QLIKVIEW INTEGRATION OVERVIEW

QlikView Technology White Paper Series

June 2011

www.qlikview.com

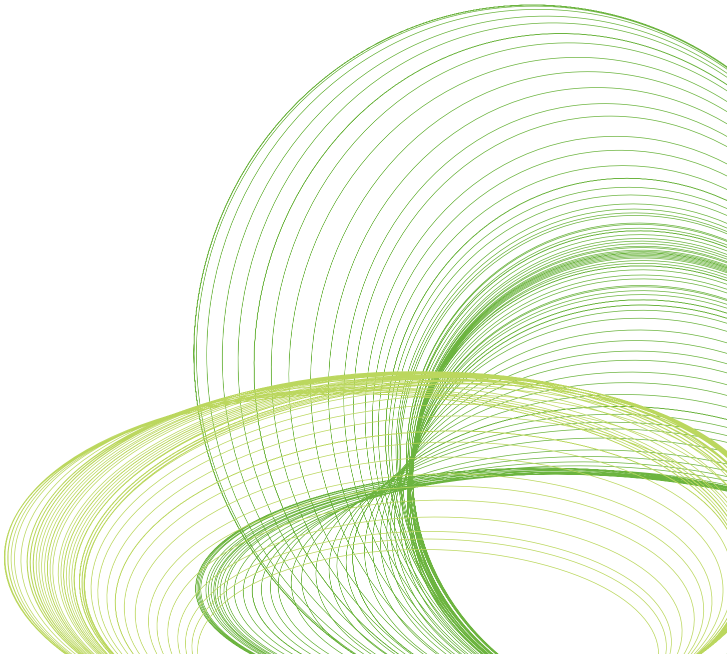


Table of Contents

Overview

Understanding how QlikView integrates with existing and proposed IT system resources such as data repositories, security infrastructure, portals, schedulers and so on, is an important element in understanding how QlikView's Business Discovery platform can benefit a company's business and technical community. This paper provides an overview of the various ways in which QlikView integrates with other common IT infrastructure pieces to provide for an optimal deployment.

QlikView can enrich existing web sites and applications, enhance portals and let you customize the way that data is visualized. QlikView can unlock data trapped in multiple sources, integrate with and benefit from existing corporate applications and security systems which all contribute to reduced systems management time.

WHO IS THIS PAPER FOR?

This paper is designed to provide QlikView Developers and IT professionals with a high-level overview of how to integrate QlikView into a typical enterprise IT environment. Further, in-depth companion pieces cover in much greater depth the more detailed aspects of integration and best practices. It is recommended to work with a local QlikView representative to understand the next level of content that is appropriate for any particular needs.

Data Source Integration

One of the key tenets of QlikView is that it can import data from a wide range of structured and unstructured sources of data and find common associations between them.

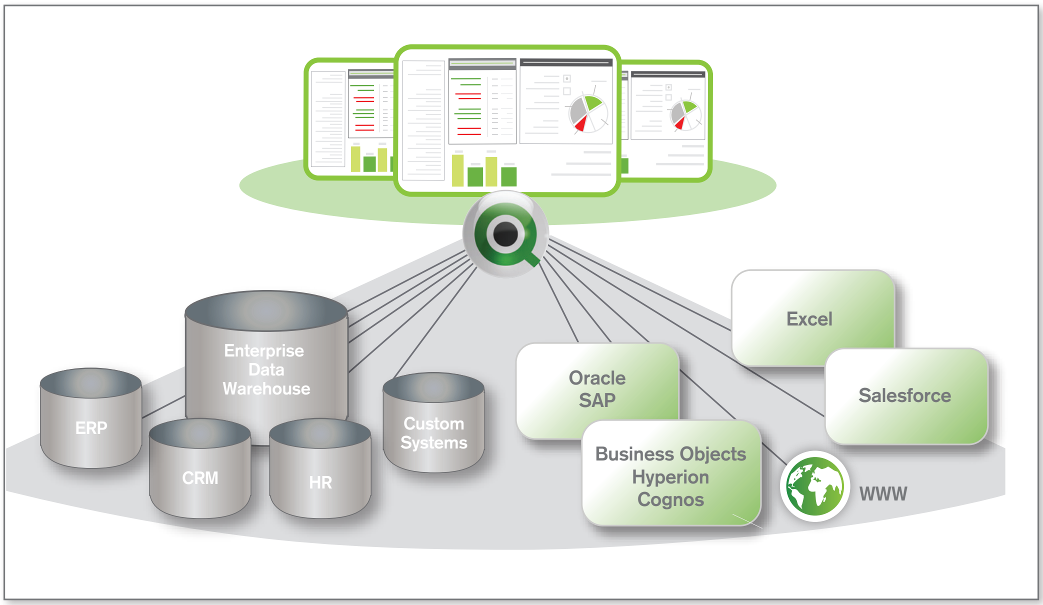


Figure 1: *QlikView can access data from a large variety of sources*

STANDARD ODBC/OLE CONNECTIONS

Many common data sources such as SQL Server and Oracle provide ODBC/OLE drivers to allow data extraction from tables and views. Additionally many other software packages use a common database such as SQL Server to store the data behind the application which is in turn available via ODBC/OLE (e.g. Microsoft Dynamics), thus further expanding the range of systems QlikView can extract data from.

QlikView can enable ODBC/OLE connections through a simple wizard to quickly extract data from source systems. QlikView uses standard SQL queries to read data and can also use Store Procedures. QlikView can also use both 32 bit and 64 bit drivers in the same application to support data extractions from legacy systems.

NON-STANDARD DATA SOURCES

An organization's data can often exist in systems where an ODBC/OLE connector is not available such as Software as a Service (SaaS) soultions like Salesforce.com. Additionally data can be stored or extracted from unstructured sources such as spreadsheets, text files and XML data files. QlikView can provide access to many different types of unstructured data sources through a range of methods.

QLIKVIEW CONNECTORS FOR SALESFORCE.COM AND SAP NETWEAVER

QlikView provides two connectors to extract data from two common data sources that do not provide typical ODBC/OLE drivers. These connectors are for Salesforce.com and SAP Netweaver and provide a straightforward method for extracting data from these systems.

SALESFORCE.COM

Salesforce.com provides web based business applications through a SaaS model. They provide a range of web services and APIs to allow access to data in the system from external applications (such as QlikView).

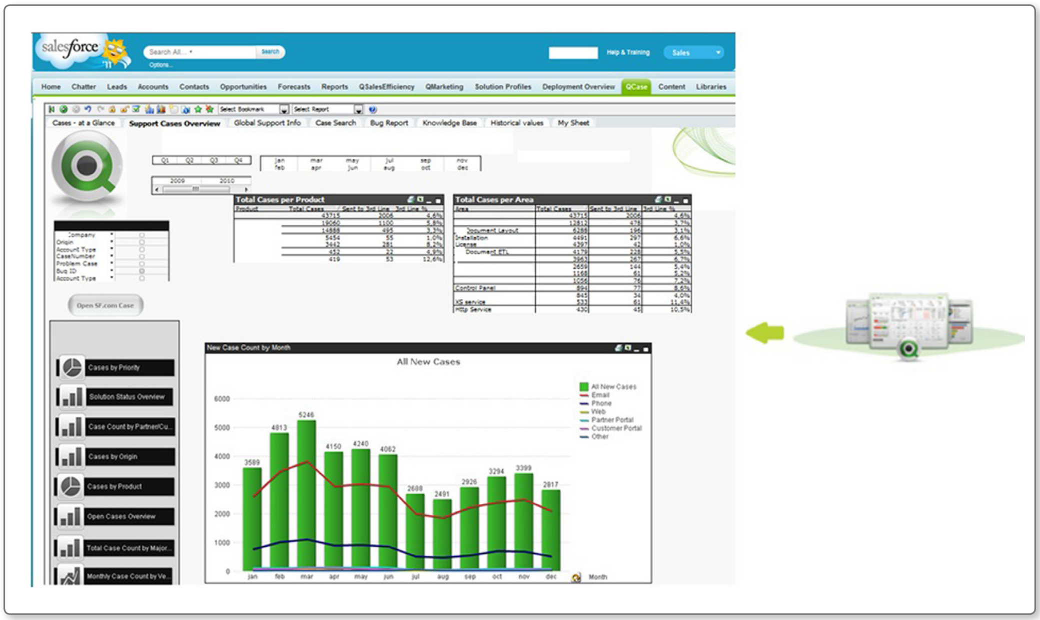
The QlikView Salesforce.com connector, when added to QlikView, allows a user to extract data from Salesforce.com in a manner that is similar to the ODBC/OLE connector method, and can use QlikView's wizard interfaces to accomplish this.

The connector can access all of the common fields in Salesforce.com as well as any custom fields an organization may have in use. The connector can be deployed to allow an end user to connect directly to Salesforce.com with their own user credentials in order to extract their own data or can be deployed using a client-server model to direct all requests via a central location.

QlikTech provides a starter template to enable QlikView users to quickly get started doing analytics using Salesforce.com data.

QLIKVIEW EMBEDDED INSIDE SALESFORCE.COM

Additionally, QlikView applications can integrate directly within the Salesforce.com user interface, providing a power analytics capability directly within a Salesforce.com deployment. For example consider viewing an account in Salesforce.com and being able to view rich analytics in the context of that account.



Total Cases per Product

Product	Total Cases	Sent to 3rd Line	3rd Line %
	43715	2006	4,6%
	19060	1100	5,8%
	14888	495	3,3%
	5454	55	1,0%
	3442	281	8,2%
	452	22	4,9%
	419	53	12,6%

Total Cases per Area

Area	Total Cases	Sent to 3rd Line	3rd Line %
	43715	2006	4,6%
	12812	478	3,7%
Document Layout	6288	196	3,1%
Installation	4491	297	6,6%
License	4397	42	1,0%
Document ETL	4179	228	5,5%
	3963	267	6,7%
	2659	144	5,4%
	1168	61	5,2%
	1056	76	7,2%
Control Panel	894	77	8,6%
	845	34	4,0%
XS service	533	61	11,4%
Http Service	430	45	10,5%

Figure 2: *: QlikView applications and objects can be embedded inside Salesforce.com*

SAP NETWEAVER

SAP provides a range of business applications that include some traditional warehouse and reporting tools. SAP stores data in its own proprietary format which is often difficult to query, requiring deep knowledge of the system to understand.

The QlikView SAP connector greatly simplifies the process of extracting data from SAP warehouses. The QlikView SAP connector includes a simple query builder and can use the metadata and field descriptions from SAP to allow users to choose fields with meaningful names, allowing users to rapidly extract and understand SAP data in QlikView.

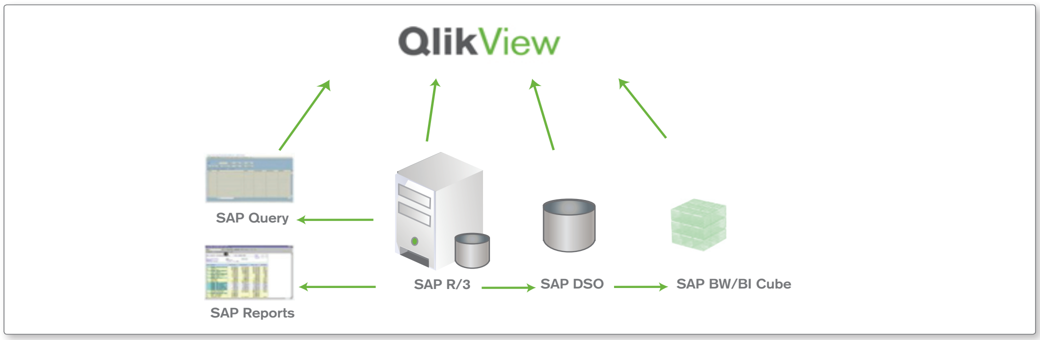


Figure 3: *SAP integration with QlikView*

The QlikView connector supports data extraction from SAP® R/3®, mySAP™ and can also leverage existing SAP BW and BEX information if required.

In addition to importing SAP data QlikView can also import the user roles and permissions data from SAP to ensure user access in QlikView mirrors that of SAP ensuring users see only the data they are permitted to see.

QlikTech provides a range of starter templates for many common SAP reporting types to get users started quickly with their deployments.

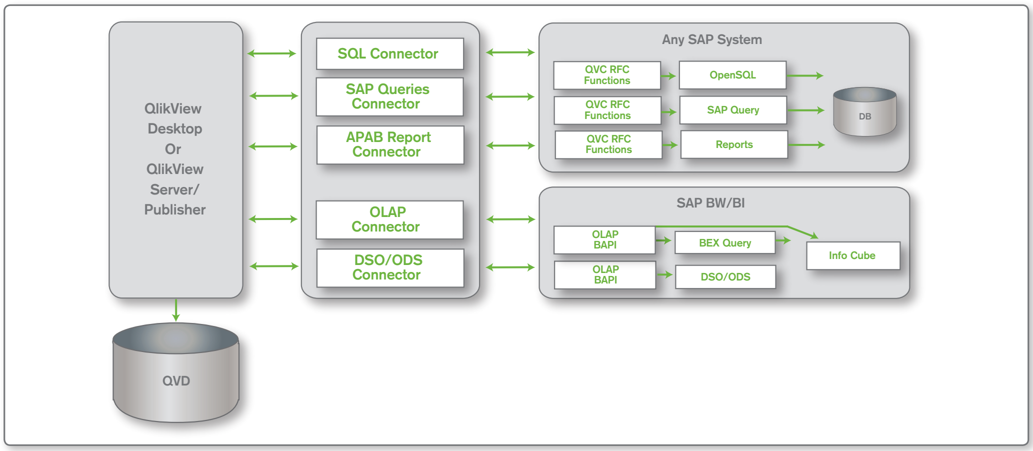


Figure 4: *SAP integration with QlikView*

READING DATA FROM WEB SERVICES, LEGACY SYSTEMS, OLAP CUBES AND MORE

It's common for data to be be stored in a location that is not reachable via ODBC/OLE, such as web service, or is in a format only accessible by a particular application such as Informatica or an OLAP source. QlikView offers a way to unlock this trapped data by using QlikView Data Exchange (also known as QVX) and can be used in two ways:

QVX AS A FILE FORMAT

Many systems can run processes to export data into several formats such as CSV or Excel. A QVX file is a file format similar to an XML file that is optimized for rapid loading of data into QlikView. The definition of how to build a QVX file is well documented to allow a third party developer to use an API to build a QVX containing data. QlikView then consumes QVX files when loading data. This essentially uses QVX to "Push" data out to QlikView. The benefit of using QVX files is that data loading times are made much shorter.

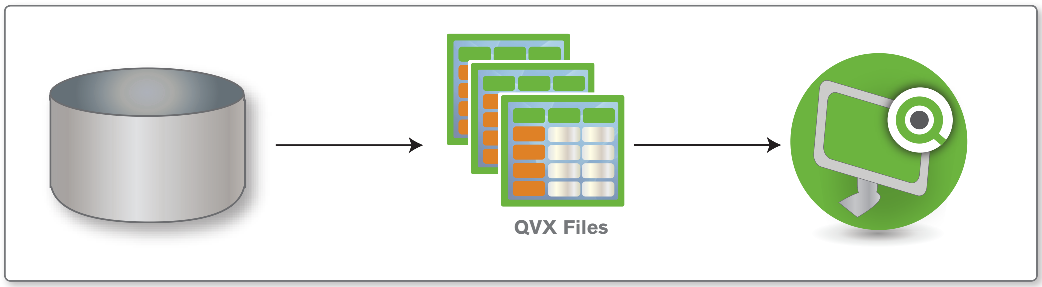


Figure 5: *QVX as a file format*

QVX AS A CONNECTOR

For systems that do not provide an ODBC/OLE or extract capability, QVX can also be used as a connector to data to allow QlikView to "Pull" data from a normally inaccessible source.

In this case a small program is created that will take a query from QlikView, translate it to a query against a source system and then return data in a QVX format. This program can be created in a number of languages and can include all the logic required to work with the APIs and services from a 3rd party system.

This enables access into existing OLAP environments and systems such as Informatica.

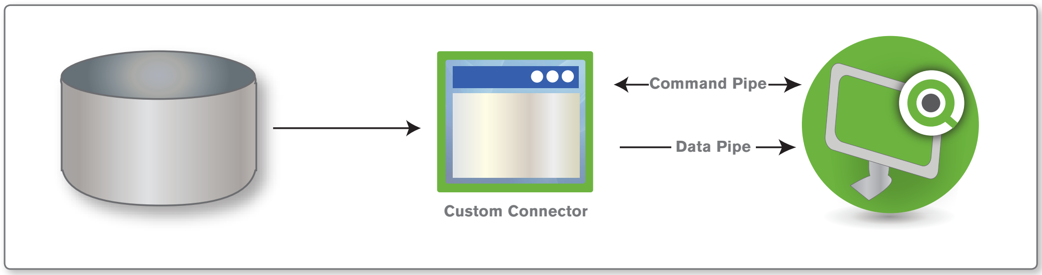
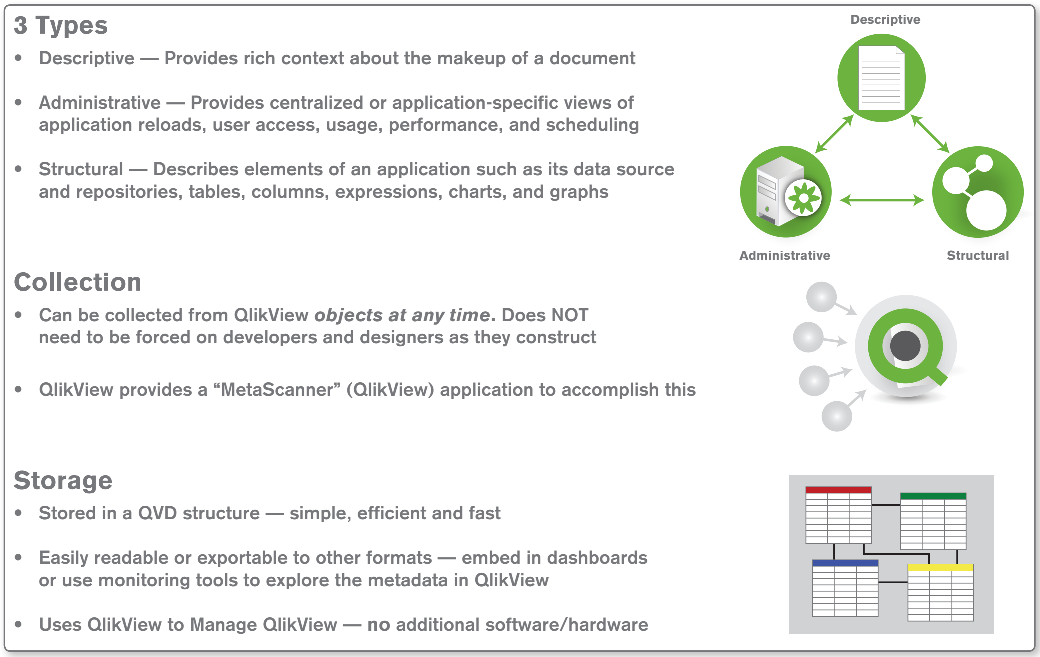


Figure 6: *QVX as a connector*

INTEGRATING META DATA WITH QLIKVIEW

The manner in which QlikView approaches metadata integration can be encapsulated with the following diagram and description. (For a more complete description and understanding of how QlikView approaches metadata, please refer to the white paper entitled "QlikView's Pragmatic Approach to Metadata")



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

Descriptive
Administrative
Structural

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

Figure 7: *QlikView metadata overview*

QlikView takes a pragmatic approach to metadata. With QlikView, metadata management is optional and retrospective and developers can introduce metadata usage over time. QlikView doesn't require a huge upfront metadata effort. QlikView handles three types of metadata—descriptive, administrative, and structural—and makes it available through a set of QVD files called the QlikView Meta Model. QlikView's metadata model is a centralized, automated collection, organization, and presentation of metadata for monitoring and distributed use within dashboards. The model is a collection of tables exported from QlikView.

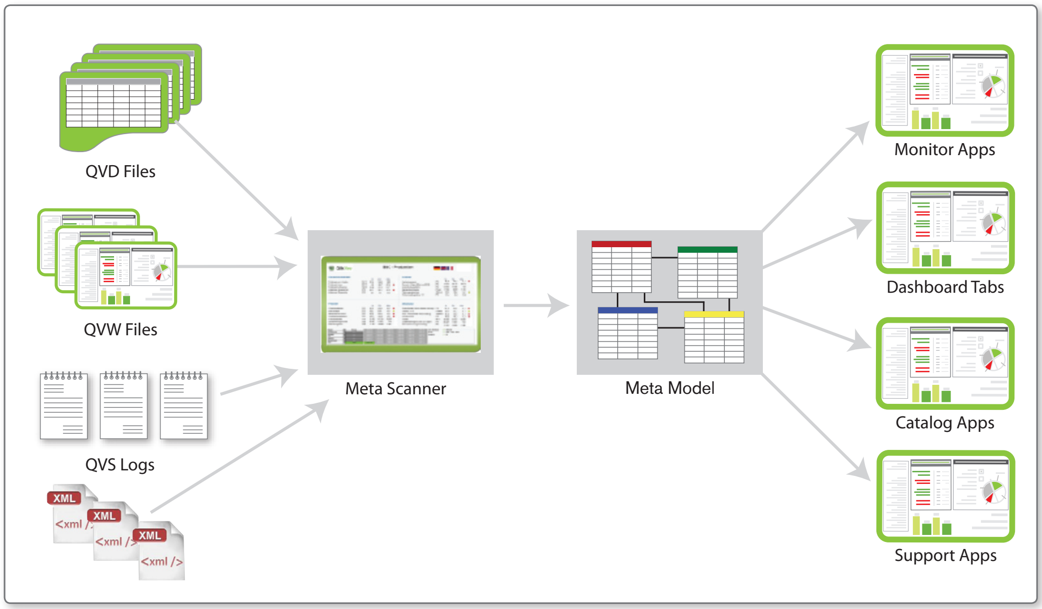


Figure 8: *QlikView MetaScanner*

Using the 'MetaScanner' application (which is itself a QlikView application), all existing QlikView files and logs in a deployment can be automatically scanned for their metadata, which is then associated in a combined metamodel (i.e. a QlikView associated data model), which then in turn can be used by custom-built QlikView applications to monitor and catalog the metadata.

This is a unique approach to metadata, moving away from the monolithic metadata repositories of the traditional BI solutions and using a streamlined, value-added collective of the most useful metadata for developers, designers, administrators and end users of QlikView.

Web Integration

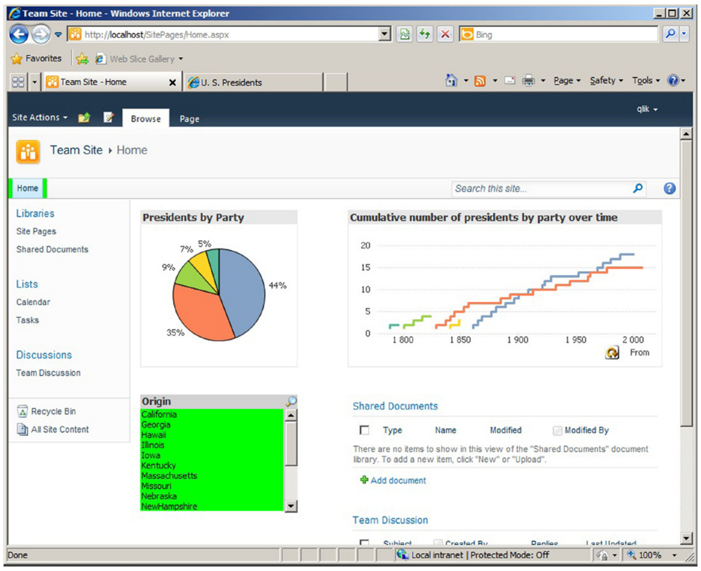
In many typical deployments of QlikView, users analyze data in isolation via a web browser and visualize data in a range of flexible ways. However, other common deployments of QlikView see QlikView visualizations and objects consumed by web sites and other applications, enhancing the ability of business users to make decisions while at the same time using other key applications and services in one environment.

Another common deployment scenario sees QlikView consuming external content and allowing users to create custom visualizations of their data using a range of technologies.

INTEGRATING QLIKVIEW WITH MICROSOFT SHAREPOINT®

Microsoft SharePoint allows developers in organizations to build portals and customize web content without needing web development skills. Developers can select from a range of pre-defined sections of content called 'web parts' and use these as building blocks to make up their web pages. For the end-user, they have a single portal or web application containing all the relevant information they need, containing a mixture of content from various source systems. One such source system can be QlikView.

QlikView supports the use of SharePoint web parts: web parts allow developers to select one or more charts or objects from a QlikView document and place them alongside their other content within the same portal. This can allow users to view their key data and KPIs from their full dashboards along with all of their other business content so they can see information without having to move from one web application to another in order to do data analysis.



There are no items to show in this view of the "Shared Documents" document library. To add a new item, click "New" or "Upload".

Figure 9: *QlikView embedded within SharePoint*

QlikView web parts remain interactive and associative with each other allowing the full power and experience of QlikView within a SharePoint intranet or portal.

INTEGRATING QLIKVIEW INTO WEB SITES USING QLIKVIEW WORKBENCH

The QlikView AJAX client is open and extensible, allowing developers to use the QlikView APIs to enhance their web sites and applications with QlikView analytics.

To help support developers to rapidly build visual, interactive analytics into their web sites, QlikTech provides the QlikView Workbench product. Workbench is a plugin to Microsoft's Visual Studio® that allows a developer to drag and drop QlikView objects into web pages written in .net with no coding effort required. This allows high levels of flexibility and control and additionally includes the ability to make QlikView objects interact with other objects and calculations present in the application.

Workbench can be used in a range of ways including showing key KPIs and analytics on a corporate intranet, providing a custom view of an individual's data and driving a powerful associative website search.

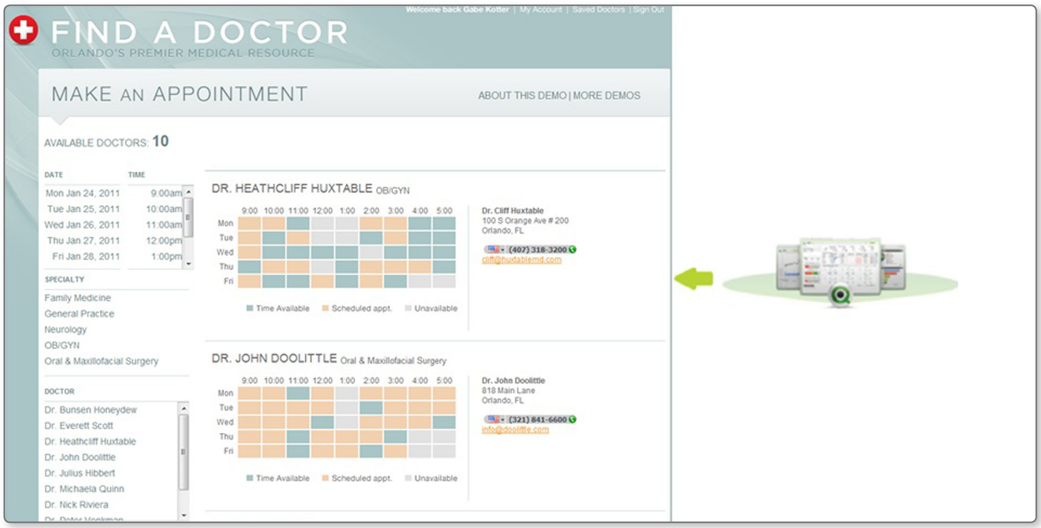


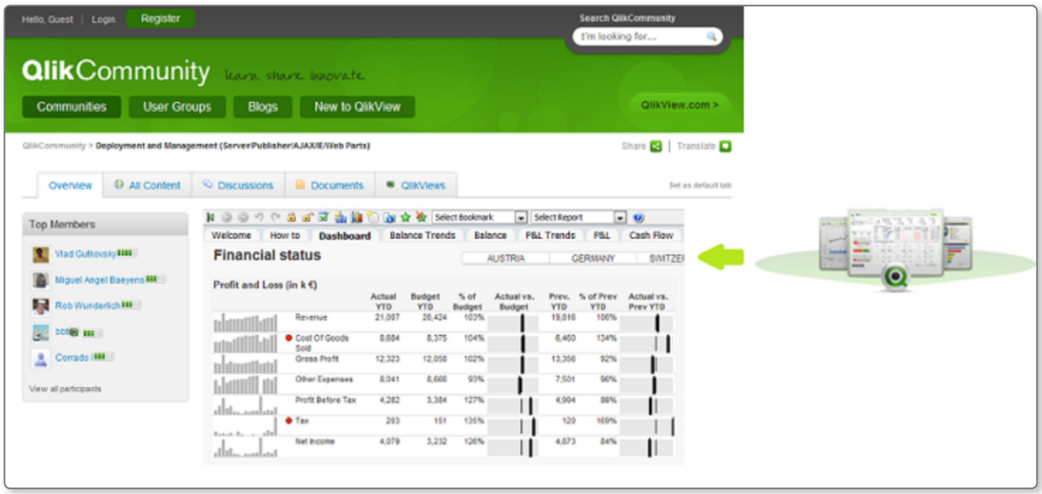
Figure 10: *QlikView embedded in a .net website*

In addition, the QlikView AJAX API's allow alternative development tools and environments to be able to consume the QlikView AJAX content. Full documentation, including examples, for this API is provided with QlikView.

DISPLAYING QLIKVIEW INSIDE OTHER WEB SITES

In many cases, displaying an entire QlikView document in the context or branding of an existing web site is required. The QlikView AJAX and plugin clients can be simply placed in a HTML container known as an iFrame, which is essentially a web page within a web page. This allows QlikView content to be integrated with any web deployment platform as no programming is required, other than the use of some standard HTML.

For example, it is possible to pass a selection state (for example a customer ID) from the parent application through to a QlikView application inside the iFrame, allowing QlikView to act on that selection state.



Financial status

Profit and Loss (in k €)

	Actual YTD	Budget YTD	% of Budget	Actual vs. Budget	Prev. YTD	% of Prev YTD	Actual vs. Prev YTD
Revenue	21,007	20,424	103%		19,816	106%	
● Cost Of Goods Sold	8,684	8,375	104%		6,460	134%	
Gross Profit	12,323	12,050	102%		13,356	92%	
Other Expenses	8,041	8,666	93%		7,501	96%	
Profit Before Tax	4,282	3,384	127%		4,994	86%	
● Tax	203	151	135%		120	169%	
Net Income	4,079	3,232	126%		4,873	84%	

***Figure 11:** : QlikView embedded in a website using iFrames*

CREATING CUSTOM VISUALIZATIONS IN QLIKVIEW

QlikView provides a range of visualization objects out of the box, but there are many cases where customers want a customized way of displaying their data or displaying content from another system. Some examples include maps, Gantt charts, or domain specific visualizations.

In QlikView a capability called Extension Objects allows a developer to develop their own ways of visualizing their data but still making use of the powerful associative data engine in QlikView. Developers can choose from a range of web technologies to build an extension object such as Flash, Javascript or plain HTML, and have these objects interact within a QlikView document just like the native QlikView charts and objects would.

INTEGRATION USING THE QLIKVIEW ACTIVEX OCX

The QlikView OCX is an ActiveX component containing the QlikView application that can be embedded into host application programs developed by 3rd party software manufacturers. It is provided on an OEM basis to provide a ‘white labeled’ version of QlikView.

This component provides full QlikView UI functionality and can be controlled and manipulated by the host application. The host application can be either from a web site or a desktop application.

Security Integration

Security is a critical element of any enterprise software deployment. Systems need to be manageable by IT professionals and need to conform to existing IT security standards and infrastructure.

QlikView is able to integrate with a series of standard security systems to authenticate and authorize users including Single Sign On (SSO) systems, Active Directory, other LDAP's and so on, allowing for integration with existing web site and portal security.

The topic of QlikView security is comprehensively covered in the QlikView Security Overview Technology White Paper but it's appropriate to cover security in relation to web integration here.

In most cases where QlikView needs to be integrated into a web site or portal the key objective is that the user should only occur once. For example, if the user has already signed in with their email address and password then they shouldn't need to provide further login details to see QlikView content. There are, of course, many ways a user can login to a web page but below are three of the most common ways. With QlikView, these may often be combined.

WINDOWS CREDENTIALS

On many corporate networks a user will sign into their PC with an Active Directory account and when they browse to web applications within their network the same login state will automatically and invisibly authenticate them with the web site. QlikView in this scenario can utilize this capability out of the box and is the default mechanism in use when using QlikView in an iFrame, SharePoint Web Parts and applications developed with QlikView WorkBench. Users can view QlikView content inside of other Windows-authenticated applications and be unaware they are using resources from two different services.

USING SINGLE SIGN-ON (SSO) INFRASTRUCTURE

SSO is a concept where a collection of services and applications can use a common sign in mechanism to log users on. It can mean that a user still logs into to every application individually with the same account but in most cases one would want a single login process to occur and then for any resource a user views, the same "login state" be used automatically.

Many web sites and portals use the 'sign in once' approach and there are many SSO products that can achieve this is different ways. Examples include Tivoli Access Manager, Microsoft ISA/ TMG and SiteMinder.

This type of deployment is used in extranet and internet scenarios but also used within large organizations for internal SSO systems. From a user perspective they will simply login, perhaps using an email address and password specified on a web page, and can then access a range of resources.

QlikView can integrate with these systems to be seamlessly displayed to a user without the need for a further login. In many cases SSO systems will be implemented as a Reverse Proxy or gateway that handles the login process and requests content on behalf of a user. When using this approach the SSO reverse proxy will pass the user's details within a HTTP header along with the request for content such as QlikView content. QlikView can use this user HTTP header to log the user in and grant access to authorized applications and grant a user license.

Please refer to the QlikView Security Overview white paper for further information

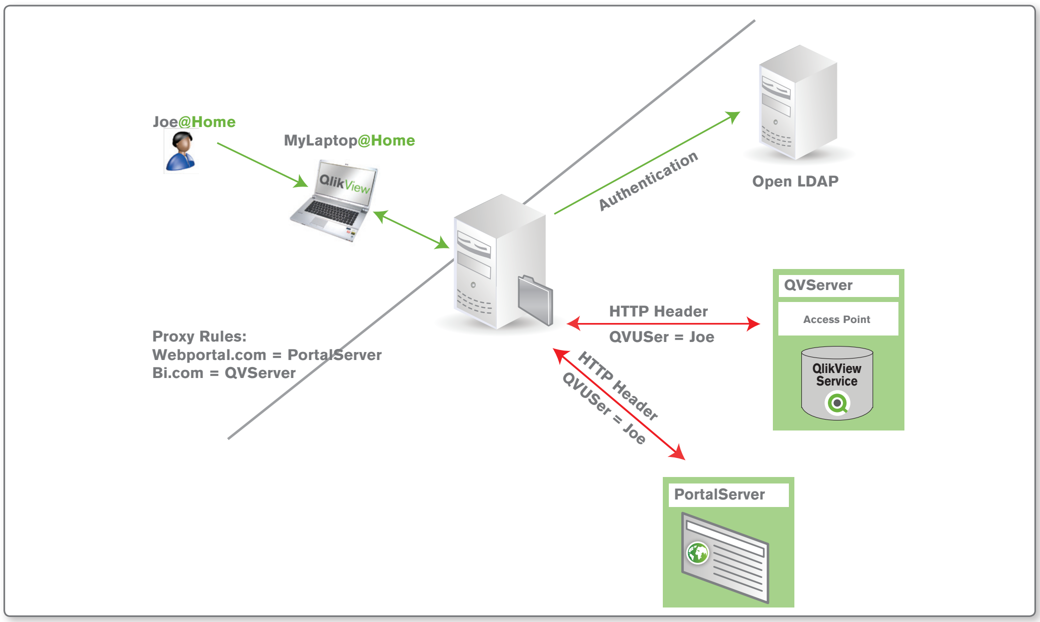


Figure 12: *Perimeter authentication using HTTP Headers*

PASSING SECURITY CONTEXT FROM A WEB APPLICATION (TICKETING)

When SSO infrastructure is not in place then many web sites will use application level authentication: i.e. the developers will build a web page to take login and password and establish a user session. This session is only applicable within the scope of that application and cannot be used elsewhere.

QlikView can integrate with these scenarios by using a method called ticketing. In this case the application has pre-authenticated a user in some way and may provide a link for the user to click on when they want to see QlikView content. The application, rather than the user, then communicates with the QlikView server to request a session ticket for the user. QlikView will then return a long unique key back to the application. The application will then redirect the user to the URL of the QlikView server adding the ticket to the end of the URL.

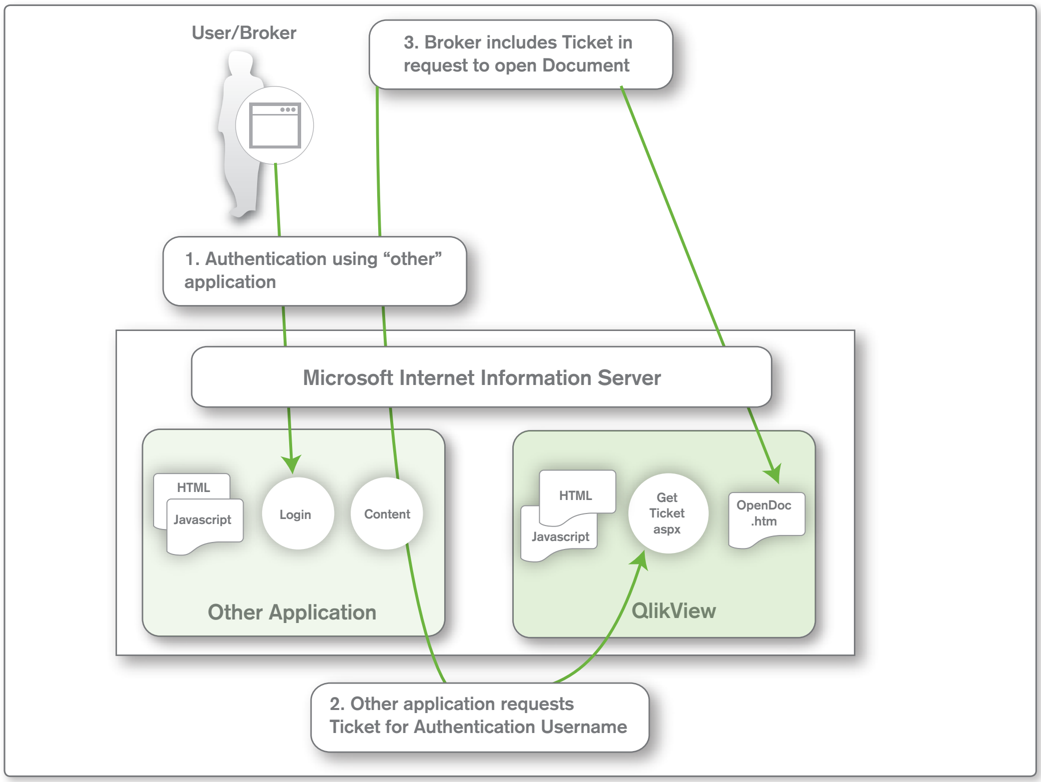

User/Broker
3. Broker includes Ticket in request to open Document
1. Authentication using "other" application
Microsoft Internet Information Server
HTML
Javascript
Login
Content
Other Application
HTML
Javascript
Get Ticket aspx
OpenDoc .htm
QlikView
2. Other application requests Ticket for Authentication Username

Figure 13: *Authentication using Ticketing*

When QlikView sees the request it will use the ticket to identify the user and their actual username and then display authorized QlikView content to them.

Ticketed authentication is most applicable to embedding QlikView content in 3rd party applications and portals, and rarely used for providing general access to QlikView.

Please refer to the QlikView Security Overview white paper for further information.

Integration with 3rd Party Systems and Processes

When an organization wants to implement any new system a key factor in choosing a product is the effort and cost of running it day to day.

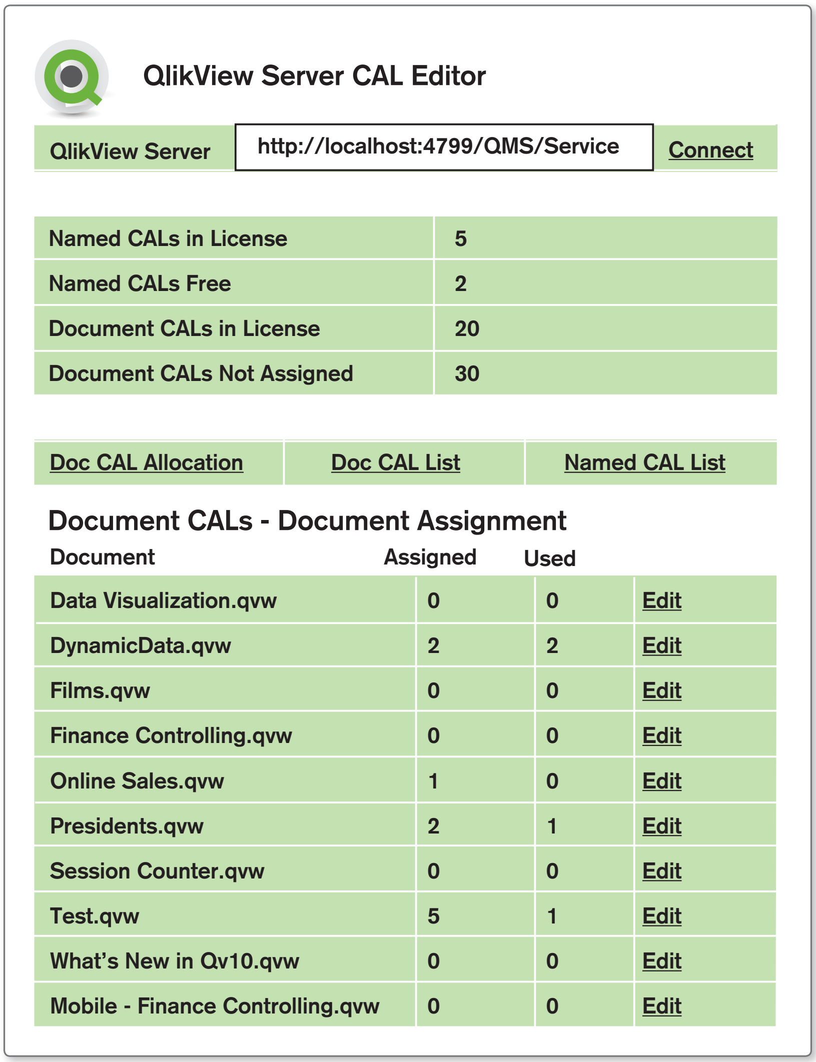
QlikView helps organizations reduce reporting costs by allowing them to rapidly develop dashboards and applications and to be flexible in dealing with business changes and demands. Additionally QlikView allows organizations to run their environment efficiently by offering a range of capabilities such as the ability to automate common tasks and integration with systems management software. This results in low maintenance time and costs for the QlikView platform, allowing IT teams to focus on other key activities or enhancing reports.

MANAGEMENT INTEGRATION

With any piece of software there are a number of common tasks that need to be performed on a regular basis. In QlikView these tasks may include adding permissions or a license for a new user, or adding new documents and applying common settings.

QlikView provides an intuitive web based management console for performing all of these tasks but when changes need to be applied on a large scale then automation of these is required. To achieve this QlikView provides programmatic control of common management tasks via a web service API. This allows administrators to automate common tasks and apply customized business logic as required using a range of technologies.

An example of management automation includes extending the scope of an employee on-boarding system to automatically add permissions and a license for QlikView or to ensure a license for a departing employee is freed up for a new user. This would remove the need to manually perform this task. Alternatively, creating a function as below to allow for licenses to be managed more efficiently depending on different customer scenarios is another example.



QlikView Server CAL Editor

QlikView Server	http://localhost:4799/QMS/Service	Connect

Named CALs in License	5
Named CALs Free	2
Document CALs in License	20
Document CALs Not Assigned	30

Document CALs - Document Assignment

Document	Assigned	Used	
Data Visualization.qvw	0	0	Edit
DynamicData.qvw	2	2	Edit
Films.qvw	0	0	Edit
Finance Controlling.qvw	0	0	Edit
Online Sales.qvw	1	0	Edit
Presidents.qvw	2	1	Edit
Session Counter.qvw	0	0	Edit
Test.qvw	5	1	Edit
What's New in Qv10.qvw	0	0	Edit
Mobile - Finance Controlling.qvw	0	0	Edit

Figure 14: *Example of custom-built license manager using QlikView Management APIs*

AUTOMATION & SCHEDULING

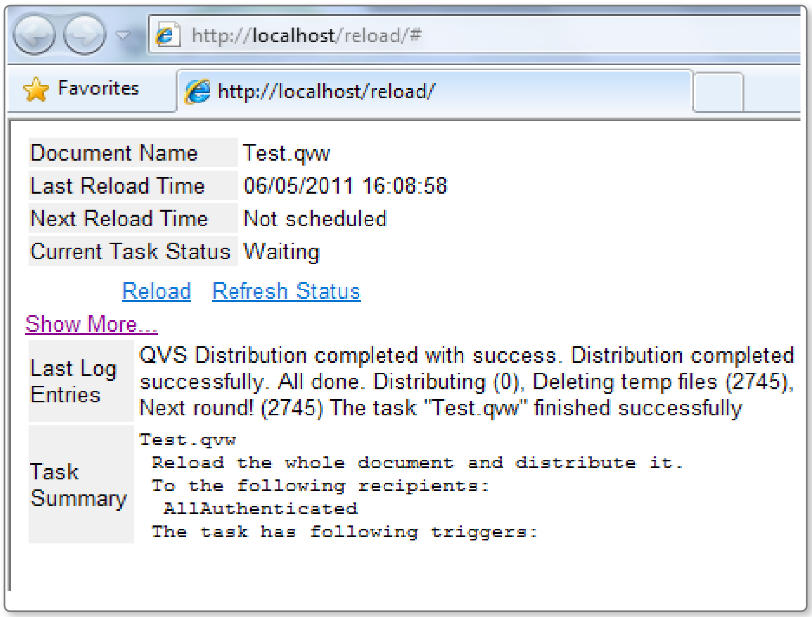
Keeping data up to date in QlikView requires a reload of a document and this can be scheduled to run as often as required using either QlikView Publisher. Often before loading data into QlikView a number of activities may be required to prepare the data; such as data cleansing or retrieving data from remote locations across the globe.

Depending on the scale of this "batch" of activities a control and scheduling system such as BMC Software's Control-M® and IBM's Tivoli®, or a process such as SQL Integration Services may be used to ensure that all the required activities are completed. Rather than simply schedule a QlikView reload it can instead be triggered by an external system as an additional step in a batch process. This ensures that QlikView users can view the most recent and reliable data.

QlikView has a feature that allows for tasks to be triggered: Event Driven Execution (EDX) allows a single reload to be launched and then polled for a completion status.

The QlikView Management API allows for custom business logic to be applied when working with tasks including the ability to create, modify and delete tasks as required.

This same functionality can be used in a range of ways beyond just scheduling, such as creating web pages to allow users to self serve reloads but with controls and security placed around this.



Document Name	Test.qvw
Last Reload Time	06/05/2011 16:08:58
Next Reload Time	Not scheduled
Current Task Status	Waiting

Reload   Refresh Status

Show More...

Last Log Entries	QVS Distribution completed with success. Distribution completed successfully. All done. Distributing (0), Deleting temp files (2745), Next round! (2745) The task "Test.qvw" finished successfully
Task Summary	Test.qvw Reload the whole document and distribute it. To the following recipients: AllAuthenticated The task has following triggers:

Figure 15: *Example of custom reload scheduler using QlikView Management APIs*

MANAGING DOCUMENTS AND SOURCE CONTROL

With any development project the ability to manage and control changes to applications is important to know what changes were made, when and by whom and also to allow multiple developers to work on applications at the same time.

A single QlikView file (.qvw) is a binary file containing a load script, data and multiple objects in the UI. Two versions of a binary file cannot be easily compared to one another to see differences and so QlikView supports two methods for overcoming this and allows developers to track and manage changes.

SOURCE CONTROL WITH PROJECT FILES

The Project Files feature is a function that "explodes" a single .qvw into multiple XML & text files. Each resulting file is the definition of an object in the QVW such as a chart, list box or the load script. Every time a QVW is opening in the developer client the objects are imported, and each time the document is saved the changed objects are then over-written.

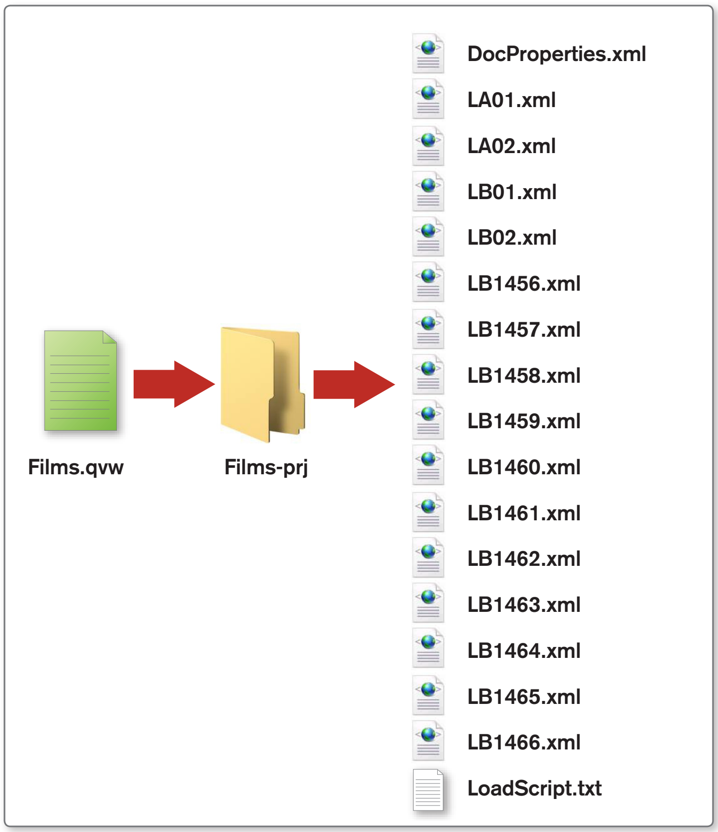


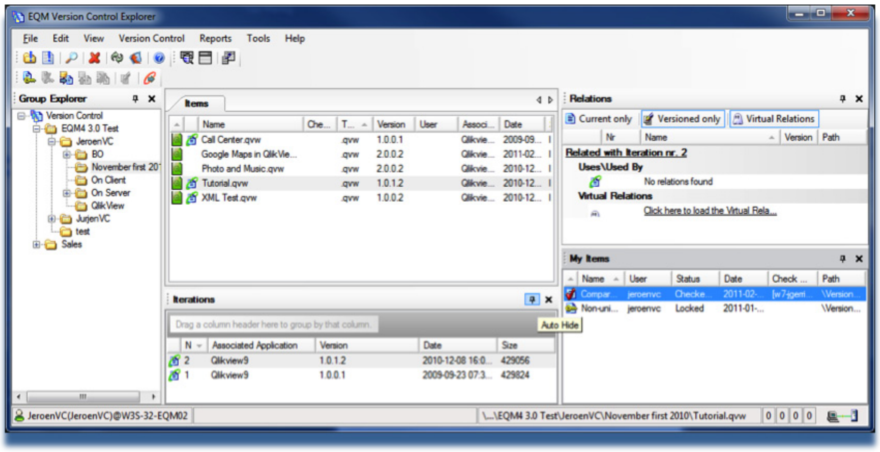
Figure 16: *QlikView project files*

These projects files can then be checked in and out of source code management system such as Apache Subversion® or Microsoft TFS so that the version history of individual objects can be retained and multiple developers can work on different sections of a document.

With this approach the data in a QVW is not exported so that the storage requirement in the source control system is kept to a minimum.

VERSION CONTROL WITH NOAD EQM4

NOAD is a QlikView partner whose product, EQM4, is an integrated version and deployment control solution for enterprise QlikView deployments. Once QlikView documents are imported into EQM4, developers can easily check-out one or more documents to their personal workspace, perform some additional development on the application and when finished, the check-in function copies the changed version from the user's workspace and stores it inside the EQM Version Control repository.



Name	Che...	T...	Version	User	Associ...	Date
Call Center.qvw		.qvw	1.0.0.1		Qlikvie...	2009-09...
Google Maps in QlikVie...		.qvw	2.0.0.2		Qlikvie...	2011-02...
Photo and Music.qvw		.qvw	2.0.0.2		Qlikvie...	2010-12...
Tutorial.qvw		.qvw	1.0.1.2		Qlikvie...	2010-12...
XML Test.qvw		.qvw	1.0.0.2		Qlikvie...	2010-12...

Drag a column header here to group by that column.

N	Associated Application	Version	Date	Size
2	Qlikview9	1.0.1.2	2010-12-08 16:0...	429056
1	Qlikview9	1.0.0.1	2009-09-23 07:3...	429824

Nr	Name	Version	Path

Related with Iteration nr. 2

Uses\Used By

No relations found

Virtual Relations

Click here to load the Virtual Rela...

Name	User	Status	Date	Check ...	Path
Compar...	jeroenvc	Checke...	2011-02-...	[w7-jgem...	\Version...
Non-uni...	jeroenvc	Locked	2011-01-...		\Version...

JeroenVC(JeroenVC)@W3S-32-EQM02 | \...\EQM4 3.0 Test\JeroenVC\November first 2010\Tutorial.qvw | 0 | 0 | 0 | 0

Figure 17: *NOAD EQM4 Version Control*

While QlikView developers are working on a new version of the QlikView application, end-users can continue using the production version of that same QlikView application without any risk of having untested changes disturbing their analysis.

Change and version history can be reviewed as well as seeing who made changes and when, and includes the ability to see the granular differences between two files.

DEPLOYMENT CONTROL

EQM4 Deployment Control enables easy rollout of multiple QlikView applications to one or more environments/QlikView Servers using a single action. For example, assume a 3 stage implementation having Development, Test and Production.

When the document is ready to be tested the change manager can deploy it to the test environment(s) immediately, or schedule it for another time.

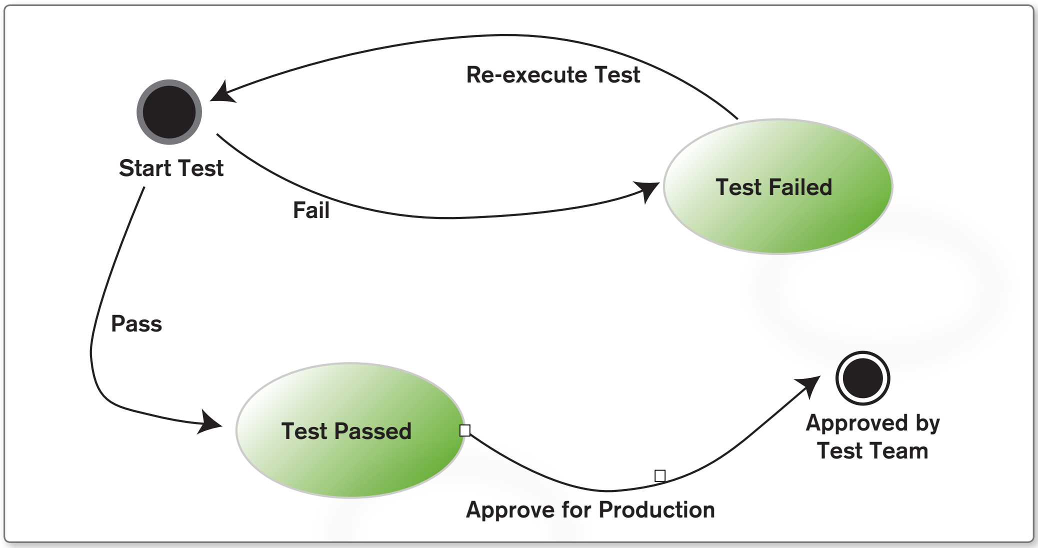


Figure 18: *QlikView deployment control workflow using EQM4 Deployment Control*

Once the QlikView document has arrived in the test environment the test manager picks it up and starts evaluating/testing it. EQM4 helps the test manager by providing reports to scope the test project. Difference analysis is a very important capability to identify the most important changes between the newly developed QlikView document and the version that's currently in use in production. Thus the test manager can focus on testing only the changed parts of the QlikView document, reducing the risk of not testing parts that actually have changed.

When the QlikView document works according to the specifications the test manager approves it and the QlikView document's state changes to 'Test Passed'. The change control manager can now approve the document for production immediately or schedule it for another time.

Conclusion

Integrating QlikView into existing IT architectures is an important element to a successful QlikView deployment, regardless of the size of the deployment. Having a fundamental understanding of each of the most important integration points, namely data integration, web-deployment integration, security integration and IT process integration will ensure that the overall deployment success for end-users and management efficiency for IT administrators will be greatly increased.