QlikView

QlikView Migration Guidelines

Contents

QlikView Migration Guidelines 1
- Summary 2
  - Roles 2
  - Definitions 2
  - Definitions of changes 2
- Overview 3
  - Communication Process 3
  - Release Strategy 3
- Details for Developers 3
- Details for Administrator 5
    - General principles for all tasks: 5
      - Admin Responsibilities 5
      - Admin Responsibilities do not include 6
    - Merging in new Reports (Layout changes) 6
    - Merging in new Script 7
    - Modifications to Publisher 7

QlikView

Summary

This is to describe the migration path of changes to production QlikView documents from the developer/professional users into production. This will need to be a coordinated effort between several people to make the process work correctly.

Roles

- Administrator – In charge of:
    - Publisher nightly jobs (Creating, modifying, maintaining)
    - Integrating New changes from Developers/Professional users into Production
    - Acting as a gateway for changes (source control)
- Developer
    - Name for either the Developer and Professional User
    - Developer - This is generally a person who develops the data model and works with the source systems (typically members of the Jackson IT group)
    - Professional User - This is generally a person who creates/modifies the charts/reports and layout (typically members of the NY BI group)
    - In this document, both the developer and professional users have the same process and are only working on different content.
- QA Users
    - This is the end user who help define report requirements and are the final validation that the reports are correct. This could be the developer or a person outside of the development team.

Definitions

- Document – A QlikView QVW file. AKA 'QVW'
- Production Candidate – A document which has all of the latest changes applied to it. This is generally a document which is in the hands of the administrator. It's the administrators responsibility to keep track of which document is the production candidate and there should only be one production candidate per document at any time.

Definitions of changes

Changes would include anything which requires a developer to open the document, edit it and save the changes. This includes changes in layout (moving things around), changing colors, adding/modifying properties on anything (list boxes, charts, graphics, etc), changes in scripting, cycle and drill groups, macro code.

This reference of changes do not include updating the data nightly through an automated process (such as publisher)

QlikView

Overview

The basic process is that a developer will make a few changes to the layout of the document and want to merge those changes into production. Once the developer has verified the changes are correct, they will notify the administrator. The administrator will integrate the changes from the developer (and possibly multiple developers) and put the document in QA for the developer to review. If the developer and end user has verified the document is correct, then the administrator can move the changes into production.

If a large number of changes are occurring, the administrator may (in coordination with the all parties involved) develop a release schedule and release many changes at once.

End User Communication Process

- A Process will need to be developed to notify end users users that new features are available in a specific application. In other words, this should be a process to notify the end users that new features are available once the development has been finished and the application is promoted to production. This could vary depending on the size of the change. If it's a simple new report, then no notification may be necessary. If it's a change to a report which is used all of the time by many people, then you may want to notify everyone with email or have a training session with the changes.

Release Strategy

- If there will be several changes from multiple developers going into production on the same document, then a release strategy should be developed to coordinate the changes to minimize the number of promotions to production and additional overhead of promoting new features to production (QA, administrative time to merge changes, possible chance for errors during promotion, etc)

Details for Developers

These details are the general steps each developer would take to develop new changes and have them added to production[1].

Once a developer receives a request to develop a change to a production document (see document "Development Server.doc") there are 2 ways to proceed

1. If there are a large number of changes which affect a large number of areas of the document
   a. The developer or professional user would contact the administrator and request the most current version of the document be given to the developer for exclusive access.

---

[1] *The Developer Details process is the general design if a developer will be working on changes with no help from any other developers. If multiple developers need to work on changes together they will need to coordinate with each other and define what changes need to happen and work out a way for that particular development task to be accomplished in coordination with each other. Because of the nature of how multiple people would work together and the number of different changes which are possible, each instance will require a unique effort, so a framework shouldn't be dictated*

This prevents any other changes to happen to the document while the developer is developing new changes. Since QlikView has no change management features, granting exclusive access to a document is completely a manual effort by the administrator.

b. The developer would make the changes by
   i. Copying the document which the administrator gives you to your working directory
   ii. Making the changes to the document
   iii. Copying the new document to the e:\QlikView Development Server Directory using the appropriate name (See the "Development Processes.doc") to allow QA users to validate the changes.
   iv. When the QA users have validated the changes, return the document to the administrator with
      1. A general list of changes. (An all-inclusive list of changes wouldn't be necessary since the administrator would take the document from the developer, validate some standards (See Details for Administrator Section of this document) and put the document directly into production.)
      2. A list of QA users who should QA the document once it's placed into the QA directory. The administrator will grant access to the document for the specific list of QA users.
c. The administrator would put the document into production after some basic checking that the application work correctly in the production environment and that the proper standards are being met (script isn't broken, all of the reports work correctly)

2. If there are small changes, or the changes are limited to a specific subset of the document
   a. The developer would copy a version of the document from the "E:\QlikView Copy of Production" to their working directory and develop changes (See the "Development Processes.doc"). It's very important to keep a list of all of the changes which were made[2].
   b. The developer must copy the document to the directory "e:\QlikView Development Server Directory" (On the Development Server) using the appropriate name (See the "Development Processes.doc") and allow QA users to validate the changes.
   c. At this point the QA user would validate the changes and the developer would notify the administrator of
      i. Where the document is located
      ii. The name of the document (exact file name)
      iii. Changes the administrator would need to merge into production

[2] The reason it's important to keep a list of all of the changes is that it will give the administrator a list of changes so the administrator doesn't have to search the entire document and every checkbox for any change to the document.

iv. Which users should see the document once it's placed into QA

d. The administrator would merge the changes of one or more developer's changes into a production build; place the build in the "E:\QlikView QA Server Directory" for QA by the developer and QA users and notify the developers and QA users of the name of the document and that it's ready for QA (The name could be different than what the developer had originally since the administrator may have merged in changes from several developers)

e. Once the developer (and optionally the QA user) has had time to QA the document, the administrator would promote it into production

Details for Administrator

Your role will encompass keeping the production environment organized and also to merge new changes into production.

These changes include changes to script, reports and Publisher jobs. To perform this role, you will need to have a firm grasp on all of the QlikView products since you will be managing the flow of data through all of them.

General principles for all tasks:

- Traditional SDLC practices should apply where the development process goes through a Development – QA – Production sequence to promote changes to production. Since QlikView doesn't have features for showing the difference between 2 different documents, comparing changes and promoting changes to production is a manual process.

- When merging in new changes from developers you'll need to make sure you keep track of which document has all of your most recent changes and merge developer's changes into that most recent document.

- You will need to avoid having multiple versions of the same document in QA at the same time. The reason for this is that if you have multiple versions of the same document in QA then when you migrate the changes to production you will have to re-merge the changes which means you need to change things which means additional QA (since changes could affect other changes)

Admin Responsibilities

- Coordinate changes into production
- Fix system problems
- Operate with good software design principles (backups, quality checks, etc)
- Enforce coding standards/practices
- Coach developers as to design principals to promote a better system

- Monitoring/tuning the QlikView Server
- Monitor changes to security groups
- Managing User CALs
- Raising issues with navigation (although they could be overruled if there is a business need)

Admin Responsibilities do not include

- Validating the numbers/data for accuracy
- Managing the QlikView projects
- Deciding what 'looks' good

Merging in new Reports (Layout changes)

When a developer notifies you of changes they should provide you with

- A location and name of the document which has the changes for you to promote to production
- A list of the changes. These changes could include
    - Layout objects (Graphs, reports, list boxes, etc)
    - Variables which the reports reference
    - Cycle and Drill objects
- A list of people who will QA the document

The process is to have a production candidate version of the document which you will work with and the newly enhanced document which the developer gave you (possibly multiple developers)

- The admin would go through the list of changes and move the changes to the new production candidate Many times this can be a simple copy/past of the new object to the production candidate and deleting the old object.
- Once you migrate the changes to the new document you will need to validate that it looks correct.
    - Check all fields – In charts and graphs, you'll want to check all fields to make sure they exist in the production candidate. Any fields which don't exist will generally turn red, or have a red X next to them in the chart properties (if it's a chart).
- Check all expressions (expressions everywhere, not just in the expression tab)
    - Check that they are valid
    - Check that they are efficient (additional data model changes may be necessary if the expressions could be made to be more efficient)
- Drill/cycle groups – There is no way to copy/paste these, they will need to be rebuilt. Also look at the sort order in the groups, the sort order could be important to the developer
- Possibly any other supporting architecture such as Island Tables, new fields, etc.

Merging in new Script

- Identify the new script and validate the script is going to work with the coding standards which have been adopted.
- Make sure it executes correctly.
- Merge the new script into the existing script. It may be possible to merge the new script with some existing script to improve efficiency instead of continually adding new script to the document

Modifications to Publisher

Users won't be giving you actual publisher items to move into production. Instead, publisher changes will come with the requirements of other enhancements.

For example, if a new request is made to reload data on a new schedule, that would be a publisher change which would need to be executed. If a new script module was added which utilized additional documents, publisher would have to be adjusted to accommodate the additional documents.

Publisher changes will be up to your discretion as to what needs to be accomplished and what timing they need to be accomplished. There are a couple of guidelines which you'll want to follow though:

- Don't 'Distribute' large documents to the QlikView Server Resource during times when users are utilizing the system. Doing this will slow the QlikView Servers dramatically Small documents (which take 30 seconds or less) are fine.
- When validating new data model changes (script changes and such), it's best to develop the publisher jobs and let them execute from publisher to validate that the data movement is occurring as it should. Without doing this, you won't know if publisher has the proper permissions and such defined to execute the jobs in their actual production environment

Development Process