Spicers

Design Overview

Date: 19/06/2015

Version: 1.0

|  |  |
| --- | --- |
| Company: | OfficeTeam/Spicers |
| Project: | Spicers QlikView Dashboards |
| Application Owner: |  |
| Application Title: | Spicers |

# Contents

[Contents 2](#_Toc422494126)

[Purpose 3](#_Toc422494127)

[Project Overview 3](#_Toc422494128)

[Source Systems 3](#_Toc422494129)

[Deployment & Security 3](#_Toc422494130)

[Solution Architecture 4](#_Toc422494131)

[QVD Creations.qvw 4](#_Toc422494132)

[Data Model – Spicers.qvw 6](#_Toc422494133)

[Data Model Diagram 7](#_Toc422494134)

[Data Modelling Transformations 7](#_Toc422494135)

[Spicers.qvw 8](#_Toc422494136)

[Variables 8](#_Toc422494137)

[Common Expressions 9](#_Toc422494138)

[User Interface 10](#_Toc422494139)

[Dashboard 10](#_Toc422494140)

[EU Category 11](#_Toc422494141)

[RDC 12](#_Toc422494142)

[Credits 13](#_Toc422494143)

[Comparative 14](#_Toc422494144)

[Ad Hoc 15](#_Toc422494145)

[Scheduled Tasks 16](#_Toc422494146)

[Data Load Details 16](#_Toc422494147)

[Loop and Reduce 18](#_Toc422494148)

# Purpose

The purpose of this document is to provide an overview of the structure and development for the Spicers QlikView application.

# Project Overview

Provide data to the business in a quick and intuitive manner to allow data interrogation without the requirement to distribute multiple spreadsheets.

Data should include performance vs budget as well as some trending analysis.

Use the QlikView Distributer to create data models based on Region/Team

## Source Systems

* Sales Invoice Data – (Transactions [tbl\_Operating\_Profit\_])
* Sales Budget Data - (Targets [tbl\_CustomerMargin])
* Customer Data – (Customers [tbl\_Master\_Customer\_Details])
* Region Data – (\*\*Gap\*\*)

## Deployment & Security

Access to the internal application is controlled using Active Directory permissions. The Following Users/Groups require access to this application:

Solution Architecture

A 3 layered approach has been taken, splitting up the data extract, load/transformation and front end design.

## QVD Creations.qvw

The QVD generator is designed to extract tables from a database and store them into QVDs for later use within the QlikView environment. The many purpose and reason behind using QVDs include:

**Increasing Load Speed**

By buffering non-changing or slowly changing blocks of input data in QVD files, script execution becomes considerably faster for large data sets.

**Decreasing Load on Database Servers**

The amount of data fetched from external data sources can also be greatly reduced. This reduces work load on external databases and network traffic. Furthermore, when several QlikView scripts share the same data it is only necessary to load it once from the source database into a QVD file. The other applications can make use of the same data via this QVD file.

**Consolidating Data from Multiple QlikView Applications**

With the Binary script statement it is possible to load data from only one single QlikView application into another one, but with QVD files a QlikView script can combine data from any number of QlikView applications. This opens up possibilities e.g. for applications consolidating similar data from different business units etc.

**Incremental Load**

In many common cases the QVD functionality can be used for facilitating incremental load, i.e. exclusively loading new records from a growing database.

Data flow for QVD Creations.qvw:

QVD

QVD

QVD

QVD

QVD Creations

**QVW**

SQL Server

**QVDs**

* tbl\_Master\_Delivery\_Types
* tbl\_Master\_CustomerNo\_AccountNo
* tbl\_Master\_Coda\_Cost\_Centres\_Spi
* tbl\_CustomerNumbers\_BusinessOwne
* tbl\_Master\_Coda\_Nominal\_Codes\_Sp
* tbl\_Master\_Coda\_Cost\_Centres\_Spi
* tbl\_Master\_Coda\_Nominal\_Codes\_Sp
* tbl\_Cost\_Centres
* tbl\_Discounts\_Customer\_Standard
* tbl\_Discounts\_Product
* tbl\_Master\_Account\_Details
* tbl\_Master\_Boss\_Level1\_Code\_Desc
* tbl\_Master\_Boss\_Level2\_Code\_Desc
* tbl\_Master\_Composite\_Codes
* tbl\_Master\_Credit\_Codes
* tbl\_Master\_Customer\_Details
* tbl\_Master\_Customer\_Number\_Chang
* tbl\_Master\_Customer\_Specific\_Sal
* tbl\_Master\_CustomerNo\_AccountNo
* tbl\_Master\_Dates
* tbl\_Master\_Dates\_WorkingDays
* tbl\_Master\_Dealer\_Groups
* tbl\_Master\_Delivery\_Charge\_Postc
* tbl\_Master\_Delivery\_Types
* tbl\_Master\_Despatch\_Codes
* tbl\_Master\_Despatch\_RDC
* tbl\_Master\_Dummy\_Division
* tbl\_Master\_Estimated\_Customer\_Re
* tbl\_Master\_EU\_Category\_Descripti
* tbl\_Master\_EU\_Product\_Sector\_inc
* tbl\_Master\_EU\_Product\_Sub\_Sector
* tbl\_Master\_Excluded\_Inter\_Co\_Cus
* tbl\_Master\_Invoice\_Types\_Focus
* tbl\_Master\_Logistics\_Full\_Standa
* tbl\_Master\_Matrix\_Pricing\_Band\_D
* tbl\_Master\_Mth\_Yr\_to\_Period
* tbl\_Master\_Order\_Status\_Definiti
* tbl\_Master\_Period\_Month
* tbl\_Master\_Price\_Indicator\_Descr
* tbl\_Master\_Product\_All\_Codes
* tbl\_Master\_Product\_Brands
* tbl\_Master\_Product\_Catalogue\_Lis
* tbl\_Master\_Product\_Category\_Desc
* tbl\_Master\_Product\_Division
* tbl\_Master\_Product\_VAT\_Rate
* tbl\_Master\_Product\_Number\_Delive
* tbl\_Master\_Promotion\_Products
* tbl\_Master\_Promotions
* tbl\_Master\_Purchase\_Class\_Descri
* tbl\_Master\_Range\_Group\_Descripti
* tbl\_Master\_Retro\_All\_Agreements
* tbl\_Master\_Spicer\_Range\_Descript
* tbl\_Master\_Selling\_Days
* tbl\_Master\_Std\_Delivery\_Chgs
* tbl\_Master\_Synergy\_Members
* tbl\_Master\_TableIDs
* tbl\_Master\_TableIDs\_for\_Pricing\_
* tbl\_Master\_Vendor\_Rates
* tbl\_Master\_Vendors
* tbl\_TableIDs\_with\_CustomerNumber
* tbl\_Working\_Days\_Calender
* qry\_Logisitics\_Full\_Data
* tbl\_Operating\_Profit

## Data Model – Spicers.qvw

The data load and transformation layer where QVDs are loaded and the transformation and joining of tables take place to create the data model.

Data Model - Spicers

**QVW**

TBF QVDs

QVD

TBF QVDs

QVD

QVD

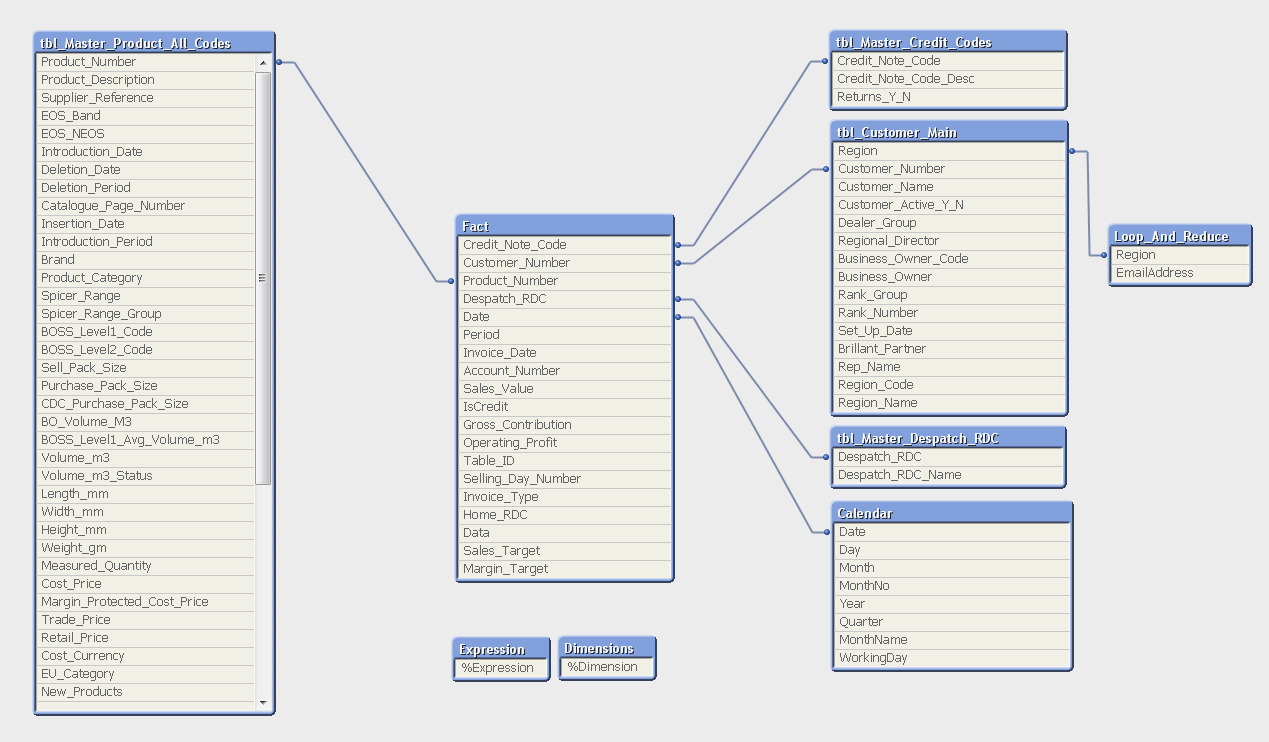
QVDs:

* tbl\_Operating\_Profit\_
* tbl\_CustomerMargin
* tbl\_Customer\_Main
* tbl\_Region
* tbl\_Master\_Credit\_Codes
* tbl\_Master\_Product\_All\_Codes
* tbl\_Master\_EU\_Category\_Descriptions
* tbl\_Master\_Despatch\_RDC
* WorkingDays

Inline:

* Loop\_And\_Reduce
* Calendar
* Dimensions
* Expressions

### Data Model Diagram



### Data Modelling Transformations

* The Fact table is a concatenation of:
  + tbl\_Operating\_Profit\_
  + tbl\_CustomerMargin
* The tbl\_Region QVD is left joined to the tbl\_Customer\_Main table
* The tbl\_Master\_Credit\_Codes QVD only brings in records where field Credit\_Note\_Code exists within the data model.
* The tbl\_Master\_EU\_Category\_Descriptions QVD is left joined to the tbl\_Master\_Product\_All\_Codes table.
* The Calendar table is built based on the Date field from the Fact table; this is both the Invoice\_Date from tbl\_Operating\_Profit\_ and TransactionDate from tbl\_CustomerMargin.
* The table WorkingDays is left joined to the Calendar table
* 2 data island tables are created to be used within the ad hoc report
  + Dimensions
  + Expressions
* All other QVDs are fully loaded

## Spicers.qvw

The front end application takes a binary load from the Data Model – Spicers.qvw.

Spicers

**QVW**

Data Model - Spicers

**QVW**

### Variables

**SET** ***vColour1*** = RGB(233,0,9); //red  
**SET** ***vColour2*** = RGB(171,144,179); //lightpurple  
**SET** ***vColour3*** = RGB(94,62,121); //darkpurple  
**SET** ***vColour4*** = RGB(255,172,56); //orange  
**SET** ***vColour5*** = RGB(26,138,144); //green  
**SET** ***vColour6*** = RGB(0,0,0); //black  
**SET** ***vColour7*** = RGB(180,180,180); //darkgrey  
**SET** ***vColour8*** = RGB(200,200,200); //lightgrey  
**SET** ***vColour9*** = RGB(255,255,255); //white  
  
**SET** ***vShowMenu*** = ''; //Show Menu Filter ie Filters/Search   
  
**SET** ***HidePrefix*** = '%'; //Hide fields prefixed with '%'

**SET** ***vMaxDate*** = =Max({<Data={'Sales'}>}Date);  
**SET** ***vYearStart*** = =YearStart(vMaxDate);  
**SET** ***vMonthStart*** = =MonthStart(vMaxDate);  
**SET** ***vR12Start*** = =AddMonths(vMaxDate+1,-12);  
**SET** ***vMaxDateLY*** = =AddYears(vMaxDate,-1);  
**SET** ***vYearStartLY*** = =YearStart(vMaxDateLY);  
**SET** ***vMaxDateLM*** = =AddMonths(vMaxDate,-1);  
**SET** ***vMonthStartLM*** = =MonthStart(vMaxDateLM);

### Common Expressions

Below is a list of commonly used expressions within the application, changing the variables within the expression you can calculate MTD/R12/LYTD/LMTD

**Sales YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >}Sales\_Value)

**Sales Target YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Sales\_Target)

**Sales % of Target YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Sales\_Value)/   
Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Sales\_Target)

**Sales variance of Target YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Sales\_Target)-  
Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Sales\_Value)

**Sales ex Credit YTD**Sum({<IsCredit={0}, Date={">=$(vYearStart) <=$(vMaxDate)"} >}Sales\_Value)

**Margin YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >}Gross\_Contribution)

**Margin Target YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Margin\_Target)

**Margin % of Target YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Gross\_Contribution)/   
Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Margin\_Target)

**Margin variance of Target YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Margin\_Target)-   
Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >} Gross\_Contribution)

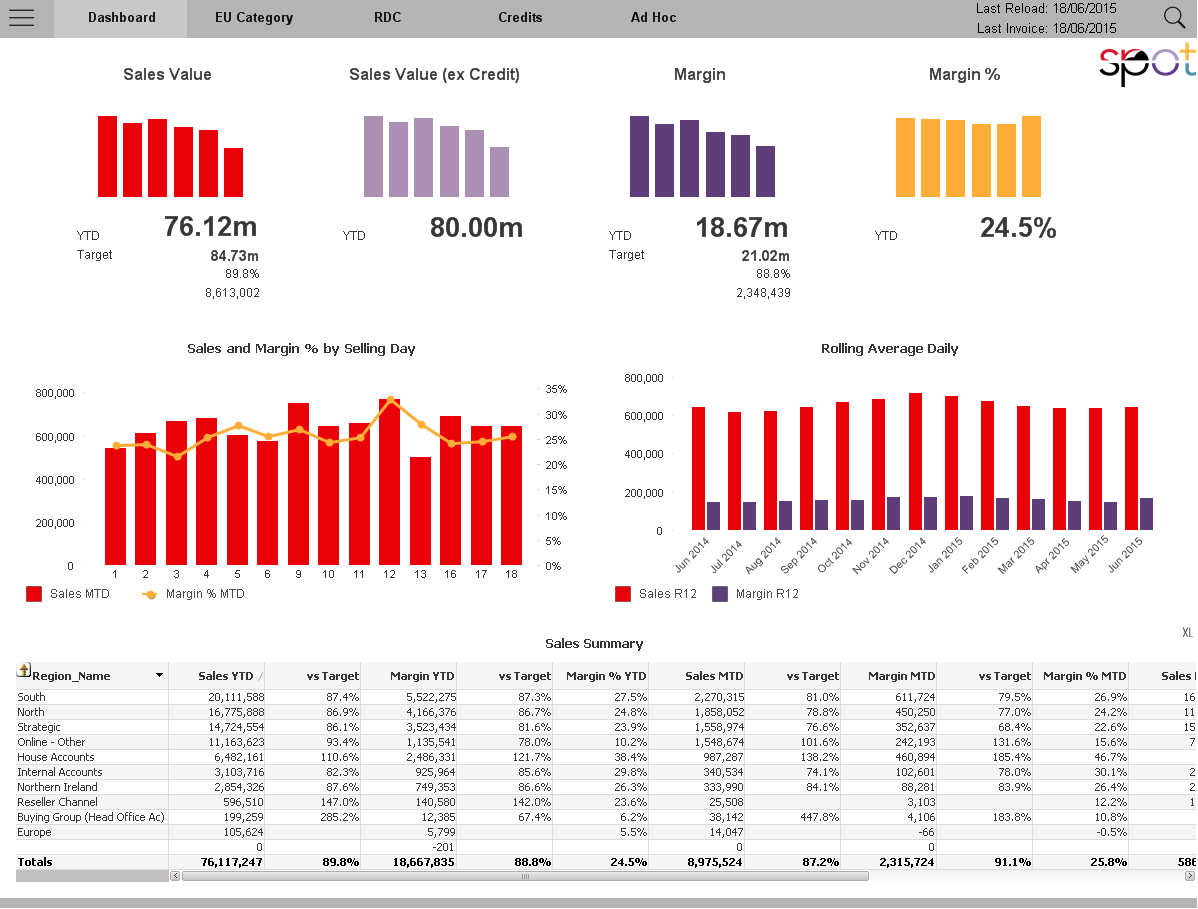
**Margin % YTD**Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >}Gross\_Contribution)/  
Sum({<Date={">=$(vYearStart) <=$(vMaxDate)"} >}Sales\_Value)

**Average Daily Sales Rolling 12 months**Sum({<Date={">=$(vR12Start) <=$(vMaxDate)"} >}Sales\_Value)/   
Sum({<Date={">=$(vR12Start) <=$(vMaxDate)"} >}WorkingDay \*-1)

**Average Daily Margin Rolling 12 months**Sum({<Date={">=$(vR12Start) <=$(vMaxDate)"} >}Gross\_Contribution)/  
Sum({<Date={">=$(vR12Start) <=$(vMaxDate)"} >}WorkingDay \*-1)

## User Interface

### Dashboard



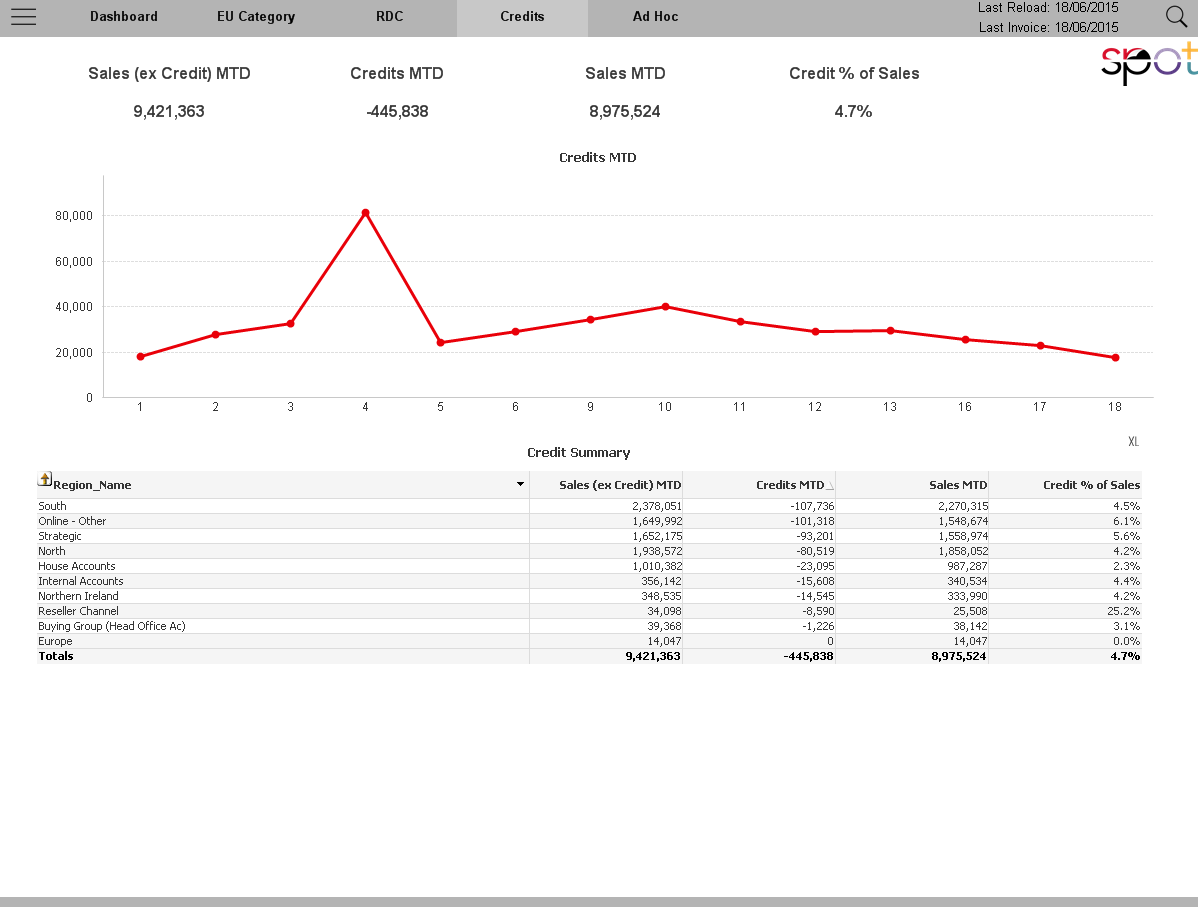
### EU Category

#### 

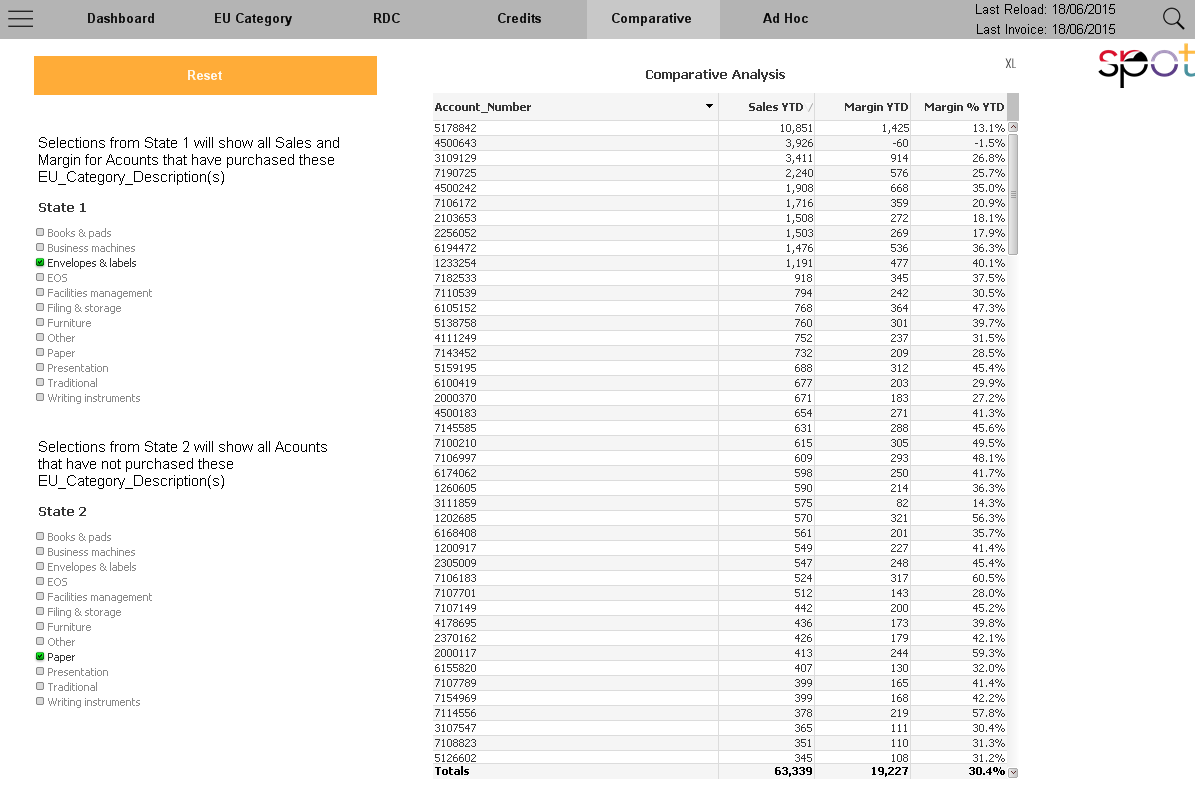
### RDC

#### 

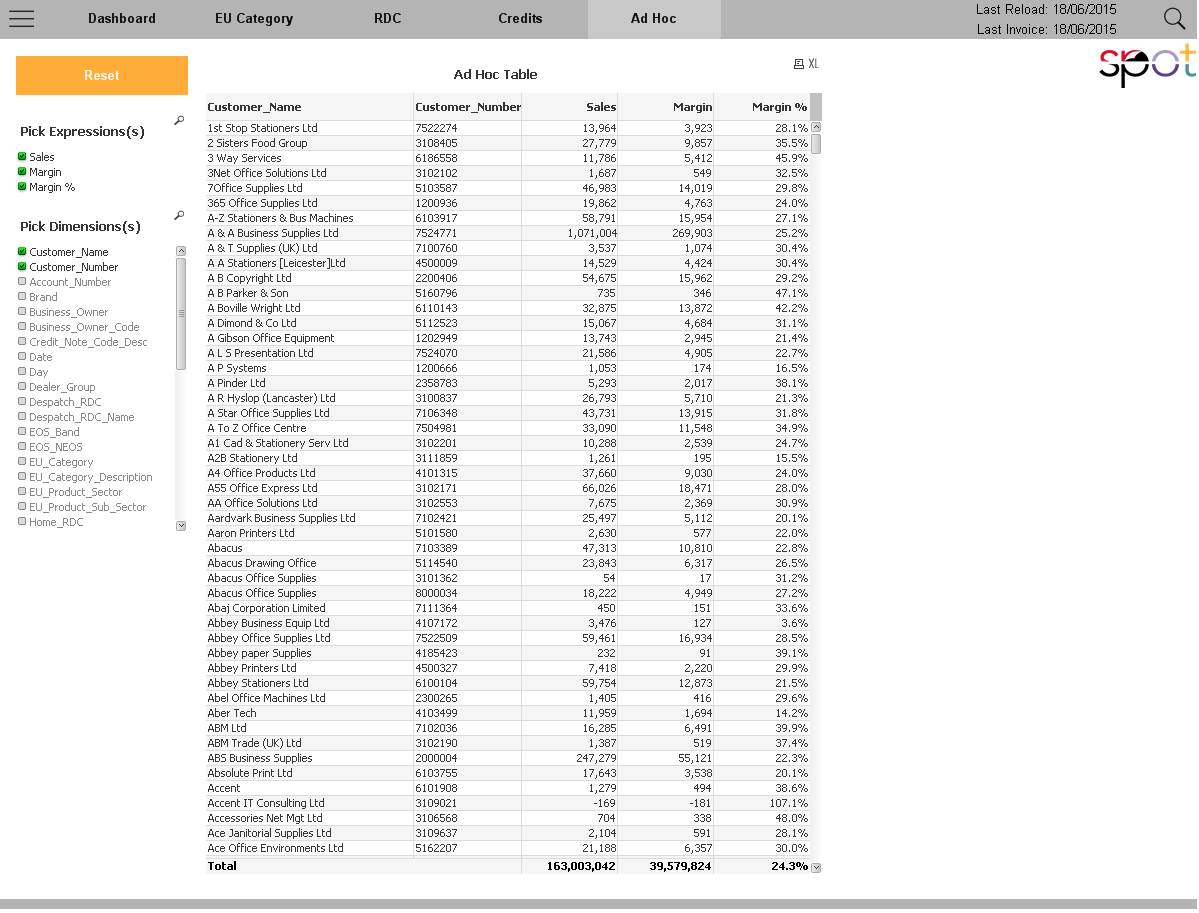
### Credits



### Comparative

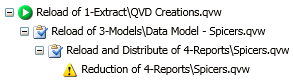


### Ad Hoc



## Scheduled Tasks

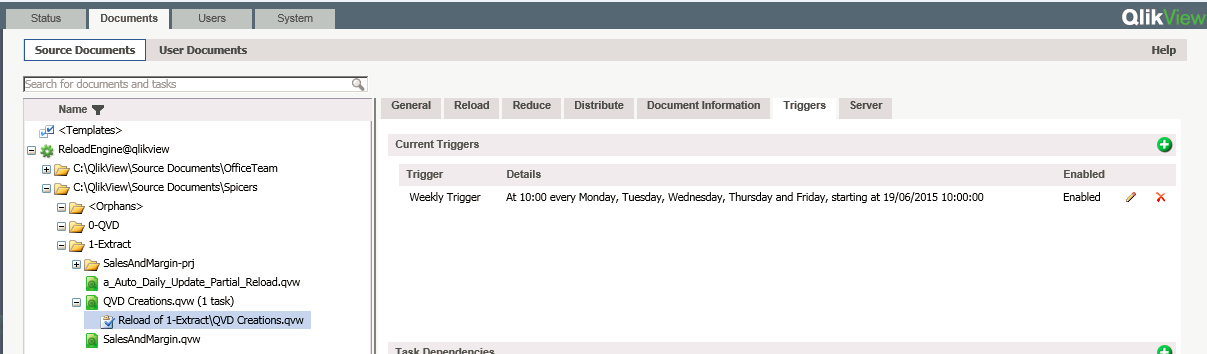
4 tasks are used to reload the application.



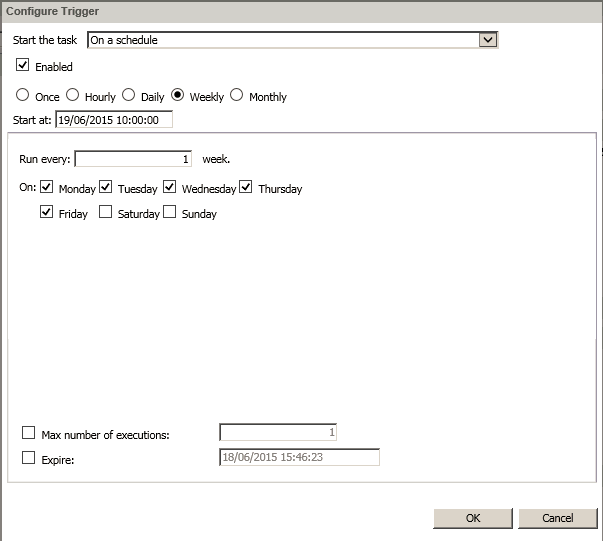
### Data Load Details

#### Data Load Frequency:

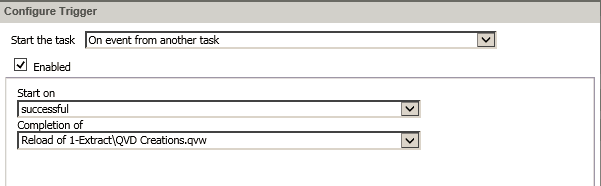
The QVD Creations.qvw task is currently scheduled Monday – Friday at 10am.



The frequency of the load can be changed according to demand. Settings to do this are in the QMC when configuring the Trigger Details.



All other tasks follow after the successfully completion of the above task.

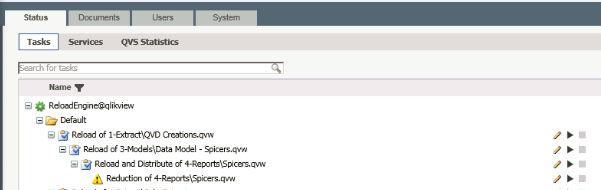


#### Data Load Type:

The QVD Creations.qvw does a full reload of the SQL tables.

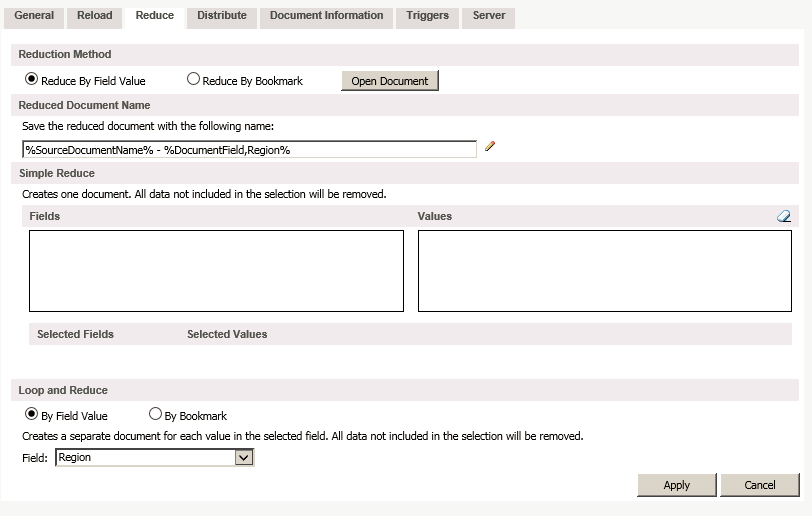
#### Restartability:

The tasks can be manually started from any stage by clicking the play icon from the status page.



### Loop and Reduce

The final task performs a Loop and Reduce over the dataset in order to create a separate QVW for each value from the Region field



Distribution secruity is then applied to the Looped and Reduced QVWs by looking back in the datamodel for recipient information. This is currently setup as an inline table in the Data Model – Spicers.qvw associcated the data on Region ans EmailAddress.

