DELL Technologies

Technical White Paper

DataIQ Auto-tag Solutions

Use cases and functional workflows

Abstract

This solutions guide explores three primary use cases for DataIQ's Auto-tag functionality. The use cases selected in this guide highlight new tools for tracking unstructured data, charge-back/show-back insights and finding non-compliant file/folder naming errors.

March 2020

H17912

Revisions

Date	Description
August 2019	Initial release
March 2020	DataIQ updated release
December 2020	DataIQ v2.1 release

Acknowledgments

This paper was produced by the following:

Author: Lynn Ragan

Table of contents

Executive summary

Businesses struggle to make sense out of the ever-growing mountains of amassed unstructured data stored within their multiple platforms. This data often spans across storage platforms deployed in different geographies and/or across multiple access protocols. Real business problems such as charge-back/show-back reporting, cross platform consumption by specific teams, and tracking of data through actual business stages of development encounter serious impediments to gaining meaningful business insights from data stored across traditional NAS and off-premise S3 cloud repositories.

DataIQ presents a flexible tag and analysis tool that yields roll-up summaries based on file and path information which reflects how data may be distributed across enterprise storage platforms. Auto-tagging methods allow content owners to gain insights according how projects, teams and processes of the business. In so doing, business context of the data moves to the fore as the primary aggregation focus and DataIQ Auto-tagging is the vehicle which makes it all possible.

1 Tagging as a business context solution

The business value of utilizing tags comes from the ability to generate reports from a business context frame of reference rather than strictly a platform view. These reports can include actionable information about resource utilization, costs, project content status, and even for tasks such as tracking down inconsistent file/folder names, problem characters, and inconsistent data placement. By using tags, data can be reported in human terms.

Usage of tags goes beyond reporting in the conventional platform utilization sense. A business now has a single pane of glass to view cross platform utilization of data assets as used within the business context. In other words, content owners can now report in terms of their teams, their projects, their role, their sequence within a multistage project.

In practical terms, a customer could report in a single view how a team with three projects underway is consuming storage across multiple on-premise NAS repositories as well as cloud hosted object archive. A bar-chart summarizing the aggregated file content grouped as project 1, project 2, project 3 might reveal that project 3 has twice as much file content as project 2 and has scattered that data across 4 different NFS/CIFS platforms.

Data which moves from one stage of analysis to another might be tracked by applying different tags such as Frontend Design, Frontend Verification, Backend Design, Backend Verification, Ready to Transfer, Post Validation, etc. Essentially, a tag is an arbitrary fully contextual label that describes the files or folders to which it is applied. Because that metadata tag is indexed and stored in the DataIQ database, it is searchable and can be presented in report form.

Ultimately, usage of business context metadata tags provides a powerful tool for

- Observing and comparing related costs of the actual file content on disk
- Gaining insight into how efficiently costly primary storage is being used for current projects and whether specific file content may be ready for archive to more cost-effective storage from within a currently active project or production stage
- Identifying data for specific deep learning or research review rather than relying on cryptic folder naming schemes
- Labeling data to be used in collaborative efforts

By doing so, DataIQ enables tactical, case by case, management decisions of critical file content. Data may be corrected, purged, relabeled or moved from one tiered platform to another on a case by case basis.

Additional data refinement tasks that can be accomplished using Auto-tagging include

- Exclude recycle bin, trash data from reporting
- Search and purge bad character usage from Folder/File names
- Clean up non-conforming folder naming
- Identify potential left-over junk data tagged for deletion

2 What is in a tag?

DataIQ data insight analytics are based on its ability to track data (files and folders) using custom metadata descriptions. These custom descriptor labels are referred to as tags. The process of automatically assigning tags to folders, files, and objects by using filtering criteria is called Auto-tagging. Auto-tag technology enables a business to employ and ensure consistent data tagging strategies rather than relying on ad-hoc human generated tags applied singly and prone to error, inconsistency, and sprawl.

A tag is simply a database entry in the DataIQ index database that indicates when a folder/file has met criteria for being tracked. Functionally, these tags can be tracked and aggregated into report views by using embedded API queries available through the DataIQ interface. These queries can also be called through secure off-host API access.

2.1 Difference from standard metadata

DataIQ Auto-tagging empowers customer workflows which are defined by business context. Instead of reporting on access time, create date, or actual file UID/GUID, all of which are limited to the storage platform the data occupies, the business can report across multiple platforms simultaneously on project status, team leader/content owner consumption, or any other business-related definition or categorization of data.

This information is not stored on disk within the storage platform, local to each file, as is standard filesystem metadata. It is stored in the DataIQ index database. The database is updated each time a new volume scan is issued.

2.2 How DataIQ implements tags

DataIQ keeps all metadata in an index database contained within the DataIQ server. This strategy offers some advantages in terms of overall functionality including

1. Easily recreated from rules in the event the DB/server is lost
2. Out of band data categorization
3. Metadata is not stored with the actual data enabling rapid adjustment of categorization
4. Massive scalability

Both tag and indexing information is stored in a database resident on the DataIQ server. Database resources are consumed as tagging strategies are applied to the data. Scaling the solution will influence the overall resources required for the DataIQ server to function smoothly and yield reasonable response time.

A tag entry in the DataIQ index database consumes roughly 1KB of storage space. Additional tags applied to the same file increase the amount of space consumed within the index database proportionally.

2.3 Types of tags

Auto-tag

Tag entries to the index database can be added in an automated manner. Auto-tag rules are defined as a series of regular expressions (regex) that are interpreted and applied to a nightly scan. Folders or files which match a (single) rule will have an entry made into the database. Tag tests may be run from within the DataIQ management UI to see how rule changes affect existing tagged entries.

Manually inserted CNtags

Content owners who may not have access to the Auto-tag configuration files are able to manually insert tags into the DataIQ index database using tag indicator files. These tag indicator files are called CNtags. A content owner or line of business user usually does have permissions to their own content on their own portion of the shared filesystem. That level of permission is sufficient to create the 0-Byte empty CNtag file which can describe the related folder content with a tag.

If a folder contains a file in the format '<category>.<tag>.cntag' then a tag with that category and tag name will be applied to the parent folder during scans.

If the cntag file is not empty, it is ignored. So, if you want this file to be used by DataIQ, ensure that the file contains no data. A folder can contain multiple cntag files. Each will cause the parent folder to be tagged with the respective tag. Optionally, for less clutter on the file system, cntag files can also be kept in a folder named cntag. In this case, the grandparent folder of the cntag files receives the tags. This option is not available unless tagging is enabled. Please contact DELL EMC for assistance.

Note: Running an auto-tag refresh does not update cntag tags. A scan is required for these tags to be applied.

3 Sample customer workflows and business challenges

The use of automatically generated tag (Auto-tag) reports is helpful for insuring the smooth flow of data from one business work sequence to another, from one team to another, and/or from one cost center to another.

Here are just three hypothetical scenarios which illustrate ways to approach applying this tool to real business problems.

3.1 Case 1: Data cost by business context

Consider an organization that wants to understand how IT resources are being consumed according to project, product development to market stage, as well as business division. Different workgroup teams are assigned to the data as the project data moves through various stages and the organization wants to keep track of resource usage and consumption via consumption-based reporting. An important factor is the business has pre-defined contextual labels established for each project status.

Application of data tags via Auto-tagging would enable the organization to monitor the overall status of the project as data moves into various temporary folder structures. The organization would also be able to correlate data stored on the filesystem with an external business schedule or business status. Use of Auto-tags helps the organization to describe the context of data age and correlate that back to project status and ultimately project cost in terms of

- Invoiceable content
- Project complete
- Work sequence complete - ready for next team/next workflow step
- Schedule awareness

3.2 Case 2: Research and charge-back/show-back

A research environment, whether an organization which hosts various resources from the scientific community, or where it has multiple inhouse research teams, is often faced with show-back and charge-back challenges. In this case, each resources group has its own definitions for grouping its data while consuming portions of shared resources. These definitions are not necessarily directed by the organization but are still connected with various cost centers.

For example, a primary investigator (PI) is being funded to conduct research. The organization needs a mechanism to track the usage of shared storage infrastructure and is required to provide show-back/charge-back reports. At the same time, data sources such as Genome sequencers, camera sensors, and microscopy data, can generate data exponentially faster than can be tracked.

An opportunity exists where data could be described from an organization standpoint in terms of a Primary investigator and relate that information to the Grant ID (QuoteNumber or Billback number).

At the same time, Primary Investigators want the ability to classify their data generating processes and aggregate their results according to their own terminologies and nomenclature rather than being restricted only to ever more obscure folder naming schemes. As a line of business user, Primary Investigators can insert their own tags to describe data in ways such as

Data Gathering Methodology – identify who or what instrument is generating the data / causing the problem

- Test sequences groupings

- Results analysis
- Teams
- Raw data repositories
- Timeline of analysis as opposed to file write times
- Results repository

3.3 Case 3: Non-conforming workflows and character cleanup

An organization may be struggling with the after-effects of merging multiple storage platforms, and the data content of multiple teams, into a consistent naming strategy. Over time, file/folder naming schemes have become very complicated, names are becoming longer and longer. Invariably, accidental insertion of irregular characters (*&%@:$;_<>, etc.) have begun to cause issues with not only file search operations, but also correctly identifying project folders for inclusion in storage consumption reports.

Common problems observed include:

- Folder clutter (inconsistent names, repetitive names, etc.)
- Orphan data
- Unfunded projects (misalignment of data to business objectives)
- Breaks in the workflow
- Data anomalies
- Problem characters used in File/Folder names
- Overly long File/Folder names

The organization wants to use DataIQ Auto-tagging to identify data irregularities which can generally be termed as non-conforming workflows.

4 Auto-tag solutions examples

For each of the cases described above, DataIQ tags may be leveraged to achieve business solutions. The way in which applying tags, and how the tag rules are structured affect the efficiency and accuracy of the generated reports. Because of DataIQ's unique approach, the reports can be aggregated according to the context of the business.

4.1 Case 1: Regular expressions for data costs and usage totals

Regular expressions are the tool used to search through and report on Folder content. By using the actual folder name value in a standard regex variable, the folder name can become a grouping label in the report output. The drawback to this strategy is that all the reporting is at the mercy of the actual folder names, including incorrect or inconsistent spelling. While this approach is often useful, the drawback can be overcome by assigning fixed values for the search results that are consistent with business terminology.

The first challenge in any filesystem structure is how the data is organized, whether on a single storage platform, or across several. In either situation, it is likely that there is data which can reported on from several different directions. In the example below, we have several parallel projects (designated PRJ or prj). Descending from the top-level folder, there are various stages of project development, indications of separate team functions, and archive repository folders.

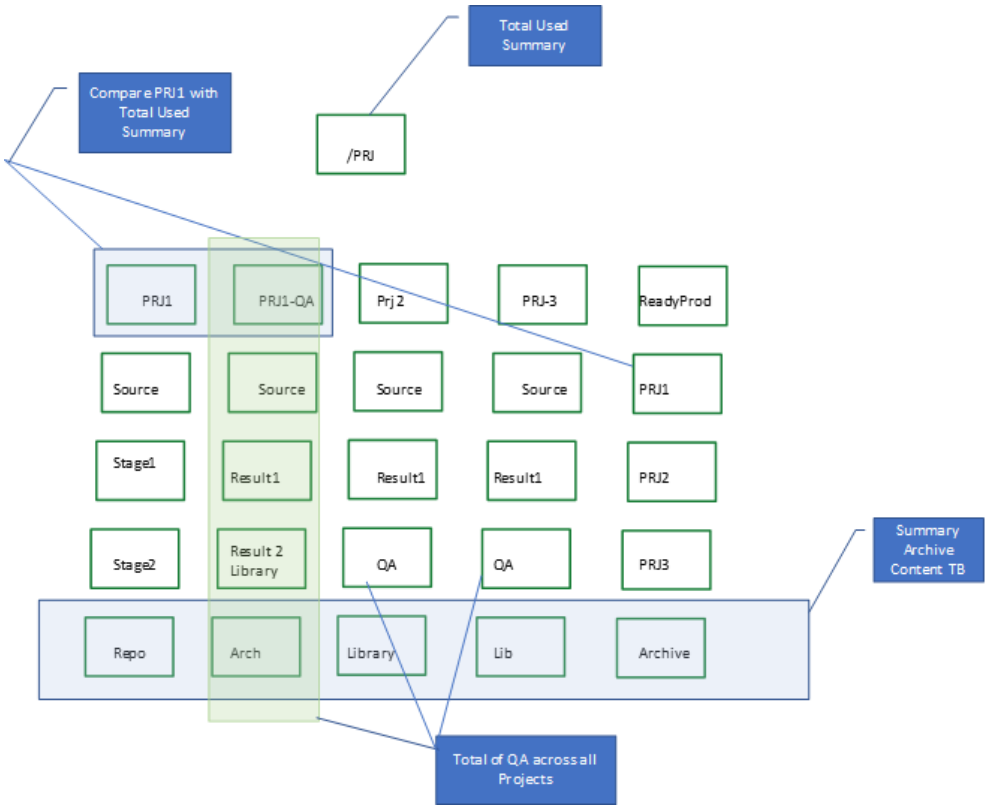

Figure 1 Autotag rules can search for related data spread across filesystems in various ways

Looking specifically at anything PRJ1 is challenging because the project has two top-level folders as well as content over in the Production top-level tree. To grab totals about that data content an Auto-tag rule with a regular expression as in this example from a common regex tester.

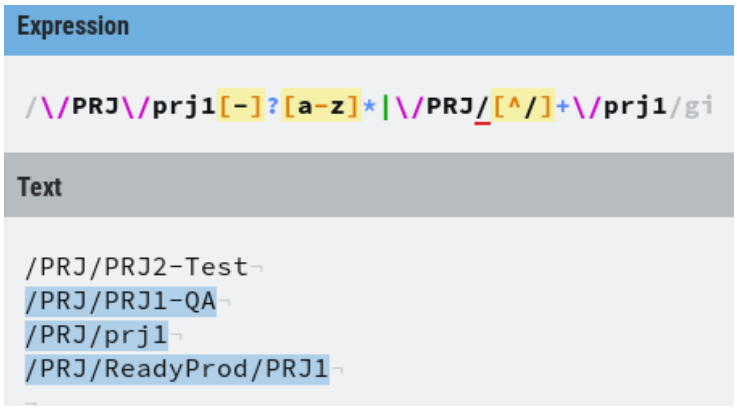

Figure 2 Testing a regular expression designed to search multiple folder trees

```
Set
match (?i)\/PRJ\/prj1[-]?[a-z]*|\/PRJ/[^/]+\/prj1
     max_depth 5
     apply_tag projects/PRJ1    #NOTE: this is a literal tag value
```

This rule looks through the volume defined as 'PRJ' and moves through the virtual path starting at the top-level directory folder of 'PRJ'. The rule descends into all related 'PRJ1' or 'prj1' folders at this level or 1 extra level down.

This could also be represented more simply in the DataIQ autotag configuration window file as

```
Set
match (?i)\/PRJ\/prj1[-]?[a-z]*
     max_depth 5
     apply_tag projects/PRJ1    #NOTE: this is a literal tag value

match (?i)\/PRJ/[^/]+\/prj1
max_depth 5
apply_tag projects/PRJ1
```

The 'set' indicator helps group all the related rules together. This property also provides a unique capability to write rules that span multiple storage platforms which may host related data folders and files scattered across separate mount points.

Expanding on the above scenario by introducing two other Isilon platforms, you might have 'PRJ1' content on mount points 'prodNFS1', 'prodNFS2', and 'prodSMB'. This does not pose a problem for DataIQ by using a group of 'match' statements keyed to each virtual path defined in the DataIQ volume definition.

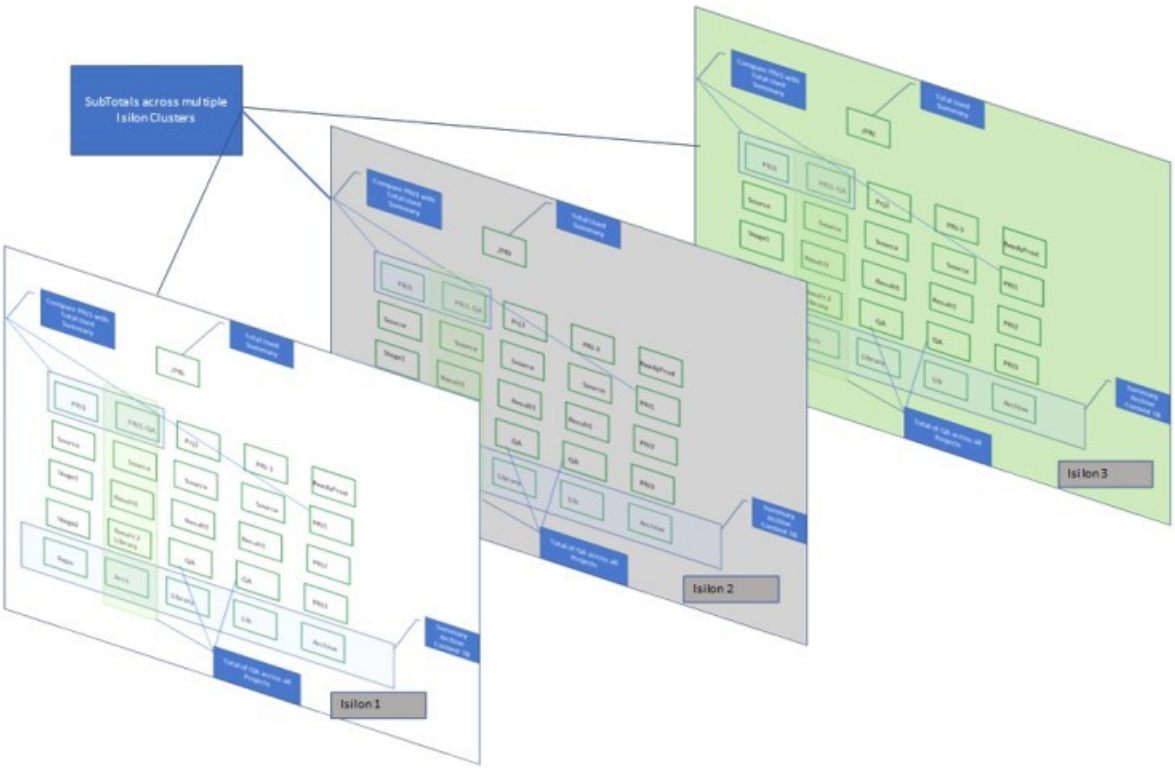

Figure 3 Regex can be used to search for similar data across multiple storage platforms

```
Set
match (?i)\/prodNFS1\/PRJ\/prj1[-]?[a-z]*
max_depth 5
apply_tag projects/PRJ1    #NOTE: this is a literal tag value

match (?i)\/prodNFS2\/PRJ\/prj1[-]?[a-z]*
max_depth 5
apply_tag projects/PRJ1    #NOTE: this is a literal tag value

match (?i)\/prodSMB\/PRJ/[^/]+\/prj1
max_depth 5
apply_tag projects/PRJ1
```

In this way, each of the three separate mount points can be gathered into a single aggregated report which summarizes total file-system space consumption for a project designated ‘PRJ1’.

4.1.1 Age based reporting insights

Additional insight can be gained by utilizing the Aged Based report capability to look for warm, cold, or frozen data within the project work areas. This can help the line of business user/manager graphically see the data life cycle process within their project folders and compare that data usage and data age against storage placement resources as assigned to that project.

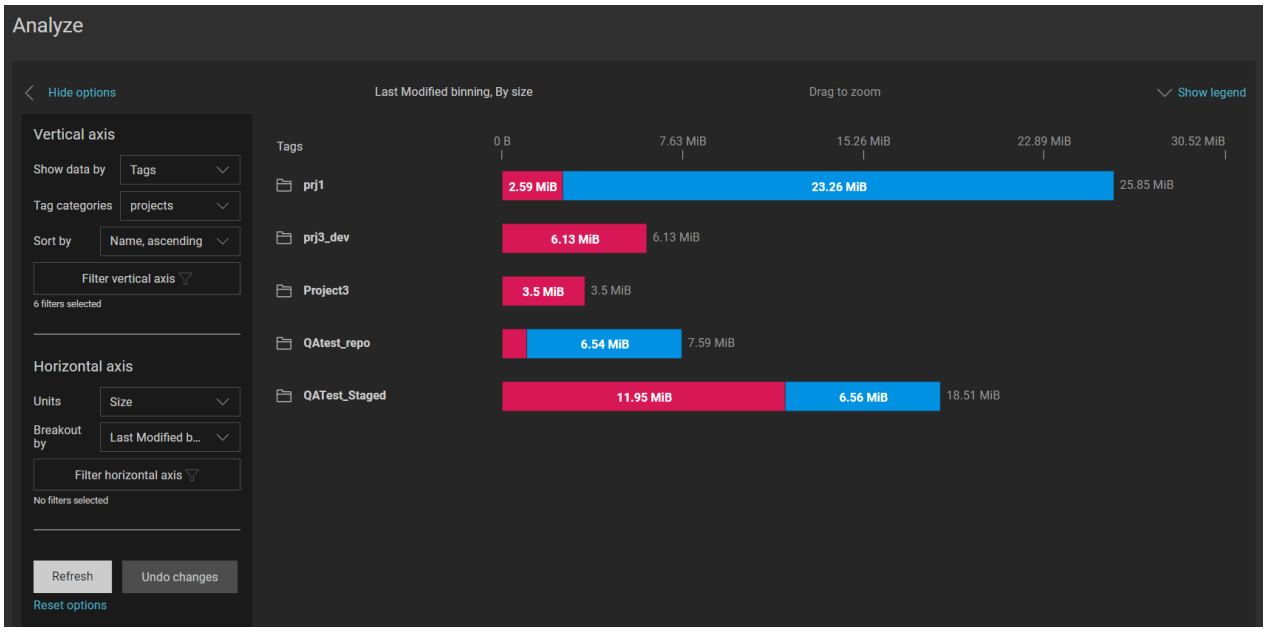

Figure 4 Report view based on project related tags across multiple platforms and then broken out by data age groups is shown below

4.2 Case 2: Research data organized using manually inserted CNtags and implied tags

In research industries, data content owners and line of business users (primary investigators) often generate data with unique filesystem naming structures, independent of other research teams hosted within the same organization. This effectively prevents an organization from both enforcing a standardized naming convention, and accurately reporting on data storage usage according to a specific quote number (Grant ID, etc.), making show-back/charge-back reporting very difficult.

DataIQ has a couple of tools to help with tracking data aside from using Auto-tag rules. Manual CNtags and Implied tags provide a means for line of business users to insert metadata tags (CNtags), using only their assigned filesystem permissions. These tags might indicate a Quote number or chargeback number, manually (or by script) into the workflow so that their data groupings can be tracked and reported for chargeback. Additional tags can also be placed to indicate data source (such as which instrument or tool) for automatically generated data.

Applying a CNtag in DataIQ uses the manual steps of adding a zero-byte (empty) file in the form of <category>.<tag>.cntag such as:

```
GrantIDs.grt49327.cntag
GrantIDs.grt873B2.cntag
```

Where ‘GrantIDS’ is the category of tags, and ‘grt49327’ is some arbitrary label for a specific grant or source of funding. The ‘cntag’ portion is the literal file extension. For example, data residing on an NFS share could easily be tagged by issuing the following command at the unix command prompt:

```
‘touch GrantIDs.grt49327.cntag’
```

These types of files may be placed within a folder (subdirectory) containing file content and perhaps other sub-folders which are related. The Parent folder in which the CNtag file is placed gets tagged according to the category and tag name of that CNtag file. If multiple tags are to be applied to the same folder content, it is possible to group all CNtag files into a sub-folder named: 'cntag'. Creating the 'cntag' subfolder applies all the contained tag labels to the Grandparent folder which contains the actual data content files, rather than to the 'cntag' sub-folder itself.

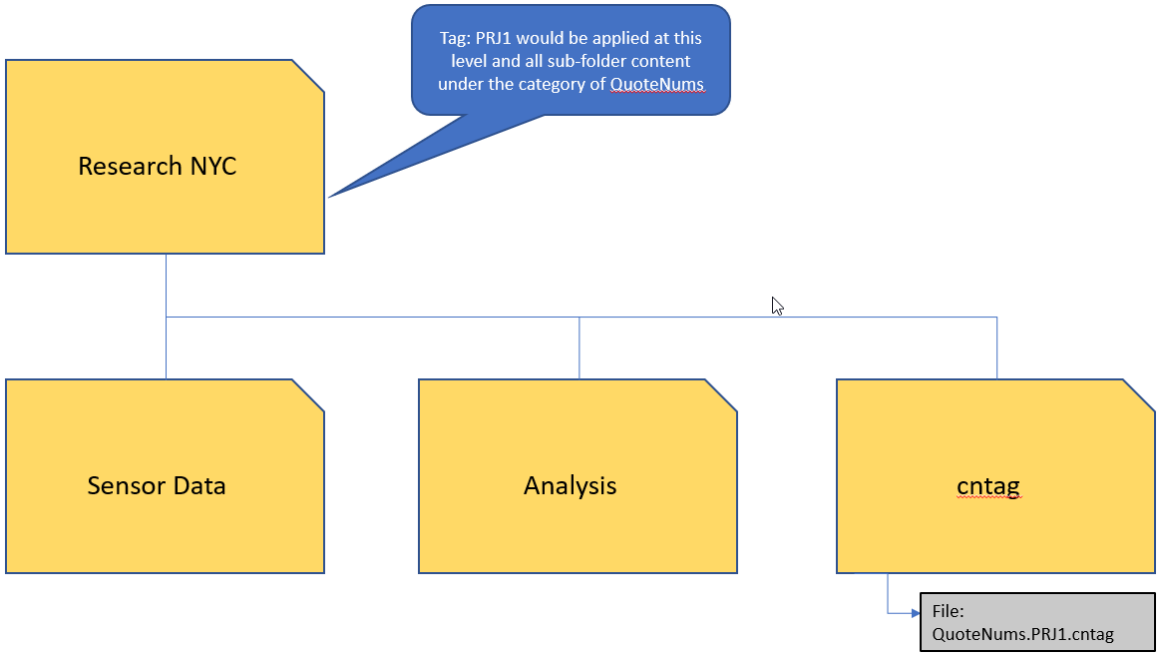

Figure 5 Sample filesystem showing relation of manual cntag subfolder and affected grandparent folder

Normal tag reports can be generated on any one of these tags and are viewable from the Analyze tab in the webUI. A Primary Investigator summary report might look like this chart:

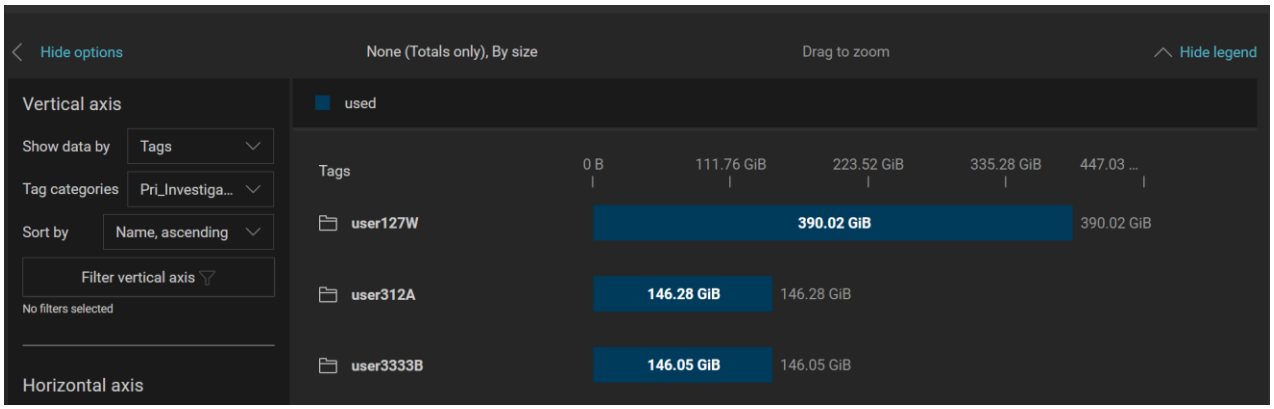

Figure 6 Tag report showing Cost/Week by tagged Primary Investigator

At the same time a chart showing usage by Quote Number can also be displayed which shows usage according to the underlying storage class. Any arbitrary context can be associated with data across several storage platforms or locations in order to provide a cohesive view of current usage totals (rollup totals) that provide insight into actual data flow between projects, between teams, between researchers, or grants.

4.3 Case 3: File/folder name consistency and data cleanup

An extremely powerful function of tag usage is the ability to help administrators clean up data (folder/file names) as it is represented in the filesystem. Using carefully crafted regular expressions (regex) it is possible to search for non-conforming folder names, including situations where irregular characters, such as Ampersand, question marks, semi-colon, were used. Once these aberrations are identified and tagged, a report can be generated which groups them together showing total consumed space as well as identifying the individual folder/files which are non-conforming so that corrective action can be taken.

Below is a sample ruleset which has been added to the autotag configuration section in the webUI Data Management configuration so that non-conforming filenames are identified during the nightly scan. In-depth discussion or explanation of the actual regex statements below is beyond the scope of this document. However, there are multiple reference sources readily available online, as well as regex testers, which can help decipher the rather cryptic nature of a regular expression. The point of showing these examples is to illustrate what may be potentially accomplished in terms of data cleanup.

In this sample offered here, a rule has been constructed which searches file names which are longer than 25 characters. A tag “over_32_chars” is applied and a roll-up summary of all files matching this criteria could be reviewed in the Analyze report window. Additionally, these files are also sorted according to their file extension specific to A/V types of content eg. avi, mp3, mp4, mov, wav, wmv. This allows the administrator to see if there is a pattern to which types of files are most often susceptible to overly long file names, and possibly to identify which workflow or application is the cause of generating these files.

```
# Pass 1: auto tag file names with more than 32 characters
# Sample directory for Pass 1
# /qumulo/GV-XSAN/GARDENVISION_AND_SIGNAGE/_PROJECTS/_LIBERTY/_assets/MEDIA DAY
2013/L13 MEDIA DAY GREEN/ L13 GR MEDIA FORD PICTURE THAT 1 001.mov

match (?i)/NFS_Madrid\/[^/]+\/.*\/([^/]{25,}[^/]+[.](avi|mp3|mp4|mov|wav|wmv))
    applies_to_files
    apply_tag file_ext/media
    apply_tag media_file_ext/$2
```

This allows for some reporting options. A general summary of the ‘media’ tag (which is part of the file_ext category) shows a summary of all the files meeting the above rule: unusually long file names (25 characters or greater) which have a file name extensions included in [avi, mp3, mp4, mov, wav, wmv].

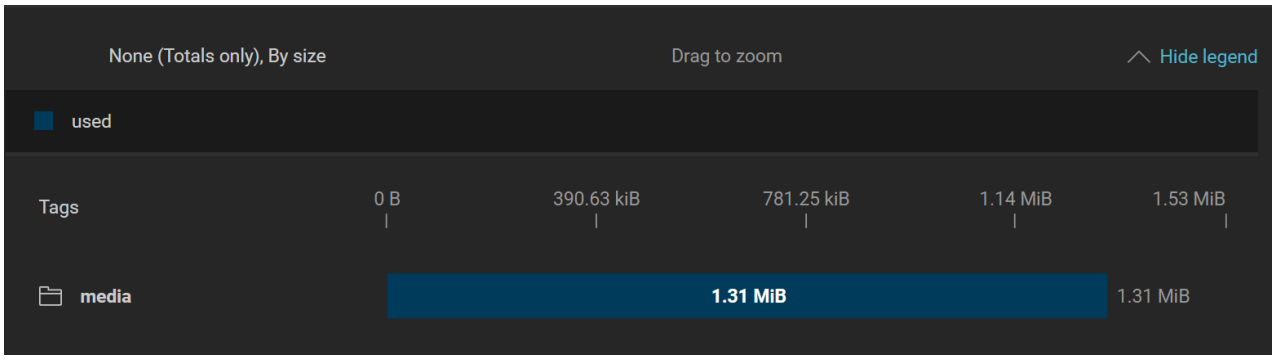

Figure 7 Analyze view of 'media' tag totals for overly long filenames

Adding a horizontal breakout criterion from the second part of the rule provides further detail into which type of media file (with overly long filenames) is consuming the most space. The rule has assigned the value of the actual file extension as the tag name so we can see how this data is broken up.

NOTE: It is highly recommended as a best practice to limit the use of file tracking, such as these examples, to search for anomalies and exception cases rather than tracking large amounts of data on a file-by-file basis. File level tracking, using the 'applies_to_files' rule can have significant and detrimental affects on the index database within DataIQ. Remember, each tag consumes 1k of database space. Use with judicious caution.

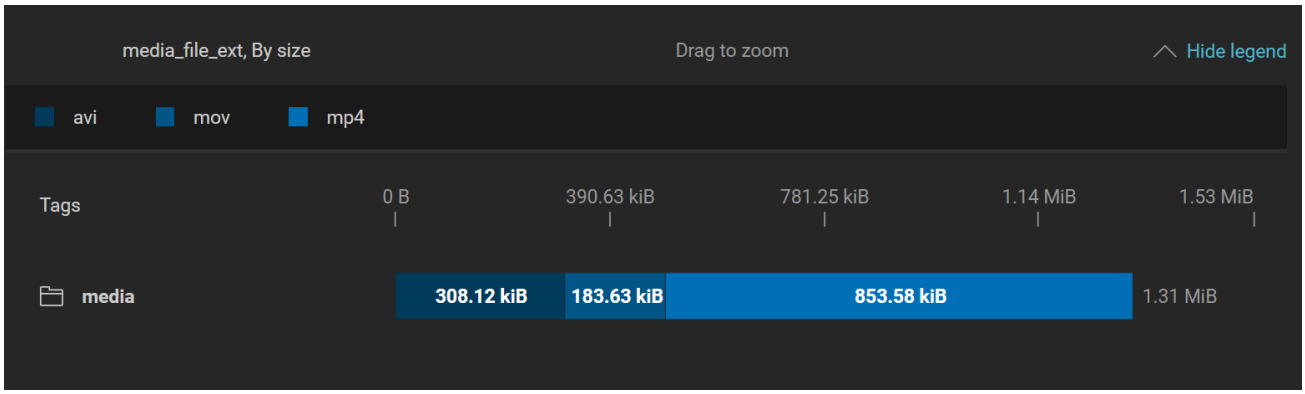

Figure 8 More detailed view of media tag by file type for overly long filenames

This view can be organized in a different way by putting the second tag rule component (apply_tag media_file_ext/$2) as the primary sort value. The drawback is that the report loses the category rollup total by size. However, it may be easier to see which file type is the greatest offender to the file-name-size standard of the organization, and what type of file is being created in such way.

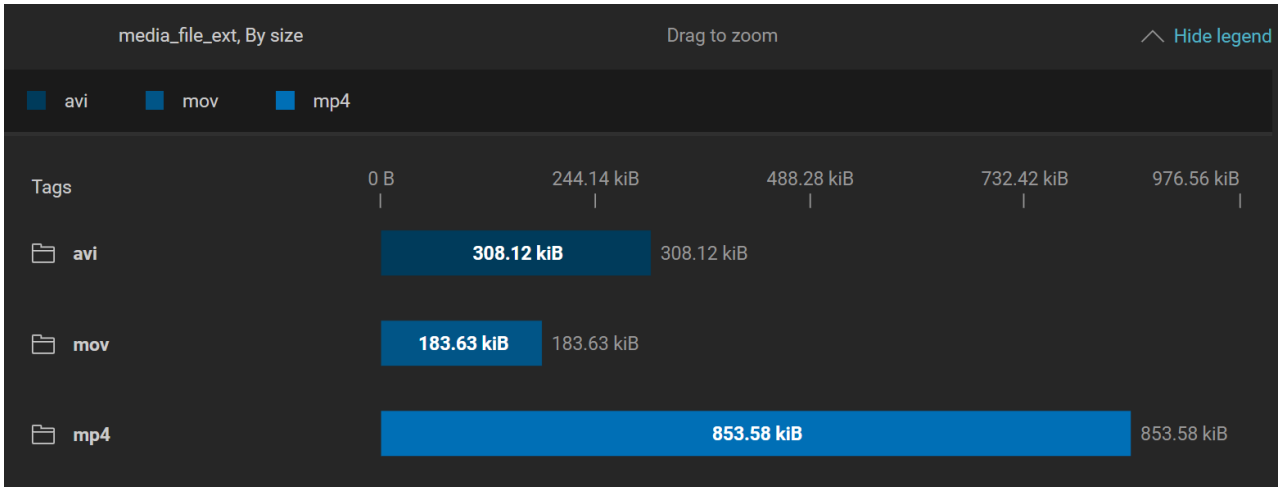

Figure 9 Alternative view of overly long filenames sorted by media type extension

Additionally, data content owners often create temporary "trash" repositories where temp data is stored during the course of project activity, however, that data has no value once the project is completed. Content owners may be hesitant to outright delete data files in-process and use these temp folders instead for later review. As projects move forward, these old-data repositories may be forgotten.

The rule below is an example of how to search for such forgotten old-data repositories for potential filesystem cleanup activities.

```
# Pass 12: auto tag common cleanup terms
# Sample directory for Pass 12
#
/Creative/PhotoAssets/0_Photo_Library/Raws/GPC/Multi_Other/Events/gp_hd4_WinterX
Games_2016/Liveworks/012616_XGames_GoPro/Trash

match
(?i)/[^/]+/(.*(trash)[^/]*|.*(archive)[^/]*|.*(delete)[^/]*|.*(junk)[^/]*|.*(bac
kup)[^/]*|.*(recycle)[^/]*|.*(temp)[^/]*)
   apply_tag cleanup/$2
   apply_tag cleanup/$3
   apply_tag cleanup/$4
   apply_tag cleanup/$5
   apply_tag cleanup/$6
   apply_tag cleanup/$7
   apply_tag cleanup/$8
```

This rule is not as complicated as it looks. Breaking this apart, it starts with a “match” statement. This tells the AutoTag routine to compare paths indicated in the rule with the matching criteria defined in the rule. The criteria conform to Java-based regular expressions (REGEX) as follows:

```
(?i)              -  Globally case insensitive

/                 - The virtual path, not the literal path

[^/]+/            - any number of folder names followed by a slash
                      (descend into folders)

()                - group output and assign to variable 1

(.*(trash)[^/]    - look for files/folders  named ‘trash’ and assign to variable 2
                                  OR
(.*(archive)[^/]  - look for files/folders  named ‘archive’ and assign to variable 3
                                  OR
(.*(delete)[^/]   - look for files/folders  named ‘delete’ and assign to variable 4
                                  OR
(.*(junk)[^/]     - look for files/folders  named ‘junk’ and assign to variable 5
                                  OR
(.*(backup)[^/]   - look for files/folders  named ‘backup’ and assign to variable 6
                                  OR
(.*(recycle)[^/]  - look for files/folders  named ‘recycle’ and assign to variable 7
                                  OR
(.*(temp)[^/]     - look for files/folders  named ‘temp’ and assign to variable 8

Apply_tag  - add variable 2 to the category cleanup and assign tag value 2 which is ‘trash’

Apply_tag – add the variable  3 to the category cleanup and assing tag value 3 which is
‘archive’
…
Etc.
```

It would look like this in the Auto-Tag configuration window under Settings -> Data Management Configuration tab.

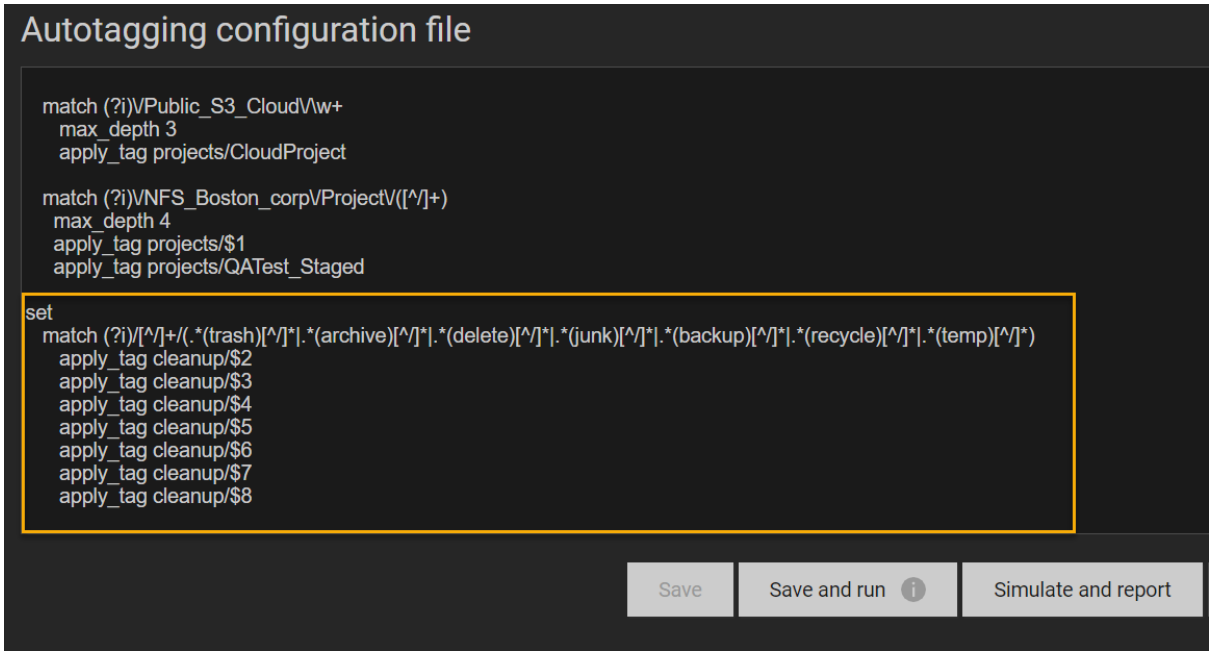

Figure 10 Example of Autotag configuration window with trash-search rule

A rule like this can be useful for identifying typical folders or file locations designated by team members as data no longer needed. Workflows often leave data iterations behind which are part of normal backup, may be dumped into folders indicating that the data is now considered: trash, junk, ready for delete, temp, or even temporary archive such as a project repository that is now no longer useful.

The resulting report might look like this example which found end-user “Junk” and “Temp” folders containing significant amounts of potentially stale data which is no longer needed.

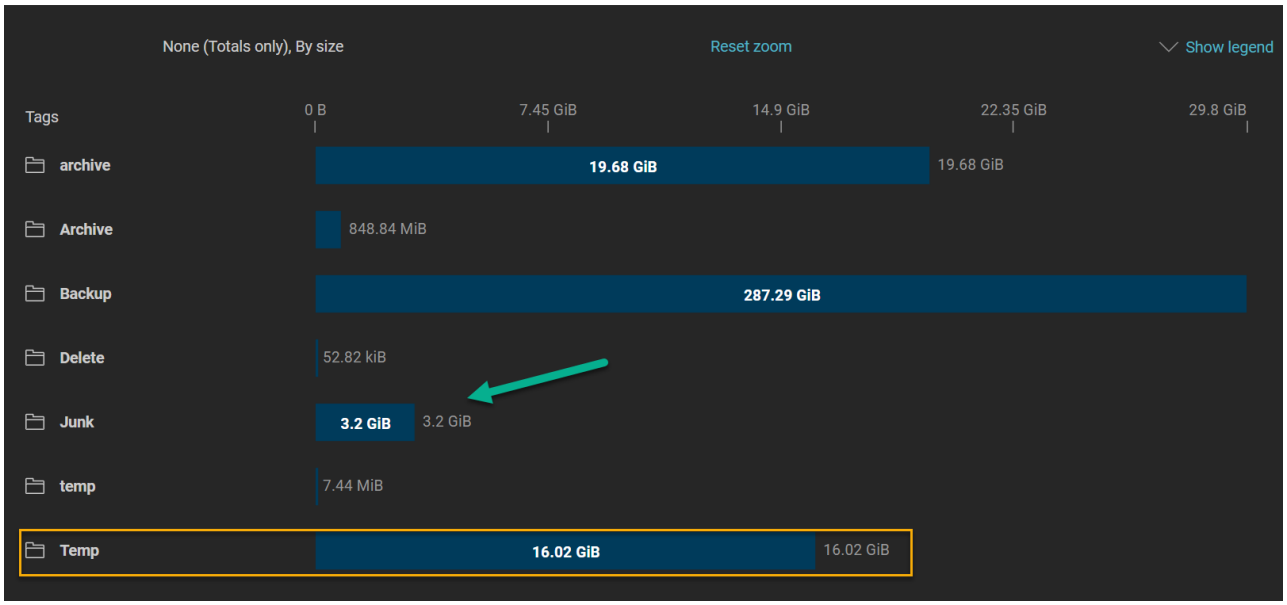

Figure 11 Example tag report summary of file/folders identified as Junk, Temp, Delete, etc.

5 Common reference architecture

A consistent deployment architecture can be applied to these three workflow examples, the same as to many others. In every case, at least one DataIQ server will exist that is licensed for the total amount of data to be scanned and managed across all relevant platforms.

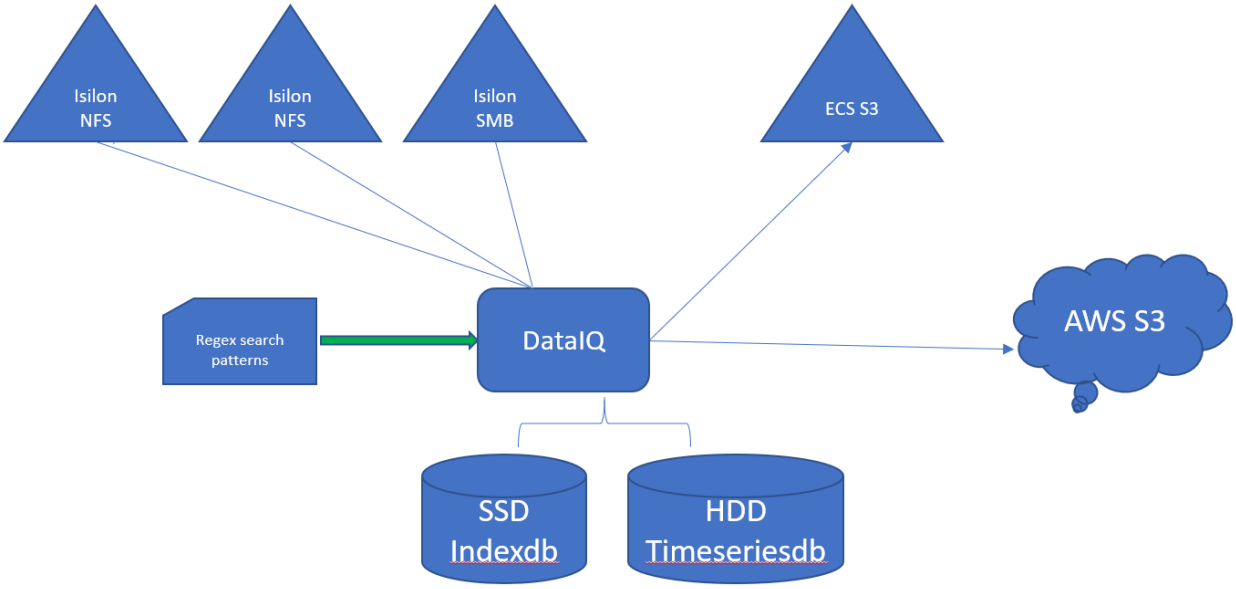

Figure 12 Common reference architecture for all use cases described in this paper

5.1 Overview

A DataIQ solution for all these use cases will have similar elements in terms Server and Clients. Since plugins such as Data Mover are out of scope for this paper, there will be no discussion of additional server-based components or deployment of data mover hosts. Covered here are the infrastructure and host elements.

In general, DataIQ relies on efficient network connectivity to each network share or S3 object platform to conduct initial volume scans. Severe network limitations will have the effect of slowing down the initial volume scan when the volume is created. DataIQ relies on standard NFS/SMB mount points. Reliable DNS with forward and backward resolution zones will be required to maintain stable network connections.

5.2 Server

In each case, a dedicated DataIQ server meeting all the minimum physical requirements is recommended. The server can be either physical or virtual. In either case, the server needs to be provisioned with an SSD tier of disk sufficient for the intended size of the Index database which contains all the file/folder indexing as well as all metadata tags. (Please reference DataIQ Best Practices Guide and DataIQ Admin Guide for further system related detail).

Keep in mind that there is always a system related cost to tagging. Therefore, tagging strategies should be carefully thought out with preference for top-down matching tag rules to minimize the demand on the DataIQ index database. Each tag represents an additional 1KB of date maintained by the database. For this reason,

often the organization will maintain a control on the type, number, and even spelling of user created tags – preferring to apply tags to data using the Auto-tagging function discussed earlier.

5.3 Line of business users

It may be necessary to provide administrative to the DataIQ interface for other users, besides system administrators, for Auto-tagging to be useful and provide benefit for the organization. However, immediate feedback to the line of business user may prove valuable as they would be able to see immediate results as tags are applied and take file management actions accordingly. DataIQ provides for three different delegated roles which may be applied to various groups of DataIQ users. User groups may also be inherited from the local Active Directory authentication service, and delegated roles may be applied to these groups as well.

DataIQ can scale upwards to many hundreds of simultaneous webUI connections provided that they are mostly not running manual re-scans or deep searches of volumes. These actions require heavier usage of server-based resources: cpu, memory, threads, etc.

The line of business interface is HTML5 based and viewable via standard web browser tools. Be sure that no intermediate firewall access rules are restricting traffic between DataIQ client and server. DataIQ is out of band to the actual data, so there is no actual file data transferred between DataIQ server and client. DataIQ does support LDAP user login with some delegated roles options defined locally by group memberships. Please consult the Admin Guide for further administrative details.

6 Configuration tips

Auto-tagging relies on properly formed Regular Expressions to find specific folder/file groups. Regex tells DataIQ what to search for and where to look in terms of

- which mounted volume
- what location in the filesystem or folder tree

Each regex must be paired with a 'Match' command and followed with the 'apply_tag' directive. (Explanation related to how regular expressions are created and formatted is beyond the scope of this paper. Many online resources exist to help with development and testing of regular expressions.)

All DataIQ Auto-tag configuration rules should be configured to test against the Virtual mount point rather than the actual mount point. The Virtual mount point is found in the Volume definition when a network filesystem resources is configured in DataIQ.

Auto-tag rules are applied to the volumes during the nightly scans. They can be tested manually using the 'Save and run' feature or 'Simulate and report' available from the UI.

6.1 Match order of precedence and tag grouping

Once a folder/file has met the qualifications of a rule in the Auto-tag configuration, and has had a tag applied, that folder/file cannot be re-tagged by another rule. This means that if multiple tags are needed to fully categorize a folder/file, they all need to be applied within the same rule. For instance, if the files of a folder need to be viewed three ways (business context, project class, team/cost center) you might create rule structure such as:

```
Match [regex rule finding all folders with names containing 'PRJ' or 'prj']
   max_depth 3
   apply_tag  {category}/DesignStage
   apply_tag  {category}/SoftwareDiv
   apply_tag  {category}/BlueTeam
```

The critical terms used in the above single rule group are:

- 'match' – the actual command to pass folder/files to the specified regex rule to see if there is a match
- 'max_depth' – maximum number of folders in the path that will be passed to the regular expression test
- 'apply_tag' – defines both the category and the actual tag to be applied to the matching folder/file
- 'category' – can be any arbitrary label of grouping such as: 'apply_tag Stage/DesignStage (or DevStage, QETestStage, ProdStage, SalesStage, or any other business label)
- 'tag' – shown above in the form of literals "DesignStage', 'SoftwareDiv', and 'BlueTeam'. These can be any arbitrary literal value which holds contextual meaning for the business. A variable can also be used in which character strings found by the regex rule can be passed and converted to its own tag value. Both literal and variable values can be combined into one report view.

From the autotag configuration notes within the UI:

```
# Notes:
#   - In apply_tag parameters, $1, $2, $3 etc are replaced by matched RE
substrings
```

```
#   - In apply_tag parameters, ${1}, ${2}, ${3} etc are replaced by
matched RE substrings
#   - In apply_tag parameters, $user and $group are replaced by the user
and group name of the file/folder respectively (Linux only)
#   - Multiple match blocks may be present within each set
#   - For backward compatibility, the set keyword may be omitted, in which
case all matches are in the same set
#   - Tags and categories specified by apply_tag will be created if not
already present
```

In this way, multiple tags can be applied to the same data. The result would be a possible report that shows total file storage associated with DesignStage across multiple projects. Or, a report could be generated which show all the folders currently under the umbrella of the Software division regardless of project. Or a report could show a grand total of all storage used by the BlueTeam.

These three views can be achieved by taking advantage of the order of precedence in which all folder/files which do not meet the rule above could be matched by another rule which may also use some of the same tag names if appropriate.

A Appendix

A.1 Blocking rules: Useful Auto-tag filesystem exclusions

Auto-tags can also be used to exclude certain types of data from summary reports. These kinds of regex matching rules are called blocking rules. Blocking rules match data files/folder but don't apply a tag. The result is that matched data will not be tagged by subsequent rules and consequently not be included in DataIQ Auto-Tag reports. This is a natural extension of the order of precedent which DataIQ employs.

Data such as Recycle Bin, Trash Can, Snapshots, '.bak' files, etc. can be excluded from usage summaries to focus on actual business usage rather than the larger storage administration concerns which encompasses all storage consumption.

Here are some regular expression examples for common types of data exclusion rules.

```
# Pass 2: Blocking Rule > do not autotag $RECYCLE.BIN
# Note: regex uses hexidecimal representation of ascii numbers
# Sample directory for Pass 2
# /4kpod07/$RECYCLE.BIN

match (?i)/[^/]+/([$]RECYCLE.BIN)
   apply_tag cleanup/RECYCLE



# Pass 3: Blocking Rule > do not autotag directories or files within the
$RECYCLE.BIN
# Note: regex uses hexidecimal representation of ascii numbers
# Sample directory for Pass 3
# /4kpod08/$RECYCLE.BIN/S-1-5-21-1881908555-2847256797-1495832542-
14694/$ROEDHHQ

match (?i)/[^/]+/[$]RECYCLE.BIN.*/([$][^/]*)
   apply_tag cleanup/RECYCLE
```

A.2 Matching rule detail

```
# Overview
#   - Autotagging allows matching rules to be used to apply tags to folders and
other tracked items
#   - Autotagging occurs whenever DataIQ! scans a file system
#   - Autotagging may also be performed manually without scanning
#   - Autotagging may be tested from the GUI without scanning or actually tagging
#   - Tags will automatically be removed if the rule no longer matches and the tag
has not been altered
#   - Autotagging relies on Java regular expressions (aka RE or regex). Google
"java regex" for syntax
#   - The following site may be of value in creating REs:
http://xenon.stanford.edu/~xusch/regexp/
#
```

```
# Format of an autotagging configuration
#
# set
#   match <re>                                  - this Java regular expression
(RE) must match the ENTIRE path (^ and $ are assumed)
#     max_depth <n>                             - maximum number of folders in
path that will be passed to RE "/vol1" == 1
#     [required_tag <cat>/<tag>]                - RE will only be used if this tag
is applied directly at volume level
#     apply_tag <cat>/<tag>                     - one or more tags to apply if
match was successful
#     apply_tag <cat>/<tag>
#     apply_tag <cat>/<tag>
#     apply_tag <cat>/<tag>
#   match ...
#
# set
#   match ...
#
# Matching rules for each item (folder or other tracked items):
#   - Sets are checked in order listed
#   - Rules within a set are checked in order listed
#   - First rule within a set to match stops further searches for that item within
that set
#   - When a set has been processed, the next set is checked
#
# Notes:
#   - In apply_tag parameters, $1, $2, $3 etc are replaced by matched RE substrings
#   - In apply_tag parameters, ${1}, ${2}, ${3} etc are replaced by matched RE
substrings
#   - In apply_tag parameters, $user and $group are replaced by the user and group
name of the file/folder respectively (Linux only)
#   - Multiple match blocks may be present within each set
#   - For backward compatibility, the set keyword may be omitted, in which case all
matches are in the same set
#   - Tags and categories specified by apply_tag will be created if not already
present
#

# example:
# designed to tag /<volume name>/<Show name> with a tag named <Show name> in
category <Show>
# /san01/PeopleAreLovely gets tag Show/PeopleAreLovely
# /san01/PeopleAreLovely/dailies does not get an additional tag because it has more
characters after the RE
#set
#  match /[^/]+/([^/]+)
#    max_depth 2
#    #required_tag extras/mark
#    apply_tag Show/$1

set
   match (?i)\/testIsilon\/projects\/(prj\d)_[a-z]+
     max_depth 4
     apply_tag projects/$1
```

```
match (?i)\/tmeIsilon\/projects\/(prj\d)_[a-z]+
  max_depth 4
  apply_tag projects/$1

match (?i)\/testCIFS\/projects\/[^/]+
  max_depth 4
  apply_tag projects/prj3
  apply_tag projects/DevTeam
```

B Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell EMC storage platforms.