



QLIKVIEW'S PRAGMATIC APPROACH TO METADATA

A QlikView Technology White Paper

Published: March, 2011

qlikview.com

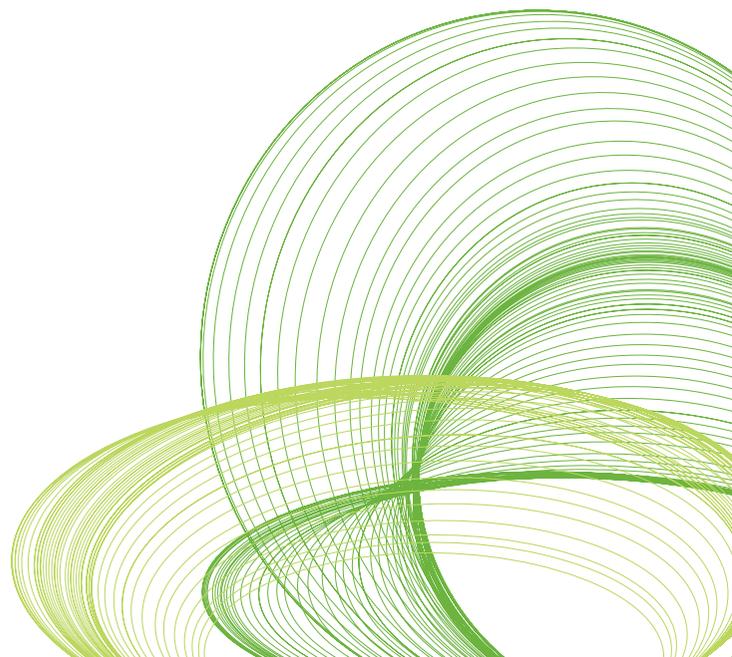


Table of Contents

Introducing QlikView's Metadata Approach	3
Metadata Is Important for Large-Scale BI Deployments	3
Traditional BI Metadata Solutions Collapse Under Their Own Weight	4
With QlikView, Metadata Is No Longer a Dirty Word	5
QlikView Metadata: Descriptive, Administrative, and Structural	6
Collecting, Organizing, and Presenting QlikView Metadata	8
QlikView Metadata Delivers Oversight and Control	10
Admins and Deployment Owners Support Users and Refine Processes	10
Developers and Data Architects Conduct Better Planning	10
Designers and Content Developers Streamline Their Work Efforts	11
Business Users Gain Insight Into Usage and System Health	11
Recommendations for Working with QlikView Metadata	11
Appendix	12
Related White Papers	13
Related Blog Articles	13

QLIKVIEW'S METADATA APPROACH

With QlikView:

- **Metadata management is optional and pervasive.** QlikView customers use metadata only when and where it adds value. QlikView creates metadata automatically. Whether or not it is used is up to the designers and developers.
- **Our focus is on QlikView itself.** QlikView's metadata focus is on helping stakeholders understand and manage the QlikView environment. Developers and designers get a clear picture of how well their QlikView applications were built and gain insights that help them maintain their applications.
- **Developers can introduce metadata usage over time.** With QlikView, developers do not have to create a metadata layer ahead of time. They can define and collect metadata after they have created, tested, and even deployed applications. Project teams can instead focus on getting the right business information and analytical tools in the hands of the right people at the right time.

Introducing QlikView's Metadata Approach

Metadata is an important element of business intelligence (BI) software deployments. Prospective customers frequently ask us how the QlikView Business Discovery platform handles metadata and how much work is involved. This *QlikView Technology White Paper* describes how QlikView delivers metadata in a way that truly supports management oversight and control — yet is simple and straightforward.

We created this paper for IT pros and tech-savvy business people who are evaluating QlikView for use in mid-sized and large organizations. Our aim is to show that QlikView's metadata approach can quickly and easily lead to improvements in QlikView process governance, troubleshooting and support, application development, data change analysis, and capacity planning.

Metadata Is Important for Large-Scale BI Deployments

For small, straightforward BI deployments, metadata management may not be important. In these environments, decision makers can quite quickly figure out answers to questions like, "Where did this data come from?" or "What does 'sales' mean in the context of that particular table?"

But for larger deployments, metadata management is critical because:

- **Users require confidence in the data.** Users need to be able to determine where the data came from (data lineage), when the data was last updated, and what the data really means (for instance, "sales" can have a number of meanings, such as product sales, total sales, net sales, or gross sales).
- **IT or BI administrators need to understand data usage.** Those responsible for managing the source data need to determine who is using the data and whether it is being used in the right way (e.g., users are respecting security and using the correct version of the data). They need to be able to conduct impact analyses on who and what will be affected when source data changes.
- **Developers want to streamline their work efforts.** BI application developers need to know what re-usable components are at their disposal. This promotes re-use and consistency, which in turn results in lower total cost of ownership.

WHAT IS METADATA, ANYWAY?

Metadata is data about data. When used effectively, metadata increases the value of data by surrounding it with context. It provides information about the content of fields in a database, table, or BI object.

Traditional BI Metadata Solutions Collapse Under Their Own Weight

Most traditional BI vendors treat metadata as a separate data and software layer for their customers to populate, query, integrate, manage, and staff. While some traditional BI offerings contain comprehensive metadata modules, these modules frequently go unused. They are work-intensive and expensive to populate and maintain.

With traditional BI software, metadata poses three major problems:

- **A huge upfront time investment is required.** As a separate add-on to traditional BI platforms, the metadata module becomes yet another headache for developers and designers. Along with learning how to use it, they also have to incorporate its use into everything they do: data architecture; extract, transform, load (ETL); queries; and staging and usage of fields.
- **Ongoing change to the metadata layer is a Herculean task.** By the time the implementation team gets around to finalizing the metadata layer, the business has changed. It becomes nearly impossible to keep the metadata layer up to date and continually meet the needs of the business.
- **The metadata solution often goes un-used.** So many steps are involved that developers and designers end up skipping steps. The result? An empty layer of software and data in the BI solution stack. The organization isn't able to reap full value from its BI investments because of the vast holes in the metadata layer.

Because this situation is so prevalent, BI stakeholders tend to view metadata as a painful hurdle the IT organization must clear before business analysts can do any serious work on analysis definition and begin meeting end user requirements. This perception persists because traditional BI solutions are based on queries and cubes. Metadata becomes the language through which these tools communicate with the data stores that feed them.

The good news is: it doesn't have to be this way.

With QlikView, Metadata Is No Longer a Dirty Word

In much the same way that QlikView has transformed BI through our focus on Business Discovery, QlikView is transforming metadata management, as well. We don't force our customers to get wrapped up in the academic process of producing and managing metadata just for metadata's sake. Our philosophy? Keep it simple. Focus on the metadata that really matters — that produces something meaningful.

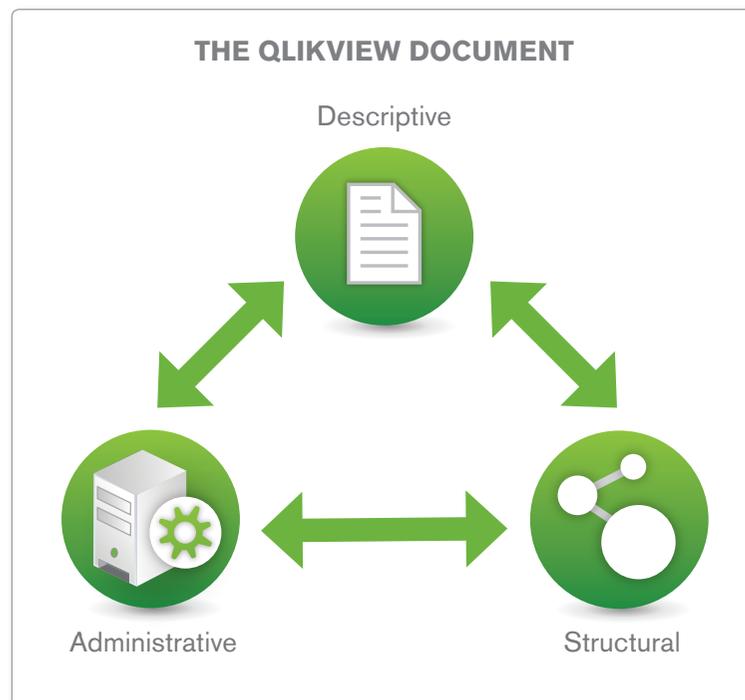
With QlikView:

- **Metadata management is optional and pervasive.** QlikView customers use metadata only when and where appropriate — where it adds value. For example, QlikView enables designers and content developers to build a prototype or sandbox application without performing any extra metadata tasks. They can then deploy that application to a production environment and simply scan the metadata from the application using an automated scanner. This approach is simple, effective, and used only when needed.
- **The focus is on QlikView itself.** Our metadata focus is on understanding and managing the QlikView environment. Developers and designers get a clear picture of how well their QlikView applications were built, and gain insights that help them maintain their applications. QlikView enables organizations to capture metadata that can deliver important insights into key functions such as QlikView application load monitoring, redundant data identification, data refresh rates, field definitions, and impact analysis for upstream (source) database changes.
- **Developers can introduce metadata usage over time.** With traditional BI solutions, developers have to manufacture metadata as part of the development process. If they wait to create metadata, it's too late. In contrast, QlikView developers do not have to create a metadata layer ahead of time. They can define and collect metadata after they have created, tested, and even deployed applications. This gives project teams the opportunity to focus on the important part of BI: getting the right business information and analytical tools in the hands of the right people at the right time — quickly and without massive amounts of overhead.

QLIKVIEW METADATA: DESCRIPTIVE, ADMINISTRATIVE, AND STRUCTURAL

QlikView handles three forms of metadata: descriptive, administrative, and structural (see Figure 1 and Figure 2). The glue that binds these three types of metadata together is the QlikView document. As administrators, developers, or business users look at the descriptive data for a QlikView document, they can also see the administrative and structural metadata.

Figure 1: QlikView captures three types of metadata for increased oversight and control



Source: QlikTech

Figure 2: QlikView handles descriptive, administrative, and structural metadata

Type of Metadata	Description	How It's Useful	How QlikView Handles It
Descriptive metadata	<ul style="list-style-type: none"> • Describes an application's origin and status • Includes information about application author, purpose, topics, search terms, history, and intended usage 	<p>Provides rich context about the makeup of an application. This context is useful to developers and designers when comparing applications, shopping for examples or code to reuse, and evaluating trends and coding practices.</p>	<p>Metadata can be added at any time to the script of individual QlikView applications, or centralized in tables and loaded in as the QlikView application loads.</p>
Administrative metadata	<ul style="list-style-type: none"> • Includes data about servers and system usage, and the people involved • Provides information on the “who, what, when, and where” of an application—who can and has accessed the application, and how and when it was used 	<p>Provides centralized or application-specific views of application refreshes (reloads), user access, usage, performance, and scheduling</p>	<p>QlikView stores this metadata automatically in the QlikView Server log files and in QlikView Publisher repository as extensible markup language (XML) files. It is collected and presented graphically in the QlikView Monitor, which gives a 360° view of the QlikView deployment.</p>
Structural metadata	<ul style="list-style-type: none"> • Provides information about the moving parts that comprise the QlikView system • Describes elements of an application such as its data sources and repositories, tables, and columns • Conveys data lineage and includes information about which files are in use 	<p>Helps developers and designers make decisions about:</p> <ul style="list-style-type: none"> • Impact of upstream database changes • Reuse of data elements within QlikView • Troubleshooting or trace-back needed to support applications that map source database tables to QlikView tables 	<p>A QlikView scanner process collects this information from all applications and centralizes the data lineage information for use in QlikView Monitor or within QlikView applications.</p>

DEFINITIONS

- QlikView document.**
 A QlikView file that contains all the elements needed to perform analysis. A QlikView application or dashboard comprises one or more QlikView documents.
- QlikView's metadata model (or "metamodel").**
 A centralized, automated collection, organization, and presentation of metadata for monitoring and distributed use within dashboards. It is a collection of tables extracted from and managed by QlikView.

COLLECTING, ORGANIZING, AND PRESENTING QLIKVIEW METADATA

QlikView's metadata approach centers on our metadata model (or "metamodel") (see Figure 3). With our approach, no additional tools are needed. Just QlikView. QlikView provides two main ways to access and work with metadata:

- QlikView Monitor for designers, developers, and administrators.**
 The QlikView Monitor application is a free example of how to use the QlikView metamodel. It is a QlikView document that contains tables of data scanned from QlikView. The three main focus areas of charts and graphs in QlikView Monitor are QlikView documents, QlikView data files, and operation of QlikView Server and QlikView Publisher. Figure 4 shows a QlikView monitoring application that utilizes the QlikView metamodel.
- Dashboard templates for business users.** QlikView makes metadata available directly in the context of QlikView dashboards. Dashboard templates load data from the metamodel and present it in the context of a QlikView dashboard, just for that dashboard. This enables stakeholders to quickly and easily access information about their QlikView document (e.g., usage, data quality, etc.). Figure 5 shows a QlikView dashboard that displays all three forms of metadata.

Figure 3: An architectural view of the QlikView metamodel

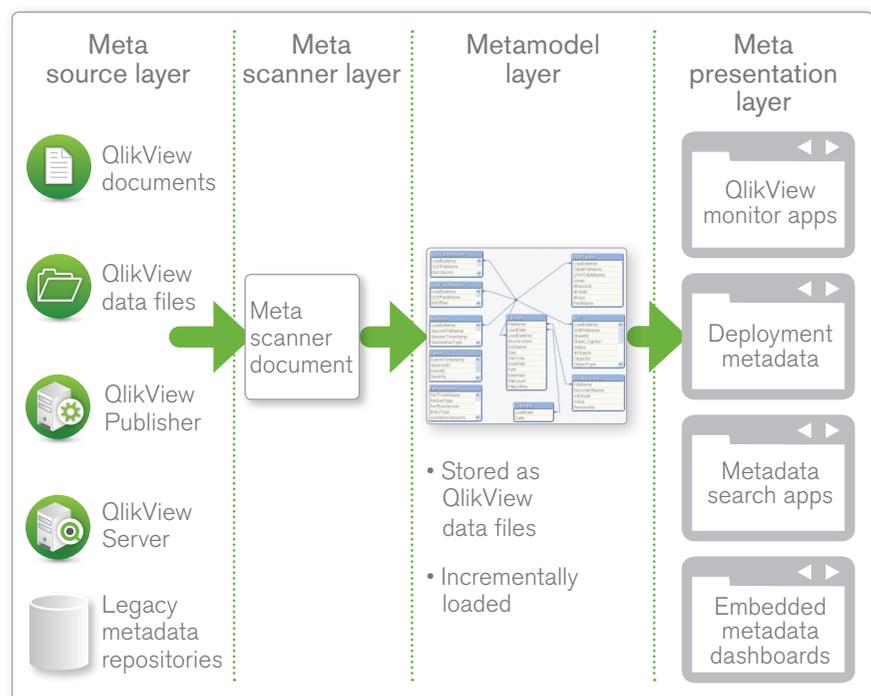


Figure 4: Example of a QlikView monitoring app using the metamodel

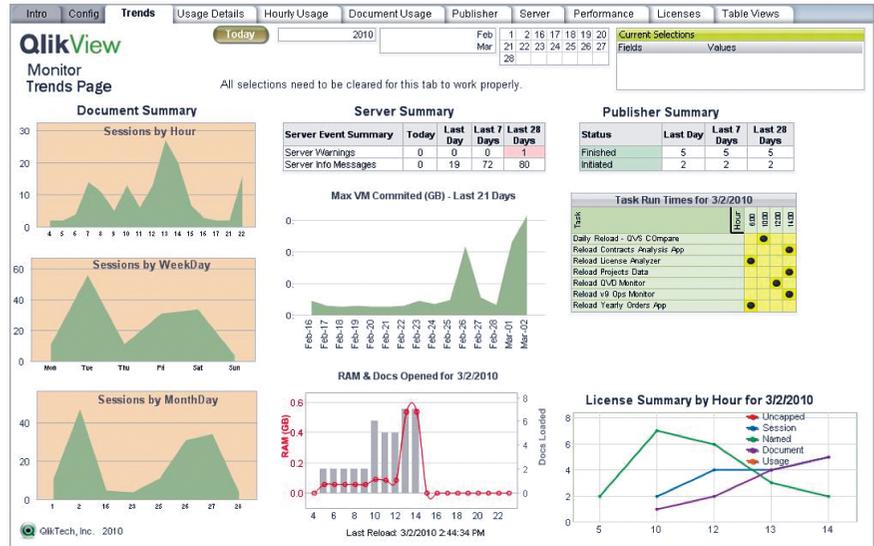
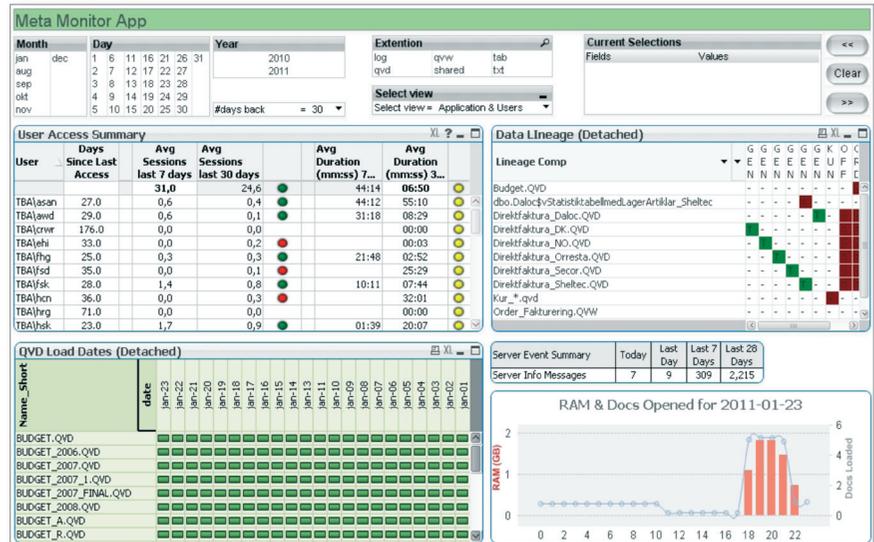


Figure 5: QlikView dashboard showing all three forms of metadata



OlikView Metadata Delivers Oversight and Control

The QlikView Business Discovery platform is known for its high adoption rate, rapid growth, and quick time to value. With an associative experience and in-memory architecture, QlikView enables rapid BI application development. QlikView's pragmatic approach to metadata balances this speed of deployment with oversight and control.

ADMINS AND DEPLOYMENT OWNERS SUPPORT USERS AND REFINE PROCESSES

With QlikView's metadata approach, administrators and deployment owners can:

- Put process governance in place for QlikView software code promotion, software license management, capacity planning, and general internal IT audits
- Determine the accuracy of an overnight QlikView data load
- Check the status of scheduled QlikView Publisher tasks and the accuracy of the steps in a batch window for data delivery.

This complete picture of the data delivery chain allows administrators and deployment owners to improve troubleshooting and support by quickly getting to the root cause of issues — without having to search through individual documents and read through scripts.

DEVELOPERS AND DATA ARCHITECTS CONDUCT BETTER PLANNING

The QlikView metadata solution enables developers and data architects to:

- Tweak the environment to accommodate changes made to upstream data. For example, a table structure may change in the data warehouse, resulting in changes that must be made to QlikView applications that depend on that underlying data.
- Improve scheduling for application updates and spot unexpected gaps in application refreshes.
- Fine-tune the content of each application to improve performance, ensure consistency, and limit negative effects on end users.
- Conduct capacity planning and growth estimates in preparation for scaling up hardware and software before limits are ever reached.

QlikView collects and accumulates structural metadata over time. The metadata shows changes in table and column sizes and row counts and allows for baselines, trending and projections needed for data capacity planning. In addition, QlikView metadata captures usage rates, license allocations, server capacities, and performance statistics, in support of server capacity planning and projections.

DESIGNERS AND CONTENT DEVELOPERS STREAMLINE THEIR WORK EFFORTS

Using descriptive and structural metadata, designers and content developers can locate duplicate data and objects in the QlikView environment. They can rationalize and simplify data usage and optimize reuse of data fields across QlikView applications. This results in fewer lines of code, cleaner deployments, and simplification of moving parts. The elimination of redundant work efforts reduces developer and end user frustration and shortens development time frames.

BUSINESS USERS GAIN INSIGHT INTO USAGE AND SYSTEM HEALTH

With QlikView metadata, business users gain insight into the usage of their dashboards and currency of their data. They can determine how often their QlikView document is used, and by how many people. Should they see something that is out of place, they can use metadata to initiate troubleshooting. For example, by looking at the dashboard metadata, a user who finds an incorrect chart can determine whether the problem lies in the data or the presentation layer (such as a chart expression).

Recommendations for Working with QlikView Metadata

Based on our work with QlikView customers, we offer a few recommendations related to metadata:

- **Take a pragmatic approach to metadata.** Data architects and business analysts familiar with traditional BI solutions often expect BI tools to force them through a complex metadata creation process (development, administration, design, etc.). We say no! Complex metadata approaches slow down BI application development and drag out the time to value. QlikView does not force end users (or designers, developers, or administrators) to hand-enter any metadata. This can be very liberating.

The time to value is quicker with QlikView than with traditional BI platforms in part because QlikView doesn't require a big metadata effort up front. QlikView offers a progression from no-touch metadata management (point to a source and the metadata scans in) to more sophisticated metadata management that requires scripting.
- **Turn on basic metadata collection.** Turning on basic metadata collection for QlikView Server/Publisher and document monitoring delivers immediate insights into data reloads, batch windows, and skipped steps or gaps in the batch window. It also provides information about server capacity, usage of QlikView documents and QlikView data files, sizing (growth of QlikView files over time), and data elements used in QlikView documents. On top of that, basic metadata collection also captures information about utilization of QlikView licenses and basic data lineage.

If desired, organizations can then augment this basic metadata with more advanced metadata collection, which requires some developer involvement. Advanced metadata collection includes document attributes, chart expression logic, and in-depth data lineage at a more granular level than the basic model provides.

- **Utilize both point-in-time and over-the-course-of-time metadata.**

Some metadata is point-in-time, focused on today. For example, point-in-time metadata can be useful for understanding how many people are using QlikView and how often. Stakeholders can also use point-in-time metadata to see the batch window and any gaps in the reload process.

Other metadata provides insight into changes that occur over time, such as changes in the number of QlikView applications or data files. This can be useful for resource planning purposes. We recommend that organizations give themselves at least three or four weeks to collect this latter form of metadata, since the value of the information increases over time.

Appendix

RELATED QLIKVIEW WHITE PAPERS

Business Discovery: The Next Generation of BI, January 2011

<http://www.qlikview.com/us/explore/resources/whitepapers/business-discovery-the-next-generation-of-bi>

QlikView Architectural Overview, October 2010

<http://www.qlikview.com/us/explore/resources/whitepapers/qlikview-architectural-overview>

The Associative Experience: QlikView's Overwhelming Advantage, October 2010

<http://www.qlikview.com/us/explore/resources/whitepapers/the-associative-experience>

How We Supported Thousands of Users Daily on a Global Scale, October 2010

<http://www.qlikview.com/us/explore/resources/whitepapers/how-we-supported-thousands-of-users-daily-on-a-global-scale>

RELATED BLOG ARTICLES

The Five Things that Make QlikView Unique, February 2, 2011

<http://community.qlikview.com/blogs/theqlikviewblog/archive/2011/02/02/qlikview-s-uniqueness-better-for-end-users.aspx>

Putting the "Business" in "Business Intelligence," December 22, 2010

<http://community.qlikview.com/blogs/theqlikviewblog/archive/2010/12/22/putting-the-quot-business-quot-in-quot-business-intelligence-quot.aspx>

Every QlikView Developer Spans BI Domains, December 3, 2010

<http://community.qlikview.com/blogs/theqlikviewblog/archive/2010/12/03/every-qlikview-developer-is-a-quot-spanner-quot.aspx>

It's the Users, Stupid, November 19, 2010

<http://community.qlikview.com/blogs/theqlikviewblog/archive/2010/11/19/it-s-the-users-stupid.aspx>

MOLAP, ROLAP . . . SCHMOLAP, October 4, 2010

<http://community.qlikview.com/blogs/theqlikviewblog/archive/2010/10/03/molap-rolap-schmolap.aspx>