the parser warns you that the condition is always true or false.

Access mode attribute

Specifies an attribute that can be used in a search expression for creating infosets that are based on certain file attributes that are collected during harvests.

Access mode attribute supports these operators: APPEARS, NOT APPEARS, and WHERE.

The Subject subattribute supports these operators: IS, IS NOT, =, !=, any(), and all().

The Readable and Writable subattributes both support these operators: IS, IS NOT, =, and !=.

For example,

- To find data objects for which you collect or maintain information about the read or write file system flags, enter `att:"Access Mode" appears`. This search returns every system-level object on CIFS or NFS volumes.
- To find data objects that are writable by either of the three types of subjects, enter `att:"Access Mode" where (Subject is any (Owner, Group,Others) and Writable is True)` or `att:"Access Mode" where (Writable is False)`. These two examples are semantically identical. If you think of the r/w/x flags as separate groups of flags (that is, `rwX | rwX | rwX`), where one is for the Owner, one is for the Group that a user might be associated with, and one is for Others in that order, then the Readable or Writable subattributes are evaluated independently for each three group of flags.
The Subject subattribute (if present) then just defines which group out of the three must be considered or ignored. The lack of the Subject subattribute indicates that the objects must be considered responsive to the expression if the Readable or Writable conditions (or any combination of them) evaluate to True in ANY of the three groups.
- To find data objects that are read-only by everyone, enter `att:"Access Mode" where (Subject is all (Owner, Group, Others) and Readable is True and Writable is False)`. The subattributes are all ANDed together and the Subject subattribute lists every possible option with the all qualifier.
- To find data objects that are read-only by at least one type of subject, enter `att:"Access Mode" where (Subject is any (Owner, Group, Others) and Readable is True and Writable is False)` or `att:"Access Mode" where (Readable is True and Writable is False)`. Again, the two examples here produce identical results for the same reason as explained in the previous bullet point.
- To find data objects that are read-only by the owner, enter `att:"Access Mode" where (Subject = Owner and Readable is True and Writable is False)`. You care only about the owner in this case, so you explicitly call Owner and only Owner out.
- To find data objects that owner can either read or write (not necessarily both), enter `att:"Access Mode" where (Subject = Owner and (Readable is True or Writable is True))`.
- To find data objects that can be written by the owner but is read-only by the group, enter `att:"Access Mode" where ((Subject = Owner and Writable = True) and (Subject = Group and Writable is False))`.

Type of object to match

For any attribute condition, you can specify what object that condition must match.

You can specify IN object-type after the expression. For example, in `att:Extension = doc=in system`, the object must be a doc file at the system level. The condition-scope can be any of these object types here.

Table 18. Available object types

Available Object Types	Description
all	The matching object can be any type of object. This object is the default.
container	The matching object must be a container object that can have other objects nested inside it. File-archive, email-archive, email, calendar, contact, and task objects are included.
simple	The matching object must not be a container object.
system	The matching object must be a system-level object.
contained	The matching object must be nested within some other container object.

Only one condition-scope can apply to any condition. If you give multiple condition-scopes, the parser issues a warning.

Filter condition reference and expression language

This appendix provides in-depth information about full-text and attribute conditions and how to use the expression language.

Multiple matching values for operators

You can give multiple matching values that use the following operators.

- =
- !=
- contains
- not contains
- matches
- not matches
- starts with
- not starts with
- ends with
- not ends with

To specify multiple matching values for these operators, give the keyword `any` or `all` after the operator, then a list of values in parentheses. The `any` keyword means that any of the values are a match. The `all` keyword means that all the values must be present. The `all` keyword is only meaningful for multivalued attributes. For example:

Table 19. Keyword examples

Example	Result
<code>att:"System metadata/Owner" = any (joe, bob, sally)</code>	Owner can be joe, bob, or sally.
<code>att:Size = any (1mb, 2mb, 1gb)</code>	Size can be 1mb, 2mb, or 1gb.
<code>att: "Modified date" = any (1 week ago, 2 months ago)</code>	

Table 19. Keyword examples (continued)

Example	Result
att: "Accessed date" = any (2009-09-19, 2009-09-20, 2009-09-21)	
att:To = all (joe, bob)	"To" must contain both joe and bob

For any, the operator must be true for at least one of the values. It is as if the user spelled out attribute oper val1 OR attribute oper val2 OR ...

- For all, the operator must be true for all the values. It is as if the user spelled out attribute oper val1 AND attribute oper val2 AND....
- For the NOT various operators, this syntax is the same as using NOT around the entire expression and the corresponding positive operator. For example, att:Size != any(1mb, 2mb, 2gb) is the same as NOT (att:Size = any(1mb, 2mb, 1gb)). The NOT matches all documents that do not match the positive operator.
- It is not an error to specify all on single-valued attributes. However, if you do, the parser warns you that the condition is always false.

At-least expressions

On occasion, it is helpful to specify that some number of conditions must be true before an object is considered to match. These expressions are called at-least expressions because at least X subexpressions must match an object before the object matches the overall expression.

The syntax for an at-least expression is to give the keywords "at least" followed by a number, and then by a list of expressions in parentheses. For example:

```
at least 2 (att: size=1mb, att: "system metadata/owner"=joe, att:
"object name"=foo.doc, att: "modified date">2009-03-28 )
```

This example specifies that at least two of the four conditions must be true before any object is considered to match the overall expression.

Each subexpression in the list can be arbitrarily complex, but normally they are one condition. Each subexpression in the list must be separated from the others with a comma.

Parent and child conditions

This parent and child conditions represent advanced function that must be undertaken only by experienced administrators.

You might want to filter for objects whose parents or children match certain conditions. For example, find all emails to Joe in a saved-emails.pst file or find emails to Joe that have an attachment that is named big-contract.pdf. To specify these kinds of queries, give the conditions for the matching objects as normal and add a prefix parent or child to the conditions that apply to the parent or child objects. Here are the expressions for the two previous examples:

```
att:"any address"=Joe and parent att: "object name"="saved-
emails.pst"
```

```
att:To=Joe and child att:"object name"="big-contract.pdf"
```

If there is more than one expression that must match the parent or child objects, you must put all the conditions inside parentheses, for example:

```
att:"any address"=Joe and parent(att: "object name"="saved-
emails1.pst" or att:"object name"="saved-emails2.pst")
```

```
att:To=Joe and child (att:"object name"="big-contract.pdf" and
att:"modified date">2009-09-07)
```

It is also possible to specify conditions on the object level of parent or child objects. An object level is the nesting level of an object. System-level objects are at level one, immediately nested objects are at level two, and so on. For example, if you want to find the non-nested emails to Joe in saved-emails.pst, you can do:

```
att:"any address"=Joe and att:"object level" = 2
and parent
att:"object name"="saved-emails.pst"
```

It specifies that the emails to Joe must be at object level 2 and be inside saved-email.pst.

It is also possible to specify the object level of the parent or child object relative to the other. For example, to find the parent object of any file that is named nested-file.doc, you can specify it this way:

```
child att: "object name" = "nested-file.doc" parent-level = child-level - 1
```

It means that nested-file.doc can itself be at any level, and the query results are the direct parent of those files.

The general form of relative parent-child level conditions is listed here.

```
parent-level operator child-level +|- number
child-level operator parent-level +|- number
parent-level operator child-level +|- number
child-level operator parent-level +|- number
```

- Plus (+) in the +|- indicates that the parent-child must be more deeply nested than the other objects.
- Minus (-) means the parent-child must be less deeply nested than the other objects.

The operator can be any of these comparisons:

Numeric Comparisons for Operators					
=	!=	<	<=	>	>=

No-inheritance conditions

The no-inheritance conditions represent advanced function that must be undertaken only by experienced administrators.

Normally, the search engine allows some objects to inherit the values of their parent objects. Most of the time this behavior is intuitive, but there are times when this behavior must be turned off. When attribute inheritance is not wanted, prefix the expression with noinheritance, and attribute inheritance is turned off for that expression. For example, this expression finds emails to Joe containing stock option in the body or any attachment:

```
att: To=joe and "stock option"
```

However, if you want to find emails to Joe with stock option in the email body itself, then you must turn off inheritance for that part of the expression:

```
noinheritance ( att: To=joe and "stock option" )
```

Analytics filter expressions

The analytics filter expressions represent advanced function that must be undertaken only by experienced administrators.

Query analytics provides a way to filter objects with certain values out of the query results. These filter expressions are specified as part of the query expression. To specify filtering, you specify the "filtered" keyword after the query expression. For example, query: StockOptionMatterBoxJoe filtered. It uses whatever filters are saved for the StockOptionMatterBoxJoe query. The StockOptionMatterBoxJoe query must have analytics that are associated with it.

It is also possible to exactly enumerate what filters must be applied to an analytics query. You specify this query by specifying what dimension values must be included or excluded. Here is the general form:

```
query: queryName filtered on include dimension=value, exclude all  
dimension=value, exclude unique dimension=value...
```

Filters come in three types: included (include), excluded (exclude all), or excluded if the values are unique (exclude unique). If no dimension value is given for include, exclude all, or exclude unique, then the saved filters of that type is used. For example, if the query expression were query: queryName filtered on include att:extension=doc, then only the objects with extension=doc is in the results. Additionally, any objects that were explicitly excluded with exclude-all or exclude-unique are filtered out.

The value for a dimension-value can also be none or other.

- The none keyword means those objects that do not have that dimension-value. For example, query: queryName filtered on include tag: department=none. This query would show only those objects that do not have some tag-value from department on them. (This case is the same as tag: department not appears.)
- The other keyword means those objects that have some other value than what is shown in the analytics for the dimension. Analytics typically show the top 10-25 values for any dimension. If you want to see those objects that have a value other than what is shown in the analytics, you can use an expression like query: queryName filtered on include att: extension=other. It shows those objects with an extension that is not shown in the analytics.

Appendix B. IBM StoredIQ discovery export

The discovery export feature provides an iterative cycle of harvesting data across volumes for generating Electronic Discovery Reference Model (EDRM) XML, Concordance Delimited ASCII Text (DAT), or Concordance DAT (Light) file exports for loading into third-party litigation review software.

This feature consists of a copy policy that copies harvested files from retention and runs and saves this data to a discovery export volume within discovery export. The resulting files detail the location of content, original and duplicate files, and key metadata that regards the content according to entered search terms and litigation needs. Native files and, optionally, their plaintext equivalents, are included to describe the content.

Concordance DAT load file

The Concordance DAT load file can be imported into a Concordance database.

The Concordance DAT export produces the following files:

DAT config file

The `siq_dat_config.CSV` provides the appropriate database schema that must be used for importing generated DAT files into Concordance.

Concordance load files

The `siq_load_file.dat` contains Concordance DAT delimited text records for each data object. The `siq_tag_file.gat` contains tag information about which objects in the DAT load file were deemed responsive.

Run files

The XML audit trail outlines the file system metadata for the source and discovery export volumes and the chain of custody for the data objects. Other files include a CSV report about all load files that are exported for the given run and a CSV user-ID-to-name mapping file.

Concordance DAT (Light)

The Light version of the Concordance load file differs from the Concordance DAT in the material that is exported from IBM StoredIQ.

For Concordance DAT (Light):

- Data-object text is not embedded in the load file, but it is instead provided in a separate text file. It is named with the DocID.
- Native files are named with their DocIDs.
- Container-folder hierarchy is not maintained. Email attachments are not placed in folders apart from their emails, and container members do not reside in folders separate from their parents.
- The first row of the load file includes the column names that are used in the load file.
- Folder contents are limited to 10,000 items.
- Load files are limited to 50,000 objects.
- Dates are provided with GMT as the default. Dates are formatted in MM-DD-YYYY. The following Date fields are provided:

- Date, specifically the main date for the object: the sent date for an email and the modified date for a non-email.
- Email Sent Date
- Email Received Date
- Modified Date
- Created Date
- The object name is provided in a separate column.
- Objects must always be grouped into separate folders by custodian.
- Email text files include the following header information: From field, To field, CC field, BCC field, Subject field, Sent Date.

EDRM XML

The EDRM XML load file can be mapped to a third-party legal review application to import all data pertinent to a legal matter.

These types of files can be mapped:

- Native files: The original source files, such as doc, ppt, txt, and optionally, plain-text versions of the original source files.
- EDRM XML load files: EDRM XML files that fully describe each data object that is produced in a discovery export run.

During the harvest procedures, the system indexes and classifies all of the files in a volume and every file is evaluated for duplicates, content-based attributes, and metadata. Content deduplication is performed automatically while it retains all document instance metadata.

An EDRM XML discovery export policy then copies objects responsive to the classification for the given legal matter and produces the EDRM XML load file. The EDRM XML describes each exported document in terms of where it was sourced from, where the corresponding native files were copied to, and important metadata about the document. The policy is incremental, in that metadata for a document instance cannot be repeated in subsequent export policy cycles unless that data changed.

IBM StoredIQ Platform captures various levels and types of metadata to provide a rich, multiview approach to a document. The information that is captured in the EDRM XML load file conforms to the EDRM XML v1.1 format and includes this information:

- Export information, which is information about the legal matter that is tied to the policy.
- Batch information, which is information about the run that exported the load file.
- Document metadata. For each document in the load file, all metadata in the EDRM XML v1.1 specifications and the attributes in the EDRM XML v1.1 Metadata Tags sheet are described when applicable. These metadata and attributes are available from <http://www.edrm.net>. In addition, metadata that is defined by IBM StoredIQ is also exported for each document to enrich the metadata available to legal review tools.
- Source location information. The source location where each document was sourced and the Custodian that is attached to that copy of the document is also described in the load file in conformity with the EDRM XML standards.

- Native copy location information. For the legal review tools to locate the individual copies of the native files that are produced for each document, the EDRM XML describes where on the target volume these native files were copied.
- Run files, which are files that cover the entire run. These files include the IBM StoredIQ Platform audit trail report in XML format, a CSV file, a CSV report, and a CSV user-ID-to- name mapping file. The CSV file details all of the EDRM XML tag definitions applicable to the produced load file. The CSV report is about all load files that are produced for the run.

Result directory and EDRM XML load file

Once a policy has completed its run, certain objects are placed within the result directory.

The directory structure of the information provides ease of review and use for manual and review tool evaluation of harvested data. Each policy definition includes a destination volume, the configuration used and destination directory under the export volume to which to copy to (optional). A directory, named after the discovery export, is created on the export volume under the specified destination directory, and all runs for the policy export data under this directory. With each export policy execution, a new discovery export run directory is created under the directory using the sequence number of the run for a name. After the completion of a policy run, the destination directory includes the following:

- The run directory holds all results for each export policy run. These directories include a documents subdirectory to hold all of the native files that were exported for the run. However, if the export was configured to deduplicate by custodian, there would be a directory named after each custodian under the run directory, and each of these in turn would have its own documents subdirectory.
- Inside each documents subdirectory, there is another level of subdirectories with numbers for names to handle exceedingly large amounts of files. Each numbered subdirectory holds up to 10,000 files.
- The documents subdirectory, that is, native files directory, includes the copies of the original source objects and the plaintext for each object (optional) in the same directory as the native copy.
- Destination plaintext files are named using the hash of the content and the file-type suffix, such as 0x7bd1b96d357e7f21d243851653517c889d6605d5.txt. The native files retain their original names from the original source.

File system directory layout

The following Discovery Export Parameters have a significant impact on the content of the EDRM XML run export.

- Destination Directory
- Limit Export process to number of Data Objects
- Save a text copy (that is extracted from data files, otherwise known as plaintext)
- Unique ID prefix, Number of Digits, and Pad Zeros

Source volume files

The following examples are based on harvesting the source volume \\qa01\groggers, starting with the directory home for one of two users: fl exus.

Although all directories cannot be seen, this list is the complete list of source volume files and directories:

- home

- home/flexus
- home/flexus/backup
- home/flexus/backup/carlota.zip
- home/flexus/backup/Spaulding.doc
- home/flexus/carlota.zip
- home/flexus/jbowers_identity.doc
- home/flexus/mybcc
- home/flexus/mybcc/to_jbowers.msg
- home/flexus/Spaulding.doc
- home/jbowers
- home/jbowers/Copy of identity.doc
- home/jbowers/from_flexus.msg
- home/jbowers/identity.doc
- home/jbowers/philosophy.txt
- home/flexus.pst

Interesting features of the source documents include the following files and directories:

- The file home/flexus/carlota.zip contains two text files, friends.txt and enemies.txt.
- The directory home/flexus/backup contains copies of two files that are found in home/flexus.
- home/flexus/mybcc/to_jbowers.msg contains a saved email message that flexus sent to jbowers. That message has a copy of Spaulding.doc as an attachment.
- home/flexus.pst contains two emails, one with subject Letter from Spaulding to lawyers and the other with the subject Philosophy of Life. Each email further has an attachment; the former has the attachment Spaulding.doc, and the latter has the attachment identity.doc.
- home/jbowers/Copy of identity.doc is a copy of home/jbowers/identity.doc. The contents of both are the same as home/flexus/jbowers_identity.doc.
- home/jbowers/from_flexus.msg contains the same email message as home/flexus/mybcc/to_jbowers.msg.

EDRM XML Production provides examples that illustrate how the discovery export feature exports files and metadata.

EDRM XML production:

The following section describes the details about the files that are generated for an EDRM XML discovery export.

Note: The discovery export action identifies duplicate files that are based on content-based digest and only exports one copy of a file per data server. The duplicate files are identified in the audit trail.

A run creates directories and files on the discovery export volume. All files for a run are located under a discovery **export-** and run-specific directory on the destination volume under the destination directory.

For each subsequent run, a new subdirectory is created. The directories are named by the run numbers 2, 3, 4, and so on.

Any number of discovery exports can use the same export directory. The output is kept separate by the use of the discovery export name, such as case_123, as the parent directory to the run directories.

This example configuration uses the following information:

Target Volume	auto-P
Destination Directory	exports
Document ID Prefix	NS, number of digits 6, with 0 padding

When a discovery export policy is configured to export EDRM XML, it exports two types of output files:

- **Native Files:** Copies of original source files such as doc, ppt, txt, and, optionally, plain-text versions of the original source files. The native files are found in the documents subdirectory.
- **Run Files:** Files that cover the entire run. These files include the EDRM XML run siq load file, the audit trail xml, the EDRM tag definition list, a csv load file report, and a user-ID-to-name csv mapping file. These files are found at the top of the discovery export run subdirectory.

The run directory contains these files:

- `siq_audit_trail.xml`: The audit trail for the discovery export policy execution. It shows the details of each source and destination object. Essentially, it is an XML encoding of the information available using the IBM StoredIQ interface to the audit trail.
- `siq_edrm_tag_definitions.csv`: This file is a csv file that lists all the tags in the EDRM XML load file that is produced by IBM StoredIQ and related information. The file has the same contents as the table of metadata tags in “EDRM XML metadata tags” on page 74.
- `siq_load_file-0-0.xml`: This file is an instance of an EDRM XML load file that is produced in the run.

Note: For performance reasons, more than one instance of an EDRM XML load file can be produced during a single run.

- `siq_loadfile_report.csv`: This csv file lists all load files that are produced by the run and status information that states whether they are valid for a load or not. The file also reports the range of document IDs produced in each of the load files and other metadata information such as size and date created.
- `siq_userid_map`: This csv file contains a list of user IDs called **uids** in the NFS domain and **SID** in the Windows Share domain with corresponding user names that map to the user IDs. The user IDs and the user names come from the produced native files for the run.

Under the documents subdirectory, an intermediate directory is named 1 that contains the files. To avoid putting too many files and directories in a single subdirectory, the files are placed into numbered subdirectories that contain no more than 10,000 files and directories.

In an EDRM XML discovery export, containers are always expanded to clearly articulate their internal structure. A directory structure is built with each level named after the container and any intermediate containers therein. The path to a

data object within the container is achieved by flattening the structure on the file system. You can see this path by looking at the directory structure.

For instance, in this image, the data objects `enemies-0.txt` and `friends-1.txt`, which are contained within the `carlota.zip` container, are flattened out under a directory named `carlota.zip.members`. Their corresponding plain-text files are next to them in the same directory.

Plaintext file is optional, but when enabled, all documents have a corresponding plain-text file. The plain-text files are named according to their SHA-1 hash value and suffix, which is `.txt`.

The complete list of files and directories for the EDRM XML export for the example is as follows:

```
//qa1/auto-P/exports/edrm global/  
//qa1/auto-P/exports/edrm global/1/  
//qa1/auto-P/exports/edrm global/1/documents  
//qa1/auto-P/exports/edrm global/1/documents/1  
//qa1/auto-P/exports/edrm global/1/documents/1/  
0x724b937d5a2750038325c8f4b48d1b5dcba77853.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/  
0xf4dd8fe9b09c834a2033849ff8a5d63f67823787.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/jbowers_identity.doc  
//qa1/auto-P/exports/edrm global/1/documents/1/Spaulding.doc  
//qa1/auto-P/exports/edrm global/1/documents/1/  
0x8d1e7cef561d7d81585af82d9f79f8a703bd96ee.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/  
0xd7eb7675f2d201c79cf821debaaeba5a208cba29.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/from_flexus.msg  
//qa1/auto-P/exports/edrm global/1/documents/1/philosophy.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/carlota.zip.members  
//qa1/auto-P/exports/edrm global/1/documents/1/carlota.zip.members/  
0x22e2105930a1ddc75b9bc9dc3d5408c3b0bac8b1.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/carlota.zip.members/  
0xfce44897add0b77c25fea85118fe47164dc828d4.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/carlota.zip.members/enemies-  
0.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/carlota.zip.members/  
friends-1.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/from_flexus.msg.members  
//qa1/auto-P/exports/edrm global/1/documents/1/from_flexus.msg.members/  
0x1564f56aecf315cdecde563605fb151356f8dcc2.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/from_flexus.msg.members/  
Spaulding-1.doc  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top  
of Personal Folders.members/  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top  
of Personal Folders.members/Story.members/  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top  
of Personal Folders.members/Story.members/Production.members  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top  
of Personal Folders.members/Story.members/Production.members/  
flexus.members  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top  
of Personal Folders.members/Story.members/Production.members/  
flexus.members/0x277bab09a965f3840cd8a14300e0fccf59f6f5a9.txt  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top  
of Personal Folders.members/Story.members/Production.members/  
flexus.members/Philosophy of Life-7.msg  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top  
of Personal Folders.members/Story.members/Production.members/  
flexus.members/bcc.members/  
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top
```

```

of Personal Folders.members/Story.members/Production.members/
flexus.members/bcc.members/Letter from Spaulding to his lawyers-9.msg
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top
of Personal Folders.members/Story.members/Production.members/
flexus.members/bcc.members/
0x2a5bccfda550233ce9bac5d1f25e52590702a539.txt
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top
of Personal Folders.members/Story.members/Production.members/
flexus.members/bcc.members/.siq_overflow
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top
of Personal Folders.members/Story.members/Production.members/
flexus.members/bcc.members/.siq_overflow/
0xd7eb7675f2d201c79cf821debaaeba5a208cba29.txt
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top
of Personal Folders.members/Story.members/Production.members/
flexus.members/bcc.members/.siq_overflow/Spaulding-8.doc
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top
of Personal Folders.members/Story.members/Production.members/
flexus.members/Philosophy of Life-7.mail.members
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top
of Personal Folders.members/Story.members/Production.members/
flexus.members/Philosophy of Life-7.mail.members/
0xf4dd8fe9b09c834a2033849ff8a5d63f67823787.txt
//qa1/auto-P/exports/edrm global/1/documents/1/flexus.pst.members/Top
of Personal Folders.members/Story.members/Production.members/
flexus.members/Philosophy of Life-7.mail.members/identity-6.doc
//qa1/auto-P/exports/edrm global/1/siq_audit_trail.xml
//qa1/auto-P/exports/edrm global/1/siq_edrm_tag_definitions.csv
//qa1/auto-P/exports/edrm global/1/siq_load_file-0-0.xml
//qa1/auto-P/exports/edrm global/1/siq_loadfile_report.csv
//qa1/auto-P/exports/edrm global/1/siq_userid_map.csv

```

Note: Duplicate Files: Although 20 data objects were originally identified for export, only 11 native files were eventually produced, along with corresponding plaintext files. Duplicates are not copied and are marked in the policy audit trail. The EDRM XML load file lists all the exported documents and related metadata according to standard EDRM specifications. Any exceptions are noted in the IBM StoredIQ EDRM metadata tags list.

EDRM XML considerations

To load data from an IBM StoredIQ EDRM XML run, begin by using the `siq_loadfile_report.csv` to locate all the load files successfully produced in the current run.

Load these files one by one, preferably in the order of increasing document IDs mentioned in the report, into the target application. All file references in each of the load files are relative to the location of the load files themselves. Therefore, it is necessary to ensure that this relative location is maintained for a successful load operation.

The EDRM XML produced in each load file is compliant with the EDRM XML Version 1.1 XML specification and can successfully validate against the EDRM XML Validation Tool version 1.1.

XML root information

The name of the discovery export is used as the **CaseID** that describes the load file itself. The discovery export description, which is entered from IBM StoredIQ application interface, is also published in the **Description** attribute.

Batch information

The batch element is associated with a single run in the application. By this definition, there is only one batch element in every load file produced. The name of the batch is the run number.

Document entries in the load file

Intermediate archives inside file archives are only present in the export all members case or if these intermediate archives are responsive to the query used to drive the run.

Email archives are never expressed in the XML as documents. Only the emails that are produced from these email archives are described in the XML.

Tag information

In addition to the EDRM XML standard metadata tags, the following information is also present in the load files in the form of IBM StoredIQ EDRM XML tags.

Tag information	Description
Responsive	Whether IBM StoredIQ deemed the document to be responsive or not, based on the query that drives the policy.
Internal path	Path leading to the object that starts from below the file system object, if it was contained within a container.
Internet Message ID	Unique ID identifying every email.

Relationship between documents

The relationships that are described between documents are as follows.

- Attachment: Direct attachments to an email are related to the email.
- Container: A member of a file archive is related to its immediate parent only if the immediate parent is exported in the load file and has a DocID of its own. In the absence of the immediate parent, the member is related to the file system-level file archive that is always present in the load file.

If an email has archives or other emails that are attached to it, the members of the archives or emails are not related to the top email. Instead, they are related to the archives or emails to which they directly belong.

There is no container relationship between an email in an email archive and the email archive itself because the email archive is not produced as a document in the load file.

EDRM XML metadata tags

Describes valid data types and EDRM XML v1.1 metadata tags.

The following data types are valid for IBM StoredIQ EDRM XML tags.

Table 20. Valid data types for EDRM XML Tags

Type	Comment
Boolean	{True/False}
DateTime	YYYY-MM-DDThh:mm:ss.sTZD
Decimal	Standard float

Table 20. Valid data types for EDRM XML Tags (continued)

Type	Comment
Integer	32-bit integer
LongInteger	64-bit integer
LongText	Text (32,767 characters or fewer)
Text	Text (255 characters or fewer)

EDRM-standard tag names are prefaced with a #. IBM StoredIQ proprietary names are prefaced with an SIQ-. In certain cases, IBM StoredIQ altered the data types for some tags from what is prescribed in the original EDRM XML v1.1 metadata tags list to support a larger range of values. It does not violate EDRM compatibility.

Table 21. EDRM data tags

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
All	BookMark URL (Box)	SIQ-Box-BookMarkURL	N/A	Text	URL that is set for a BookMark object.
	Collaborator (Box)	SIQ-Box-Collaborator	N/A	Text	Collaborator of this object.
	Comments (Box)	SIQ-Box-Comment	N/A	Text	A comment that was entered in a file in Box.
	Content created at (Box)	SIQ-Box-ContentCreatedAt	N/A	DateTime	Date that the content for this object was created.
	Content modified at (Box)	SIQ-Box-ContentModifiedAt	N/A	DateTime	Date that the content for this object was modified.
	Created At (Box)	SIQ-Box-CreatedAt	N/A	DateTime	Date that the object was created.
	Created By (Box)	SIQ-Box-CreatedBy	N/A	Text	The user that created a file.
	Custom (Box)	SIQ-Box-CUSTOM	N/A	Text	Custom metadata that is set on this object.
	Description (Box)	SIQ-Box-Description	N/A	Text	The description that is entered for the data object.
	Item Status (Box)	SIQ-Box-ItemStatus	N/A	Text	Item status for the data object, whether it is deleted or not.
	Modified At (Box)	SIQ-Box-ModifiedAt	N/A	DateTime	Date that the object was modified.
	Modified By (Box)	SIQ-Box-ModifiedBy	N/A	Text	The user that modified a file.
	Owned By (Box)	SIQ-Box-OwnedBy	N/A	Text	The user that owns a file.
	SharedLink Download Count (Box)	SIQ-Box-SharedLink-DownloadCount	N/A	Text	Download count for the shared link for this object.
	SharedLink Vanity URL (Box)	SIQ-Box-SharedLink-VanityURL	N/A	Text	Vanity URL of the shared link for this object.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	SharedLink Link Access (Box)	SIQ-Box-SharedLink-LinkAccess	N/A	Text	Access level of the shared link for this object.
	SharedLink Unshared At (Box)	SIQ-Box-SharedLink-UnsharedAt	N/A	Text	The date that the shared link for this object was unshared.
	SharedLink Password Enabled (Box)	SIQ-Box-SharedLink-PasswordEnabled	N/A	Boolean	Whether passwords are required to access the shared link for this object.
	SharedLink Preview Count (Box)	SIQ-Box-SharedLink-PreviewCount	N/A	Text	Preview count for the shared link for this object.
	SharedLink Download URL (Box)	SIQ-Box-SharedLink-DownloadURL	N/A	Text	Download URL of the shared link for this object.
	SharedLinkURL (Box)	SIQ-Box-SharedLink-URL	N/A	Text	URL of the shared link for this object.
	Tag (Box)	SIQ-Box-Tag	N/A	Text	Tags that are applied to this object.
	Internal Path	SIQ- Internal Path	N/A	LongText	This path is the path that leads to an object contained inside a container, relative to the file system-level container.
	Language	#Language	Text	Text	This document is the ISO 639-1 (two-character) primary language of the document. If omitted, then English (EN) can be assumed.
	Responsiveness	SIQ-Responsive-ness	N/A	Boolean	It indicates whether IBM StoredIQ deemed the object to be responsive for the discovery export run.
	All Day Event Flag (SharePoint)	SIQ-SP-AllDay-Event	N/A	Boolean	It indicates whether the Event is an all-day Event or not.
	Announcement Expiry Date (SharePoint)	SIQ-SP-Announcement-Expiry	N/A	DateTime	The expiry date set for the announcement, if any.
	Approval status (SharePoint)	SIQ-SP- Moderation Status	N/A	Text	It provides information about the moderation status of the specified Windows SharePoint Services object, one of Approved, Denied, Draft, Pending, or Scheduled.
	Approver comments (SharePoint)	SIQ-SP- Approver Comments	N/A	LongText	These comments are the moderation comments of the specified Windows SharePoint Services weblog object.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	Ask me about (SharePoint)	SIQ-SP-UserAskMeAbout	N/A	Text	The query topics for which a SharePoint User is open or available. Multiple topics are separated into their individual tags.
	Assigned To (SharePoint)	SIQ-SP- TaskAssignedTo	N/A	Text	The user name to which the task was assigned.
	Assistant (SharePoint)	SIQ-SP-UserAssistant	N/A	Text	The assistant for the user.
	Author (SharePoint)	#Author	N/A	Text	It represents the specified author of the Windows SharePoint Services object.
	Average Rating (SharePoint)	SIQ-SP-Average-Rating	N/A	Decimal	The average of all ratings that are applied to the object.
	Birthdate (SharePoint)	SIQ-SP-UserBirthdate	N/A	Text	The birth date of SharePoint user (MM-DD).
	Blog or list name (SharePoint)	SIQ-SP- BlogOrListName	N/A	Text	It is the name of the blog or list to which the object belongs.
	Check in comment (SharePoint)	SIQ-SP-Checkin-Comment	N/A	LongText	It is the check in comments of the specified Windows SharePoint Services object.
	Checked out to (SharePoint)	SIQ-SP-Checked-OutTo	N/A	Text	It represents the designated user who checked out the Windows SharePoint Services object by using version control.
	Colleague (SharePoint)	SIQ-SP-UserColleague	N/A	Text	The colleagues for a user. Multiple colleagues are separated into their individual tags.
	Comments (SharePoint)	#Comments	N/A	Text	It provides comments that are associated with the specified Windows SharePoint Services object. Multiple comments are separated into their individual tags.
	Contact Business Address (SharePoint)	SIQ-SP-Contact-Work-Address	N/A	LongText	The work address of the contact.
	Contact Business Phone (SharePoint)	SIQ-SP-Contact-WorkPhone	N/A	Text	The work phone of the contact.
	Contact Company (SharePoint)	#Company	N/A	Text	The company of the contact.
	Contact Email (SharePoint)	SIQ-SP-Contact-Email	N/A	Text	The email address of the contact.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	Contact First Name (SharePoint)	SIQ-SP-Contact-FirstName	N/A	Text	The given name of the contact.
	Contact Full Name (SharePoint)	SIQ-SP- Contact FullName	N/A	Text	The full name of the contact.
	Contact Home Phone (SharePoint)	SIQ-SP- Contact HomePhone	N/A	Text	The home phone of the contact.
	Contact Job Title (SharePoint)	SIQ-SP-Contact-JobTitle	N/A	Text	The job title of the contact.
	Contact last name (SharePoint)	SIQ-SP-UserLast-Name	N/A	Text	The surname of the user.
	Contact Mobile Phone (SharePoint)	SIQ-SP-Contact-Mobile- Phone	N/A	Text	The mobile number of the contact.
	Contact Web Page (SharePoint)	SIQ-SP- Contact WebPage	N/A	Text	The web page URL of the contact with its associated description.
	Content type (SharePoint)	SIQ-SP-ContentType	N/A	Text	It is the content type of the specified Windows SharePoint Services object.
	Custom properties (SharePoint)	SIQ-SP-CUSTOM-Attribute Name, where Attribute-Name is the name of the Custom Attribute with white spaces replaced by '_'	N/A	LongText	It provides all custom properties that are applied to specified Windows SharePoint Services object.
	Declared record time (SharePoint)	SIQ-SP-Declared RecordDate	N/A	DateTime	It is the date at which the item was declared a record.
	Department (SharePoint)	SIQ-SP-User Department	N/A	Text	The department of the user.
	Description (SharePoint)	SIQ-SP-Description	N/A	LongText	It provides alternative text for specified Windows SharePoint Services object.
	Discussion Last Updated (SharePoint)	SIQ-SP-Discussion-Last-Updated	N/A	DateTime	The time stamp for when the discussion was last updated.
	Discussion Title (SharePoint)	SIQ-SP-Discussion-Title	N/A	Text	The title of the post for which the reply was written.
	Dotted-line Manager (SharePoint)	SIQ-SP-UserDotted LineManager	N/A	Text	The dotted-line manager for the user.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	Editor (SharePoint)	SIQ-SP-Editor	N/A	Text	This information is the editor name or information that is associated with a person who is referenced by a Windows SharePoint Services contact object.
	Event Category (SharePoint)	#Category	N/A	Text	The category that is associated with the event.
	Event Date (SharePoint)	SIQ-SP-EventDate	N/A	DateTime	The start date for the event.
	Event Duration (SharePoint)	SIQ-SP-Duration	N/A	Integer	The duration in seconds for the event.
	Event End Date (SharePoint)	SIQ-SP-EventEnd-Date	N/A	DateTime	The end date for the event.
	Event Location (SharePoint)	SIQ-SP-Event-Location	N/A	Text	The location for the event.
	Event Type (SharePoint)	SIQ-SP-EventType	N/A	Text	The event type, which has one of the following values: <ul style="list-style-type: none"> • Single Event • Recurring Event • Deleted Recurring Event • Edited Recurring Event
	Hire Date (SharePoint)	SIQ-SP-UserHire-Date	N/A	DateTime	The hire date for SharePoint user.
	Interests (SharePoint)	SIQ-SP-User Interests	N/A	Text	The interests that are listed by SharePoint user. Multiple interests are separated into their individual tags.
	Issue ID (SharePoint)	SIQ-SP-IssueID	N/A	LongInteger	The unique ID for the issue in the list.
	Keywords (SharePoint)	#Keywords	N/A	Text	It is the keyword summary of the specified Windows SharePoint Services object. Multiple keywords are separated into their individual tags.
	Link URL (SharePoint)	SIQ-SP-LinkURL	N/A	LongText	The URL that forms the basis of the link, plus an optional description. In the absence of the description, the URL is repeated.
	Litigation hold (SharePoint)	SIQ-SP-Litigation-Holds	N/A	Text	It is the list of all legal holds placed on an object. Multiple litigation holds are separated into their individual tags.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	Managed keywords (SharePoint)	#Keywords	N/A	Text	It describes all managed keywords that are applied to specified Windows SharePoint Services object. Multiple keywords are separated into their individual tags.
	Manager (SharePoint)	SIQ-SP-User-Manager	N/A	Text	The manager of the user.
	Member Of (SharePoint)	SIQ-SP-UserMemberOf	N/A	Text	It indicates the sites for which a SharePoint User is a member. Multiple sites are separated into their individual tags.
	Message ID (SharePoint)	SIQ-SPMessage-ID	N/A	Text	The unique message ID (similar to those on an email) associated with a post on a discussion board.
	Office (SharePoint)	SIQ-SP-UserOffice	N/A	Text	The office information for SharePoint user.
	Office CommunicatorAddress (SharePoint)	SIQ-SP-UserOffice-CommunicatorAddress	N/A	Text	The office communicator address for user.
	Office Location (SharePoint)	SIQ-SP-UserOffice Location	N/A	Text	The office-location information for SharePoint user.
	Organization (SharePoint)	SIQ-SP-User-Organization	N/A	Text	The organization to which the SharePoint user belongs.
	Original expiration date (SharePoint)	SIQ-SP- Record Expiration- Date	N/A	DateTime	It is the date at which the retention on the object is set to expire.
	Past Projects (SharePoint)	SIQ-SP-User PastProjects	N/A	Text	The list of past projects that are conducted by SharePoint user. Multiple projects are separated into their individual tags.
	Percent Complete (SharePoint)	SIQ-SP- TaskPercent Complete	N/A	Integer	The completion percentage of the task as an integer.
	Post Category (SharePoint)	#Category	N/A	Text	This category is associated with the post. Categories are defined as managed objects on the blog site for use with posts.
	Predecessors (SharePoint)	SIQ-SP- Task Predecessors	N/A	Text	The names of tasks that have a defined precedence to the task. Multiple predecessors are separated into their individual tags.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	Preferred Name (SharePoint)	SIQ-SP- User Preferred Name	N/A	Text	The preferred name of the user.
	ProfileId (SharePoint)	SIQ-SP-User-ProfileId	N/A	Text	The unique identifier for the user profile.
	Public URL (SharePoint)	SIQ-SP-UserProfile-PublicUrl	N/A	LongText	The public URL to access a SharePoint user profile.
	Published Date (SharePoint)	SIQ-SP-Published-Date	N/A	DateTime	The date on which the post was published.
	Publishing Contact (SharePoint)	SIQ-SP- Publishing Contact	N/A	Text	The user name of the contact.
	Publishing Contact Email (SharePoint)	SIQ-SP- Publishing Contact- Email	N/A	Text	The email address of the contact.
	Publishing Contact Name (SharePoint)	SIQ-SP-Publishing-Contact-Name	N/A	LongText	The friendly name of the contact.
	Publishing level (SharePoint)	SIQ-SP-Publishing-Level	N/A	Text	It is the version-control alert level information of the specified Windows SharePoint Services object, one of Published, Draft, or Checkout.
	Quicklinks (SharePoint)	SIQ-SP-UserQuick links	N/A	Text	The Quick Links entry that is configured on a user profile. Multiple links are separated into their individual tags.
	Rating Count (SharePoint)	SIQ-SP-Rating-Count	N/A	Integer	The total number of ratings that are applied to this object.
	Record restrictions (SharePoint)	SIQ-SP- Record-Restrictions	N/A	Text	Describes the restrictions that are applied to the record by the policies in the Records Center: <ul style="list-style-type: none"> • None • BlockDelete • BlockEdit
	Recurrence Event (SharePoint)	SIQ-SP-Recurring-Event	N/A	Boolean	It indicates whether an event is a recurring event.
	Related Issues (SharePoint)	SIQ-SP-Related-Issues	N/A	Text	Issues that are marked as related to an issue. Multiple issues are separated into their individual tags.
	Routing rule description (SharePoint)	SIQ-SP-Routing-Rule-Description	N/A	LongText	This description is associated with the object that matched a routing rule.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	Schools (SharePoint)	SIQ-SP-User-Schools	N/A	Text	The schools that are listed by SharePoint user. Multiple schools are separated into their individual tags.
	Skills (SharePoint)	SIQ-SP-UserSkills	N/A	Text	The skills that are listed by SharePoint user. Multiple skills are separated into their individual tags.
	Status (SharePoint)	SIQ-SP-TaskOr-Issue-Status	N/A	Text	It is either the task status or the issue status. These options for task status: <ul style="list-style-type: none"> • "Not Started" • "In Progress" • "Completed" • "Deferred" • "Waiting on someone else" These options are for issue status: <ul style="list-style-type: none"> • "Active" • "Resolved" • "Closed"
	Status Bar (SharePoint)	SIQ-SP-StatusBar	N/A	DateTime	It identifies a field that contains information about the status bar for a specified SharePoint Foundation object.
	Status Message (SharePoint)	SIQ-SP-UserStatus Message	N/A	Text	The most recent status message on SharePoint user profile.
	Survey Completed (SharePoint)	SIQ-SP- Survey Completed	N/A	Boolean	It indicates whether the survey response was completed.
	Task Due Date (SharePoint)	SIQ-SP-TaskDue-Date	N/A	DueTime	The due date for a task.
	Task Priority (SharePoint)	SIQ-SP-TaskPriority	N/A	Text	The priority of the task, which is usually one of "(1) High," "(2) Normal," or "(3) Low." The Administrator has the option of editing the choices.
	Task Start Date (SharePoint)	SIQ-SP-TaskStart-Date	N/A	DateTime	The start date for a task.
	Thread Index (SharePoint)	SIQ-SP-Thread-Index	N/A	LongText	The unique index into the discussion to help determine the position of the post in the threading hierarchy of the discussion.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	Title (SharePoint)	#Title	N/A	Text	It represents information about the occupational title of a specified Windows SharePoint Services contact object.
	User name (SharePoint)	SIQ-SP-UserName	N/A	Text	The user name that is associated with a SharePoint user profile.
	URL (SharePoint)	SIQ-SP-URL	N/A	LongText	It represents information about the server-relative URL for the specified Windows SharePoint Services object.
	Version (SharePoint)	SIQ-SP-Version	N/A	Decimal	It represents the version string that is associated with the user interface of the specified Windows SharePoint Services object.
	Virus status (SharePoint)	SIQ-SP-VirusStatus	N/A	Text	It is the virus-scan status of a specified Windows SharePoint Services object: <ul style="list-style-type: none"> • Clean • Virus reported • Virus reported; can be removed • Virus previously reported; removed since • Virus reported; clean failed • Virus reported; delete recommended • Timeout last processing for Virus
	Wiki Page Categories (SharePoint)	#Category	N/A	Text	They are the managed metadata values that describe categories for the wiki page. Multiple categories are separated into their individual tags.
	Wiki Page Comment (SharePoint)	#Comments	N/A	LongText	This comment describes the Wiki page. There can be only one.
	Work City (SharePoint)	SIQ-SP- Contact WorkCity	N/A	Text	The city of the contact.
	Work Country (SharePoint)	SIQ-SP- Contact-Work-Country	N/A	Text	The country of the contact.
	Work Fax (SharePoint)	SIQ-SP-Contact-WorkFax	N/A	Text	The work fax number of the contact
	Work State (SharePoint)	SIQ-SP-Contact-WorkState	N/A	Text	The work state of the contact.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	Work ZIP (SharePoint)	SIQ-SP-Contact-WorkZip	N/A	Text	The postal code of the contact.
File	Author	#Author	Text	Text	The (Office) document author.
	Category	#Category	Text	Text	The (Office) document category.
	Comments	#Comments	Text	LongText	The (Office) document comments. Note: All Office document metadata can be longer than 255 characters.
	Company	#Company	Text	Text	The (Office) document company.
	Creation Time	#Date-Created	DateTime	DateTime	The date the file was created.
	Date Last Printed	#Date-Printed	DateTime	DateTime	The date the file was last printed.
	FileName	#FileName	Text	LongText	The name of the original file. Note: File names can be longer than 255 characters.
	File Extension	#File- Extension	Text	Text	The extension of the original file.
	File Size	#FileSize	LongInteger	LongInteger	The size of the original file in bytes.
	Keywords	#Keywords	Text	LongText	The (Office) document keywords.
	Last Accessed Time	#Date-Accessed	DateTime	DateTime	The date the file was last accessed.
	Last Modified Time	#Date-Modified	DateTime	DateTime	The date the file was last modified.
	Subject	#Subject	Text	Text	The (Office) document subject.
	Title	#Title	Text	Text	The (Office) document title.
Message	Attachment Count	#Attachment Count	Integer	Integer	The number of attachments the email has.
	Attachment Names	#Attachment Names	Text	LongText	The concatenated list of attachment names that are separated by semicolons. Note: The list of attachment names can be longer than 255 characters.
	BCC	#BCC	Text	LongText	The blind carbon-copied recipients of the message in the same format as "From." Multiple recipients are separated into their individual tags.

Table 21. EDRM data tags (continued)

Document Type	Metadata Property	Tag Name	EDRM Data Type	Data Type	Description
	CC	#CC	Text	LongText	The carbon-copied recipient of the message in the same format as "From." Multiple recipients are separated into their individual tags.
	Flag Status	#FlagStatus	Text	Text	The (Outlook) flag status.
	From	#From	Text	LongText	The sender of the message in the EDRM-recommended RFC2821 format (either "your-name@domain.com" or "Your Name <your-name@domain.xyz>") Note: Email addresses can have 64-character account names and 255-character domain names.
	Full Email Header	#Header	Text	LongText	The message header of the message. Note: The header can be longer than 64,000 characters; however, it is truncated in that case.
	Has Attachments	#Has Attachments	Boolean	Boolean	It indicates whether the email has attachments.
	Importance	#Importance Flag	Boolean	Boolean	Indicates whether the email was sent with high importance.
	Internet MessageID	SIQ-Internet MessageID	N/A	Text	The universal Internet message ID assigned to the email. Note: The message ID is usually in the form unique_id@domain.
	Message Class	#Message-Class	Text	Text	The (Outlook) message class.
	Read Flag	#ReadFlag	Boolean	Boolean	It indicates if the email was read.
	Received Date	#Date Received	DateTime	DateTime	The date the message was received.
	Sent Date	#DateSent	DateTime	DateTime	The date the message was sent.
	Subject	#Subject	Text	Text	The subject of the message.
	To	#To	Text	LongText	The recipient of the message in the same format as "From." Multiple recipients are separated into their individual tags.

Appendix C. Metadata by attribute type

This reference guide describes metadata that is organized by attribute type.

The column headings in the attribute tables are listed as follows:

Table 22. Column headings in attribute table

Name	Name of the attribute.
Dense	Dense or Sparse, signifying whether an attribute occurs on an object.
Multiple	Single or Multiple Instance, signifying whether the attribute can occur multiple times (Y) or one time (N).
Type	Indicates whether the attribute is a string, integer, Boolean, or date.
Normalized	Indicates whether the pipeline has a normalization liability when you are extracting the attribute. In general, normalizing the format of an attribute produces richer results.
Description	Provides a description of the attribute (optional).

System metadata attributes

This table describes system metadata for Windows Share and NFS attributes.

Table 23. System metadata attributes

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Examples	Supported Operators	Value Supported
Accessed Date	Y	N	Date	Y	A time stamp that indicates the last accessed time of the file (YYYY-MM-DD hh:mm:ss)	att:"/Library/Attributes/System metadata/Accessed date" > "2015-08-09" IN all att:"/Library/Attributes/System metadata/Accessed date" >= 12 weeks ago IN all	=, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Container path	Y	N	String	N	Path of an object relative to the container that it lives in.	att:"/Library/Attributes/System metadata/Container path" CONTAINS any ("pst") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match.	Text
Content type description	Y	N	String	N	Derived from the content of an object; a description of the application with which the object was created.	att:"/Library/Attributes/System metadata/Content type description" CONTAINS any ("word") IN all	is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 23. System metadata attributes (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Examples	Supported Operators	Value Supported
Content type mismatch	Y	N	Boolean	N	True if the object's extension does not match the content type that is derived from the object.	att:"/Library/Attributes/System metadata/Content type mismatch" = true IN all	=, !=	True, false
Created date	Y	N	Date	Y	A time stamp that indicates the creation time of the file (YYYY-MM-DD hh:mm:ss)	att:"/Library/Attributes/System metadata/Created date" < "2016-01-06" IN all att:"/Library/Attributes/System metadata/Created date" > 5 years ago IN all	=, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Data object type (is known)	Y	N	Boolean	N	True if the data type of the object can be mapped to a known file extension in the system.	att:"/Library/Attributes/System metadata/Data object type (is known)" != true IN all	=, !=	True, false
Extension	Y	N	String	N	The extension of the data object.	att:"/Library/Attributes/System metadata/Extension" IS any ("doc") IN all	is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Group	Y	N	String	N	The name of the group owner of the file that was resolved by a user directory service. If the group identifier was not resolved, it can be because no directory service was configured or an error occurred. The field contains the string representation of the group ID.	att:"/Library/Attributes/System metadata/Group" BEGINS WITH any ("a") IN all	is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Modified date	Y	N	Date	Y	A time stamp that indicates the last modified time of the file (YYYY-MM-DD hh:mm:ss)	att:"/Library/Attributes/System metadata/Modified date" < 7 years ago IN all att:"/Library/Attributes/System metadata/Modified date" <= "2009-01-07" IN all	=, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 23. System metadata attributes (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Examples	Supported Operators	Value Supported
Object category	Y	N	String	N	General file type of the object.	att:"/Library/Attributes/System metadata/Object category" IS any (Collaboration) IN all (att:"/Library/Attributes/System metadata/Object category" IS any (Collaboration , Presentation) IN all)	is, is not	Collaboration, computer programming, container files and archives, emails, images, miscellaneous, multimedia, presentation, spreadsheets, system files, word processing
Object ID	Y	N	String	N	Name of the data object (file) that is located within the context of the specified volume.	att:"/Library/Attributes/System metadata/Object ID" CONTAINS any ("cm85") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Object name	Y	N	String	N	Unique data object identifier that provides the data object's "handle" to the system where the data object resides.	att:"/Library/Attributes/System metadata/Object name" CONTAINS any ("stock") IN all	is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Owner	Y	N	String	N	The user name of the file owner that was resolved by a user directory service. If the user identifier cannot be resolved, perhaps it is because no directory service was configured or an error occurred. The field contains the string representation of the user ID.	att:"/Library/Attributes/System metadata/Owner" IS any ("qa\mtran") IN all IN all	is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Owner (is known)	Y	N	Boolean	N	True if the object's owner attribute can be mapped to a known user name in the system.	att:"/Library/Attributes/System metadata/Owner (is known)" = false IN all	=, !=	True, false
Size	Y	N	Integer	N	The size of the file in bytes.	att:"/Library/Attributes/System metadata/Size" > 1 MB IN all	=, !=, <, <=, >, >=, between, not between	Number; bytes, KB, MB, GB
Size (on disk)	Y	N	Integer	N	The amount of physical cluster space that a data object occupies.	att:"/Library/Attributes/System metadata/Size (on disk)" IS BETWEEN 2 MB and 4 MB IN all	=, !=, <, <=, >, >=, between, not between	Number; bytes, KB, MB, GB

Table 23. System metadata attributes (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Examples	Supported Operators	Value Supported
Source container path	Y	N	String	N	The container path of the object on the source volume.	att:"/Library/Attributes/System metadata/Source container path" CONTAINS any ("msg") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Source volume	Y	N	String	N	The original primary source location.	att:"/Library/Attributes/System metadata/Source volume" CONTAINS any ("bmorgan") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
System path	Y	N	String	N	All attributes that reference the data object's location.	att:"/Library/Attributes/System metadata/System path" CONTAINS any ("pst") IN all	is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Documentum server metadata attributes

Documentum attributes are listed in the following table.

Name	Dense	Multiple	Type	Normalized	Description
ACL Domain (Documentum)					The repository owner name or user name that created the Access Control List
ACL Name (Documentum)					Name uniquely identifying the Access Control List on the Documentum server
Authors (Documentum)	N	N	String	N	The creator of the data object, as shown in Document Properties.
Checked out to (Documentum)	N	N	String	N	Name of person the data object is checked out to for editing.

Name	Dense	Multiple	Type	Normalized	Description
Creation Date	Y	N	Date	Y	A time stamp that indicates the creation time of the file.
File Size	Y	N	Int	N	The size of the file in bytes.
Format (Documentum)					Application type of the data object's content.
Group	Y	N	String	N	The group name of the file owner as resolved by a Documentum server.
Immutable	N	N	String	N	Denotes whether a data object can be change or deleted.
Keywords (Documentum)	N	N	String	N	Keywords that are associated with the data object.
Lock Date	Y	N	Date	N	Date that machine was locked (YYYY-MM-DD)
Lock Machine	N	N	String	N	Name of the client machine on which users are working when they lock an object (by a check out or branch operation). If the server is unable to resolve the name, r_lock_attribute is set to the client machine's IP address instead.
Modified Date	Y	N	Date	Y	A time stamp that indicates the modification time of the file.
ObjectID	Y	N	String	N	Unique data object identifier provides the data object's "handle" to the system where the data object is found.

Name	Dense	Multiple	Type	Normalized	Description
Object Name (Documentum)	Y	N	String	N	Name of the data object (file) located within the context of the specified volume.
Owner	Y	N	String	N	The user name of the file owner as resolved by a Documentum server.
Parent document ID	Y	N	String	N	Data object parent ID
Parent document name	Y	N	String	N	Data object parent name
Retain until	Y	N	Date	N	Date data object is released from retention. Set on the Documentum server not through the application.
Retention date (Documentum)	Y	N	Date	N	Date data object was placed on hold. Set on the Documentum server not through the application.
RPS markup name					Retention Policy Services retention policy markup name
RPS policy name					Retention Policy Services retention policy name
Subject (Documentum)	N	N	String	N	The subject of the data object. By default, it is the subject of the Office document as shown in Document Properties. Configurable by the user when it is checked in.

Name	Dense	Multiple	Type	Normalized	Description
Title (Documentum)	N	N	String	N	The title of the data object. By default, it is the title of the Office document as shown in Document Properties. Configurable by the user when it is checked in.
Version (Documentum)	Y	N	String	N	The version of the data object from a Documentum server perspective.

Note: Access Time not Reset: When harvesting Documentum volumes, IBM StoredIQ does not reset the access time back to the original value before the harvest. The reason is due to a known issue. Resetting the volume access time on IBM StoredIQ causes Documentum to reset its files' modification time and date to the time and date of the harvest.

Centera server metadata attributes

Metadata for Centera server attributes is listed in the following table.

Table 24. Metadata for Centera server attributes

Name	Dense	Multiple	Type	Normalized	Description
Access Date	Y	N	Date	Y	A time stamp that indicates the last accessed time of the data object.
Creation Date	Y	N	Date	Y	A time stamp that indicates the creation time of the data object.
File Extension	Y	N	String	N	The data object's extension.
File Name	Y	N	String	N	Name of the data object located within the context of the specified volume.
File Path	Y	N	String	N	All attributes that reference the data object's location.
File Size	Y	N	Int	N	The size of the data object in bytes.

Table 24. Metadata for Centera server attributes (continued)

Name	Dense	Multiple	Type	Normalized	Description
Litigation Hold	N	Y	String	N	Name of the litigation hold that is applied to the data object. There can be multiple litigation holds applied to a data object.
Modified Date	Y	N	Date	Y	A time stamp that indicates the modification time of the data object.
Object ID	Y	N	String	N	Unique data object identifier provides the data object's "handle" to the system where the data object is found.
Retention Class	N	N	String	N	Name of the retention class that is applied to the data object.
Retention Period	N	N	String	N	The length of time a data object is set to be retained.
Retention Policy Name	N	N	String	N	The name of the policy that last set retention for a data object.
Retention Tag	N	N	String	N	Tag defined in copy policy that is retained in the copied and harvested data object.
Source Volume	Y	N	String	N	The originating server and share name from which an object was copied to Centera.

Hitachi HCAP Server metadata attributes

This table describes metadata for Hitachi HCAP Servers attributes.

Table 25. Metadata for Hitachi HCAP

Name	Dense	Multiple	Type	Normalized	Description
Access Date	Y	N	Date	Y	A time stamp that indicates the last accessed time of the data object.
Creation Date	Y	N	Date	Y	A time stamp that indicates the creation time of the data object.
File Extension	Y	N	String	N	The data object's extension.
File Name	Y	N	String	N	Name of the data object located within the context of the specified volume.
File Path	Y	N	String	N	All attributes that reference the data object's location.
File Size	Y	N	Int	N	The size of the data object in bytes.
Litigation Hold	N	Y	String	N	Name of the litigation hold that is applied to the data object. There can be multiple litigation holds applied to a data object.
Modified Date	Y	N	Date	Y	A time stamp that indicates the modification time of the data object.
Object ID	Y	N	String	N	Unique data object identifier provides the data object's handle to the system where the data object is found.
Retention Period	N	N	String	N	The length of time a data object is set to be retained.
Retention Policy Name	N	N	String	N	The name of the policy that last set retention for a data object.

Table 25. Metadata for Hitachi HCAP (continued)

Name	Dense	Multiple	Type	Normalized	Description
Source Volume	Y	N	String	N	The originating server and share name from which an object was copied to the Hitachi HCAP.
Access Date	Y	N	Date	Y	A time stamp that indicates the last accessed time of the data object.

Discovery Accelerator metadata attributes

This table describes metadata for Discovery Accelerator attributes.

Table 26. Metadata for Discovery Accelerator

Name	Dense	Multiple	Type	Description
Latest Review Status	N	N	String	All system-level Accelerator objects have one of the following statuses: Produced, Reviewed, Unreviewed, Pending, Questioned.
Review Mark Names	N	Y	String	Mark names set for each Review Status that the object was put into. Review marks are user-defined, and an object might have multiple marks.
Review Comments	N	Y	String	Comment (text) added by a Reviewer in Discovery Accelerator for this object. An object might have multiple review comments.
Reviewers	N	Y	String	Log in ID of a Reviewer who either added a Comment or a Status/Mark on the object. An object might have multiple Reviewers.
Message Type	N	N	String	All system-level Accelerator objects have one of the following messages: Unknown, Exchange, Instant Messaging, Bloomberg, Fax, Lotus, FSA, SMTP

FileNet metadata attributes

This table describes metadata for FileNet attributes.

Table 27. Metadata for FileNet

Name	Dense	Multiple	Type	Description
FoldersFiledIn	Y	Y	Object	The list of folders where this document is filed.
Id	N	Y	GUID	The unique object ID. References FileNet Id.
Name	N	N	String	This attribute returns the value of the designated name property for the object or its ID if there is no name property. References FileNet Name.
DateContentLastAccessed	N	N	Date/time	The date and time when this content of a document was last accessed. References FileNet DateContentLastAccessed.
DateCreated	N	N	Date/time	The date and time this object was created. References FileNet DateCreated.
DateLastModified	N	N	Date/time	The date and time when this object was last modified. References FileNet DateLastModified.
ContentSize	N	N	Float	The size (in bytes) of the captured content that is associated with this object. References FileNet ContentSize.
Owner	N	N	String	The security owner of the object. References FileNet Owner.
ContentRetentionDate	N	N	Date/time	The date until which the object must be retained, as determined by the content storage subsystem. References FileNet ContentRetentionDate.
Creator (FileNet)	N	N	String	The name of the user who created this object. References FileNet Creator.
DateCheckedIn	N	N	Date/time	The date and time this object was checked in. References FileNet DateCheckedIn.
DocumentTitle	N	N	String	Different from Document.name. References FileNet DocumentTitle.

Table 27. Metadata for FileNet (continued)

Name	Dense	Multiple	Type	Description
IsCurrentVersion	N	N	Boolean	Indicates whether it is the current version in the version series. It is not a separate attribute, but rather used to set the Current string in attribute VersionStatus (FileNet). References FileNet IsCurrentVersion.
IsFrozenVersion	N	N	Boolean	Indicates whether user properties or content can be modified. It is not a separate attribute, but rather used to set the Frozen string in attribute VersionStatus (FileNet). References FileNet IsFrozenVersion.
IsInExceptionState	N	N	Boolean	Indicates whether the lifecycle of a document is in exception state. References FileNet IsInExceptionState.
IsReserved	N	N	Boolean	Indicates whether a user reserved the right to check in a version after this version. References FileNet IsReserved.
LastModifier	N	N	String	The name of the user who last modified this object. References FileNet LastModifier.
MajorVersionNumber	N	N	Integer	The major version number for this document version. References FileNet MajorVersionNumber.
MimeType	N	N	String	The MIME type of the document. References FileNet MimeType.
MinorVersionNumber	N	N	Integer	The minor version number for this document version. References FileNet MinorVersionNumber.
ReleasedVersion	N	N	Boolean	Indicates whether user properties or content can be modified.
VersionStatus	N	N	Integer	The version status of this document version. These status options are listed here. <ul style="list-style-type: none"> • In Process • Current version • Released • Superseded • Frozen

Email metadata attributes

This table describes metadata for email attributes.

Table 28. Metadata for email

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Any address	N	Y	String	Y	Any recipient or the sender of an email object	att:"/Library/Attributes/Email/Any address" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Attachments				Y	Whether the data object has attachments. True if message has attachments	att:/Library/Attributes/Email/Attachments = true IN all	Appears, Does Not Appear, =, !=	True, False

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
BCC					Recipients that are blind-copied on the email data object	att:/Library/Attributes/Email/BCC CONTAINS any ("mtran@qa.local") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
CC				Y	Recipients that are copied on the email data object	att:/Library/Attributes/Email/CC MATCHES any ("logesh@qa.local") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Conversation index	N	N	String	N	A value that indicates the relative position of this message within a conversation thread	att:"/ Library/ Attributes/ Email/ Conversation index" APPEARS IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (any)
Conversation topic	N	N	String	N	The topic of the first message in a conversation thread	(att:"/ Library/ Attributes/ Email/ Conversation topic" CONTAINS any ("data expert" , "matter") IN all)	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
Expire time	N	N	Date	N	The date and time at which the messaging system can invalidate the content of a message (YYYY-MM-DD)	att:"/ Library/ Attributes/ Email/Expire time" DOES NOT APPEAR IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
From				Y	The sender of the email data object	att:/Library/Attributes/Email/From IS any ("testadmin@qa.local") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches , Does Not Match	Text
Importance					The importance level that is assigned by the sender of the email data object: High, Normal, Low	att:/Library/Attributes/Email/Importance IS any ("low") IN all or att:/Library/Attributes/Email/Importance IS any ("high") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches , Does Not Match	Text
Internet message ID	N	N	String	Y	Corresponds to the message ID field as specified in [RFC2822]	att: "/Library/Attributes/Email/Internet message ID" CONTAINS any ("journal.report.generator") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches , Does Not Match	Text

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Is-part-of					TRUE if object is part of email body, email attachment, miscellaneous email item's body or miscellaneous email item's attachment	att:"/ Library/ Attributes/ Email/Is-part-of" IS any ("Email archive") IN all	Is, Is Not	See Note below this table.
Message create time	N	N	Date	Y	Date the message was created (YYYY-MM-DD)	att:"/ Library/ Attributes/ Email/ Message create time" < "2015-05-05" IN all	att:"/ Library/ Attributes/ Email/ Message create time" < "2015-05-05" IN all	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Message modify time	N	N	String	Y	Date the message was last modified (YYYY-MM-DD)	att:"/ Library/ Attributes/ Email/ Message modify time" > "2015-05-01" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Priority	N	N	String	Y	The order or speed at which the message must be sent by the messaging system software: Urgent, Normal, Low	att:/Library/Attributes/Email/Priority IS any ("urgent") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
Proof of Delivery Requested	N	N	Boolean	N	TRUE if a message sender requests proof of delivery for a particular recipient	att: "/Library/Attributes/Email/Proof of delivery requested" = true IN all	Appears, Does Not Appear, =, !=	True, False
Proof of Submission Requested	N	N	Boolean	N	TRUE if a message sender requests proof that the message transfer system was submitted a message for delivery to the originally intended recipient	att: "/Library/Attributes/Email/Proof of submission requested" = true IN all	Appears, Does Not Appear, =, !=	True, False

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Provider Submit Time	N	N	Date	Y	The date and time a transport provider passed the message to its underlying messaging system (YYYY-MM-DD)	att:"/ Library/ Attributes/ Email/ Provider submit time" IS BETWEEN 15 months ago and 13 months ago IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Received by	N	N	String	N	Message recipient	att:"/ Library/ Attributes/ Email/ Received by" IS any ("testadmin @qa.local") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
Received date					Date the message was received (YYYY-MM-DD)	att:"/ Library/ Attributes/ Email/ Received date" <= "2015-05-25" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Received Representing	N	N	String	N	The display name for the messaging user that is represented by the recipient	att:"/ Library/ Attributes/ Email/ Received representing" IS any ("testuser @qa.local") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
Reply Message ID	N	N	String	Y	The original message's PR_INTER NET_MESSAGE _ID property value	att:"/ Library/ Attributes/ Email/Reply message ID" CONTAINS any ("@ex2010.qa.local") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
Reply Message time	N	N	Date	N	Time when the reply was sent (YYYY-MM-DD)	att:"/ Library/ Attributes/ Email/Reply message time" != any (14 months ago) IN all IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Reply Requested	N	N	Boolean	Y	TRUE if a message sender requests a reply from a recipient	att:"/ Library/ Attributes/ Email/Reply requested" = true IN all	Appears, Does Not Appear, =, !=	True, False
Sensitivity	Y	N	String	N	The sensitivity level that is assigned by the sender of the email data object: Normal, Personal, Private, Confidential	att:/Library/ Attributes/ Email/ Sensitivity IS any ("Confidential") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
Sent date				Y	Date the message was sent (YYYY-MM-DD)	att:"/ Library/ Attributes/ Email/Sent date" IS BETWEEN 2 years ago and 23 months ago IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Sent Representing	N	N	String	N	Contains the display name for the messaging user that is represented by the sender	att:"/ Library/ Attributes/ Email/Sent representing" CONTAINS any ("QA test administrator") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
Subject				Y	The subject of the email data object	att:/Library/ Attributes/ Email/ Subject CONTAINS any ("Legal user removed from matter named \"TC_ eDiscovery UserMgmt-Matter-1\") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
To				Y	The recipient/s of the email data object	att:/Library/ Attributes/ Email/To CONTAINS any ("nkumar @qa.local") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text

Table 28. Metadata for email (continued)

Name	Dense	Multiple	Type	Available for Non Microsoft Email	Description	Filter Code Example	Supported Operators	Value Supported
Transport Message Headers	N	N	Date	Y	Contains transport-specific message envelope information	att:"/ Library/ Attributes/ Email/ Transport message headers" CONTAINS any ("testing mule") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text

Note: These are the supported values for Is-part-of: Chatter Collaboration Group Post, Chatter Comment, Chatter Private Message, Chatter User Profile Post, Email, Email archive, Email attachment, Email body, File container, Jive Blog Post, Jive Blog Post Comment, Jive Discussion, Jive Discussion Reply, Jive Document, Jive Document Comment, Misc. email, Misc. email attachment, Misc. email body, SharePoint announce-ment, SharePoint (any), SharePoint Calendar, Contact, or Task, SharePoint Discussion Reply, SharePoint Discussion Thread, SharePoint Issue, SharePoint Link, SharePoint Note, SharePoint Post, SharePoint Post Comment, SharePoint Survey, SharePoint User Profile, SharePoint User Profile Noteboard, SharePoint Wiki Page

Data object property attributes

This table describes Data Object Property Attributes. Data object property attributes are stored inside the data object.

Table 29. Data object property attributes

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Abstract	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Abstract" CONTAINS any ("option") IN all or att:"/Library/Attributes/Data object properties/Abstract" CONTAINS any ("valuation") IN all or att:"/Library/Attributes/Data object properties/Abstract" CONTAINS any ("stock") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Account	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Account" MATCHES any ("8904") IN all or att:"/Library/Attributes/Data object properties/Account" CONTAINS any ("RTC") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Address	N	Y	String	Metadata stored inside the data object itself.	att: "/Library/Attributes/Data object properties/Address" CONTAINS any ("Sacramento") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Application version	N	Y	String	Metadata stored inside the data object itself.	att: "/Library/Attributes/Data object properties/Application version" IS any ("3.0") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Attachments	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Attachments" CONTAINS any ("MS Word") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Authorization	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Authorization" MATCHES any ("true") IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Backup date	N	Y	String	Metadata stored inside the data object itself.	att: "/Library/Attributes/Data object properties/Backup date" CONTAINS any ("2008") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Base file location	N	Y	String	Metadata stored inside the data object itself.	att: "/Library/Attributes/Data object properties/Base file location" MATCHES any ("Atlanta") IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Bill to	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Bill to" CONTAINS any ("Engineering") IN all or att:"/Library/Attributes/Data object properties/Bill to" CONTAINS any ("Sales") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Blind copy	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Blind copy" IS any ("yes") IN all or att:"/Library/Attributes/Data object properties/Blind copy" APPEARS < 1 IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Bytes	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Bytes" IS any ("45000") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Carbon copy	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Carbon copy" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Category	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Category" IS NOT any ("Sales") IN all or att:"/Library/Attributes/Data object properties/Category" IS any ("Engineering") IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Characters	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Characters" CONTAINS any ("valid") IN all or att:"/Library/Attributes/Data object properties/Characters" BEGINS WITH any ("in") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Characters (with spaces)	N	Y	String	Metadata stored inside the data object itself.	att: "/Library/Attributes/Data object properties/Characters (with spaces)" IS any ("invalid") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Checked by	N	Y	String	Metadata stored inside the data object itself.	att: "/Library/Attributes/Data object properties/Checked by" ENDS WITH any ("James") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Client	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Client" CONTAINS any ("dry valley") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	(=, >, >=, <, <=, any, all)
Comment	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Comment" DOES NOT CONTAIN any ("yes") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Company	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Company" MATCHES any ("valley") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Completed date	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Completed date" CONTAINS any ("2005") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Created	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/ Created" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Department	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/ Department" IS NOT any ("Engineering") IN all or att:"/Library/Attributes/Data object properties/ Department" IS any ("Sales") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Destination	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Destination" MATCHES any ("Storage") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Disposition	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Disposition" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Division	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Division" IS any ("Atlantic") IN all or att:"/Library/Attributes/Data object properties/Division" IS any ("west") IN all or att:"/Library/Attributes/Data object properties/Division" IS any ("midwest") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Document number	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Document number" CONTAINS any ("64") IN all or att:"/Library/Attributes/Data object properties/Document number" CONTAINS any ("ATO") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Editor	N	Y	String	Metadata stored inside the data object itself.	att: "/Library/Attributes/Data object properties/Editor" CONTAINS any ("Farr") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Forward to	N	Y	String	Metadata stored inside the data object itself.	att: "/Library/Attributes/Data object properties/Forward to" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Group	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Group" CONTAINS any ("North") IN all IN all or att:"/Library/Attributes/Data object properties/Group" IS any ("Atlantic") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Heading pairs	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Heading pairs" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Hidden slides	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Hidden slides" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Keyword	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Keyword" CONTAINS any ("att") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Language	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Language" IS NOT any ("French") IN all and att:"/Library/Attributes/Data object properties/Language" IS any ("German") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Last saved	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Last saved" MATCHES any ("1988") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Last saved by	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Last saved by" CONTAINS any ("Mary") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Lines	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Lines" IS any ("invalid") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Links dirty	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Links dirty" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Mailstop	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Mailstop" CONTAINS any ("5") IN all IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Manager	N	Y	String	Metadata stored inside the data object itself.	att:"/Library/Attributes/Data object properties/Manager" MATCHES any ("Cort") IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Matter	N	Y	String		att:"/Library/Attributes/Data object properties/Matter" MATCHES any ("open") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Multimedia clips	N	Y	String		att:"/Library/Attributes/Data object properties/Multimedia clips" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Notes	N	Y	String		att:"/Library/Attributes/Data object properties/Notes" CONTAINS any ("valid") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Office	N	Y	String		att:"/Library/Attributes/Data object properties/Office" IS NOT any ("New York") IN all and att:"/Library/Attributes/Data object properties/Office" IS any ("Seattle") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Operator	N	Y	String		att:"/Library/Attributes/Data object properties/Operator" MATCHES any ("Cor") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Owner	N	Y	String		att:"/Library/Attributes/Data object properties/Owner" CONTAINS any ("Cor") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Pages	N	Y	String		att:"/Library/Attributes/Data object properties/Pages" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Paragraphs	N	Y	String		att:"/Library/Attributes/Data object properties/Paragraphs" CONTAINS any ("valid") IN all	Appears, Does Not Appear, Appears X-times, Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Presentation format	N	Y	String		att:"/Library/Attributes/Data object properties/Presentation format" IS any ("Powerpoint") IN all	Appears, Does Not Appear, Appears X-times, Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Primary author	N	Y	String		att: "/Library/Attributes/Data object properties/Primary author" MATCHES any ("Cortez") IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Printed	N	Y	String		att: "/Library/Attributes/Data object properties/Printed" IS any ("valid") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Project	N	Y	String		att: "/Library/Attributes/Data object properties/Project" IS any ("Galaxy") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Publisher	N	Y	String		att: "/Library/Attributes/Data object properties/Publisher" CONTAINS any ("Brinks") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Purpose	N	Y	String		att:"/Library/Attributes/Data object properties/Purpose" MATCHES any ("doc") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Received from	N	Y	String		att:"/Library/Attributes/Data object properties/Received from" MATCHES any ("Cor") IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Recorded by	N	Y	String		att:"/Library/Attributes/Data object properties/Recorded date" IS any ("July 3, 2000") IN all or att:"/Library/Attributes/Data object properties/Recorded by" IS any ("cortez") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Recorded date	N	Y	String		att:"/Library/Attributes/Data object properties/Recorded date" IS any ("July 3, 2000") IN all or att:"/Library/Attributes/Data object properties/Recorded by" IS any ("cortez") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Reference	N	Y	String		att:"/Library/Attributes/Data object properties/Reference" CONTAINS any ("ARI") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Revision date	N	Y	String		att:"/Library/Attributes/Data object properties/Revision date" IS any ("September 15, 2009") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Revision notes	N	Y	String		att:"/Library/Attributes/Data object properties/Revision notes" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Scale crop	N	Y	String		att:"/Library/Attributes/Data object properties/Scale crop" CONTAINS any ("50") IN all and att:"/Library/Attributes/Data object properties/Scale crop" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Secondary author	N	Y	String		att:"/Library/Attributes/Data object properties/Secondary author" MATCHES any ("Farr") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Section	N	Y	String		att:"/Library/Attributes/Data object properties/Section" IS NOT any ("SALES") IN all or att:"/Library/Attributes/Data object properties/Section" IS any ("ACCT") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	(=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Security	N	Y	String		att:"/Library/Attributes/Data object properties/Security" IS any ("High") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Slides	N	Y	String		Text (=, >, >=, <, <=, any, all)	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Source	N	Y	String		att:"/Library/Attributes/Data object properties/Source" CONTAINS any ("Support") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Status	N	Y	String		att:"/Library/Attributes/Data object properties/Status" IS any ("Published") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Subject	N	Y	String		att: "/Library/Attributes/Data object properties/Subject" CONTAINS any ("pharm") IN all	Appears Does not appear Appears X times Appears unique X-times Is Is not Contains Does not contain Begins with Does not begin with Ends with Does not end with Matches Does not match	Text
Title	N	Y	String		att: "/Library/Attributes/Data object properties/Title" DOES NOT CONTAIN any ("medical") IN all and att: "/Library/Attributes/Data object properties/Title" CONTAINS any ("user guide") IN all	Appears Does not appear Appears X times Appears unique X-times Is Is not Contains Does not contain Begins with Does not begin with Ends with Does not end with Matches Does not match	Text

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Title of parts	N	Y	String		att:"/Library/Attributes/Data object properties/Title of parts" CONTAINS any ("user guide") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Total editing time	N	Y	String		att:"/Library/Attributes/Data object properties/Total editing time" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Type	N	Y	String		att: "/Library/Attributes/Data object properties/Type" CONTAINS any ("text") IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Typist	N	Y	String		att: "/Library/Attributes/Data object properties/Typist" MATCHES any ("Farr") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Version date	N	Y	String		att:"/Library/Attributes/Data object properties/Version date" CONTAINS any ("2014") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Version notes	N	Y	String		att:"/Library/Attributes/Data object properties/Version notes" APPEARS IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

Table 29. Data object property attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Version number	N	Y	String		att:"/Library/Attributes/Data object properties/Version number" IS any ("3.0") IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)
Words	N	Y	String		att:"/Library/Attributes/Data object properties/Words" APPEARS IN all IN all	Appears, Does Not Appear, Appears X-times, Appears Unique X-times, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text (=, >, >=, <, <=, any, all)

SharePoint attributes

This table describes SharePoint metadata attributes.

Table 30. Metadata for SharePoint

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
All day event (SharePoint)	N	N	Boolean	N	True if the Event is an all-day Event	att:"/Library/Attributes/SharePoint/All day event (SharePoint)" = true IN all	Appears, does not appear, =, !=	True, false
Announcement expiry date (SharePoint)	N	N	Date	N	Expiry date set for the announcement.	att:"/Library/Attributes/SharePoint/Announcement expiry date (SharePoint)" IS BETWEEN "2015-02-01" and "2016-01-12" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Approval status (SharePoint)	N	N	String	N	Information about moderation status of the specified Windows SharePoint object.	att:"/Library/Attributes/SharePoint/Approval status (SharePoint)" IS any ("approved") IN all		Text (Approved, denied, draft, pending or scheduled.)
Approver comments (SharePoint)	N	N	String	N	The moderation comments of the specified SharePoint weblog object.	att:"/Library/Attributes/SharePoint/Approver comments (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Ask me about (SharePoint)	N	N	String	N	SharePoint user profile's <i>ask me about</i> topics.	att:"/Library/Attributes/SharePoint/Ask me about (SharePoint)" CONTAINS any ("test") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Assistant (SharePoint)	N	N	Compound attribute	N	SharePoint assistant.	(att:"/Library/Attributes/SharePoint/Assistant (SharePoint)" WHERE (Name CONTAINS any ("jone")) IN all)	Appears, does not appear; where (account, email, id, name, url) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Author (SharePoint)	Y	N	String	N	Represents the specified author of the SharePoint object. It appears as Created By in the SharePoint UI.	att:"/Library/Attributes/SharePoint/Author (SharePoint)" CONTAINS any ("satwik") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Author account (SharePoint)	Y	N	String	N	The account name of the Author for the data object.	att:"/Library/Attributes/SharePoint/Author account (SharePoint)" CONTAINS any ("winnie") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Birthday (SharePoint)	N	N	Integer	N	SharePoint user's birthday (mm-dd).	att:"/Library/Attributes/SharePoint/Birthday (SharePoint)" IS any ("02-07") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Number

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Blog or List ID (SharePoint)	Y	N	Integer	N	ID of the SharePoint blog or list that the object belongs to.	att:"/Library/Attributes/SharePoint/Blog or List ID (SharePoint)" CONTAINS any ("45") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Number
Blog or List Name (SharePoint)	Y	N	String	N	Name of the SharePoint blog or list that the object belongs to.	att:"/Library/Attributes/SharePoint/Blog or List name (SharePoint)" CONTAINS any ("enron") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Category (SharePoint)	Y	N	Integer	N	Category associated with the Event.	att:"/Library/Attributes/SharePoint/Category (SharePoint)" CONTAINS any ("2") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Check in comment (SharePoint)	N	N	String	N	The check-in comments of the specified SharePoint object.	att:"/Library/Attributes/SharePoint/Check in comment (SharePoint)" CONTAINS any ("update") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Checked Out To	Y	N	String	N	Name of the user who currently has the data object checked out. This attribute is empty if the file is not checked out.	att:"/Library/Attributes/SharePoint/Checked out to (SharePoint)" CONTAINS any ("winnie") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Colleague (SharePoint)	N	N	Compound attribute	N	SharePoint colleague.	(att:"/Library/Attributes/SharePoint/Colleague (SharePoint)" WHERE (Name CONTAINS any ("jsitz")) IN all)	Appears, does not appear, appears X numbers of times, appears unique X times; where (account, email, id, name, url) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Comment post title (SharePoint)	N	N	String	N	Title of the post the comment was written for.	att:"/Library/Attributes/SharePoint/Comment post title (SharePoint)" CONTAINS any ("postman") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Comments (SharePoint)	N	N	String	N	Comments that are associated with the SharePoint object.	att:"/Library/Attributes/SharePoint/Comments (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact company (SharePoint)	N	N	String	N	Company of the contact.	att:"/Library/Attributes/SharePoint/Contact company (SharePoint)" IS any ("someco") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact email (SharePoint)	N	N	String	N	Email address of the contact – free form text.	att:"/Library/Attributes/SharePoint/Contact email (SharePoint)" CONTAINS any ("someco") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Contact first name (SharePoint)	N	N	String	N	First name of contact.	att:"/Library/Attributes/SharePoint/Contact first name (SharePoint)" IS any ("joe") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact full name (SharePoint)	N	N	String	N	Full name of the contact.	att:"/Library/Attributes/SharePoint/Contact full name (SharePoint)" IS any ("joe bob") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact home phone (SharePoint)	N	N	String	N	Home phone of the contact.	att:"/Library/Attributes/SharePoint/Contact home phone (SharePoint)" CONTAINS any ("512") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Contact job title (SharePoint)	N	N	String	N	Job title of the contact.	att:"/Library/Attributes/SharePoint/Contact job title (SharePoint)" CONTAINS any ("qae") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact last name (SharePoint)	N	N	String	N	Last name of the contact.	att:"/Library/Attributes/SharePoint/Contact last name (SharePoint)" IS any ("bob") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact mobile phone (SharePoint)	N	N	String	N	Mobile phone number of the contact.	att:"/Library/Attributes/SharePoint/Contact mobile phone (SharePoint)" IS any ("5123345145") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Contact preferred name (SharePoint)	N	N	String	N	Preferred name of contact.	att:"/Library/Attributes/SharePoint/Contact preferred name (SharePoint)" CONTAINS any ("jsitz") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact web page (SharePoint)	N	N	String	N	Web page URL of the contact with its associated description.	att:"/Library/Attributes/SharePoint/Contact web page (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact work address (SharePoint)	N	N	String	N	Work address of contact.	att:"/Library/Attributes/SharePoint/Contact work address (SharePoint)" CONTAINS any ("clay road") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Contact work city (SharePoint)	N	N	String	N	Work city of contact.	att:"/Library/Attributes/SharePoint/Contact work city (SharePoint)" IS any ("austin") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact work country (SharePoint)	N	N	String	N	Work country of contact.	att:"/Library/Attributes/SharePoint/Contact work country (SharePoint)" CONTAINS any ("USA") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact work fax (SharePoint)	N	N	String	N	Work fax number of contact.	att:"/Library/Attributes/SharePoint/Contact work fax (SharePoint)" CONTAINS any ("512") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Contact work phone (SharePoint)	N	N	Text	N	Work phone of contact.	att:"/Library/Attributes/SharePoint/Contact work phone (SharePoint)" IS any ("512-295-5555") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact work state (SharePoint)	N	N	String	N	Work state of contact.	att:"/Library/Attributes/SharePoint/Contact work state (SharePoint)" IS any ("pennsylvania") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Contact work ZIP (SharePoint)	N	N	String	N	Work ZIP code of contact.	att:"/Library/Attributes/SharePoint/Contact work ZIP (SharePoint)" IS any ("78610") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Content type (SharePoint)	Y	N	String	N	The content type of the SharePoint object.	att:"/Library/Attributes/SharePoint/Content type (SharePoint)" IS any ("post") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Created (SharePoint)	Y	N	Date	Y	A time stamp that indicates the creation time of the file.	att:"/Library/Attributes/SharePoint/Created (SharePoint)" IS BETWEEN "2015-05-03" and "2016-01-13" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Custom properties (SharePoint)	N	N	String	N	Custom properties created by a user on a SharePoint object.	att:"/Library/Attributes/SharePoint/Custom properties (SharePoint)/Name" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Declared record date (SharePoint)	N	N	Date	N	Date at which the item was declared a record (e.g. 2010-02-18)	att:"/Library/Attributes/SharePoint/Declared record date (SharePoint)" DOES NOT APPEAR IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Department (SharePoint)	N	N	String	N	Department name.	att:"/Library/Attributes/SharePoint/Department (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Description (SharePoint)	N	N	String	N	Alternative text for the object.	att:"/Library/Attributes/SharePoint/Description (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Discussion last updated (SharePoint)	Y	N	Date	N	Date when the discussion was last updated.	att:"/Library/Attributes/SharePoint/Discussion last updated (SharePoint)" < "2016-01-13" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Discussion message ID (SharePoint)	Y	N	String	N	Unique message ID associated with the discussion on the discussion board.	att:"/Library/Attributes/SharePoint/Discussion message ID (SharePoint)" CONTAINS any ("bbaf") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Discussion status bar (SharePoint)	N	N	Date	Y	Field containing information about the status bar for a SharePoint foundation object (date)	att:"/Library/Attributes/SharePoint/Discussion status bar (SharePoint)" = any ("2010-10-08") IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Discussion thread index (SharePoint)	Y	N	String	N	Unique identifier to determine the position of the post in the threading hierarchy of the discussion.	att:"/Library/Attributes/SharePoint/ Discussion thread index (SharePoint)" CONTAINS any ("0x0") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Discussion title (SharePoint)	Y	N	String	N	Title of the discussion post that the reply was written for.	att:"/Library/Attributes/SharePoint/ Discussion title (SharePoint)" CONTAINS any ("serverfault") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Dotted-line manager (SharePoint)	N	N	Compound attribute	N	SharePoint dotted line manager.	att:"/Library/Attributes/SharePoint/Dotted-line manager (SharePoint)" APPEARS IN all	Appears, does not appear, appears X numbers of times, appears unique X times; where (account, email, id, name, url) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Editor (SharePoint)	Y	N	String	N	The editor name or information that is associated with a person who is referenced by a Windows SharePoint Services contact object. It appears as Modified By in the SharePoint UI.	att:"/Library/Attributes/SharePoint/Editor (SharePoint)" CONTAINS any ("satwik") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Editor account (SharePoint)	Y	N	String	N	The account name of the editor for the data object.	att:"/Library/Attributes/SharePoint/Editor account (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Event date (SharePoint)	N	N	Date	Y	Start date for the event.	att:"/Library/Attributes/SharePoint/Event date (SharePoint)" <= "2016-01-18" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Event duration (SharePoint)	N	N	Integer	N	Duration for the event in seconds.	att:"/Library/Attributes/SharePoint/Event duration (SharePoint)" > 10 IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Number
Event end date (SharePoint)	N	N	Date	Y	End date of the event.	att:"/Library/Attributes/SharePoint/Event end date (SharePoint)" < "2015-06-07" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Event location (SharePoint)	N	N	String	N	Location for the event.	att:"/Library/Attributes/SharePoint/Event location (SharePoint)" IS any ("room 666") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Event type (SharePoint)	N	N	String	N	Event type	att:"/Library/Attributes/SharePoint/Event type (SharePoint)" IS any ("Deleted Recurring Event") IN all	Appears, does not appear, is, is not	Single event, recurring event, deleted recurring event, edited recurring event
Hire date (SharePoint)	N	N	Date	Y	SharePoint date of hire.	att:"/Library/Attributes/SharePoint/Hire Date (SharePoint)" APPEARS IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Interests (SharePoint)	N	N	String	N	SharePoint user's interest	att:"/Library/Attributes/SharePoint/Interests (SharePoint)" CONTAINS any ("development") IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Issue ID (SharePoint)	Y	N	Integer	N	Unique ID of the issue.	att:"/Library/Attributes/SharePoint/Issue ID (SharePoint)" APPEARS IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Numbers
Keywords (SharePoint)	N	N	String	N	The keyword summary of the object.	att:"/Library/Attributes/SharePoint/Keywords (SharePoint)" IS any ("favorite") IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Link URL (SharePoint)	Y	N	String	N	URL that forms the basis of the link, plus an optional description. In the absence of a description, the URL is repeated.	att:"/Library/Attributes/SharePoint/Link URL (SharePoint)" CONTAINS any ("enron") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Litigation hold (SharePoint)	N	N	String	N	List of all legal holds placed on an object.	att:"/Library/Attributes/SharePoint/Litigation hold (SharePoint)" APPEARS IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Managed keywords (SharePoint)	N	N	String	N	All managed keywords applied to an object.	att:"/Library/Attributes/SharePoint/Managed keywords (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Managed metadata tags (SharePoint)	N	N	String	N	All managed metadata tags applied to an object.	att:"/Library/Attributes/SharePoint/Managed metadata tags (SharePoint)" APPEARS IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Manager (SharePoint)	N	N	Compound attribute	N	SharePoint manager.	(att:"/Library/Attributes/SharePoint/Manager (SharePoint)" WHERE (Account APPEARS) IN all)	Appears, does not appear, appears X numbers of times, appears unique X times; where (account, email, id, name, url) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Member of (SharePoint)	N	N	String	N	SharePoint user profile memberships.	att:"/Library/Attributes/SharePoint/Member of (SharePoint)" IS any ("internal blog") IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Modified (SharePoint)	Y	N	Date	Y	Date object was modified.	att:"/Library/Attributes/SharePoint/Modified (SharePoint)" IS BETWEEN "2015-09-01" and "2016-01-19" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Modified by (SharePoint)	Y	N	String	N	Identifies the user who last modified the object.	att:"/Library/Attributes/SharePoint/Modified by (SharePoint)" CONTAINS any ("winnie") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Organization (SharePoint)	N	N	String	N	SharePoint organization	att:"/Library/Attributes/SharePoint/Organization (SharePoint)" APPEARS IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Original expiration date	N	N	Date	Y	Date at which the retention on the object is set to expire (eg. 2010-02-18)	att:"/Library/Attributes/SharePoint/Original expiration date (SharePoint)" < 3 weeks ago IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Past projects (SharePoint)	N	N	String	N	SharePoint user's past projects.	att:"/Library/Attributes/SharePoint/Past Projects (SharePoint)" CONTAINS any ("accounting") IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Post category (SharePoint)	N	N	String	N	Category associated with a post. Categories are defined as managed objects on the blog site for use with posts.	att:"/Library/Attributes/SharePoint/Post category (SharePoint)" CONTAINS any ("test automation") IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Public URL (SharePoint)	Y	N	String	N	URL of the SharePoint user profile.		Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Published date (SharePoint)	N	N	Date	Y	Date the post was published.	att:"/Library/Attributes/SharePoint/Published date (SharePoint)" = any ("2015-03-25") IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Publishing level (SharePoint)	N	N	String	N	Version control alert level information for the object.	att:"/Library/Attributes/SharePoint/Publishing level (SharePoint)" IS any ("draft") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Published, draft, or checkout
Quick link (SharePoint)	N	N	String	N	SharePoint quick link.	att:"/Library/Attributes/SharePoint/Quick link (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Rating (SharePoint)	N	N	Compound attribute	N	SharePoint rating.	att:"/Library/Attributes/SharePoint/Rating (SharePoint)" APPEARS IN all	Appears, does not appear, appears X number of times, appears unique X times; where (title, url, value) appears, is, is not, contains, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Record restrictions (SharePoint)	N	N	String	N	Restrictions applied to the record by policies in the record center.	att:"/Library/Attributes/SharePoint/Record restrictions (SharePoint)" CONTAINS any ("blockedit") IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	None, BlockDelete, BlockEdit
Recurring event (SharePoint)	N	N	Boolean	N	True if event is recurring.	att:"/Library/Attributes/SharePoint/Recurring event (SharePoint)" = true IN all	Appears, does not appear, =, !=	True, false

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Referred URL (SharePoint)	N	N	String	N	Represents a URL referred to by an object.	att:"/Library/Attributes/SharePoint/Referred URL (SharePoint)" APPEARS IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Related issues (SharePoint)	N	N	String	N	Issues marked as related to the issue.	att:"/Library/Attributes/SharePoint/Related issues (SharePoint)" APPEARS IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Routing rule description (SharePoint)	N	N	String	N	Description associated with the object that matched a routing rule.	att:"/Library/Attributes/SharePoint/Routing rule description (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Schools (SharePoint)	N	N	String	N	SharePoint user's schools.	att:"/Library/Attributes/SharePoint/Schools (SharePoint)" IS any ("harvard") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Site owner (SharePoint)	Y	N	Compound attribute	N	SharePoint site owner	(att:"/Library/Attributes/SharePoint/Site owner (SharePoint)" WHERE (Name CONTAINS any ("jsitz")) IN all)	Appears, does not appear; where (Account, email, ID, name, url) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Skills (SharePoint)	N	N	String	N	SharePoint user's skills	att:"/Library/Attributes/Skills (SharePoint)" APPEARS IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Status message (SharePoint)	N	N	String	N	SharePoint user's status message.	att:"/Library/Attributes/SharePoint/Status Message (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Survey completed (SharePoint)	N	N	Boolean	N	True if the SharePoint survey response was completed.	att:"/Library/Attributes/SharePoint/Survey completed (SharePoint)" = true IN all	Appears, does not appear, =, !=	True, false
Tag (SharePoint)	N	N	Compound attribute	N	The tag that appears in user profile.	(att:"/Library/Attributes/SharePoint/Tag (SharePoint)" WHERE (URL APPEARS) IN all)	Appears, does not appear, appears X times, appears X unique times; where (name, url) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Task assigned (SharePoint)	N	N	String	N	Usernames to whom the task has been assigned.	att:"/Library/Attributes/SharePoint/Task assigned to (SharePoint)" CONTAINS any ("klay") IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Task due date (SharePoint)	Y	N	Date	N	Due date for task.	att:"/Library/Attributes/SharePoint/Task due date (SharePoint)" IS BETWEEN "2015-09-01" and "2015-10-01" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Task or issue status (SharePoint)	Y	N	Boolean	N	Status of the task or issue.	att:"/Library/Attributes/SharePoint/Task or Issue status (SharePoint)" IS any ("Not Started") IN all	Appears, does not appear, is, is not	Not started, in progress, completed, deferred, waiting on someone else, active, resolved, closed
Task percent complete (SharePoint)	N	N	Integer	N	Completion percentage of the task.	att:"/Library/Attributes/SharePoint/Task percent complete (SharePoint)" < 50 IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Number

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Task predecessors (SharePoint)	N	Y	String	N	Names of tasks that have a defined precedence to the task.	att:"/Library/Attributes/SharePoint/Task predecessors (SharePoint)" CONTAINS any ("marketing collateral") IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Task priority (SharePoint)	N	N	Boolean	N	Priority of the task.	att:"/Library/Attributes/SharePoint/Task priority (SharePoint)" IS any ("(1) High") IN all	Appears, does not appear, is, is not	1. High 2. Normal 3. Low
Task start date (SharePoint)	N	N	Date	Y	Start date for task.	att:"/Library/Attributes/Task/Task start date" IS BETWEEN 1 months ago and 6 months ago IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Title (SharePoint)	Y	N	String	N	The title of the data object. By default, this attribute is the title of the Office document as shown in Document Properties. Configurable by the user during check-in.	att:"/Library/Attributes/SharePoint/Title (SharePoint)" CONTAINS any ("ECM") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
URL	Y	N	String	N	The URL that points to the data object on SharePoint.	att:"/Library/Attributes/SharePoint/URL (SharePoint)" CONTAINS any ("winnie") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
User name (SharePoint)	N	N	String	N	SharePoint user name	att:"/Library/Attributes/SharePoint/User name (SharePoint)" CONTAINS any ("jsitz") IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Version	Y	N	String	N	The version of the data object, from a SharePoint server perspective.	att:"/Library/Attributes/SharePoint/Version (SharePoint)" IS BETWEEN "1" and "5" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	True, false, number
Virus status (SharePoint)	N	N	String	N	The virus scan of a specified Windows SharePoint object.	att:"/Library/Attributes/SharePoint/Virus status (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	0 = Clean 1 = Virus reported 2 = Virus reported, can be removed 3 = Virus previously reported, removed since 4 = Virus reported, clean failed 5 = Virus reported, delete recommended 6 = Timeout reported last processing virus
Wiki average rating (SharePoint)	N	N	Integer	N	Average of all ratings applied to the object.	att:"/Library/Attributes/SharePoint/Wiki average rating (SharePoint)" APPEARS IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Number

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Wiki page categories (SharePoint)	N	Y	String	N	Categories associated with a wiki page. Categories are defined as managed objects on the wiki page.	att:"/Library/Attributes/SharePoint/Wiki page categories (SharePoint)" APPEARS IN all	Appears, does not appear, appears X number of times, appears unique X times, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Wiki page comment (SharePoint)	N	N	String	N	Comment describing the wiki page. There can only be one per page.	att:"/Library/Attributes/SharePoint/Wiki page comment (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Wiki publishing contact (SharePoint)	N	N	String	N	User name of contact.	att:"/Library/Attributes/SharePoint/Wiki publishing contact (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Wiki publishing contact email (SharePoint)	N	N	String	N	Email address of the contact: free form text.	att:"/Library/Attributes/SharePoint/Wiki publishing contact email (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Wiki publishing contact name (SharePoint)	N	N	String	N	Friendly name of contact – free form text.	att:"/Library/Attributes/SharePoint/Wiki publishing contact name (SharePoint)" APPEARS IN all	Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 30. Metadata for SharePoint (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Operators Supported	Value Supported
Wiki rating count (SharePoint)	N	N	Integer	N	Total number of ratings applied to an object.	att:"/Library/Attributes/SharePoint/Wiki rating count (SharePoint)" APPEARS IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	True, false, number

Calendar metadata attributes

This table describes metadata attributes for calendar items.

Table 31. Metadata for calendar

Attribute Name	Dense	Multiple	Type	Normalized	Description
Event duration	No	No	Integer	No	Time in minutes
Event end date	No	No	String	Yes (ISO 8601)	Date a calendar entry end (YYYY-MM-DD)
Event location	No	No	String	Yes (lowercase)	Location field in a calendar entry
Event start date	No	No	String	Yes (ISO 8601)	Date that a calendar entry begins (YYYY-MM-DD)
is Meeting	No	No	Boolean	No	True if the calendar entry is a meeting (Any entry that has attendees)

Contact metadata attributes

This table describes metadata attributes for contacts.

Table 32. Metadata for contact

Attribute Name	Dense	Multiple	Type	Normalized	Description
Business address	No	No	String	Yes (lowercase)	Business address that is listed for a contact entry
Business phone	No	No	String	Yes (lowercase)	Business phone that is listed for a contact entry
Company	No	No	String	Yes (lowercase)	Company that is listed for a contact entry
Contact job title	No	No	String	Yes (lowercase)	Job title that is listed for a contact entry

Table 32. Metadata for contact (continued)

Attribute Name	Dense	Multiple	Type	Normalized	Description
Contact keywords	No	No	String	Yes (lowercase)	Keywords that are listed for a contact entry
Department	No	No	String	Yes (lowercase)	Department that is listed for a contact entry
Email display name	No	No	String	Yes (lowercase)	Email display name that is listed for a contact entry
Full name	No	No	String	Yes (lowercase)	Full name that is listed for a contact entry
Home address	No	No	String	Yes (lowercase)	Home address that is listed for a contact entry
Home phone	No	No	String	Yes (lowercase)	Home phone that is listed for a contact entry
IM address	No	No	String	Yes (lowercase)	Instant Messenger address that is listed for a contact entry
Mobile phone	No	No	String	Yes (lowercase)	Mobile phone number that is listed for a contact entry
Office location	No	No	String	Yes (lowercase)	Office location that is listed for a contact entry
Pager	No	No	String	Yes (lowercase)	Pager number that is listed for a contact entry
Web page	No	No		Yes (lowercase)	Web page that is listed for a contact entry

Digital signatures metadata attributes

This table describes attributes that are used in IBM StoredIQ data analysis.

Table 33. Metadata for digital signatures

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Supported Operators	Value Supported
Encryption type	No	No	Enum	Yes (lowercase)	Lotus Notes' or 'General'	att: "/ Library / Attributes / Digital signatures / Encryption type" IS any (General) IN all	Appears, Does Not Appear, Is, Is Not	General, Lotus Notes, Rights Management Protected
Hash	Yes	No	String	Yes (lowercase)	Hash value for the object. Hash is not computed for data objects on volumes that have only a system metadata index.	att: "/ Library / Attributes / Digital signatures / Hash" DOES NOT CONTAIN any ("00") IN all IN all	Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
NSRL digital signature	Yes	No	Boolean	No	True if the object's hash value matches an item on the NSRL hash list. The Determine whether data objects have NSRL digital signature option must be selected on the Harvester settings pane for each data server prior to harvest.	att: "/ Library / Attributes / Digital signatures / NSRL digital signature" != true IN all	=, !=	True, False

Processing metadata attributes

This table describes attributes that are used in data processing.

To harvest processing metadata attributes, the **Binary processing** options must be selected in the harvester settings on each data server. For more information about configuring harvester settings, see *Configuring harvester settings*.

Table 34. Metadata for processing data

Attribute Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Supported Operators	Value Supported
Binary error code	No	Yes	String	Yes (lowercase)	When binary processing is performed, the following codes apply: Indexing of binary text failed, Binary text partially indexed, Binary content tagging failed, Max entities reached in binary text, Binary text extraction failed, Binary full processing completed	att:"/ Library/ Attributes/ Processing/ Binary error code" APPEARS IN all	Appears, Does Not Appear, Appears X-Times, Appears Unique X-Times, Is, Is Not	Text (any, all, =, >, >=, <, <=, Binary content tagging, failed, Binary full processing completed, Binary text extraction failed, Binary text partially indexed, Indexing of binary text failed, Max entities reached in binary text)
Binary processing state	Yes	No	Integer	Yes	State of binary processing: Not Performed, Successful, Partial, Failed	att:"/ Library/ Attributes/ Processing/ Binary processing state" IS any (Successful) IN all	Is, Is Not	Text (Not Performed, Successful, Partial, Failed)

Table 34. Metadata for processing data (continued)

Attribute Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Supported Operators	Value Supported
Content error code	No	Yes	String	Yes (lowercase)	When the content processing state is "Partial" or "Failed," the following codes apply: Configured skipped data object, Unsupported file format, Configured max. object size, Indexing failed, Partially indexed, Unknown file format, Content tagging failed, Max entities reached, Encrypted or password protected, Text extraction failed, No text in this file format, Could not read source object, Max entity value length reached	att:"/ Library/ Attributes/ Processing/ Content error code" IS any ("Configured skipped data object") IN all	Appears, Does Not Appear, Appears X-Times, Appears Unique X-Times, Is, Is not	Text (=, >, >=, <, <=, Configured max object size, Configured skipped data object, Content tagging failed, Could not read source object, Encrypted or password protected, Indexing failed, Max entities reached, Max entity value length reached, No text in this file format, Partially indexed, Text extraction failed, Unknown file format, Unsupported file format)
Content processing state	Yes	No	Integer	Yes	State of content processing: Not Performed, Successful, Partial, Failed	att:"/ Library/ Attributes/ Processing/ Content processing state" IS any (Successful) IN all	Is, Is Not	Text (Failed, Not Performed, Partial, Successful)

Table 34. Metadata for processing data (continued)

Attribute Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Supported Operators	Value Supported
System error code	No	Yes		Yes (lowercase)	When system processing state is "Partial," the following code applies: User/Group lookup failed	att:"/ Library/ Attributes/ Processing/ System error code" IS NOT any ("User/ Group lookup failed") IN all	Appears, Does Not Appear, Appears X-Times, Appears Unique X-Times, Is, Is not	Text (=, >, >=, <, <=, User/Group lookup failed)
System processing state	Yes	No	Integer	Yes	State of system processing: Successful, Partial	att:"/ Library/ Attributes/ Processing/ System processing state" IS any (Successful) IN all	Is, Is Not	Text (Failed, Not Performed, Partial, Successful)

Task metadata attributes

This table describes metadata for task attributes.

To harvest task metadata attributes, the **Harvest miscellaneous email items** in the harvester settings on each data server must be selected prior to harvest. For more information about configuring harvester settings, see Configuring harvester settings.

Table 35. Metadata for task items

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Supported Operators	Value Supported
Task complete	No	No	Boolean	No	True if the task is marked as complete	att:"/ Library/ Attributes/ Task/Task complete" = false IN all	Appears, Does not appear, =, !=	True, false
Task due date	No	No	String	Yes (ISO 8601)	Date the task is marked as being due (YYYY-MM-DD)	att:"/ Library/ Attributes/ Task/Task due date" IS BETWEEN "2013-05-01" and "2013-05-15" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)

Table 35. Metadata for task items (continued)

Name	Dense	Multiple	Type	Normalized	Description	Filter Code Example	Supported Operators	Value Supported
Task owner	No	No	String	Yes	Creator of the task	att:"/ Library/ Attributes/ Task/Task owner" CONTAINS any ("gwilliams") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text
Task percent complete	No	No	Integer	Yes	Percent complete of a task	att:"/ Library/ Attributes/ Task/Task percent complete" = any (75) IN all	Appears, Does Not Appear, =, !=, <, <=, >, >=, Is Between, Is Not Between	Number
Task start date	No	No	String	Yes (ISO 8601)	Date the task is marked as being started (YYYY-MM-DD)	att:"/ Library/ Attributes/ Task/Task start date" IS BETWEEN "2013-05-01" and "2013-05-31" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Task status	No	No	String	Yes	Status of a task	att:"/ Library/ Attributes/ Task/Task status" CONTAINS any ("Complete") IN all	Appears, Does Not Appear, Is, Is Not, Contains, Does Not Contain, Begins With, Does Not Begin With, Ends With, Does Not End With, Matches, Does Not Match	Text

Attribute value types

This table describes the value and instance-based expressions that can be used with a query condition, depending on the type of attribute.

Table 36. Attribute value types

Value Type	Dense		Sparse	
	Single Instance	Multi-instance	Single Instance	Multi-instance
String			Appears	Appears
			Appears uniquely	Appears uniquely
			Does not appear	Does not appear
	Is	Is	Is	Is
	Is not	Is not	Is not	Is not
	Contains	Contains	Contains	Contains
	Does not contain	Does not contain	Does not contain	Does not contain
	Begins with	Begins with	Begins with	Begins with
	Ends with	Ends with	Ends with	Ends with
	Matches regular expression	Matches regular expression	Matches regular expression	Matches regular expression
		at least n times (n>0)		at least n times
		at most n times (n>0)		at most n times
	exactly n times (n=0)		exactly n times	
Date			Appears	Appears
			Appears uniquely	Appears uniquely
			Does not appear	Does not appear
	Is	Is	Is	Is
	Between x and y days ago	Between x and y days ago	Between x and y days ago	Between x and y days ago
	At least x days ago	At least x days ago	At least x days ago	At least x days ago
	At most x days ago	At most x days ago	At most x days ago	At most x days ago
		at least n times (n>0)		at least n times
		at most n times (n>0)		at most n times
		exactly n times (n=0)		exactly n times
			Appears	Appears
	Int			Appears

Table 36. Attribute value types (continued)

Value Type	Dense		Sparse	
	Single Instance	Multi-instance	Single Instance	Multi-instance
			Appears uniquely	Appears uniquely
			Does not appear	Does not appear
	Equals	Equals	Equals	Equals
	At least	At least	At least	At least
	At most	At most	At most	At most
	Between	Between	Between	Between
		at least n times (n>0)		at least n times
		at most n times (n>0)		at most n times
		exactly n times (n=0)		exactly n times
	Not performed		Not performed	
	Successful		Successful	
	Partial		Partial	
	Failed		Failed	

IBM Lotus Notes metadata attributes

This table describes metadata for IBM Lotus Notes attributes.

Table 37. Metadata for IBM Lotus Notes

Attribute Name	Dense	Multiple	Type	Normalized	Description
Attachments					Whether the data object has attachments. True if message has attachments
Bcc, BlindCopyTo, EnterBlindCopyTo					Recipients that are blindcopied on the email data object
CC, CcopyTo, EnterCopyTo					Recipients that are copied on the email data object
Thread_Topic	N	N	String	N	The topic of the first message in a conversation thread
From					The sender of the email data object

Table 37. Metadata for IBM Lotus Notes (continued)

Attribute Name	Dense	Multiple	Type	Normalized	Description
Importance					The importance level that is assigned by the sender of the email data object: High, Normal, Low
Message-ID	N	N	String	Y	Corresponds to the message ID field as specified in [RFC2822]
Is-part-of					TRUE if object is part of email body, email attachment, miscellaneous email item's body or miscellaneous email item's attachment
Create, ComposedDate	N	N	Date	Y	Date the message was created (YYYY-MM-DD)
DeliveredDate	N	N	Date	Y	
Modify	N	N	String	Y	Date the message was last modified (YYYY-MM-DD)
DeliveryPriority	N	N	String	Y	The order or speed at which the message must be sent by the messaging system software: Urgent, Normal, Low
ReturnReceipt	N	N	Boolean	N	TRUE if a message sender requests proof of delivery for a particular recipient
Sensitivity					The sensitivity level that is assigned by the sender of the email data object: Normal, Personal, Private, Confidential
Sent, Posted					Date the message was sent (YYYY-MM-DD)

Table 37. Metadata for IBM Lotus Notes (continued)

Attribute Name	Dense	Multiple	Type	Normalized	Description
SentRepresenting-Name	N	N	String	N	Contains the display name for the messaging user that is represented by the sender
Subject					The subject of the email data object
To, SendTo					The recipient/s of the email data object
Attachments					Whether the data object has attachments. True if message has attachments

Enterprise vault metadata attributes

This table describes metadata for enterprise vault attributes.

Table 38. Metadata for enterprise vault

Name	Dense	Multiple	Type	Description
Archive date (Enterprise Vault)	N	N	Date	Date that the object was archived (YYYY-MM-DD).
Author (Enterprise Vault)	N	N	String	The creator of the object, as shown in the archive properties
Content fingerprint (Enterprise Vault)	N	N	String	Content fingerprint of the object
Custom ID (Enterprise Vault)	N	N	String	Application-specific ID for the object
Custom properties (Enterprise Vault)	N	Y	String	Application-specific information for the object
Dedupe fingerprint (Enterprise Vault)	N	N	String	Deduplication fingerprint of the object
Domino UNID (Enterprise Vault)	N	N	String	Domino Universal NoteID
FSA original name (Enterprise Vault)	N	N	String	File system archive original name
Keywords (Enterprise Vault)	N	Y	String	Keywords that are associated with the data object
Message type (Enterprise Vault)	N	N	String	Data type of the object (DOC, XLS, MSG)

Table 38. Metadata for enterprise vault (continued)

Name	Dense	Multiple	Type	Description
Missing content reason (Enterprise Vault)	N	N	Numeric String	Reason that object content is not available in the archive
Original location (Enterprise Vault)	N	N	String	Original location of the archived object
Retention category (Enterprise Vault)	N	N	String	Name of retention category that is applied to the object
Retention expires (Enterprise Vault)	N	N	Date String	Date that the retention on the object expires (YYYY-MM-DD)

Jive metadata attributes

This table describes attributes that are used in Jive items.

Table 39. Metadata for Jive

Attribute Name	Dense	Multiple	Type	Normalized	Description
Author ID (Jive)	No	No	String	No	The unique ID of the author within Jive.
Author Name (Jive)	No	No	String	No	The author's full name within Jive.
Author Username (Jive)	No	No	String	No	The author's user name within Jive.
Content ID (Jive)	No	No	String	No	The unique ID of content within Jive.
Content URL (Jive)	No	No	String	No	The content's URL within Jive.
Discussion Question (Jive)	No	No	Boolean	No	Indicates whether the Jive discussion was marked as a question.
Like Count (Jive)	No	No	Date	No	Indicates that the number of times that content was marked as "liked" on Jive.
Place ID (Jive)	No	No	String	No	The unique ID of the parent place on Jive.
Place Name (Jive)	No	No	String	No	The name of the parent place on Jive.
Place URL (Jive)	No	No	String	No	The URL of the parent place on Jive.

Table 39. Metadata for Jive (continued)

Attribute Name	Dense	Multiple	Type	Normalized	Description
Published Date (Jive)	No	No	Date	No	The post's published date on Jive.
Reply Count (Jive)	No	No	Date	No	Indicates that the number of replies posted to content on Jive.
Status (Jive)	No	No	String	No	Indicates a discussion's status on Jive.
Subject (Jive)	No	No	String	No	Indicates content's subject on Jive.
Version (Jive)	No	No	String	No	Indicates a document's version on Jive.
View Count (Jive)	No	No	Date	No	Indicates that the number of times content was viewed on Jive.

Chatter metadata attributes

This table describes attributes that are used in Chatter items

Table 40. Metadata for Chatter

Attribute Name	Dense	Multiple	Type	Normalized	Description
Object ID (Chatter)	No	No	String	No	The unique ID of the object within Chatter.
Object type (Chatter)	No	No	String	No	The type of object within Chatter.
Parent group ID (Chatter)	No	No	String	No	The unique ID of the parent group or profile within Chatter.
Parent group name (Chatter)	No	No	String	No	The name of the parent group or profile within Chatter.
Parent post ID (Chatter)	No	No	String	No	The unique ID of the parent post within Chatter.
Parent post type (Chatter)	No	No	Boolean	No	The object type of the parent post within Chatter.
Created by user ID (Chatter)	No	No	Date	No	The unique ID of the creator within Chatter.

Table 40. Metadata for Chatter (continued)

Attribute Name	Dense	Multiple	Type	Normalized	Description
Created by username (Chatter)	No	No	String	No	The user name of the creator within Chatter.
Created date (Chatter)	No	No	Datetime	No	The posted date within Chatter.
Comment count (Chatter)	No	No	Integer	No	The number of comments for a post within Chatter.
Like count (Chatter)	No	No	Integer	No	The number of "likes" for a post or comment within Chatter.
Title (Chatter)	No	No	String	No	The title for links and attachments within Chatter.
Link URL (Chatter)	No	No	String	No	The content of links posted within Chatter.

Box attributes

The Box attributes are listed in the following table.

Table 41. Box attributes

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Bookmark URL (Box)	N	N	String	URL set for a Bookmark object.	att:"/Library/Attributes/Box/Bookmark URL (Box)" CONTAINS any ("google") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Collaborator (Box)	N	Y	Compound attribute	Collaborator set on an object. Note: In the case that a document is incrementally reharvested after only an attribute is updated on Box, the change of the attribute is not reflected in indexes. It is necessary to reharvest with the full harvest option.	att:"/Library/Attributes/Box/Collaborator (Box)" APPEARS > 3 IN all	appears, does not appear, appears X times (=, >, >=, <, <=) appears unique x times (=, >, >=, <, <=), where (Collaborator id, collaborator login, collaborator name, collaborator role, collaborator type)	Text

Table 41. Box attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Collaborator ID	N	N	String (sub-attribute for Collaborator (Box))	Box user ID. Note: In the case that a document is incrementally reharvested after only an attribute is updated on Box, the change of the attribute is not reflected in indexes. It is necessary to reharvest with the full harvest option.	(att:"/Library/Attributes/Box/Collaborator (Box)" WHERE ("Collaborator Id" APPEARS) IN all)	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Collaborator Login	N	N	String (sub-attribute for Collaborator (Box))	Primary email address of the Box user. Note: In the case that a document is incrementally reharvested after only an attribute is updated on Box, the change of the attribute is not reflected in indexes. It is necessary to reharvest with the full harvest option.	(att:"/Library/Attributes/Box/Collaborator (Box)" WHERE ("Collaborator Login" CONTAINS any ("nirmal")) IN all)	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Collaborator Name	N	N	String (sub-attribute for Collaborator (Box))	Box user name. Note: In the case that a document is incrementally reharvested after only an attribute is updated on Box, the change of the attribute is not reflected in indexes. It is necessary to reharvest with the full harvest option.	(att:"/Library/Attributes/Box/Collaborator (Box)" WHERE ("Collaborator Name" BEGINS WITH any ("ni")) IN all)	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Collaborator Role	N	N	String (sub-attribute for Collaborator (Box))	Access levels defined by Box 'owner', 'co-owner', 'editor', 'viewer', 'previewer', 'uploader', and 'previewer uploader' or 'viewer uploader'. Note: In the case that a document is incrementally reharvested after only an attribute is updated on Box, the change of the attribute is not reflected in indexes. It is necessary to reharvest with the full harvest option.	(att:"/Library/Attributes/Box/Collaborator (Box)" WHERE ("Collaborator Role" IS any ("editor")) IN all)	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	

Table 41. Box attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Collaborator Type	N	N	String (sub-attribute for Collaborator (Box))	Box collaborator type. Note: In the case that a document is incrementally reharvested after only an attribute is updated on Box, the change of the attribute is not reflected in indexes. It is necessary to reharvest with the full harvest option.	(att:"/Library/Attributes/Box/Collaborator (Box)" WHERE ("Collaborator Type" IS any ("user")) IN all)	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Comment (Box)	N	Y	Compound attribute	A comment that was entered in a file in Box.	(att:"/Library/Attributes/Box/Comment (Box)" WHERE ("Created by name" CONTAINS any ("nirmal")) IN all)	appears, does not appear, appears X times (=, >, >=, <, <=) appears unique x times (=, >, >=, <, <=), where (Created at, Created by id, Created by login, Created by name, message)	Text
Content created at (Box)	N	N	Date	Date that the content for this object was created.	att:"/Library/Attributes/Box/Content created at (Box)" IS BETWEEN "2015-09-01" and "2015-09-15" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Content modified at (Box)	N	N	Date	Date that the content for this object was modified.	att:"/Library/Attributes/Box/Content modified at (Box)" IS NOT BETWEEN "2014-03-15" and "2014-03-31" IN all	Appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Created at (Box)	N	N	Date	Date that this object was created on the server of Box.	att:"/Library/Attributes/Box/Created at (Box)" IS BETWEEN "2015-12-01" and "2016-01-08" IN all	appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Created by (Box)	N	N	Compound attribute	The user that created a file.	(att:"/Library/Attributes/Box/Created by (Box)" WHERE (Name CONTAINS any ("kumar")) IN all)	appears, does not appear; where (id, login, name, type) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Custom properties (Box)	N	Y	String	Custom metadata set on this object.	att:"/Library/Attributes/Box/Custom properties (Box)/Title" APPEARS IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 41. Box attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Description (Box)	N	N	String	The description that is entered for the data object.	att:"/Library/Attributes/Box/Description (Box)" CONTAINS any ("marketing") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Item status (Box)	N	N	String	Item status for the data object, whether it is deleted or not.	att:"/Library/Attributes/Box/Item status (Box)" IS any ("active") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Modified at (Box)	N	N	Date	Date that this object was modified on the server of Box.	att:"/Library/Attributes/Box/Modified at (Box)" IS BETWEEN "2015-10-14" and "2015-10-21" IN all	appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Modified by (Box)	N	N	Compound attribute	The user that modified a file.	(att:"/Library/Attributes/Box/Modified by (Box)" WHERE (Name CONTAINS any ("nirmal")) IN all)	appears, does not appear; where (id, login, name, type) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Owned by (Box)	N	N	Compound attribute	The user that owns a file.	(att:"/Library/Attributes/Box/Owned by (Box)" WHERE (Name CONTAINS any ("morgan")) IN all)	appears, does not appear; where (id, login, name, type) Appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Shared link access (Box)	N	N	String	Access level of the shared link for this object.	att:"/Library/Attributes/Box/Shared link access (Box)" IS any ("open") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Shared link download count (Box)	N	N	Integer	Download count for the shared link for this object.	att:"/Library/Attributes/Box/Shared link download count (Box)" IS BETWEEN 0 and 1 IN all	appears, does not appear, =, !=, <, <=, >, >=, between, not between	Number
Shared link download URL (Box)			String	Download URL of the shared link for this object.	att:"/Library/Attributes/Box/Shared link download URL (Box)" CONTAINS any ("boxnote") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Table 41. Box attributes (continued)

Name	Dense	Multiple	Type	Description	Filter Code Example	Supported Operators	Value Supported
Shared link password enabled (Box)	N	N	Boolean	Whether passwords are required to access the shared link for this object.	att:"/Library/Attributes/Box/Shared link password enabled (Box)" != false IN all	appears, does not appear, =, !=	True, false
Shared link preview count (Box)	N	N	Integer	Preview count for the shared link for this object.	att:"/Library/Attributes/Box/Shared link preview count (Box)" APPEARS IN all	appears, does not appear, =, !=, <, <=, >, >=, between, not between	Number
Shared link unshared (Box)	N	N	Date	The date that the shared link for this object was unshared.	att:"/Library/Attributes/Box/Shared link unshared at (Box)" IS BETWEEN "2015-06-15" and "2016-01-11" IN all	appears, does not appear, =, !=, <, <=, >, >=, between, not between	Absolute date, Relative date (days ago, weeks ago, months ago, years ago, days ahead, weeks ahead, months ahead, years ahead)
Shared link URL (Box)	N	N	String	URL of the shared link for this object.	att:"/Library/Attributes/Box/Shared link URL (Box)" CONTAINS any ("marketing") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Shared link vanity URL (Box)	N	N	String	Vanity URL of the shared link for this object.	att:"/Library/Attributes/Box/Shared link vanity URL (Box)" CONTAINS any ("pricelist") IN all	appears, does not appear, is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text
Tag (Box)	N	Y	String	Tags that are applied for this object.	att:"/Library/Attributes/Box/Tag (Box)" IS any ("expenses") IN all	appears, does not appear, appears X times (=, >, >=, <, <=) appears unique x times (=, >, >=, <, <=), is, is not, contains, does not contain, begins with, does not begin with, ends with, does not end with, matches, does not match	Text

Note: For information about compound attributes, see “Conditions for compound attributes” on page 61.

Content Manager attributes

This table describes attributes that are used in IBM Content Manager items.

Table 42. Content Manager attributes

Name	Dense	Multiple	Type	Description
Name	N	N	String	The name of the item. It is the value of the ICM\$NAME attribute in IBM Content Manager, if this attribute is in the item type.
Itemtype name	Y	N	String	The name of the item type that this document is stored in.
Creator	Y	N	String	The userid of the creator of the document. It maps to the CREATEUSERID attribute in IBM Content Manager.

Table 42. Content Manager attributes (continued)

Name	Dense	Multiple	Type	Description
Last modifier	Y	N	String	The userid of the last modifier of the document. It maps to the LASTCHANGEDUSERID attribute in IBM Content Manager.
Version number	Y	N	Integer	The version number of the document. It maps to the VERSIONID attribute in IBM Content Manager
Semantic type	Y	N	String	The semantic type of the document. It maps to the SEMANTICTYPE attribute in IBM Content Manager.
Original file name	N	N	String	The original file name for this document. For document model item types, it is the original file name for the first base part. It maps to the ORIFNAME attribute in IBM Content Manager.
Retention date	N	N	Date/ time	The retention date of the document. It is the date that the document is retained until, if the item type was enabled for retention. It maps to the ICM\$RETENTIONDATE in IBM Content Manager.
Expiration date	N	N	Date/ time	The expiration date of the document. It is the date when the document expires and might be automatically deleted. It is used only if the item type was enabled for expiration. It maps to the ICM\$AUTODELETEDATE in IBM Content Manager.
Managed record flag	Y	N	Integer	This flag indicates whether the document is declared as a record. It maps to the SYSROOTATTRS.ICMMANAGEDRECORD in IBM Content Manager.
Mime type	N	N	String	The mime type of the document. If it is a document model item type, it is the mime type of the first base part.
Number of content elements	Y	N	Integer	The number of parts that this document contains.
Custom properties	N	N	String	The custom metadata for this document. It includes all user-defined and system attributes that are not included in the properties above or in the file system metadata.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

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

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

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

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

The client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

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

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the

names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: © Copyright IBM Corp. 2004, 2015. All rights reserved.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

[Depending upon the configurations deployed, this Software Offering may use [session] [and] [persistent] cookies that collect each user's [name,] [user name,] [password,] [profile name,] [position,] or [other personally identifiable information] for purposes of [session management,] [authentication,] [enhanced user usability,] [single sign-on configuration] [or other usage tracking or functional purposes.] These cookies [cannot be disabled] [can be disabled, but disabling them will also [likely] eliminate the functionality they enable].

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Index

A

- access mode attribute 62
- action 4
 - run 40
- actions 29, 39
- actions on data 39
- analytics filter expressions 66
- ASCII characters 55
- at-least expressions 64
- attribute 62
- attribute condition 57
- attribute conditions 58
- audit report 14
- auto-classification model 15

B

- batch element 74
- batch information 74
- boolean operator 22
 - syntax 22
- Box attributes 199
- build 18
 - filter 18

C

- cancel 40
 - scheduled action 40
- category view
 - options 35
- clone 18
 - filter 18
- code view 22
- collapse 29
 - hierarchy 29
- compound attribute 58
- Concordance DAT 67
- Concordance DAT (Light) 67
- condition operators 58
- conditions 58, 60, 61, 65
 - compound attributes 61
 - date attributes 59
 - enumeration attributes 58
 - multivalued sparse or dense attributes 61
 - no-inheritance 65
 - number attributes 60
 - parent and child conditions 64
 - sparse attributes 60
- configure 24
 - columns 24
- contained data 4
- Content Manager attributes 203
- create 9, 19, 20, 21, 22, 27, 28
 - auto-classification filter 22
 - date filter 19
 - email filter 21
 - file filter 19
 - full-text filter 21

- create (*continued*)
 - infoset 9
 - infoset with a symmetric difference 28
 - intersection infoset 27
 - name filter 20
 - size filter 20
 - subtraction infoset 28
 - union infoset 27
- created view
 - options 35
- Custom CSV Export report 44
- customized reports 4

D

- DAT 67
- DAT config file 67
- data map 33
 - Data Map Details 33
 - Group By 33
 - View By 33
 - Viewing x of x total object 33
- data object property attributes 110
- data object viewer 15
- data objects 39
- data overlay 36
- data reports 41
- Data Source Type 33, 35
- data source types 33
- delete 12, 17
 - filter 17
 - infoset 12
- dense attribute 58
- discovery export 67, 74
- document entries 74
- download 36
 - CSV data 36
- duplicate identification infoset 30
- duplicate infoset 30
- Duplicate Operations 30

E

- edit 11, 18
 - filter 18
 - infoset 11
- EDRM XML 67, 68, 69, 70, 73, 74
 - metadata tags 74
- EDRM XML tags
 - valid data types 74
- EDRM XML v1.1 metadata tags 74
- enhancements 37
- exception object 50
- exceptions 49
 - Event 49
 - Exception Objects 49
 - Types 49
- Execution Log 13
- expand 29

- expand (*continued*)
 - hierarchy 29
- export 50
 - exception object list 50
- expression language 51, 63

F

- feedback 15
- file system directory layout 69
- filter 3, 17, 22, 24
 - administration 17
 - attribute 17
 - condition reference 51, 63
 - filters 17
 - format 44
 - .csv file 44
 - manifest file 44
 - full-text filter 54
 - full-text macro 54

G

- generate 45
 - system report 45
 - term hit report 45
- Generate Report link 14
- getting started 7
- global deduplication 70
- group 35
 - infoset 35
- Group By 35

H

- harvest 2
- hash value 30

I

- IBM StoredIQ Platform index 2
- index types 2
 - full-text index 2
 - metadata index 2
- information set 2
- infoset 2, 3, 4, 9, 17, 25, 27, 37
 - intersection 25
 - subtraction 26
 - symmetric difference 26
 - system infoset 2
 - union 25
 - user infoset 2
- infoset ancestry 12, 13
- infoset and data 33
- Infoset Details window 26
- infoset management 9
- interrupt 10
 - infoset creation 10

K

- key concept 2
 - action 2
 - filter 2
 - index 2
 - information set 2
 - infoset 2
 - overlay 2
 - report 2
 - scope operation 2
 - set operation 2

L

- Last Accessed View
 - options 36
- Last Modified View
 - options 36
- legal
 - notices 205
 - trademarks 208
- library 17, 18
- load file 67, 68

M

- macro 54
- metadata 87, 93, 199
 - attribute type 87
 - Box attributes 199
- metadata attributes 87, 90
 - calendar 186
 - Centera server 93
 - Chatter 198
 - contact 186
 - data processing 189
 - Discovery Accelerator 96
 - documentum server 90
 - email 99
 - enterprise vault 196
 - FileNet 97
 - Hitachi HCAP 95
 - IBM Content Manager 203
 - IBM Lotus Notes 194
 - Jive 197
 - SharePoint 148
 - source volume files 69
 - system 87
 - task 191
 - value types 193
- metadata tags 74
- multiple matching values 63
- multiple terms or term phrase
 - syntax 23
- multiple terms or term phrase syntax 22
- multivalued attribute 58
- My Requests 47

N

- no-inheritance conditions 65
- notices
 - legal 205

O

- operators 63
- overlay 3

P

- parent and child conditions 64
- preview 24
 - filter results 24
- preview filter results 24
- primary infoset 28, 30
- Primary Infoset 47
- proximity
 - syntax 23

R

- reference 57
- refine
 - refine an infoset 34
- Refine 33
- relationship between documents 74
- report 4, 41
 - compliance report 41
 - content collector manifest report 41
 - data assessment report 41
 - data topology report 41
 - overlay hit report 41
 - report types 4, 41
- reports 4, 41
 - types of 4
 - types of reports 41
- request management 47
- result directory 69
- run 13, 40
- run file 67

S

- schedule 40
 - action 40
- scope operation 4
 - Collapse 4
 - Expand 4
- scope operations 29
- search 11, 17, 22, 37, 40, 44, 50
 - exception object 50
 - filter 17
 - infoset 11
 - report 44
 - specific action 40
 - specific enhancement 37
- search syntax 22
- secondary infoset 28
- select 36
 - data overlay 36
- Select Data Overlay 33
- Set Operations 25
- single-valued attribute 58
- size view
 - options 35
- sparse attribute 58
- specification of attribute conditions 58
- string attributes 58
- symmetric infoset 28

- syntax 22
 - boolean operator 22
 - multiple terms or term phrase 23
 - precedence 24
 - proximity 23
 - wildcard 23
- syntax primer 22

T

- tag information 74
- target infoset 30
- term hit report 45
- trademarks 208

U

- user reports 44

V

- view 12, 13, 15, 18, 50
 - Action Details 13
 - action log of infoset 13
 - Ancestor Filter Details 13
 - Ancestor Infoset Details 13
 - data objects 15
 - Details tab 12, 13
 - exception object 50
 - infoset ancestry 13
 - infoset details 12
 - source code 18
- Viewing data with different views 35

W

- wildcard
 - syntax 23
- workflow 13

X

- XML root information 74



Printed in USA

SC27-5691-09

