QlikView

GOVERNANCE OVERVIEW

A QlikView Technology White Paper

December 2011

qlikview.com

QlikView

Table of Contents

Overview

Governance of QlikView deployments is an important topic for any size deployment, but has particular significance for larger deployments involving teams of developers, large numbers of applications and changing data considerations. For QlikView system administrators, it's important to have a controlled development and data environment and to have a 360-degree view of their deployment.

This paper discusses two principle aspects of governance of QlikView deployments: Application governance and Data governance. Application governance refers to both the development of applications as well as the usage of applications, in a deployment. Data governance refers to understanding what data is being used and what its lineage is.

Who is this paper for?

This paper is designed to provide QlikView Developers and IT professionals with a high-level overview of the best practices and tools available to provide proper and consistent governance over QlikView deployments, small or large.

Application Governance

This section summarizes the best practices and available materials to address questions around governance of developing and deploying QlikView applications as well as governance over the usage of applications in a deployment. It does not cover data governance; that is covered in the next section.

DEVELOPMENT

For the purposes of this paper, it's convenient to consider QlikView as an application development environment and, as such, one that is subject to the rigors of development best practices that exist within all software development environments. These best practices extend to considerations of basic application development best practices, handling teams of developers working on a project, application certification and deployment migrations.

General Development Best Practices

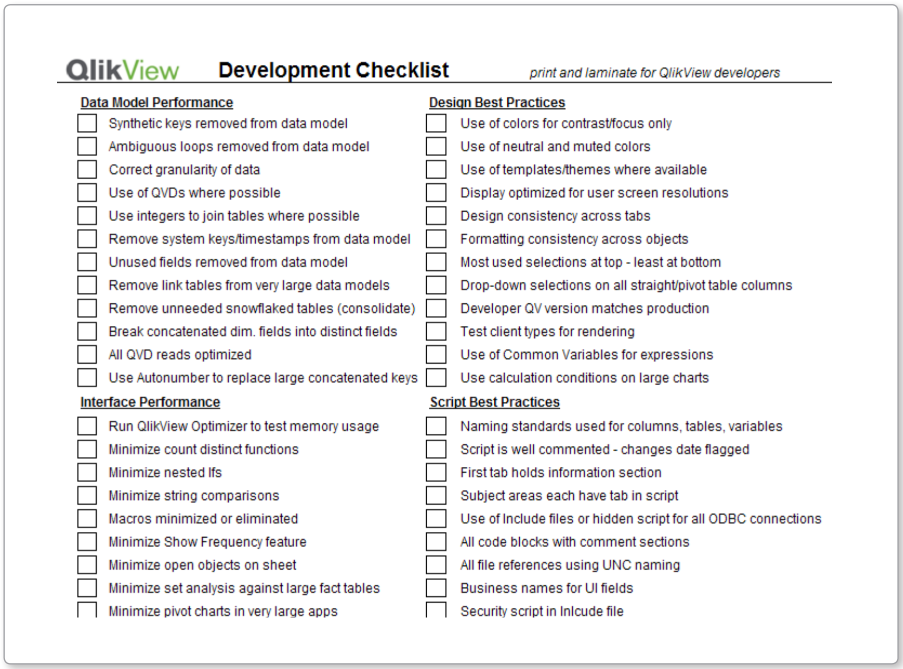
The document '*QlikView Best Practices – Development'* (located here) provides a comprehensive overview of the best practices recommended for conducting any development of QlikView applications. It is a reference manual for QlikView developers who's areas of expertise range from data modeling to scripting to UI design. The document is designed to facilitate a much clearer understanding of the methodologies and practices that are optimal for producing highly usable, highly optimized and highly configurable QlikView applications, whether used by small departments or by large enterprises.

A partial list of contents includes: UI Design, Scripting, Data Models, Variables, Security, Naming Standards and Testing and Certification.

In addition to this document, QlikTech have produced a free QlikView Developer toolkit and accompanying video that outlines the design best practices that should be followed when producing a QlikView application. These can be found at http://demo.qlikview.com under the Video Demos tab.

Every good development process should follow a checklist approach, to ensure that no critical pieces or processes are neglected. The document entitled *'QV Developer Checklist'* (located here) is a simple yet effective list of key areas that a developer should consider, ranging from data model performance to interface performance to scripting best practices.

Figure 1: Development Checklist



QlikView **Development Checklist** *print and laminate for QlikView developers*

Data Model Performance

- □ Synthetic keys removed from data model
- □ Ambiguous loops removed from data model
- □ Correct granularity of data
- □ Use of QVDs where possible
- □ Use integers to join tables where possible
- □ Remove system keys/timestamps from data model
- □ Unused fields removed from data model
- □ Remove link tables from very large data models
- □ Remove unneeded snowflaked tables (consolidate)
- □ Break concatenated dim. fields into distinct fields
- □ All QVD reads optimized
- □ Use Autonumber to replace large concatenated keys

Interface Performance

- □ Run QlikView Optimizer to test memory usage
- □ Minimize count distinct functions
- □ Minimize nested Ifs
- □ Minimize string comparisons
- □ Macros minimized or eliminated
- □ Minimize Show Frequency feature
- □ Minimize open objects on sheet
- □ Minimize set analysis against large fact tables
- □ Minimize pivot charts in very large apps

Design Best Practices

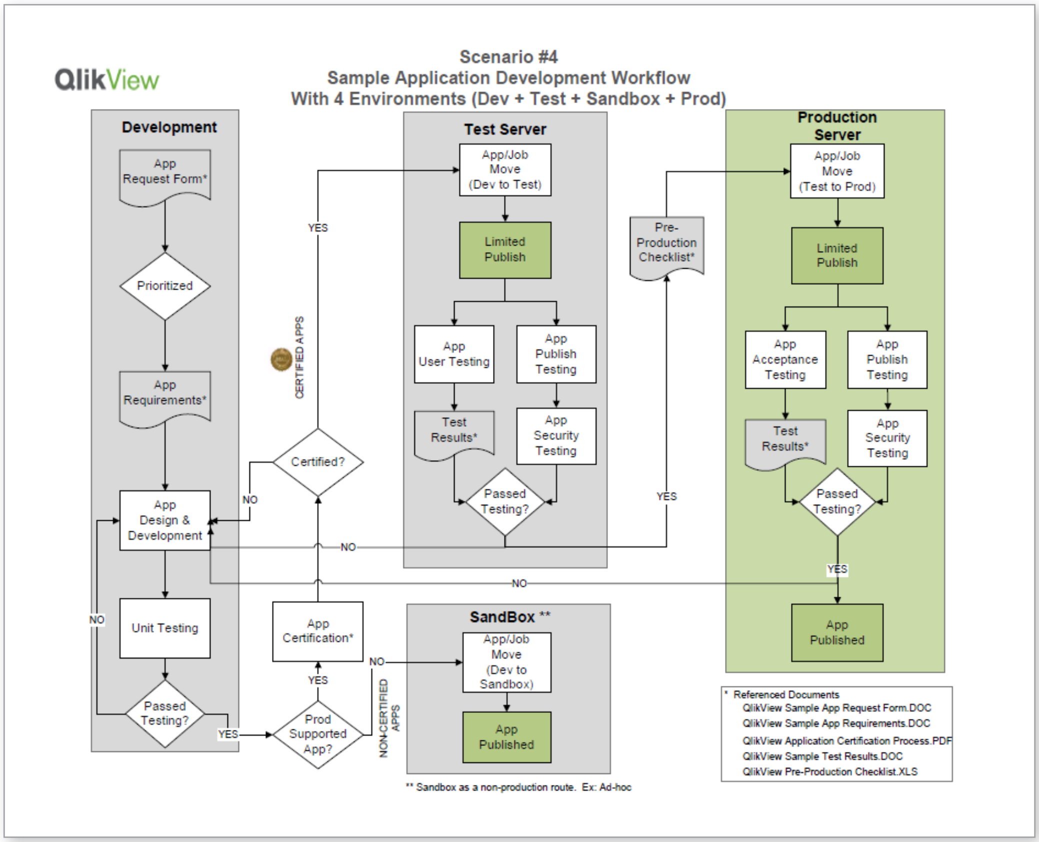
- □ Use of colors for contrast/focus only
- □ Use of neutral and muted colors
- □ Use of templates/themes where available
- □ Display optimized for user screen resolutions
- □ Design consistency across tabs
- □ Formatting consistency across objects
- □ Most used selections at top - least at bottom
- □ Drop-down selections on all straight/pivot table columns
- □ Developer QV version matches production
- □ Test client types for rendering
- □ Use of Common Variables for expressions
- □ Use calculation conditions on large charts

Script Best Practices

- □ Naming standards used for columns, tables, variables
- □ Script is well commented - changes date flagged
- □ First tab holds information section
- □ Subject areas each have tab in script
- □ Use of Include files or hidden script for all ODBC connections
- □ All code blocks with comment sections
- □ All file references using UNC naming
- □ Business names for UI fields
- □ Security script in Inlcude file

Application development workflows are necessary within any development project and must be adhered to especially when dealing with environments that include Dev-Test-Pre-Prod-Prod type environments. Having a workflow in place to determine when an application can/should be promoted to a test server or a production server is important to the success of any deployment. The document, *'App Dev Workflow Scenarios'* (located here) provides 4 example scenarios that development teams could follow, depending on their particular deployment circumstances and governance infrastructure.

Figure 2: Example Application Development Workflow Scenario

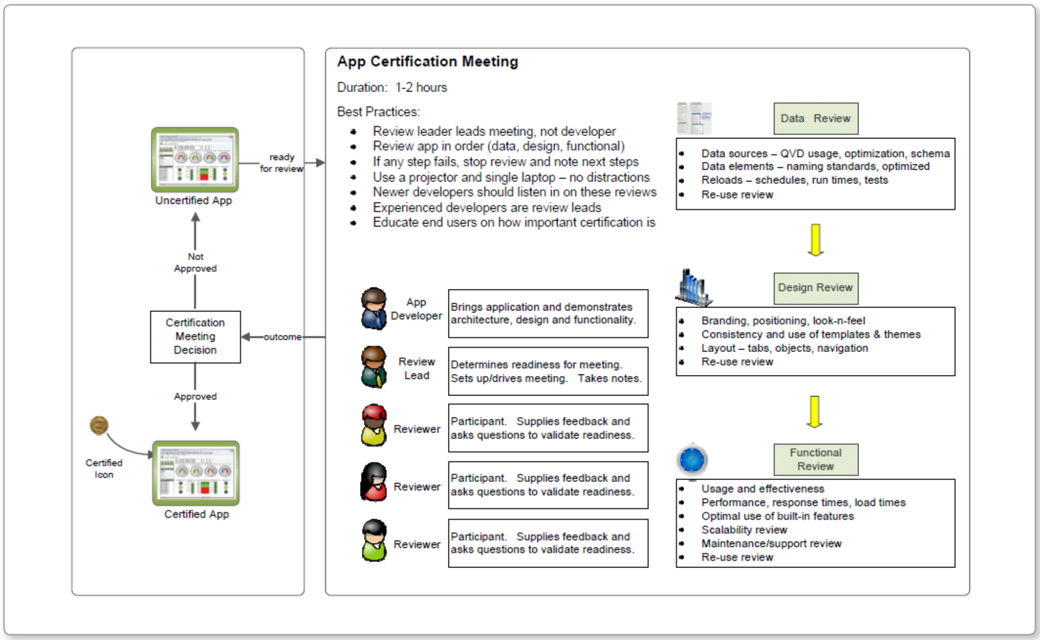


© 2011 QlikTech

Having an application certification process in place is important for any QlikView deployment. As the usage of QlikView grows within an organization, QlikView applications can proliferate rapidly. As with any such environment, the quality and relevance of the applications can vary across the entire deployment. As a result, it's important for the users of the applications to have an indication as to the quality of the application and whether it is the most current version. In short, what business users desire is to know whether an application has been 'certified' or not by their development team(s). Certification indicates that an application has followed standard development methodologies, has been reviewed by the correct people and has passed functional use tests.

Certification is determined by the best practices and governance procedures in place within your organization, however there exists a document, entitled *'Application Certification Process'* (located here) to help guide organizations who are considering a certification process for their QlikView deployments.

Figure 3 : QlikView Application Certification



App Certification Meeting

Duration: 1-2 hours

Best Practices:

- Review leader leads meeting, not developer
- Review app in order (data, design, functional)
- If any step fails, stop review and note next steps
- Use a projector and single laptop – no distractions
- Newer developers should listen in on these reviews
- Experienced developers are review leads
- Educate end users on how important certification is

App Developer: Brings application and demonstrates architecture, design and functionality.

Review Lead: Determines readiness for meeting. Sets up/drives meeting. Takes notes.

Reviewer: Participant. Supplies feedback and asks questions to validate readiness.

Reviewer: Participant. Supplies feedback and asks questions to validate readiness.

Reviewer: Participant. Supplies feedback and asks questions to validate readiness.

Data Review

- Data sources – QVD usage, optimization, schema
- Data elements – naming standards, optimized
- Reloads – schedules, run times, tests
- Re-use review

Design Review

- Branding, positioning, look-n-feel
- Consistency and use of templates & themes
- Layout – tabs, objects, navigation
- Re-use review

Functional Review

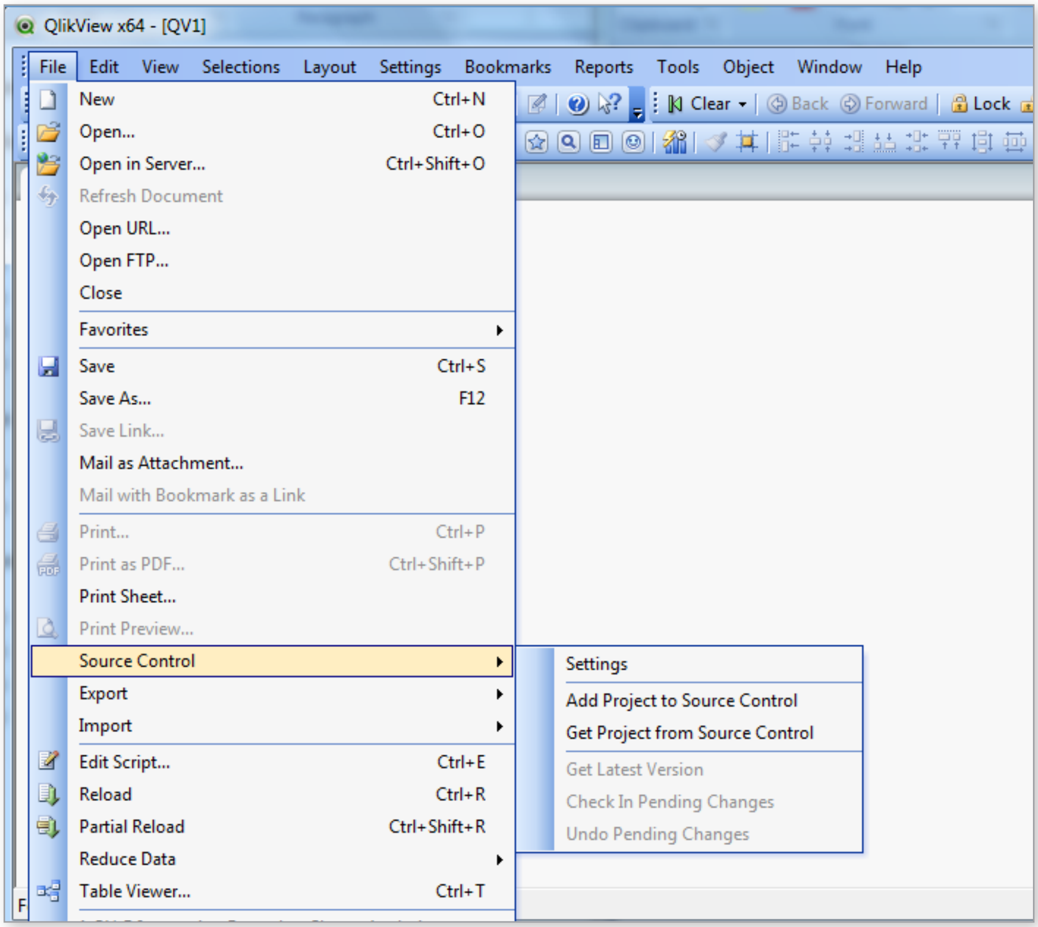
- Usage and effectiveness
- Performance, response times, load times
- Optimal use of built-in features
- Scalability review
- Maintenance/support review
- Re-use review

DEVELOPMENT TEAMS

It's very common for a team or teams of QlikView developers to work in concert to produce and maintain QlikView applications. In these types of environments it's important to maintain strict control and visibility over the various work products produced during a project by individual team members. Having a robust application control system (i.e. source control system) in place is critical to ensure developers are working on the most current and authorized version of the application(s).

Source control integration was introduced in version 11 of QlikView. With this integration, development teams can now integrate with a source control system like Microsoft TFS to maintain project control.

Figure 4: Source Control Integration in QlikView



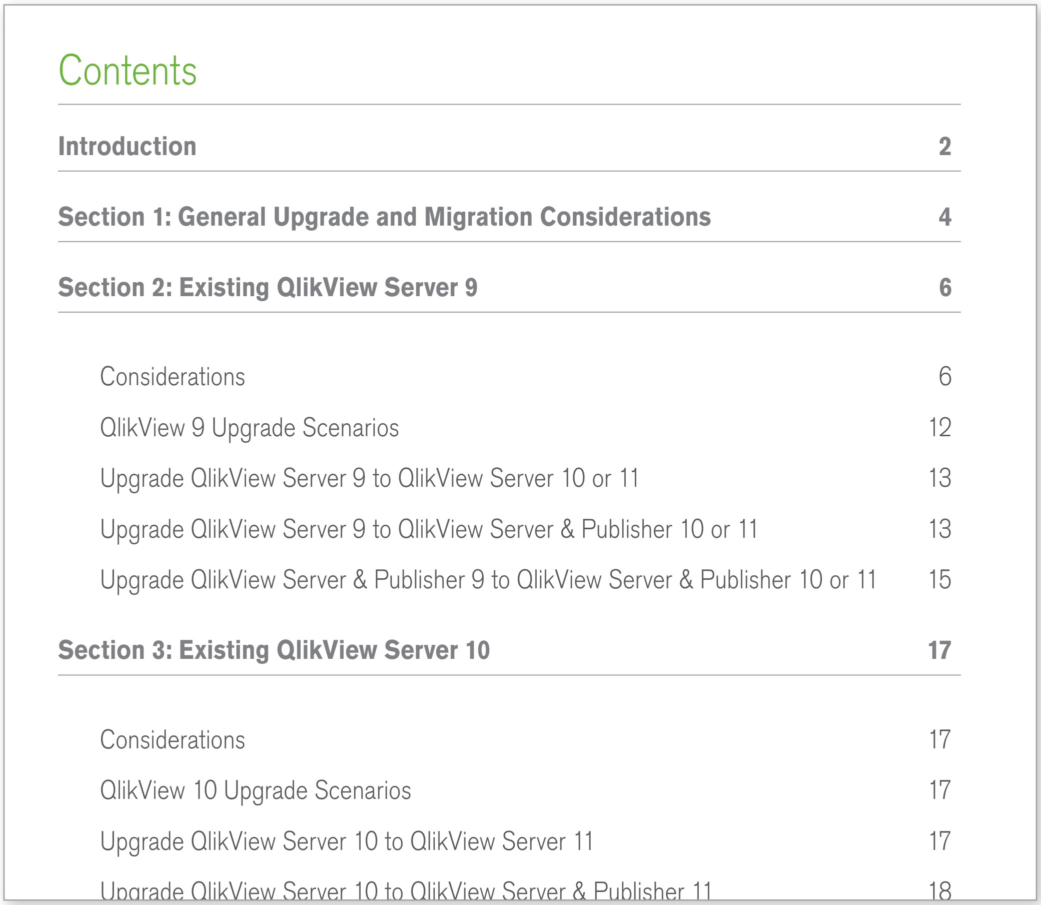
© 2011 QlikTech

In addition, QlikTech partner NOAD provides a complete change management and deployment workflow solution that is integrated with QlikView, called EQM4. For more information, go to www.noadbi.com

UPGRADES AND MIGRATIONS

Deployment upgrades to new versions, as well as migrations to new server infrastructures are a part of a QlikView administrator's role. To help facilitate this process, a document entitled *'QlikView 11 Upgrade and Migrations'* (located here) has been produced as a step-by-step guide to moving from QlikView 9 and 10 versions to QlikView 11. It also includes best practices for migrating from a single server environment to a multi-server environment with dispersed services.

Figure 5: QV11 Upgrade and Migrations document



Contents

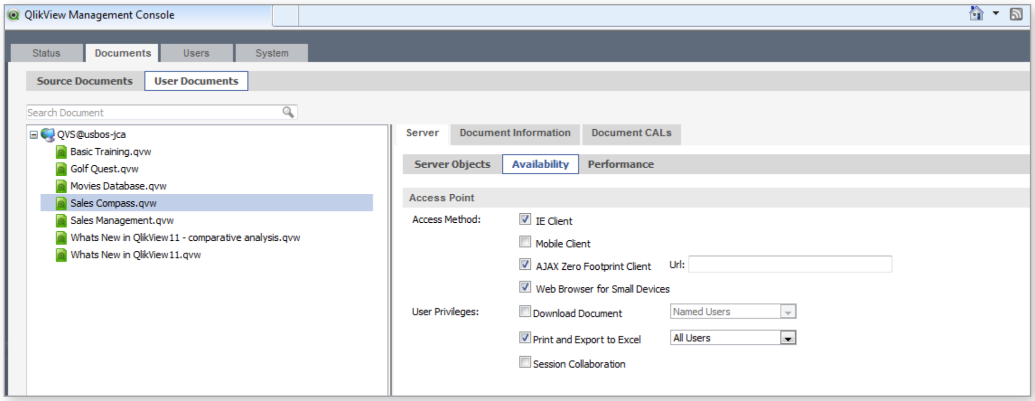
© 2011 QlikTech

USAGE

Understanding the usage of QlikView deployments is important for a number of reasons. Firstly, it's important to understand the system usage, from a hardware and bandwidth perspective, so that QlikView administrators can adequately provision the correct hardware and network infrastructure to support a growing deployment. Secondly, it's important from an auditing and compliance perspective, especially in those industries where these requirements are critical, such as healthcare, finance and government.

The QlikView Management Console (QMC) allows system administrators to manage QlikView deployments. One of the tasks that can be performed allows system administrators to determine which users that are allowed to access applications and data, which users are allowed to download applications and data and which developers are allowed to access the data packages (QVD files) used with QlikView Publisher. It also can control which applications (and their data) can be downloaded, and which cannot.

Figure 6: QlikView Management Console



QVS@usbos-jca
- Basic Training.qvw
- Golf Quest.qvw
- Movies Database.qvw
- Sales Compass.qvw
- Sales Management.qvw
- Whats New in QlikView11 - comparative analysis.qvw
- Whats New in QlikView11.qvw

Access Point

Access Method:
- ☑ IE Client
- ☐ Mobile Client
- ☑ AJAX Zero Footprint Client Url:
- ☑ Web Browser for Small Devices

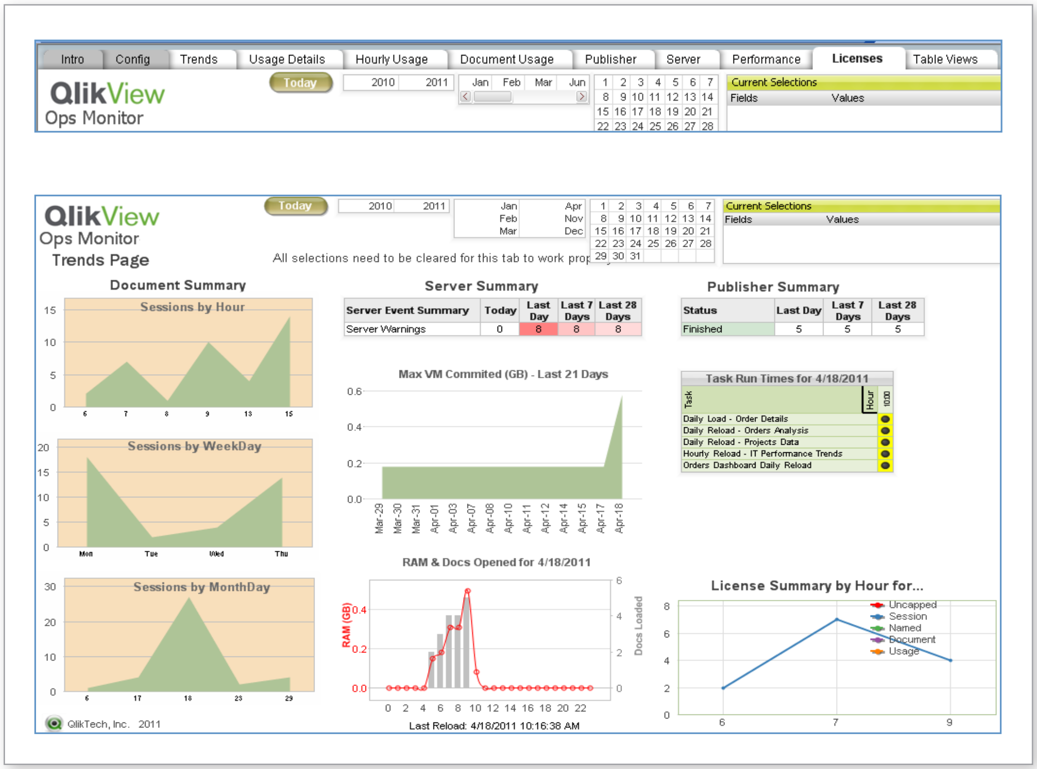
User Privileges:
- ☐ Download Document Named Users
- ☑ Print and Export to Excel All Users
- ☐ Session Collaboration

© 2011 QlikTech

QlikTech has produced a tool that allows a system administrator to monitor the complete operations of their QlikView deployment in a visual and interactive way. The *'QlikView Ops Monitor'* (located here) is an application build using QlikView that will read the log files from a QlikView Server and Publisher deployment and will provide the results to an administrator in a visual, interactive environment. The administrator simply configures the application to read the appropriate log files and will display the results instantly. Administrators can analyze a wide range of factors, including (but not limited to):

- Performance characteristics (including RAM and CPU usage)
- Task results
- Data volumes generated
- User access data
- Document usage
- License usage

Figure 7: QlikView Ops Monitor application



All selections need to be cleared for this tab to work prop

Last Reload: 4/18/2011 10:16:38 AM

Server Summary

Server Event Summary	Today	Last Day	Last 7 Days	Last 28 Days
Server Warnings	0	8	8	8

Publisher Summary

Status	Last Day	Last 7 Days	Last 28 Days
Finished	5	5	5

Task Run Times for 4/18/2011

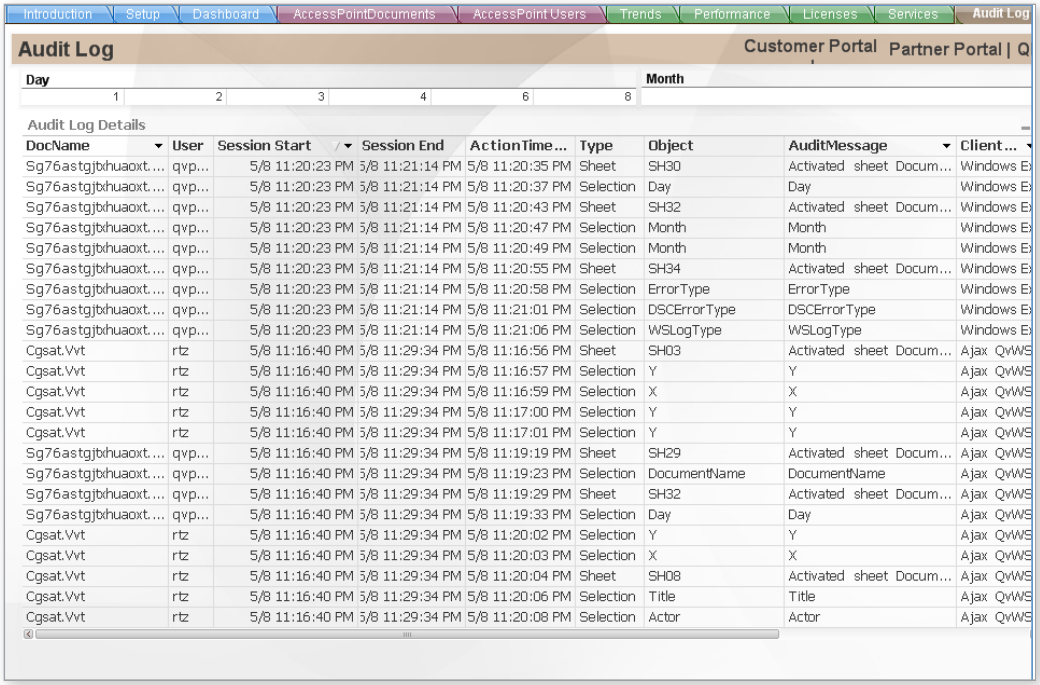
Task	10:00
Daily Load - Order Details	●
Daily Reload - Orders Analysis	●
Daily Reload - Projects Data	●
Hourly Reload - IT Performance Trends	●
Orders Dashboard Daily Reload	●

© 2011 QlikTech

This free utility enables a QlikView system administrator to both optimize their deployment but also provide them with an oversight capability.

In addition, QlikView administrators very often need to provide auditing results to meet regulatory requirements. QlikView deployments have built-in auditing capabilities, where user activity can be logged down to the object-level of interaction. That is, audit logs can be written to establish who clicked on what, and when. Using the *'QlikView Systems Monitor'* application (located here), these logs can be read and analyzed easily.

Figure 8: Audit log analysis of a QlikView deployment



Audit Log Details

DocName	User	Session Start	Session End	ActionTime...	Type	Object	AuditMessage	Client...
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:20:35 PM	Sheet	SH30	Activated sheet Docum...	Windows E
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:20:37 PM	Selection	Day	Day	Windows E
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:20:43 PM	Sheet	SH32	Activated sheet Docum...	Windows E
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:20:47 PM	Selection	Month	Month	Windows E
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:20:49 PM	Selection	Month	Month	Windows E
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:20:55 PM	Sheet	SH34	Activated sheet Docum...	Windows E
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:20:58 PM	Selection	ErrorType	ErrorType	Windows E
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:21:01 PM	Selection	DSCErrorType	DSCErrorType	Windows E
Sg76astgjtxhuaoxt....	qvp...	5/8 11:20:23 PM	5/8 11:21:14 PM	5/8 11:21:06 PM	Selection	WSLogType	WSLogType	Windows E
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:16:56 PM	Sheet	SH03	Activated sheet Docum...	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:16:57 PM	Selection	Y	Y	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:16:59 PM	Selection	X	X	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:17:00 PM	Selection	Y	Y	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:17:01 PM	Selection	Y	Y	Ajax QvWS
Sg76astgjtxhuaoxt....	qvp...	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:19:19 PM	Sheet	SH29	Activated sheet Docum...	Ajax QvWS
Sg76astgjtxhuaoxt....	qvp...	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:19:23 PM	Selection	DocumentName	DocumentName	Ajax QvWS
Sg76astgjtxhuaoxt....	qvp...	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:19:29 PM	Sheet	SH32	Activated sheet Docum...	Ajax QvWS
Sg76astgjtxhuaoxt....	qvp...	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:19:33 PM	Selection	Day	Day	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:20:02 PM	Selection	Y	Y	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:20:03 PM	Selection	X	X	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:20:04 PM	Sheet	SH08	Activated sheet Docum...	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:20:06 PM	Selection	Title	Title	Ajax QvWS
Cgsat.Vvt	rtz	5/8 11:16:40 PM	5/8 11:29:34 PM	5/8 11:20:08 PM	Selection	Actor	Actor	Ajax QvWS

SECURITY

Providing access to the right data for the right people is a critical component of any QlikView deployment. Equally as important is ensuring that no unauthorized access to data is permitted. QlikTech has addressed this topic in its comprehensive white paper, *'QlikView Security Overview Technology White Paper'* (located here). QlikView provides a variety of mechanisms to safeguard data and applications, including integration with existing Single Sign-On (SSO) systems, integration with Active Directory and other LDAP providers, as well as native row-level security. Using a combination of these methods – highlighted in the video series *'QlikView Security Overview Video Series'* (located here) – QlikView deployments are made highly secure and controlled.

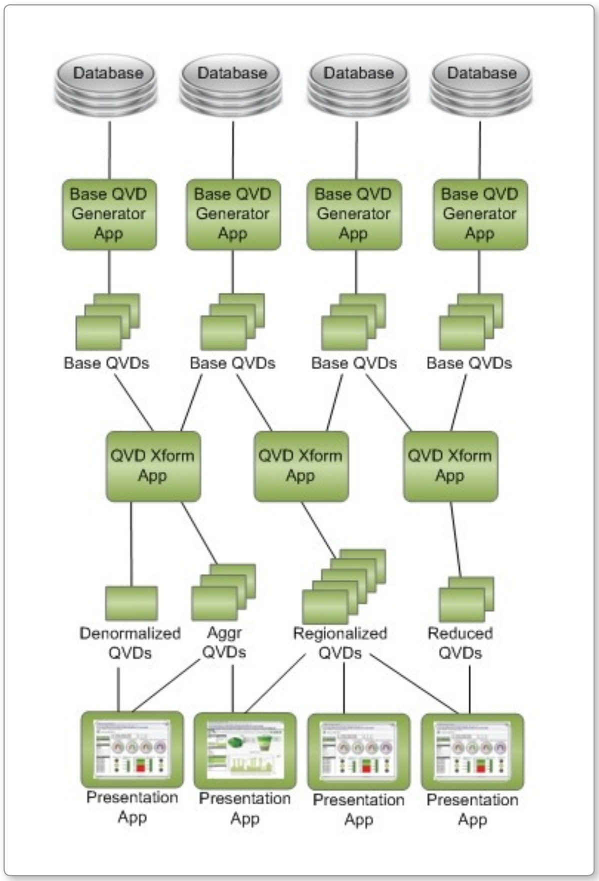
Data Governance

This section summarizes the best practices and available documents to address questions around data governance within QlikView deployments.

QVD LAYER

Almost all medium to large QlikView deployments incorporate a so-called *'QlikView Data (QVD) layer'* to stage data for rapid reload into front-line QlikView applications. The QVD layer includes QVD files that contain highly compressed and efficient data models, often organized around key metrics such as time, department, function or other user-defined metric. QVD files are created by extracting data from sources such as databases, flat files, transactional systems and other QVD files. These QVD files are then used throughout a variety of different front-line applications. They also can change, based on data reload schedules and changing business requirements. Therefore, it is critical for a QlikView administrator to understand the data lineage of and changes made to their QVD layer.

Figure 9: Multi-staged QVD environment



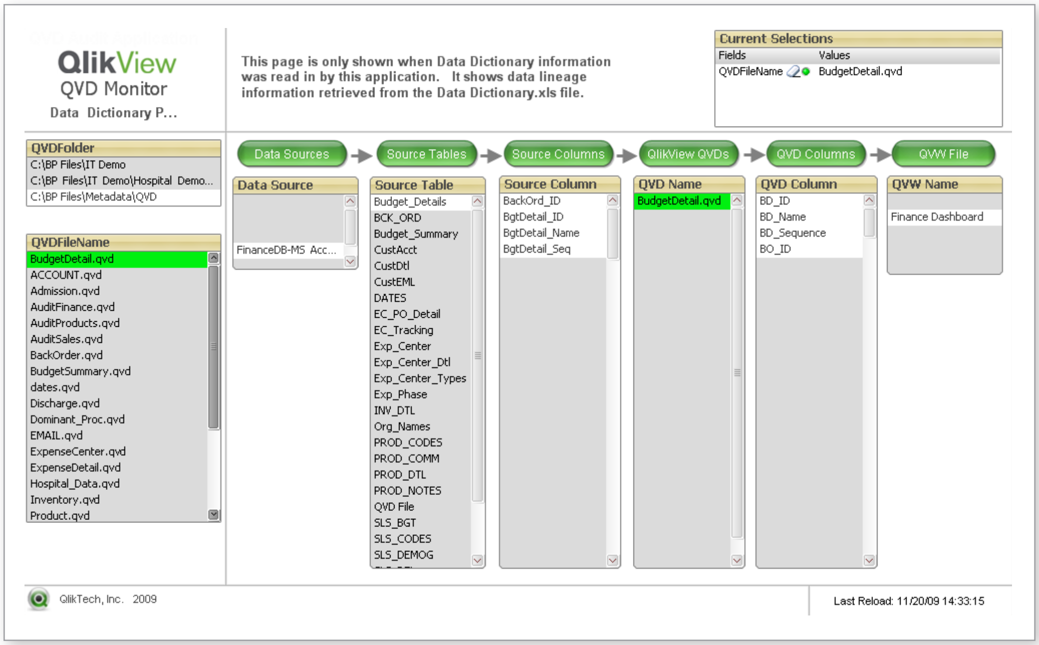
© 2011 QlikTech

The *'QVD Monitor'* application (located here) is a free utility designed to allow administrators to conduct proactive data monitoring of their entire QVD infrastructure. It provides a comprehensive view of the QVD layer, including (but not limited to):

- Changes to QVD files
- Source data tracking and lineage
- Trend analysis of QVD sizes and usage
- Field-level lookup
- Reload date tracking

This application allows an administrator to monitor QlikView QVD files and their associated metadata. They can track data such as load date/times, sizes, record counts, column counts, column names, and trends for each. A set of charts is included to highlight changes in data and to help identify any anomalies or exceptions that might occur.

Figure 10: QVD Data lineage display



This page is only shown when Data Dictionary information was read in by this application. It shows data lineage information retrieved from the Data Dictionary.xls file.

Current Selections

Fields	Values
QVDFileName	BudgetDetail.qvd

QVDFolder
C:\BP Files\IT Demo
C:\BP Files\IT Demo\Hospital Demo...
C:\BP Files\Metadata\QVD

QVDFileName
BudgetDetail.qvd
ACCOUNT.qvd
Admission.qvd
AuditFinance.qvd
AuditProducts.qvd
AuditSales.qvd
BackOrder.qvd
BudgetSummary.qvd
dates.qvd
Discharge.qvd
Dominant_Proc.qvd
EMAIL.qvd
ExpenseCenter.qvd
ExpenseDetail.qvd
Hospital_Data.qvd
Inventory.qvd
Product.qvd

Data Source
FinanceDB-MS Acc...

Source Table
Budget_Details
BCK_ORD
Budget_Summary
CustAcct
CustDtl
CustEML
DATES
EC_PO_Detail
EC_Tracking
Exp_Center
Exp_Center_Dtl
Exp_Center_Types
Exp_Phase
INV_DTL
Org_Names
PROD_CODES
PROD_COMM
PROD_DTL
PROD_NOTES
QVD File
SLS_BGT
SLS_CODES
SLS_DEMOG

Source Column
BackOrd_ID
BgtDetail_ID
BgtDetail_Name
BgtDetail_Seq

QVD Name
BudgetDetail.qvd

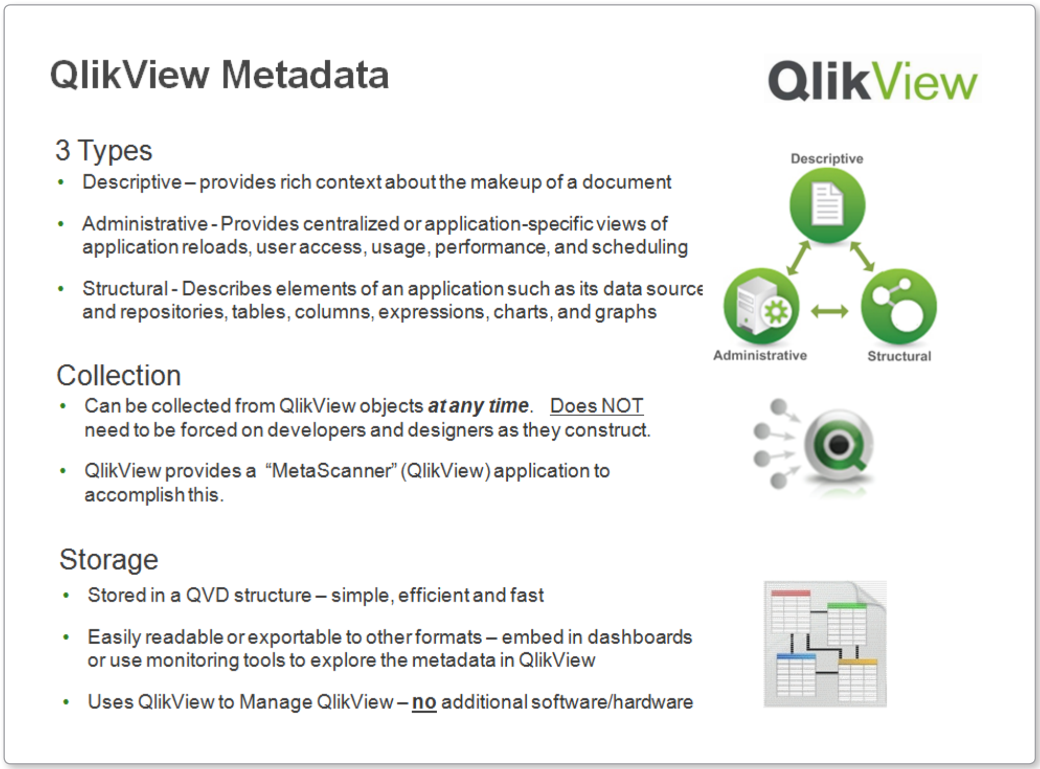
QVD Column
BD_ID
BD_Name
BD_Sequence
BO_ID

QVW Name
Finance Dashboard

QlikTech, Inc. 2009

Last Reload: 11/20/09 14:33:15

Figure 11: QVD Data size trend monitor



QlikView Metadata

QlikView

3 Types

- Descriptive – provides rich context about the makeup of a document
- Administrative - Provides centralized or application-specific views of application reloads, user access, usage, performance, and scheduling
- Structural - Describes elements of an application such as its data source and repositories, tables, columns, expressions, charts, and graphs

Collection

- Can be collected from QlikView objects ***at any time***. Does NOT need to be forced on developers and designers as they construct.
- QlikView provides a "MetaScanner" (QlikView) application to accomplish this.

Storage

- Stored in a QVD structure – simple, efficient and fast
- Easily readable or exportable to other formats – embed in dashboards or use monitoring tools to explore the metadata in QlikView
- Uses QlikView to Manage QlikView – **no** additional software/hardware

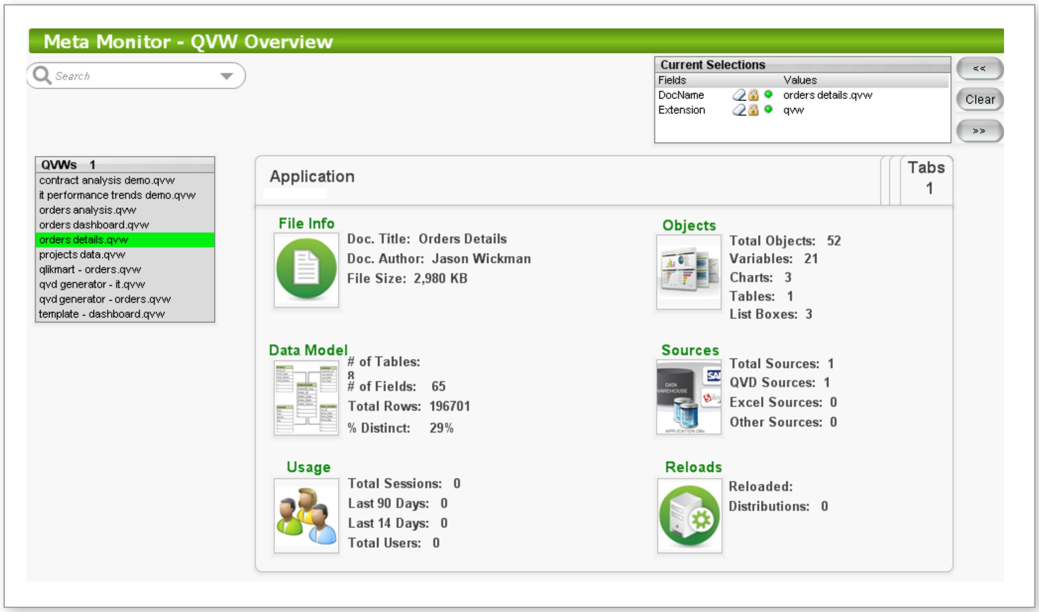
METADATA

The topic of metadata is an important one and is usually high on the list of relevant topics when discussing the governance of any BI deployment, whether with QlikView or another solution. As outlined in the comprehensive white paper on this topic, *'QlikView's Pragmatic Approach to Metadata'* (located here), metadata management with QlikView is both optional and pervasive. That is, a QlikView system admin can chose to implement a metadata monitoring environment (or not, depending on their requirements), and – if they chose to do so – can tap into metadata that is automatically created with any QlikView deployment. This pragmatic approach gives system administrators all the flexibility they need, when they need it.

In addition to the white paper, two utilities exist that allow a QlikView system administrator to extract the metadata from a deployment and, once extracted, analyze it and monitor it for changes that occur over time. The *'MetaScanner'* application (located here) will scan through a deployment, extracting the descriptive, administrative and structural metadata that exists in a deployment, and the *'MetaMonitor'* application (located here) will provide administrators a clear picture of their entire deployment. Examples include (but are not limited to):

- Application-level
    - Number of objects
    - Number of variables
    - Number of QVD sources
    - Reload schedule
    - Usage stats
- Deployment-level
    - Server info messages
    - VM committed
    - RAM usage
    - Session statistics
    - Publisher reload events tracking

Figure 12: Application-level statistics with MetaMonitor



Meta Monitor - QVW Overview

Current Selections

Fields	Values
DocName	orders details.qvw
Extension	qvw

QVWs 1

- contract analysis demo.qvw
- it performance trends demo.qvw
- orders analysis.qvw
- orders dashboard.qvw
- orders details.qvw
- projects data.qvw
- qlikmart - orders.qvw
- qvd generator - it.qvw
- qvd generator - orders.qvw
- template - dashboard.qvw

File Info

Doc. Title: Orders Details
Doc. Author: Jason Wickman
File Size: 2,980 KB

Objects

Total Objects: 52
Variables: 21
Charts: 3
Tables: 1
List Boxes: 3

Data Model

# of Tables: 8
# of Fields: 65
Total Rows: 196701
% Distinct: 29%

Sources

Total Sources: 1
QVD Sources: 1
Excel Sources: 0
Other Sources: 0

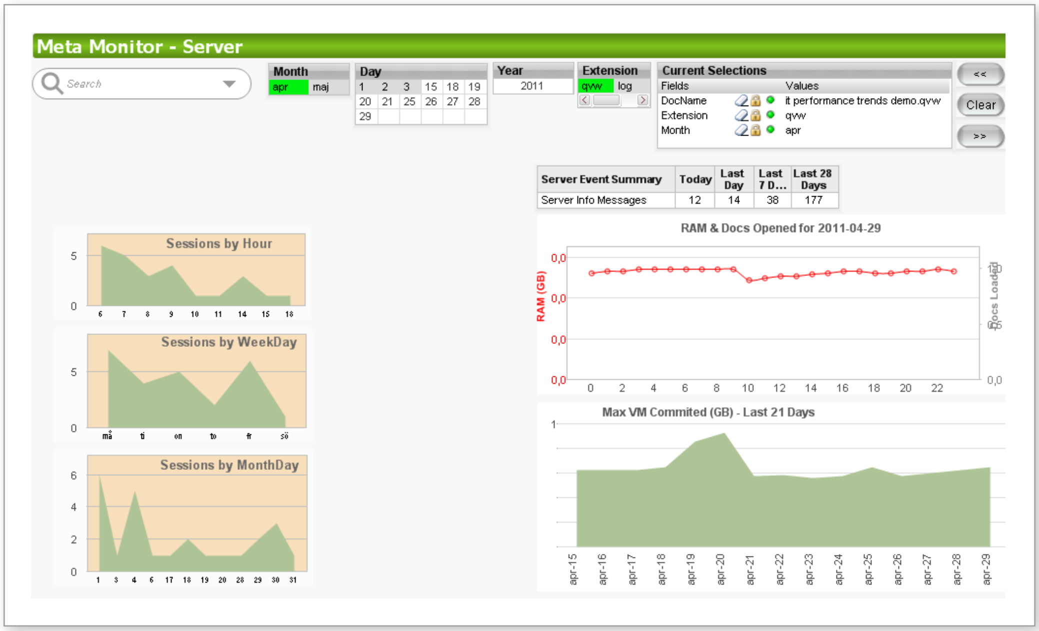
Usage

Total Sessions: 0
Last 90 Days: 0
Last 14 Days: 0
Total Users: 0

Reloads

Reloaded:
Distributions: 0

Figure 13: Deployment-level statistics with MetaMonitor



Meta Monitor - Server

Month	
apr	maj

Day					
1	2	3	15	18	19
20	21	25	26	27	28
29					

Year
2011

Extension	
qvw	log

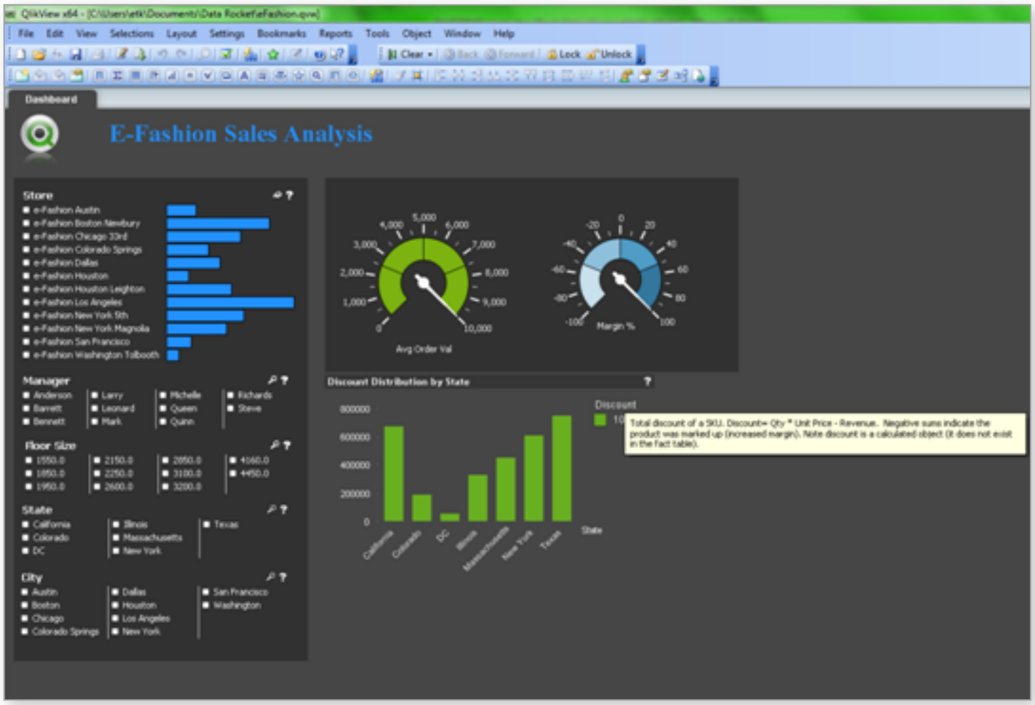
Current Selections

Fields	Values
DocName	it performance trends demo.qvw
Extension	qvw
Month	apr

Server Event Summary	Today	Last Day	Last 7 D…	Last 28 Days
Server Info Messages	12	14	38	177

QlikView can also access the metadata from third party systems, provided there are sufficient access privileges and API's in place on the third party systems to allow their metadata to be extracted. By doing this, QlikView can inherit the business rules, calculations and data lineage information without having the create them again. Figure 14 shows a QlikView application that is using metadata extracted from a Business Objects Universe using a connector build by a QlikTech partner, DataRoket.

Figure 14: QlikView application using metadata from Business Objects



Total discount of a SKU. Discount= Qty * Unit Price - Revenue. Negative sums indicate the product was marked up (increased margin). Note discount is a calculated object (it does not exist in the fact table).

Conclusion

This paper has outlined the available tools and documents for providing robust governance over QlikView deployments. Topics such as application governance, including both development and usage have been covered, along with governance of data. For a better understanding of your own specific governance requirements, please connect with your local QlikTech representative to set up a meeting.

Appendix

GOVERNANCE FILES

All files are located in the QlikCommunity, located here (http://community.qlikview.com/community)

- QlikView Best Practices – Development v0.5.pdf
- QV Developer Checklist.xls
- App Dev Workflow Scenarios.pdf
- Application Certification Process.pdf
- QlikView Ops Monitor.qvw
- QlikView Systems Monitor.qvw
- QlikView Security Overview Technology White Paper.pdf
- QVD Monitor.qvw
- QlikView's Pragmatic Approach to Metadata.pdf
- MetaScanner.qvw
- MetaMonitor.qvw