

# SSRS

SQL Server Reporting Services



Rainer Stropek

software architects gmbh

Web <http://www.timecockpit.com>  
Mail [rainer@timecockpit.com](mailto:rainer@timecockpit.com)  
Twitter @rstropek



**time cockpit**  
Saves the day.

# Further Readings

- ▶ Reporting Services (SSRS) in MS Technet  
<http://technet.microsoft.com/en-us/library/ms159106.aspx>
- ▶ SQL Server Reporting Services Team Blog  
<http://blogs.msdn.com/b/sqlrsteamblog/>
- ▶ SQL Server Data Tools Team Blog  
<http://blogs.msdn.com/b/ssdt/>
- ▶ Videos about SSRS on Channel9  
<http://channel9.msdn.com/search?term=reporting+services&type=All>
- ▶ Book: Professional SQL Server 2012 Reporting Services  
[Link to Amazon](#)

# Resources

- ▶ SQL Server 2012 Downloads

Express Ed.: <http://www.microsoft.com/en-us/download/details.aspx?id=29062>

Trial Ed.: <http://www.microsoft.com/en-us/download/details.aspx?id=29066>

- ▶ SQL Server Data Tools BI for Visual Studio 2012

<http://www.microsoft.com/en-us/download/details.aspx?id=36843>

- ▶ SQL Server 2012 Report Builder

<http://www.microsoft.com/en-us/download/details.aspx?id=29072>

- ▶ AdventureWorks Sample Databases

Recommended install: AdventureWorks, AdventureWorksDW

<http://msftdbprodsamples.codeplex.com/>

- ▶ Sample code from book [Professional SQL Server 2012 Reporting Services](#)

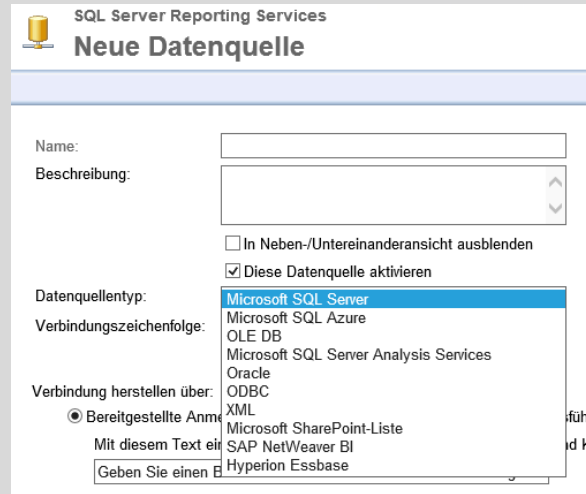
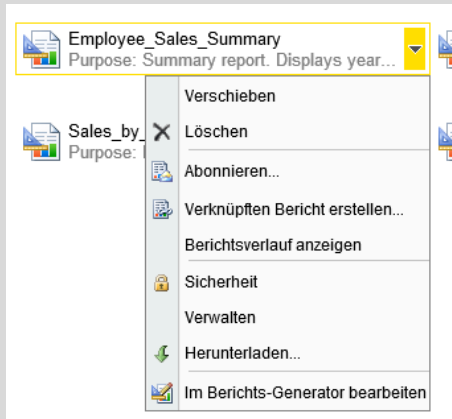
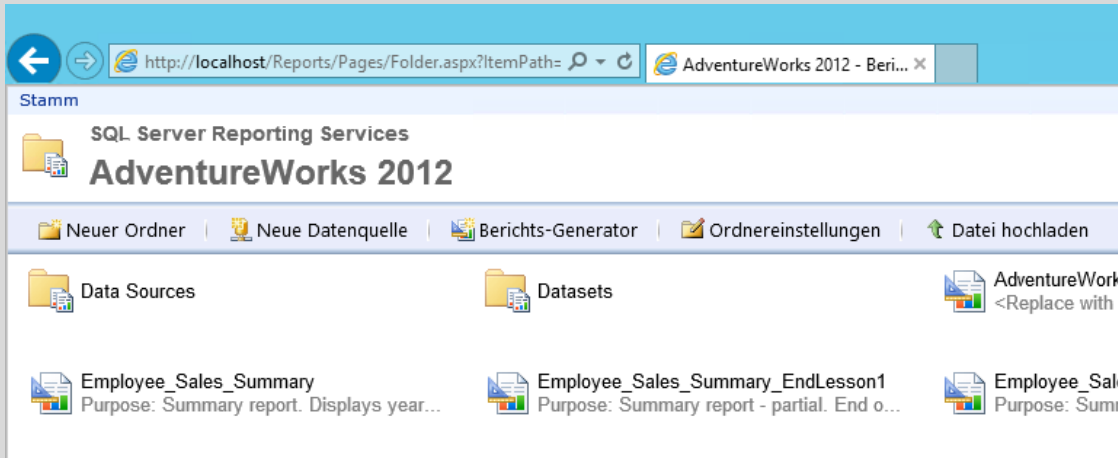
<http://www.wrox.com/WileyCDA/WroxTitle/Professional-Microsoft-SQL-Server-2012-Reporting-Services.productCd-1118101111.html>

# Introduction

SQL Server Reporting Services Basics

# What is SSRS?

- ▶ **Server components that execute reports**
  - Get data from data source
  - Render report to a certain target format (e.g. PDF, Excel, HTML, etc.)
- ▶ **Web portal for managing reports**
  - Management tasks (upload report, schedule report execution, etc.)
  - Trigger report execution incl. interaction (drill down, links between reports, etc.)
- ▶ **SharePoint integration**
- ▶ **API**
  - [Web services](#)
  - [URL-based API](#)
  - [.NET components for WinForms and ASP.NET](#)



# Managing Reports

Report Manager

*http://myserver/Reports*

## Manage SSRS Items

Reports

[Data Sources](#)

[Subscriptions](#)

[Users and permissions](#)

## Run reports

Server-based reporting (as opposed to local mode and RDLC)

# Report Manager

A lap around SSRS Report Manager

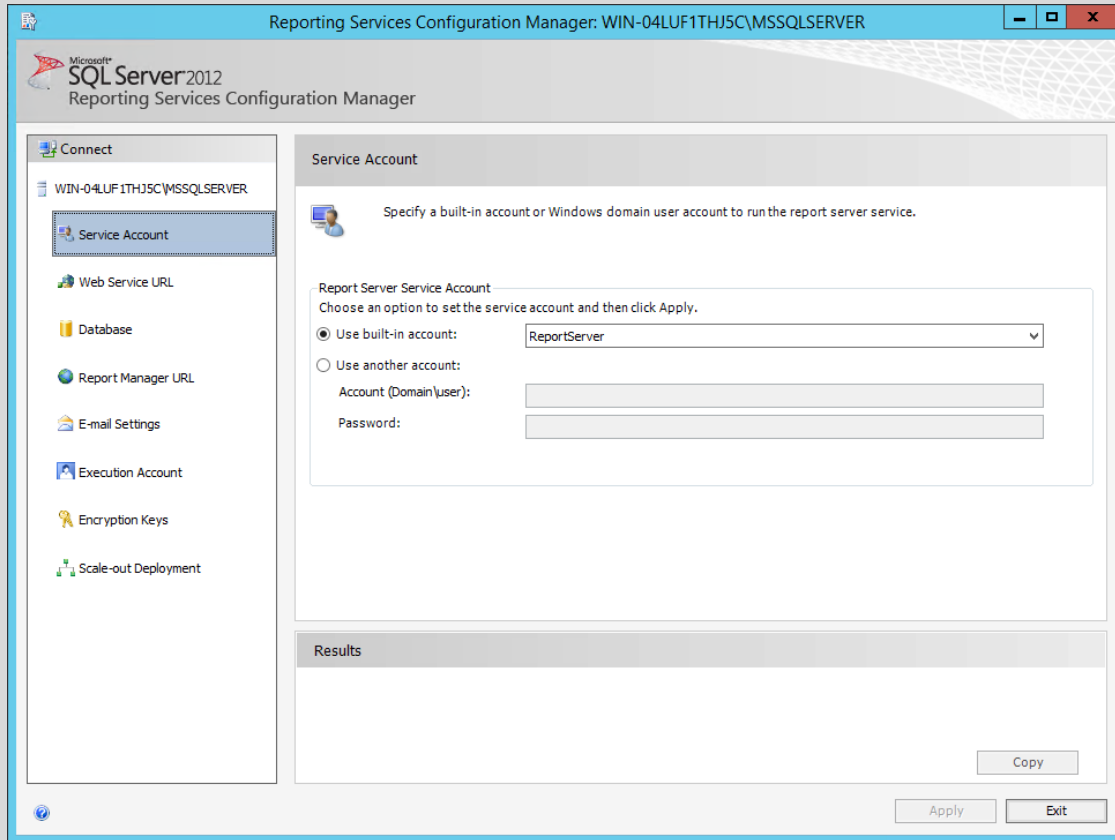
# Demo

# Config Manager

Configure server-wide settings for SSRS

For system administrators

See also [Technet](#)





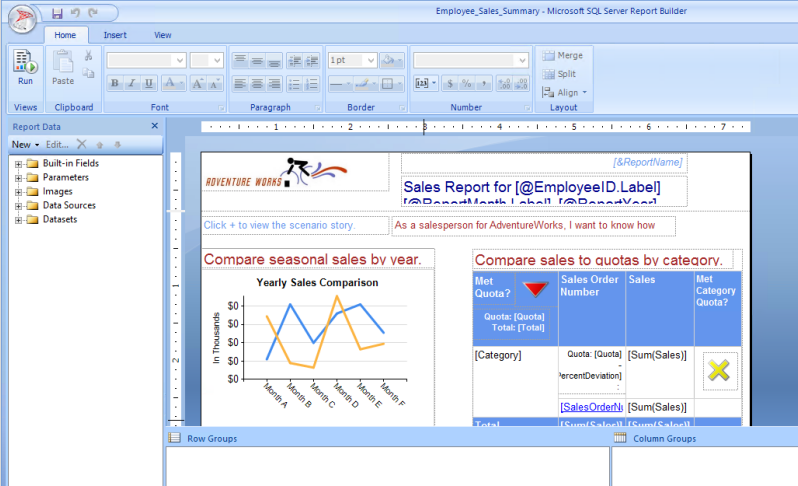
# Report Builder

Office-like UI

Report creation front-end  
for end users

3rd party alternative  
available

Compatibility on RDL-level



Design View

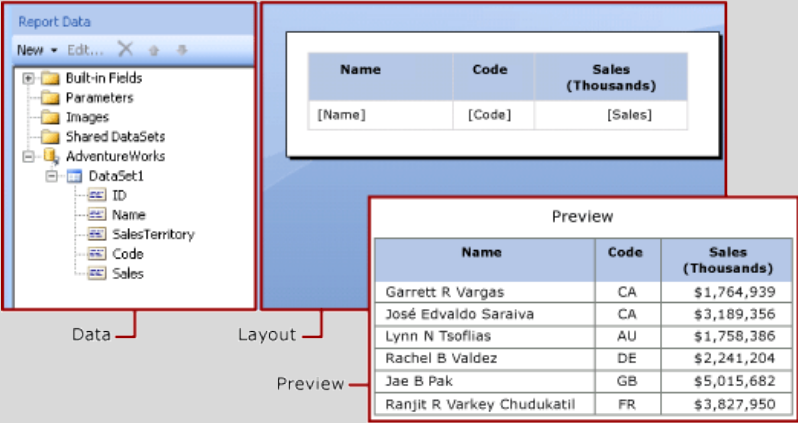
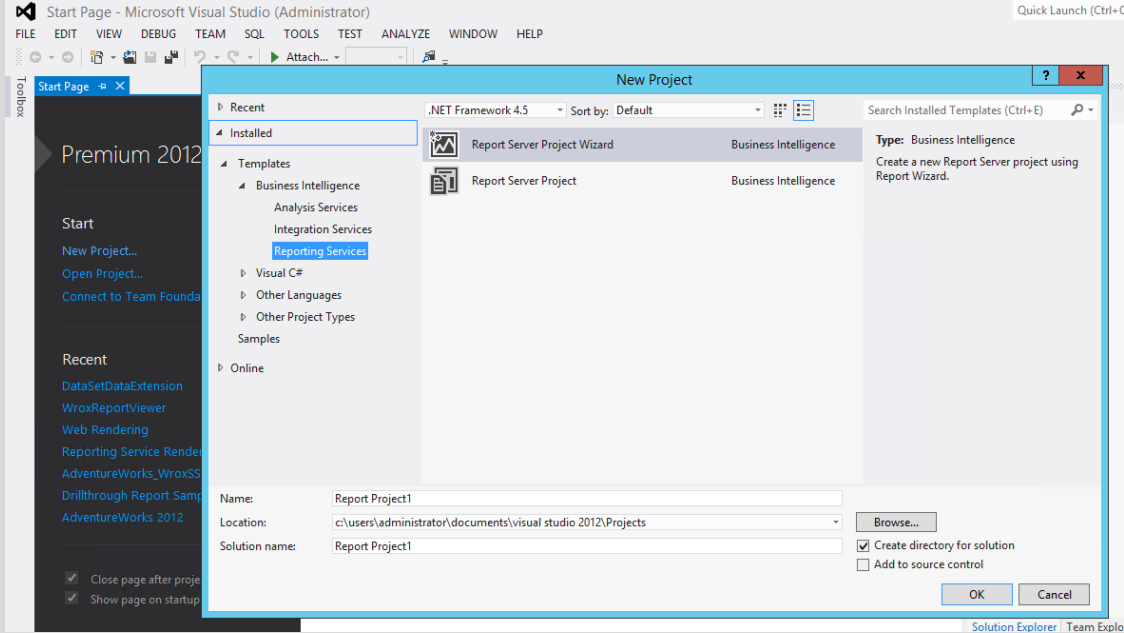


Image source (lower image): [Technet](http://Technet)



# SSDT

SQL Server Data Tools – BI  
aka SQL Server BI Studio

Visual Studio extension  
Report Designer

Report creation UI for  
developers and power  
users

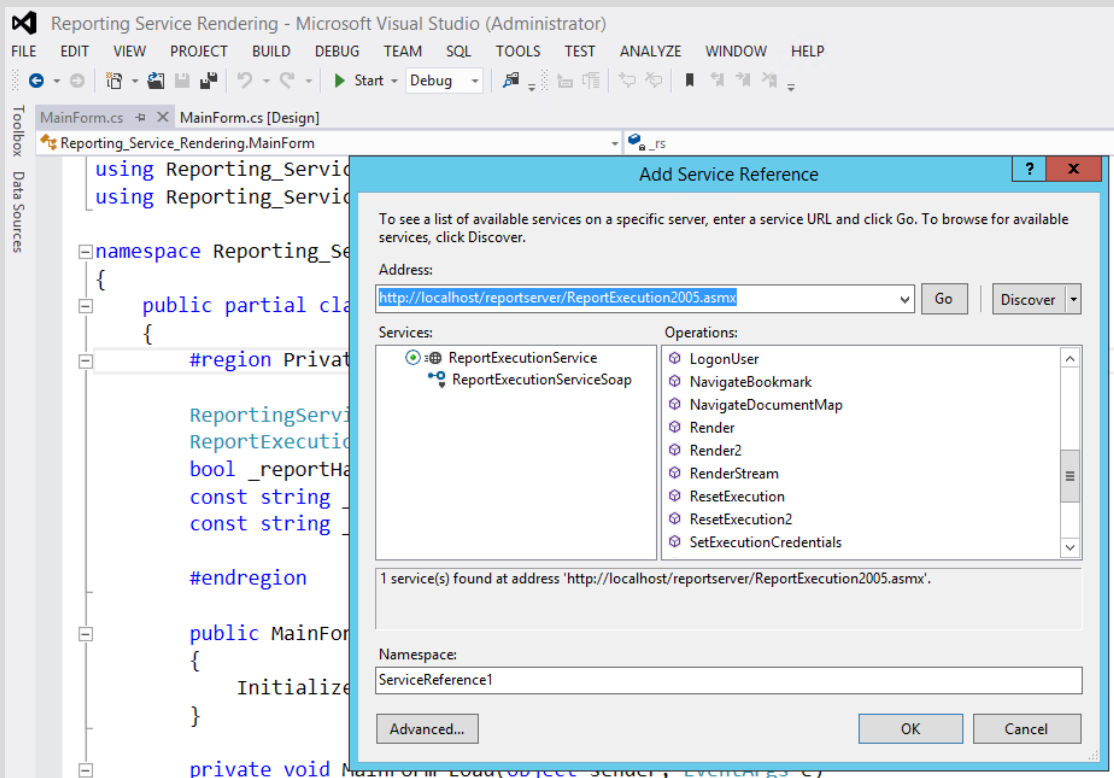
# Demo

## Report Creation

A lap around SSRS Report Builder  
and SSDT

Report Builder

Visual Studio with SSDT



# Web Services

Manage SSRS and render reports

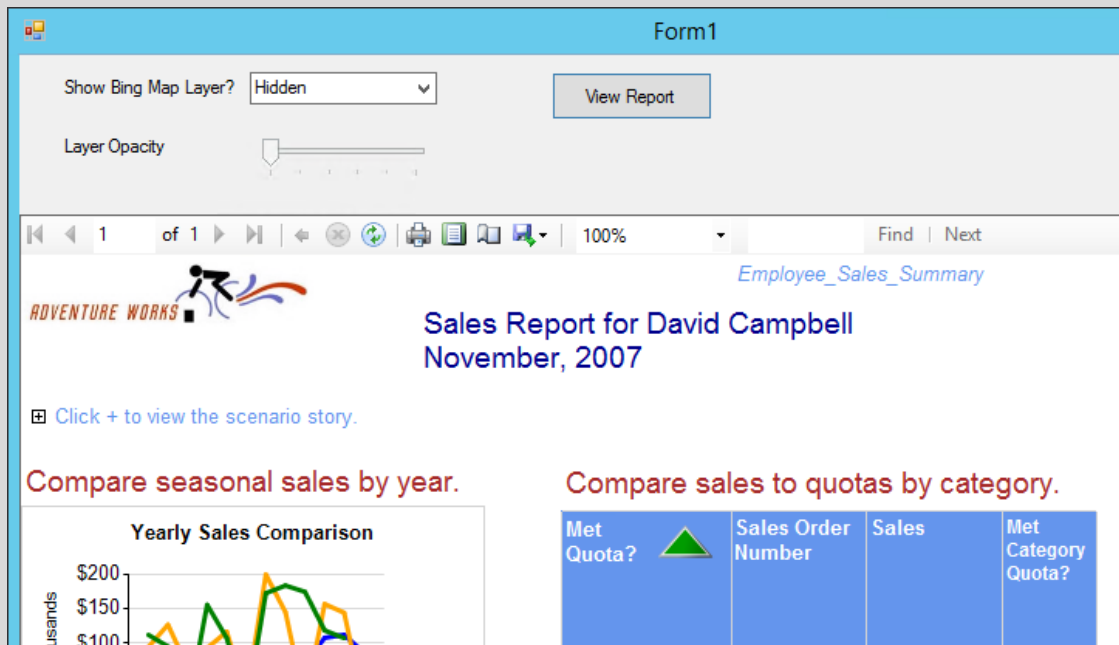
Set of SOAP web services

<http://myserver/reportserver/ReportExecution2005.asmx>

<http://myserver/reportserver/ReportService2010.asmx>

Detailed reference see

[Technet](#)



# Report Viewer

WinForms/WPF und ASP.NET

## Report viewer control

C:\Program Files (x86)\Microsoft  
Visual Studio 10.0\  
ReportViewer

## WinForms

Can be used in WPF, too (see  
[MSDN](#))

Detailed reference see  
[MSDN](#)

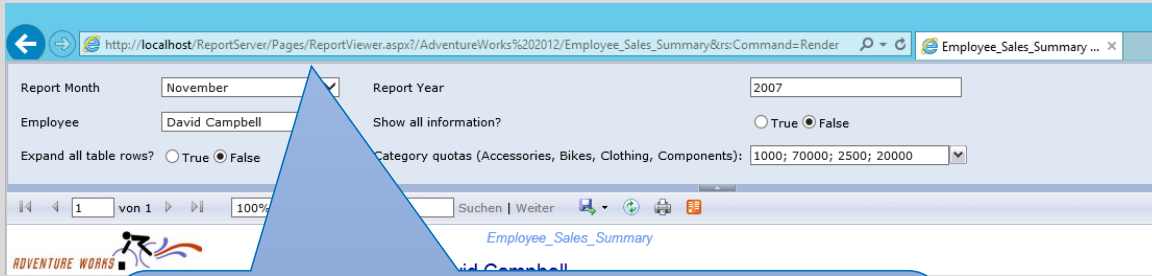
# URL Access

## Launch reports using URL

Details see MSDN, chapter *URL Access (SSRS)*

Detailed reference see

[Technet](#)



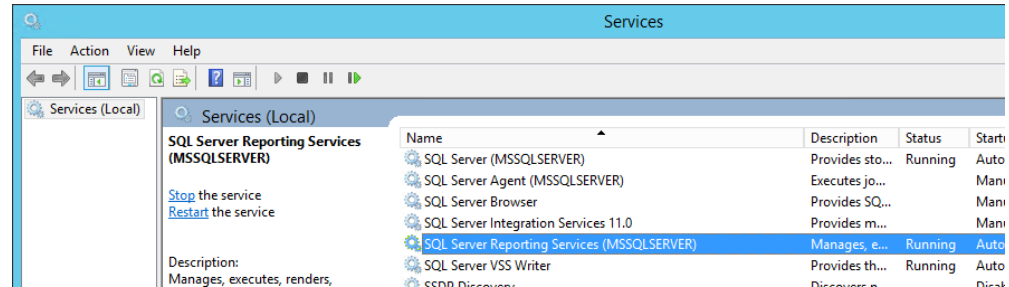
http://localhost/ReportServer/Pages/ReportViewer.aspx?  
/AdventureWorks%202012/Employee\_Sales\_Summary  
&rs:Command=Render

# Installation

SSRS Installation Basics, Server Architecture

# Installation

- ▶ Fully integrated in SQL Server installation
- ▶ Native Mode or SharePoint Mode  
SharePoint mode not covered here
- ▶ Windows service
- ▶ Support for multiple instances





# Editions

## ▶ Developer Edition

Full feature set at a reduced price  
For non-production use only!  
Can be installed on client OS, too

## ▶ Different editions for production use

FEATURE	ENTERPRISE, BUSINESS INTELLIGENCE EVALUATION, DEVELOPER	STANDARD	WEB	EXPRESS ADVANCED
Reporting Services Windows service	Yes	Yes	Yes	Yes
SharePoint Integrated mode	Yes	Yes		
Scale-out topologies	Yes			
Report Data Alerts	Yes			
Power View	Yes			
Role-based security	Yes	Yes	Yes, limited	Yes, limited

FEATURE	ENTERPRISE, BUSINESS INTELLIGENCE EVALUATION, DEVELOPER	STANDARD	WEB	EXPRESS ADVANCED
Custom security extensions	Yes	Yes	Yes	Yes
Export to Word, Excel, PDF, and images	Yes	Yes	Yes	Yes
Remote and nonrelational data sources	Yes	Yes		
Data source, delivery, and rendering extensibility	Yes	Yes		
Report delivery	Yes	Yes		
Report history, scheduling, subscription, and caching	Yes	Yes		
Data-driven subscriptions	Yes			
SSDT Report Designer	Yes	Yes	Yes	Yes
Report Builder	Yes	Yes		
Report Manager	Yes	Yes	Yes	Yes
Server memory (minimum)	1 GB	512 MB	512 MB	256 MB (32-bit) 512 MB (64-bit)
Server memory (recommended)	2+ GB	2+ GB	2+ GB	1+ GB
Server memory (maximum utilized)	Unlimited	64 GB	64 GB	4 GB
Supported CPU architectures	32-bit 64-bit	32-bit 64-bit	32-bit 64-bit	32-bit 64-bit
Maximum number of CPU cores used by a single Reporting Services instance	Unlimited	16 cores	16 cores	4 cores
CPU speed (minimum)	1.4 GHz	1.4 GHz	1.4 GHz	1.4 GHz
CPU speed (recommended)	2.0+ GHz	2.0+ GHz	2.0+ GHz	2.0+ GHz

# Server Architecture

Note: SSRS does not use IIS  
IIS and SSRS can co-exist on a single server

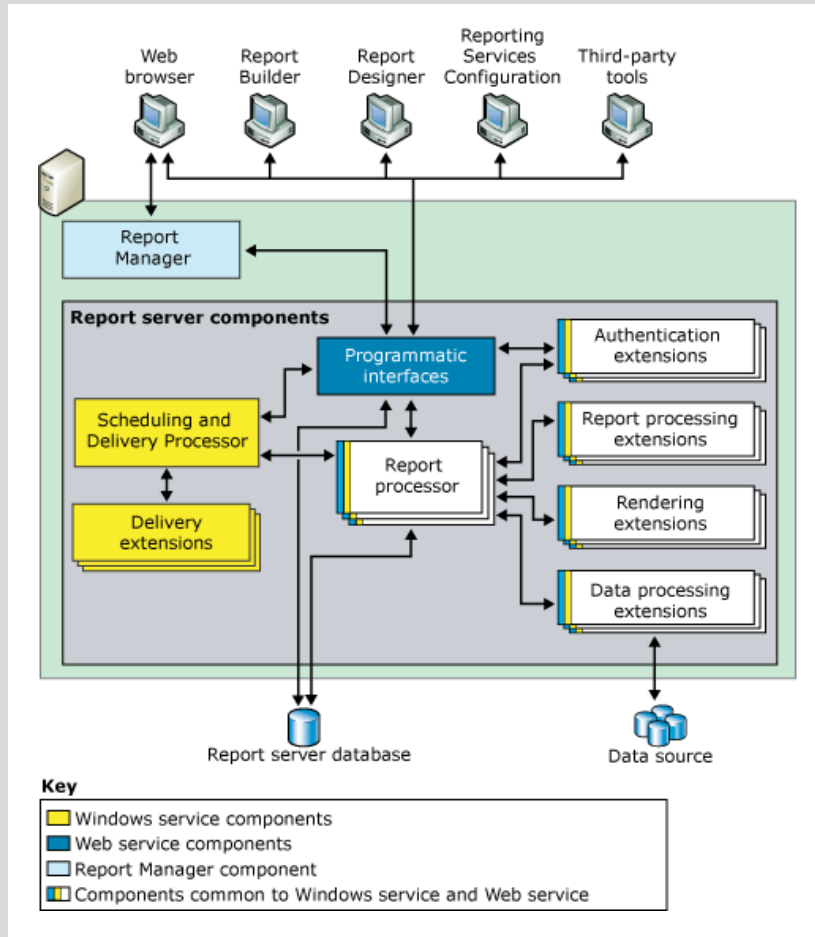
## Security Sublayer

Kerberos is preferred (Windows Authentication)

Basic Authentication is possible in combination with TLS/SSL

## Highly extensible

Detailed reference see [Technet](#)



# Data Sources and Rendering Extensions

## ► Data Sources

- Microsoft SQL Server
- Microsoft SQL Server Analysis Services
- OLE DB data sources
- ODBC data sources
- Oracle
- XML data sources
- Etc. (details see [Technet](#))

## ► Rendering Extensions

- CSV
- Excel
- Word
- HTML
- PDF
- TIFF
- XML
- Atom

Details see [Technet](#)

# Demo

## Sources, Rendering

Adding different Data  
Sources

Rendering reports into  
different formats

Object Explorer

Connect

WIN-04LUF1THJ5C (SQL Server 11.0.3128 - WIN-04LUF1THJ5C\Admini)

- Databases
  - System Databases
  - Database Snapshots
  - AdventureWorks\_WroxSSRS2012
  - AdventureWorks2012
  - AdventureWorksDW\_WroxSSRS2012
  - ReportServer
    - Database Diagrams
    - Tables
    - Views
    - Synonyms
    - Programmability
    - Service Broker
    - Storage
    - Security
    - ReportServerTempDB
  - Security
  - Server Objects
  - Replication
  - AlwaysOn High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent (Agent XPs disabled)

Object Explorer Details

WIN-04LUF1THJ5C (SQL Server 11.0.3128 - WIN-04)

Name	Schema
ActiveSubscriptions	dbo
Batch	dbo
CachePolicy	dbo
Catalog	dbo
ChunkData	dbo
ChunkSegmentMapping	dbo
ConfigurationInfo	dbo
DataSets	dbo
DataSource	dbo
DBUpgradeHistory	dbo
Event	dbo
ExecutionLogStorage	dbo
History	dbo
Keys	dbo
ModelDrill	dbo
ModelItemPolicy	dbo
ModelPerspective	dbo
Notifications	dbo
Policies	dbo
PolicyUserRole	dbo
ReportSchedule	dbo
Roles	dbo
RunningJobs	dbo

# Databases

## *ReportServer*

Main data store for SSRS

## *ReportServerTempDB*

Session and caching data

Detailed reference see

[Technet](http://technet.microsoft.com)

# Report Design

Creating Reports with SSRS

# Report Elements

- ▶ **Data Connection aka Data Source**  
Connection to the underlying data source (e.g. SQL Server)  
Shared or Embedded Data Sources
- ▶ **Data Set**  
Data returned from the data source  
Shared (i.e. cached) or Embedded Data Sets
- ▶ **Report Parameters**  
Used to filter and control the report data  
Also used for passing parameters in links (interactivity)

# Report Elements

## ▶ Data Regions

[Table, Matrix, or List](#)

[Chart](#)

[Gauge](#)

[Indicators](#)

[Map \(ESRI shapefiles, Bing Map Tiles\)](#)



# Demo

## Create a Report

Using the Matrix Wizard

Create a Data Source

Create a Data Set

```
SELECT * FROM  
vSalesByTimeAndTerritory
```

Create a Matrix

Format the report

Publish the report

<http://myserver/reportsserver>

```
CREATE view [dbo].[vSalesByTimeAndTerritory] as
select top 100 percent
    d.CalendarYear
    , d.CalendarQuarter
    , d.MonthNumberOfYear
    , d.EnglishMonthName MonthName
    , st.SalesTerritoryGroup
    , st.SalesTerritoryRegion
    , st.SalesTerritoryCountry
    , sum(s.SalesAmount) SalesAmt
    , sum(s.OrderQuantity) OrderQty
    , sum(s.Freight) Frieght
from
    FactResellerSales s inner join DimDate d
        on s.OrderDateKey = d.DateKey
    inner join DimSalesTerritory st
        on s.SalesTerritoryKey = st.SalesTerritoryKey
group by
    d.CalendarYear
    , d.CalendarQuarter
    , d.MonthNumberOfYear
    , d.EnglishMonthName
    , st.SalesTerritoryGroup
    , st.SalesTerritoryRegion
    , st.SalesTerritoryCountry
order by
    d.CalendarYear
    , d.CalendarQuarter
    , d.MonthNumberOfYear
    , st.SalesTerritoryGroup
    , st.SalesTerritoryRegion
    , st.SalesTerritoryCountry
```

# Demo

Prerequisites

Create a view with revenue data

*AdventureWorksDW2012* DB

# Demo

## Matrix Wizard

**New Table or Matrix**

### Arrange fields

Arrange fields to group data in rows, columns, or both, and choose values to display. Data expands across the page in column groups and down the page in row groups. Use functions such as Sum, Avg, and Count on the fields in the Values box.

**Available fields**

- CalendarYear
- CalendarQuarter
- MonthNumberOfYear
- MonthName
- SalesTerritoryGroup
- SalesTerritoryRegion
- SalesTerritoryCountry
- SalesAmt
- OrderQty
- Frieght

**Column groups**

- SalesTerritoryCountry

**Row groups**

- CalendarYear
- MonthName

**Σ Values**

- Sum(SalesAmt)
- Sum(OrderQty)

Help < Back Next > Cancel

# Demo

## Create a Report

Manual matrix creation

Build matrix report manually

Group property window

Correct sorting

Property pane

Available for all report items

Report Formatting

# Table and Matrix

- ▶ **Tables**  
Static columns  
Dynamic rows

Sales Territory	Full Name	ID2003	ID2004
Northwest	Pamela O. Anziani-Wolfe	\$900365.58	\$1656492.88
	David R. Campbell	\$1377431.33	\$1930885.56
	Tabi A. Menna-Arnan		
<b>Total</b>			

Table Report

Sales by Area and Year		North America		2003	2004	Total
		US	CA			
Clothing	Caps	\$9798.76	\$2801.06	\$9099.82	\$1001.50	\$10101.32
	Tights	\$51391.51	\$8233.90	\$59625.41	\$755.00	\$60180.41
Components	Chains	\$4972.00	\$701.83	\$5673.83	\$75	\$5748.83
	Cranksets	\$105118.64	\$18515.99	\$123634.63	\$728	\$124362.63
	Touring Frames	\$767130.33	\$196141.85	\$963272.18	\$6034	\$969306.18

Matrix Report

**Congratulations!**

Michael G Blythe 275

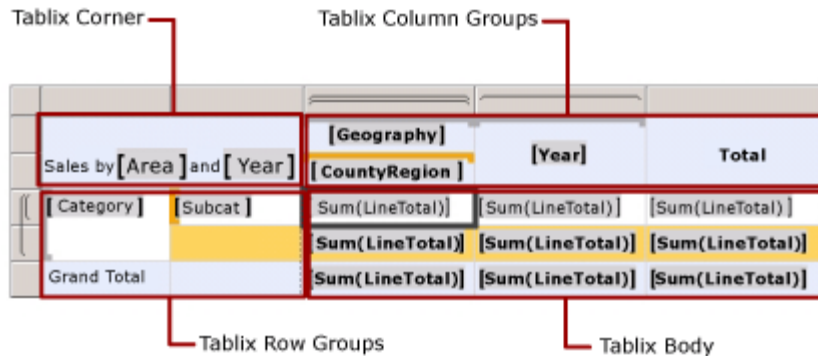
Hello Michael,  
Congratulations on an outstanding year!  
Your sales of \$1951086.83 makes you the TOP Adventure Bikes sales person in 2004.  
Bob & Craig



List Report

- ▶ **Matrix**  
Dynamic columns  
Dynamic rows

- ▶ **Tablix**  
Under the hood, everything is a tablix  
You can change from table to matrix and back as you like



# Report Structure

Click to add title

Calendar Year	Month Name	[Sales Amount]
[CalendarYear]	[MonthName]	[SalesAmount]
Total		[SumSalesAmount]

- Insert Column
- Delete Columns
- Add Group
  - Inside Group - Left
  - Inside Group - Right
- Column Group
  - Outside Group - Left
  - Outside Group - Right
- Column Visibility...
- Tablix Properties...

[&ExecutionTime]

New Edit... X

- Built-in Fields
- Parameters
- Images
- Data Sources
  - DW
    - Sales
      - CalendarYear
      - CalendarQuarter
      - MonthNumberOfYear
      - MonthName
      - SalesTerritoryGroup
      - SalesTerritoryRegion
      - SalesTerritoryCountry
      - SalesAmt
      - OrderQty
      - Friight

Click to add title

Calendar Year	Header
[CalendarYear]	Data

- LastName
- EnglishProductName
- Sum\_SalesAmount

# Tablix Structure

## Visual Indicators

The diagram illustrates a Tablix structure with the following columns: Date, Order, Product, Qty, and Line Total. The rows are grouped as follows:

- Parent group:** The first row, containing [Date], [Order], [Product], [Qty], and [Line Total].
- Child group:** The second and third rows, both containing [Sum(Qty)] and [Sum(LineTotal)].

Visual indicators include:

- Parent group:** Indicated by a red line pointing to the first row.
- Child group:** Indicated by a red line pointing to the second and third rows.
- Details group:** Indicated by a red line pointing to the [Qty] and [Line Total] columns.
- Innermost group indicator:** A red line pointing to the [Qty] cell in the first row.

	Date	Order	Product	Qty	Line Total
Parent group	[Date]	[Order]	[Product]	[Qty]	[Line Total]
Child group				[Sum(Qty)]	[Sum(LineTotal)]
Child group				[Sum(Qty)]	[Sum(LineTotal)]

The diagram illustrates a complex Tablix structure with the following columns: Sales by, Geography, Year, and Total. The rows are grouped as follows:

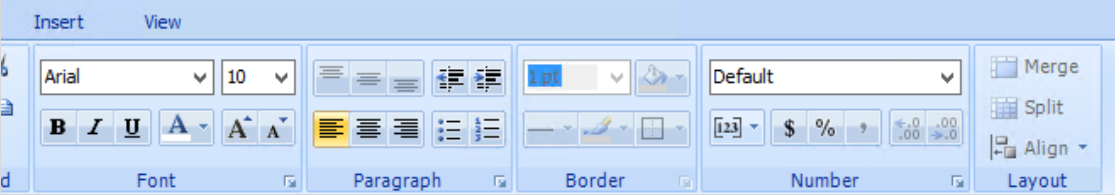
- Row group:** The first row, containing [Area] and [Year], [CountryRegion], [Year], and Total.
- Row group:** The second and third rows, both containing [Sum(LineTotal)] and [Sum(LineTotal)].
- Nested row group:** The fourth row, containing Subtotal, [Sum(LineTotal)], [Sum(LineTotal)], and [Sum(LineTotal)].
- Grand total row:** The fifth row, containing Grand Total, [Sum(LineTotal)], [Sum(LineTotal)], and [Sum(LineTotal)].

Visual indicators include:

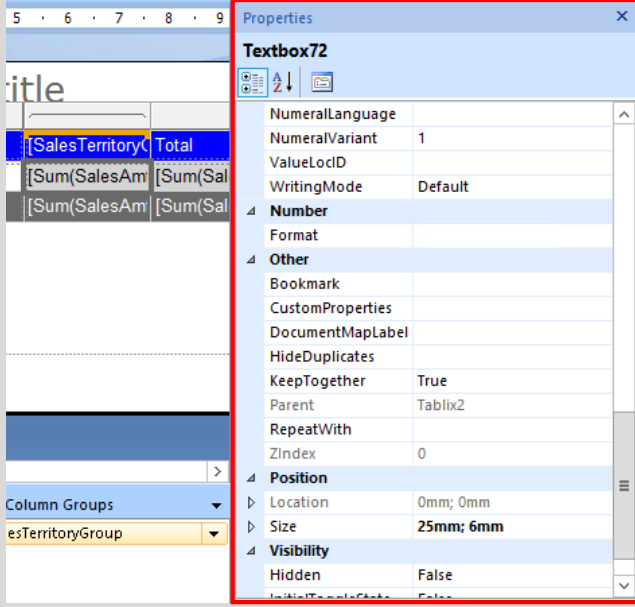
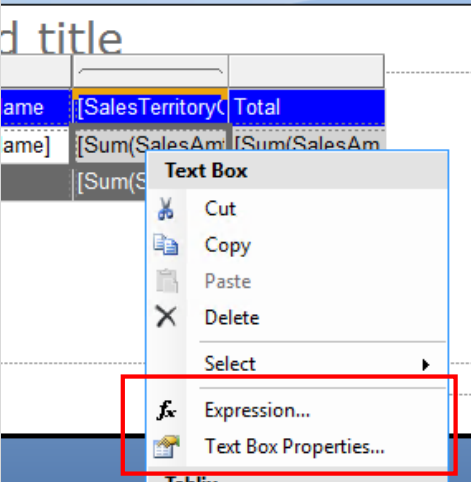
- Corner handle:** A red line pointing to the top-left corner of the first row.
- Corner title:** A red line pointing to the text "Sales by" above the first row.
- Column handles:** Red lines pointing to the top of the columns.
- Column groups:** Red lines pointing to the top of the columns.
- Total column:** A red line pointing to the Total column.
- Row handles:** Red lines pointing to the left of the rows.
- Row group:** A red line pointing to the first row.
- Nested row group:** A red line pointing to the fourth row.
- Group total row:** A red line pointing to the third row.
- Grand total row:** A red line pointing to the fifth row.

	Sales by	Geography	Year	Total
Row group	[Area] and [Year]	[CountryRegion]	[Year]	Total
Row group	[Category]	[Subcat]	[Sum(LineTotal)]	[Sum(LineTotal)]
Row group		Subtotal	[Sum(LineTotal)]	[Sum(LineTotal)]
Nested row group	Grand Total		[Sum(LineTotal)]	[Sum(LineTotal)]

# Formatting



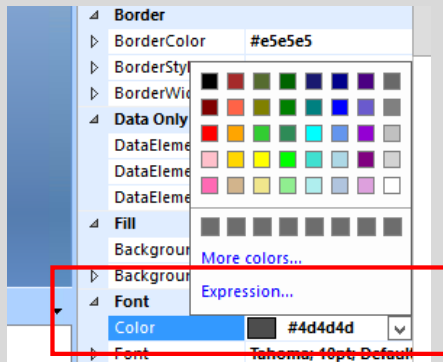
Ribbon



Property Window

Property pane





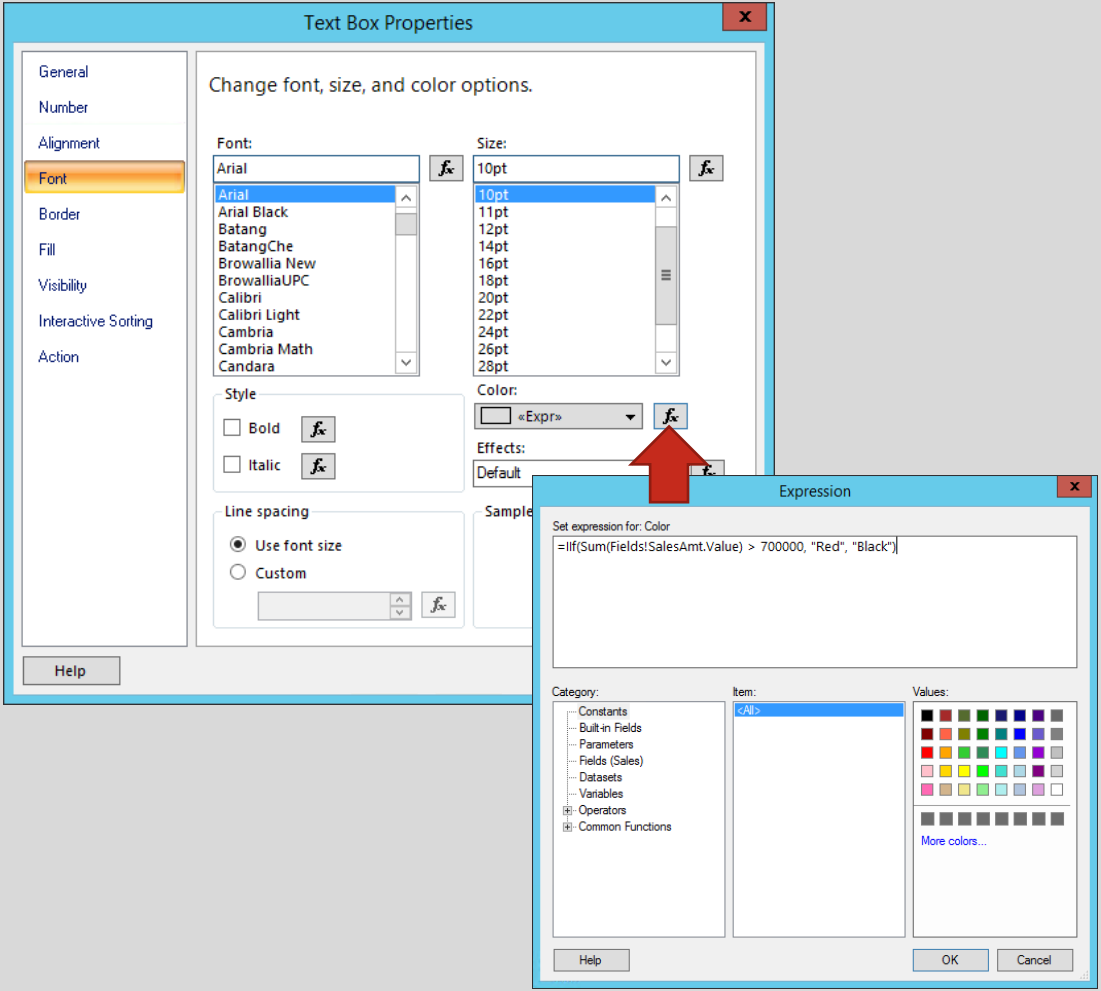
# Formatting

Expressions in property pane

Use [expressions](#) to add dynamic formatting

# Formatting

Tip: Use formulas for property values for dynamic formatting

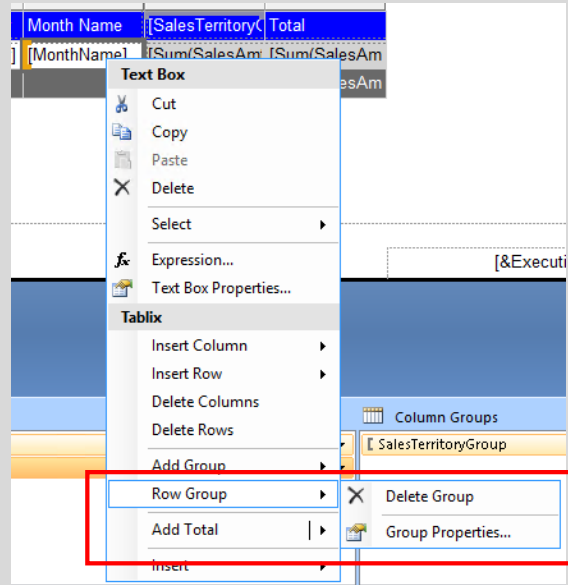
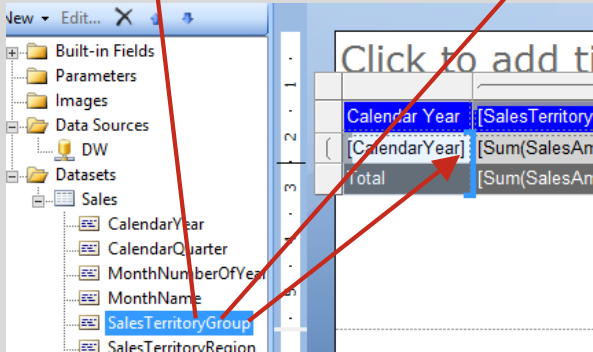
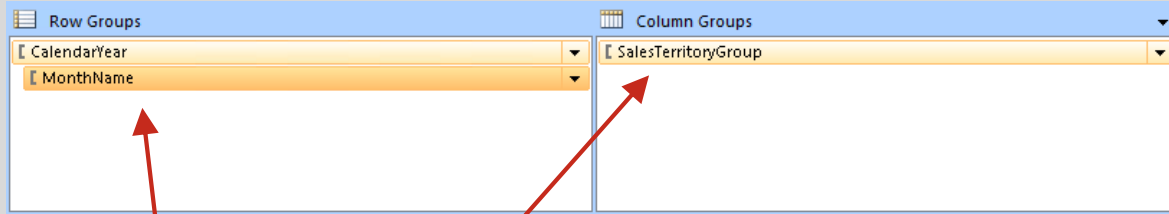


# Grouping

## Defining groups

- Grouping pane
- Drag & drop columns

- ## Fine-tune grouping
- properties with group property window
  - E.g. sorting

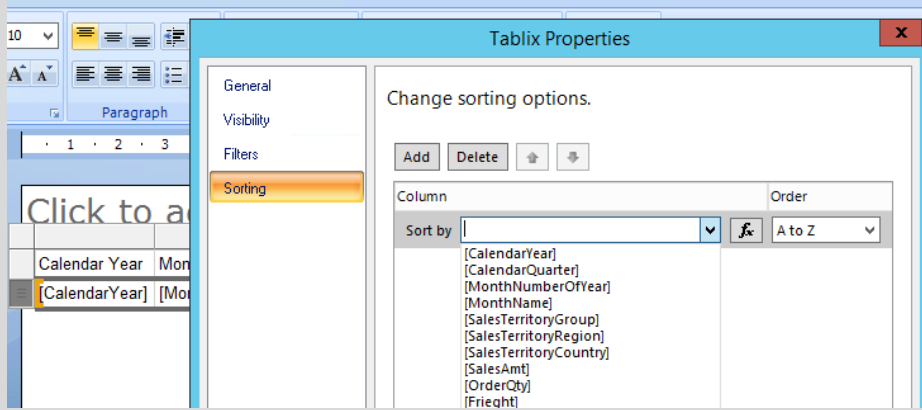


# Demo

Interactive Reports

Interactive Sorting

Drill-Down Reports



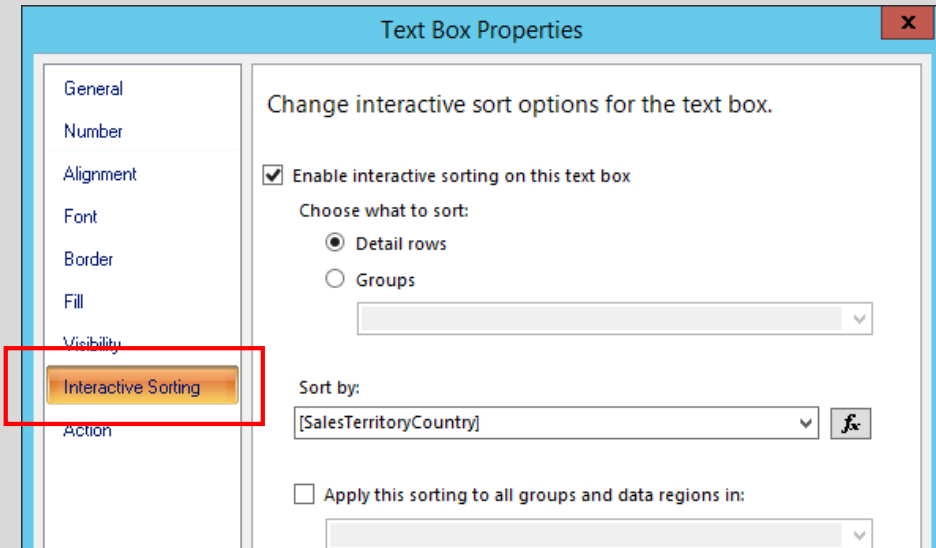
# Sorting

Sort details or groups

## Interactive sorting

Added to a cell in column headers

Can sort groups or details



# Drill-Down Reports

Adding Interactivity

Toggle visibility of a group

new

Paragraph

Click to add title

Calendar Year	Month Name
Total	
[CalendarYear]	Subtotal
	[MonthName]

Row Groups

- (CalendarYear)
- (MonthName)
- (Details)

Column Groups

Group Properties

Change display options.

When the report is initially run:

Show

Hide

Show or hide based on an expression

Display can be toggled by this report item:

CalendarYear

Help OK Cancel

ADVENTURE WORKS cycles

Australia		
Canada		
France		
Germany		
Northwest		
Southwest		
United Kingdom		

ADVENTURE WORKS cycles

Australia		
Canada		
Central		
France		
Germany		
Northwest		
Southwest		
Shu Ito		
Linda Mitchell		
United Kingdom		

ADVENTURE WORKS cycles

Shu Ito			\$8,761,727.29
Linda Mitchell			\$13,975,741.46
SD45784			\$5,061.87
SD45559			\$168.61
SD45532			\$43,167.84
SD45554			\$33,797.43
SD45555			\$87,266.26
SD45664			\$83,297.69
SD45566			\$1,159.98
SD45572			\$16,190.49
SD45531			\$5,697.17
SD45533			\$1,948.05
SD45339			\$36,821.21
SD45343			\$27,940.33
SD45311			\$24,259.49

# Data Access

Details about data access

# Data Sources

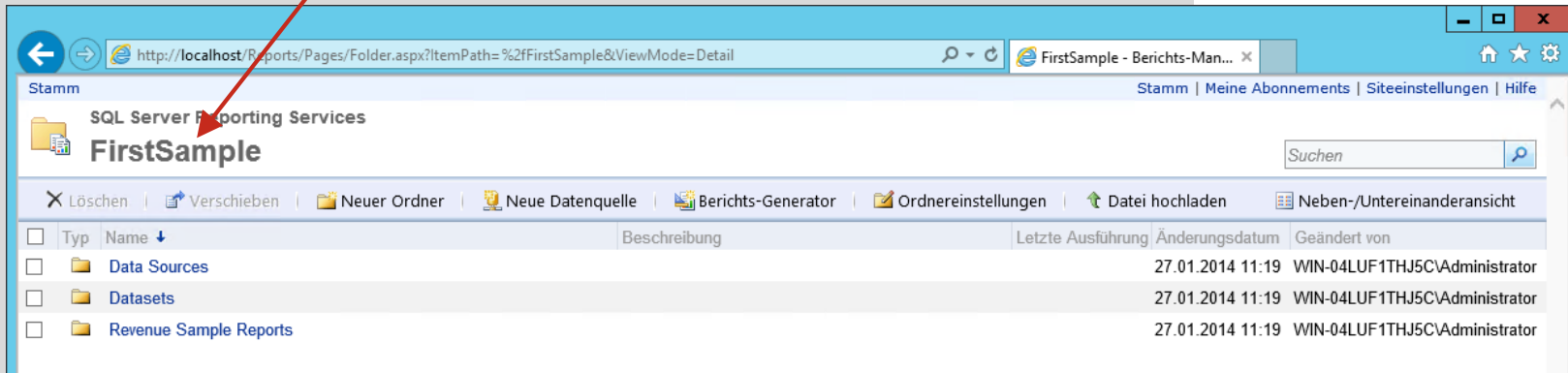
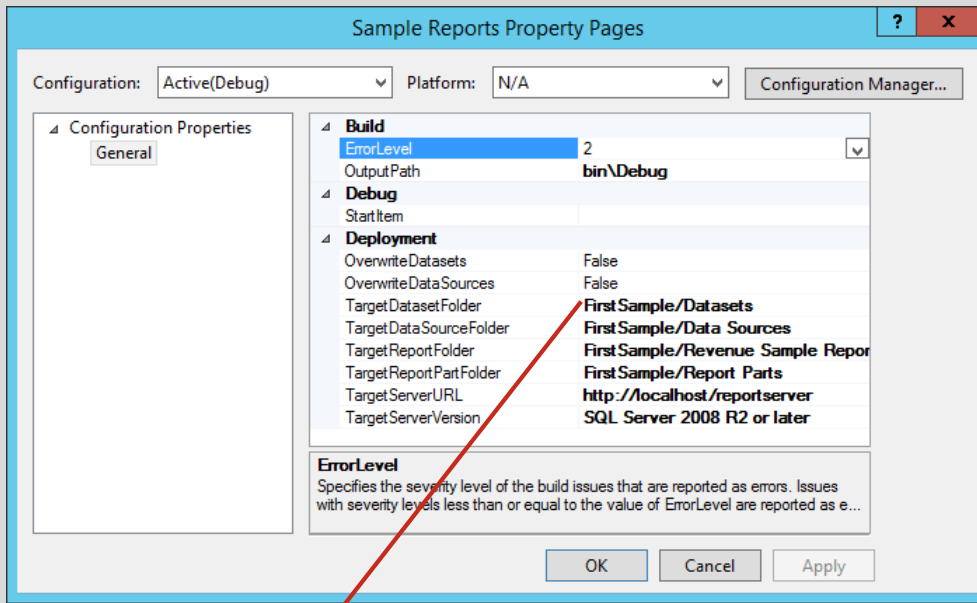
- ▶ Represent a connection to a database
- ▶ Embedded data source
  - Only available for a single report
  - Can be used by multiple Data Sets
- ▶ Shared data sources
  - Available for multiple reports
  - Often easier to maintain (e.g. after moving to a new DB server)
- ▶ **Tip: Prefer Shared Data Sources**
  - Simplifies maintenance

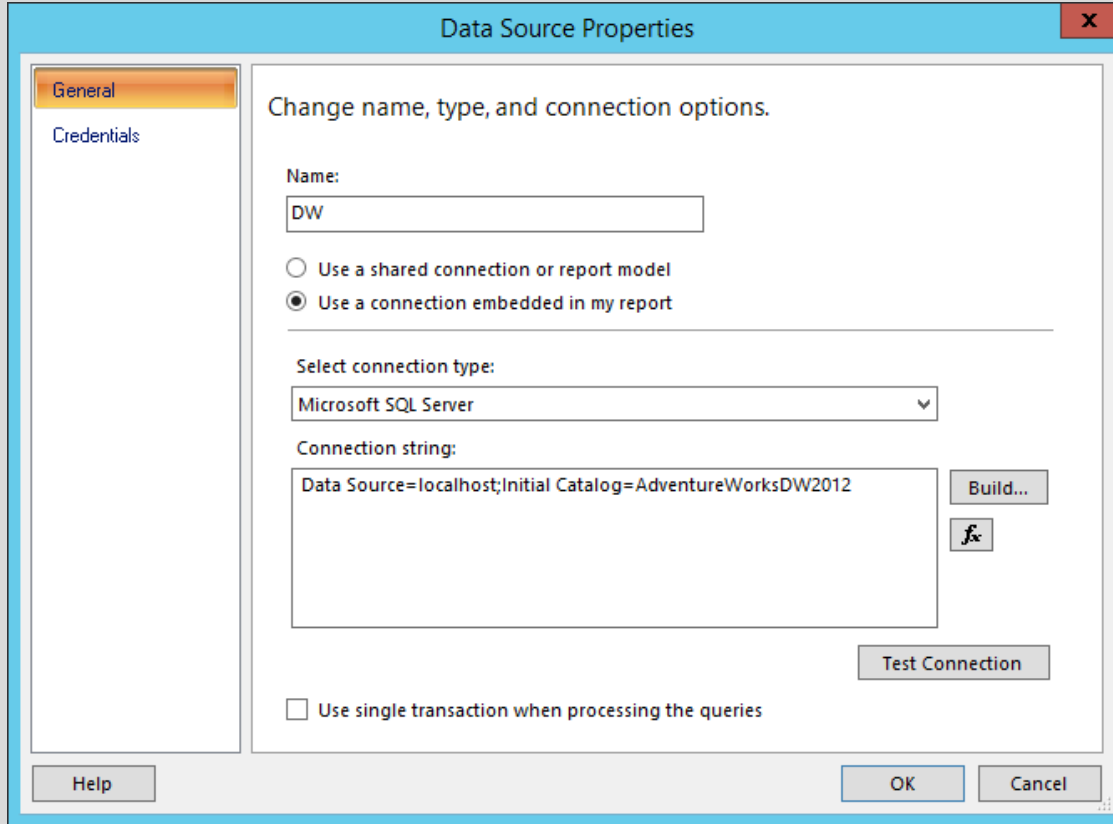


Demo

# Deployment in VS

Deployment settings





# Data Sources

Shared or embedded

## Types

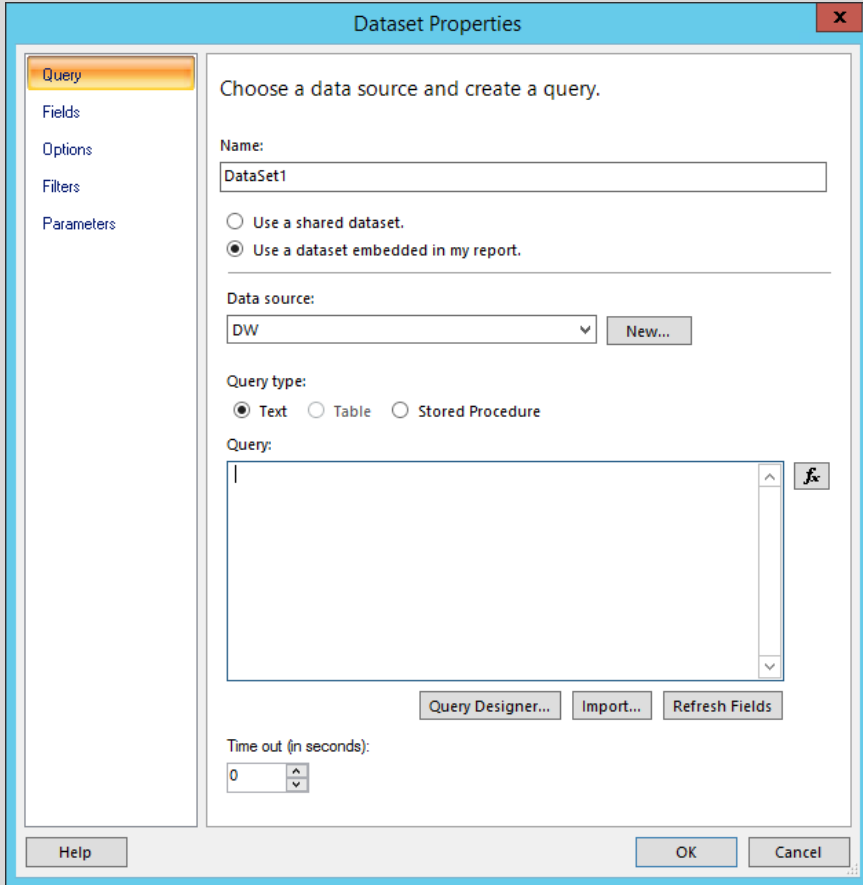
SQL Server, Oracle, etc.

## Connection string

Note: Can be defined using an expression

## Credentials

Tip: Prefer Windows Authentication



# Data Sets

## SQL SELECT Statement

Query type *Text*

*Query Designer* available

Full table

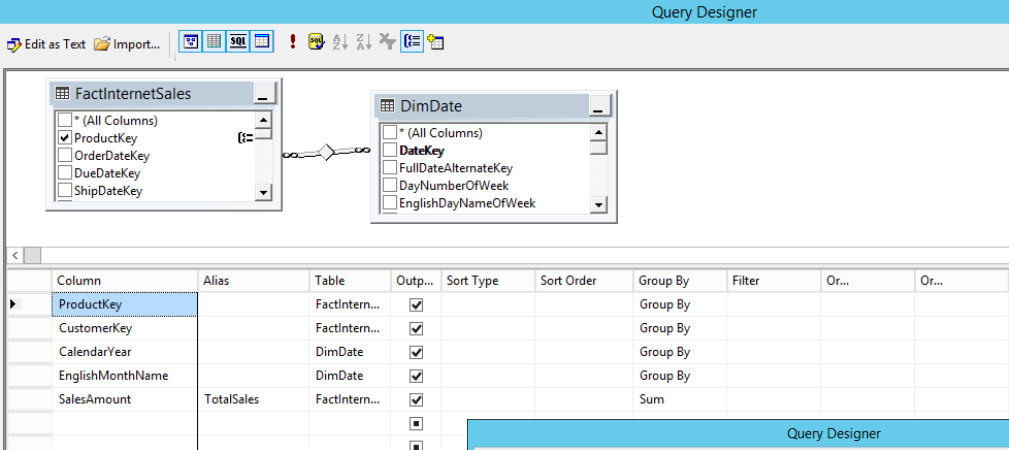
Stored procedure

# Queries

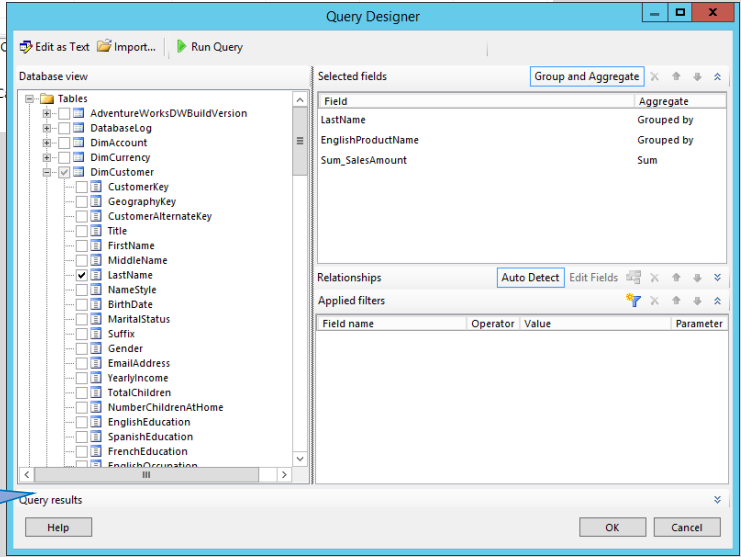
Query as text  
T-SQL

## Query Builders

Note: Query builders in VS and Report Builder are different  
Detailed reference see [Technet](#)



```
SELECT FactInternetSales.ProductKey, FactInternetSales.CustomerKey, DimDate.CalendarYear, FactInternetSales.SalesAmount FROM FactInternetSales INNER JOIN DimDate ON FactInternetSales.OrderDateKey = DimDate.DateKey GROUP BY FactInternetSales.ProductKey, FactInternetSales.CustomerKey, DimDate.CalendarYear
```



Query builder in Visual Studio

Query builder in Report Builder

# Demo

## Complex Data Set

Complex query with

[parameters](#)

Including NULL handling

```
select year(soh.OrderDate) as OrderYear,  
       month(soh.OrderDate) as OrderMonth,  
       p.ProductLine,  
       p.Name as ProductName,  
       st.Name as TerritoryName,  
       sum(sod.OrderQty * sod.UnitPrice) as Revenue  
from   Sales.SalesOrderHeader soh  
       inner join Sales.SalesOrderDetail sod  
         on soh.SalesOrderID = sod.SalesOrderID  
       inner join Production.Product p  
         on sod.ProductID = p.ProductID  
       inner join Sales.Customer c  
         on soh.CustomerID = c.CustomerID  
       inner join Sales.SalesTerritory st  
         on c.TerritoryID = st.TerritoryID  
where  year(soh.OrderDate) = @OrderYear  
       and (p.ProductLine=@ProductLine or @ProductLine is null)  
       and (p.ProductID = @ProductID or @ProductID is null)  
       and (st.TerritoryID = @TerritoryID or @TerritoryID is null)  
group by year(soh.OrderDate),  
         month(soh.OrderDate),  
         p.ProductLine,  
         p.Name,  
         st.Name
```

## Query

# Query

Dataset Properties

Choose a data source and create a query.

Name:  
Orders

Use a shared dataset.  
 Use a dataset embedded in my report.

Data source:  
OLTP New...

Query type:  
 Text  Table  Stored Procedure

Query:

```
select  year(soh.OrderDate) as OrderYear,
        month(soh.OrderDate) as OrderMonth,
        p.ProductLine,
        p.Name as ProductName,
        st.Name as TerritoryName,
        sum(sod.OrderQty * sod.UnitPrice) as Revenue
from    Sales.SalesOrderHeader soh
        inner join Sales.SalesOrderDetail sod on soh.SalesOrderID = sod.SalesOrderID
        inner join Production.Product p on sod.ProductID = p.ProductID
        inner join Sales.Customer c on soh.CustomerID = c.CustomerID
        inner join Sales.SalesTerritory st on c.TerritoryID = st.TerritoryID
```

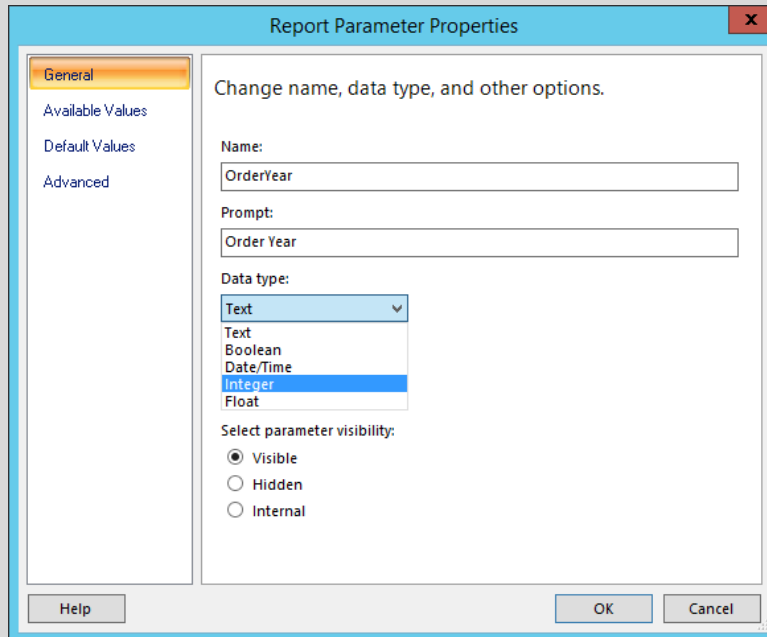
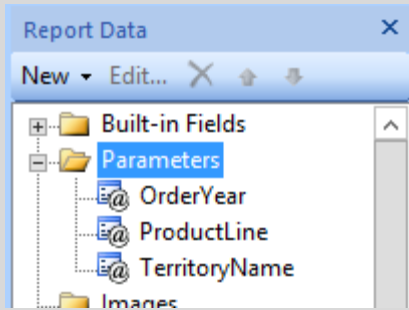
Query Designer... Import... Refresh Fields

Time out (in seconds):  
0

Help OK Cancel



# Parameters



Name: AvailableProductLines

Use a shared dataset.  
 Use a dataset embedded in my report.

Data source: OLTP

Query type:  Text  Table  Stored Procedure

Query: select distinct ProductLine from Production.Product

Report Parameter Properties

General  
Available Values  
Default Values  
Advanced

Choose the available values for this parameter.

Select from one of the following options:

None  
 Specify values  
 Get values from a query

Dataset: AvailableProductLines

Value field: ProductLine

Label field: ProductLine

Help

Order Year 2005

Territory Name <Select a Value>

Product Line M  
(Null)  
M  
R  
S  
T

# Parameters

List of values

Get LOV from Data Set

Detailed reference see [Technet](#)

Query:

```
select TerritoryID, Name from Sales.SalesTerritory
```

```
union  
select null, null
```

### Report Parameter Properties

General

Available Values

Default Values

Advanced

Change name, data type, and other options

Name:

TerritoryID

Prompt:

Territory ID

Data type:

Integer

Allow blank values ("")

Allow null value

Allow multiple values

Order Year

2005

Territory ID

(Null)

Australia

Canada

Central

France

Germany

Northeast

Northwest

Southeast

Southwest

United Kingdom

(Null)

Total

962,7

# Parameters

List of values

*Null* handling in LOVs

Query:  
select distinct ProductLine from Production.Product  
union  
select null

### Report Parameter Properties

Change name, data type, and other options.

**General**  
Available Values  
Default Values  
Advanced

Name: ProductLine

Prompt: Product Line

### Dataset Properties

Choose a data source and create a query.

Name: AvailableProducts

Use a shared dataset.  
 Use a dataset embedded in my report.

Data source: OLTP New...

Query type:  
 Text  Table  Stored Procedure

Query:  
select ProductID, Name from Production.Product where ProductLine =  
@ProductLine or @ProductLine is null  
union  
select null, null

# Parameters

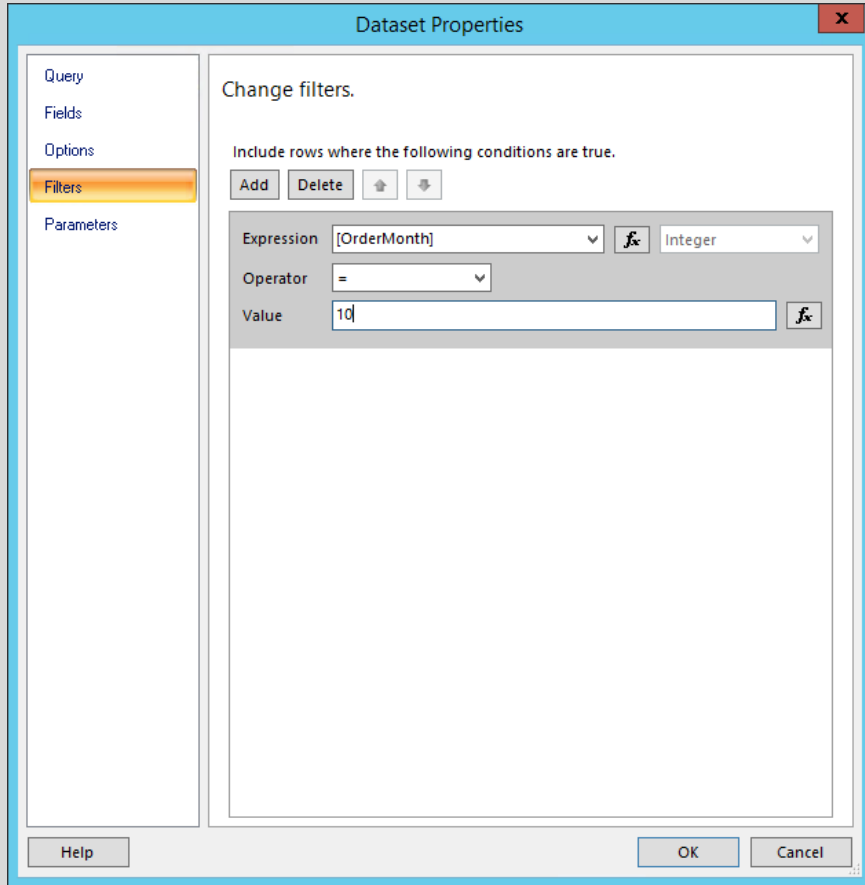
List of values

## Cascaded Parameter LOVs

SSRS cares for auto-refreshing of LOVs

Detailed reference see

[Technet](#)



## Data Set Filters

Filtering done by SSRS, not  
at the DB level  
Can still use parameters

Used for data set caching  
with long-running  
queries

# Data Set Best Practices

- ▶ Use shared data sources instead of embedded ones
- ▶ Use shared data sets for reusing complex queries
- ▶ Filter on the DB-level whenever possible
  - Reduces network traffic
  - Reduces load on SSRS servers

# Advanced Reporting

Advanced feature for report building

```

CREATE view [dbo].[vResellerSalesProdTerrDate] as
select top 100 percent
    d.CalendarYear
    , d.CalendarQuarter
    , d.MonthNumberOfYear
    , d.EnglishMonthName MonthName
    , pc.EnglishProductCategoryName Category
    , sc.EnglishProductSubcategoryName Subcategory
    , p.EnglishProductName ProductName
    , st.SalesTerritoryGroup
    , st.SalesTerritoryRegion
    , st.SalesTerritoryCountry
    , sum(s.SalesAmount) SalesAmt
    , sum(s.OrderQuantity) OrderQty
    , sum(s.Freight) Frieght
from
    FactResellerSales s inner join DimDate d on s.OrderDateKey = d.DateKey
    inner join DimProduct p on s.ProductKey = p.ProductKey
    inner join DimProductSubcategory sc on p.ProductSubcategoryKey = sc.ProductSubcategoryKey
    inner join DimProductCategory pc on sc.ProductCategoryKey = pc.ProductCategoryKey
    inner join DimSalesTerritory st on s.SalesTerritoryKey = st.SalesTerritoryKey
group by
    d.CalendarYear
    , d.CalendarQuarter
    , d.MonthNumberOfYear
    , d.EnglishMonthName
    , pc.EnglishProductCategoryName
    , sc.EnglishProductSubcategoryName
    , p.EnglishProductName
    , st.SalesTerritoryGroup
    , st.SalesTerritoryRegion
    , st.SalesTerritoryCountry
order by
    d.CalendarYear
    , d.CalendarQuarter
    , d.MonthNumberOfYear
    , pc.EnglishProductCategoryName
    , sc.EnglishProductSubcategoryName
    , p.EnglishProductName
    , st.SalesTerritoryGroup
    , st.SalesTerritoryRegion
    , st.SalesTerritoryCountry

```

# Demo

## Prerequisites

### Create a view with revenue data

*AdventureWorksDW2012* DB



```
SELECT CalendarYear, CalendarQuarter, MonthNumberOfYear,
       MonthName, Category, Subcategory, ProductName,
       SalesTerritoryGroup, SalesTerritoryRegion,
       SalesTerritoryCountry, SalesAmt, OrderQty, Freight
FROM vResellerSalesProdTerrDate
```

Year	Quarter	Month	Sales Amt	Order Qty
[CalendarYear]	[CalendarQuarter]	[MonthName]	um(SalesAmt)]	Sum(OrderQty)]
	Total		um(SalesAmt)]	um(OrderQty)]
Total			um(SalesAmt)]	um(OrderQty)]

## Hands-On Lab

Create this report and format it appropriately

# Built-in Aggregation Functions

- Note: If you do not specify an aggregation function although there are multiple values, the *first* value of the group is displayed

Function	Description
Avg	Returns the average of all non-null numeric values specified by the expression, evaluated in the given scope.
Count	Returns a count of non-null values specified by the expression, evaluated in the context of the given scope.
CountDistinct	Returns a count of all distinct non-null values specified by the expression, evaluated in the context of the given scope.
Max	Returns the maximum value of all non-null numeric values specified by the expression, in the context of the given scope. You can use this for specifying a chart axis maximum value to control the scale.
Min	Returns the minimum value of all non-null numeric values specified by the expression, in the context of the given scope. You can use this for specifying a chart axis minimum value to control the scale.
StDev	Returns the standard deviation of all non-null numeric values specified by the expression, evaluated in the given scope.
StDevP	Returns the population standard deviation of all non-null numeric values specified by the expression, evaluated in the context of the given scope.
Sum	Returns the sum of all the non-null numeric values specified by the expression, evaluated in the given scope.
Union	Returns the union of all the non-null spatial data values of type <b>SqlGeometry</b> or <b>SqlGeography</b> that are specified by the expression, evaluated in the given scope.
Var	Returns the variance of all non-null numeric values specified by the expression, evaluated in the given scope.
VarP	Returns the population variance of all non-null numeric values specified by the expression, evaluated in the context of the given scope.

Demo

# Scopes

Aggregate functions and scopes

# Scope Demo Report

Sales Amount	Sum(Sales Amount)	Sum(Fields !Sales Amount.Value, "DataSet")
159.00€	29,358,677.22€	29,358,677.22€

Name of the Data Set

2007

Sales Amount	Sum(Sales Amount)	Sum(Fields !Sales Amount.Value, "DataSet")
159.00€	9,791,080.30€	29,358,677.22€

Name of the details grouping of year list

Order Month	Sum(Sales Amount)	Sum(Fields !Sales Amount.Value, "YearDetails")
1	438,885.17€	9,791,080.30€
2	489,090.34€	9,791,080.30€
3	485,574.79€	9,791,080.30€
4	506,399.27€	9,791,080.30€
5	562,772.56€	9,791,080.30€
6	554,799.23€	9,791,080.30€
7	886,668.84€	9,791,080.30€
8	847,413.51€	9,791,080.30€
9	1,010,258.13€	9,791,080.30€
10	1,080,449.58€	9,791,080.30€
11	1,196,981.11€	9,791,080.30€
12	1,731,787.77€	9,791,080.30€
Total revenue in 2007	9,791,080.30€	
Revenue of Jan. in 2007	438,885.17€	Sum(IIf(Fields!OrderMonth.Value=1,Fields!SalesAmount.Value,CDec(0.00)))

List grouped by OrderYear

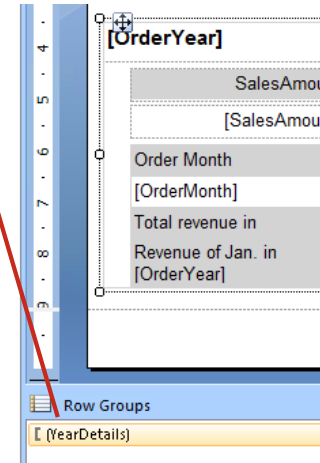
2008

Sales Amount	Sum(Sales Amount)	Sum(Fields !Sales Amount.Value, "DataSet")
159.00€	9,770,899.74€	29,358,677.22€

Order Month	Sum(Sales Amount)	Sum(Fields !Sales Amount.Value, "YearDetails")
1	1,340,244.95€	9,770,899.74€
2	1,462,479.83€	9,770,899.74€

## Scopes

Use *Lists* to combine report items and data regions



# Scopes

Report Definition

## Scope Demo Report

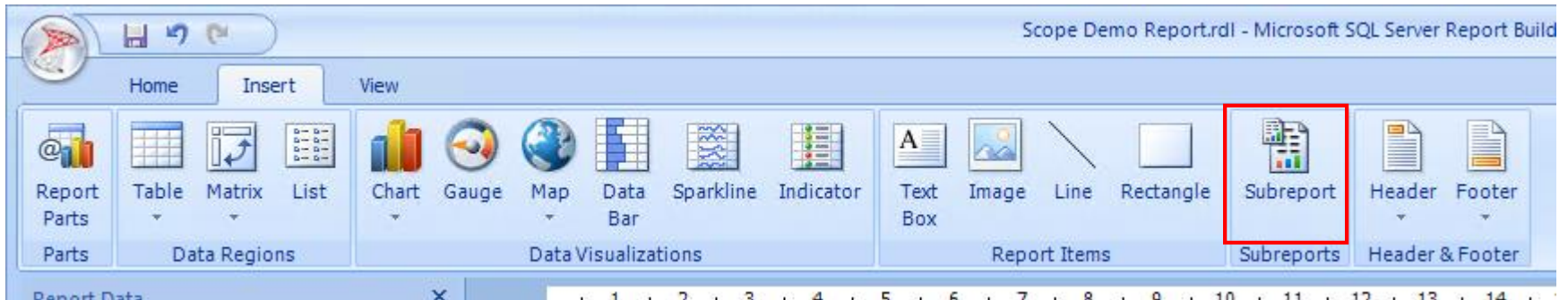
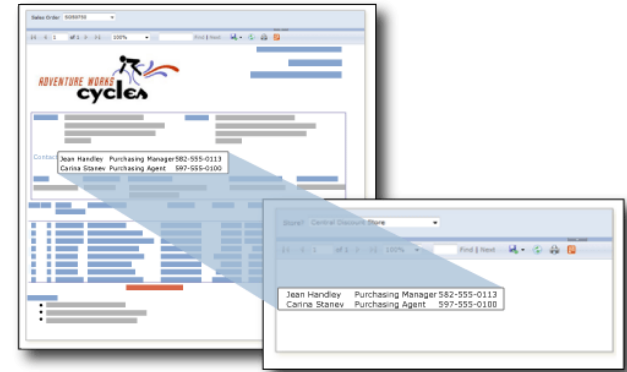
SalesAmount	Sum(SalesAmount)	Sum(Fields!SalesAmount.Value, "DataSet")
[SalesAmount]	[Sum(SalesAmount)]	«Expr»

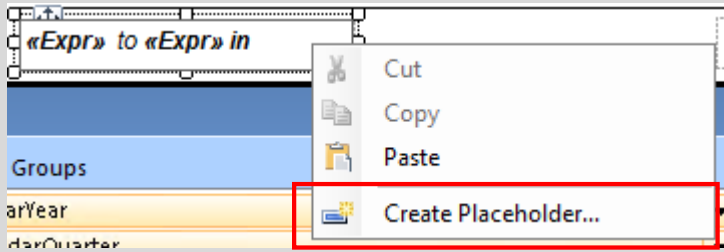
### [OrderYear]

SalesAmount	Sum(SalesAmount)	Sum(Fields!SalesAmount.Value, "DataSet")
[SalesAmount]	[Sum(SalesAmount)]	«Expr»
Order Month	Sum(SalesAmount)	Sum(Fields!SalesAmount.Value, "YearDetails")
[OrderMonth]	[Sum(SalesAmount)]	«Expr»
Total revenue in	[Sum(SalesAmount)]	
Revenue of Jan. in [OrderYear]	«Expr»	Sum(Iif(Fields!OrderMonth.Value=1,Fields! SalesAmount.Value,CDec(0.00)))

# Sub-Reports

- ▶ Combine multiple reports  
Pass parameters to sub report
- ▶ Avoid if possible, use DWH-approach instead  
High load on DB servers  
Rendering problems
- ▶ Detailed reference see [Technet](#)



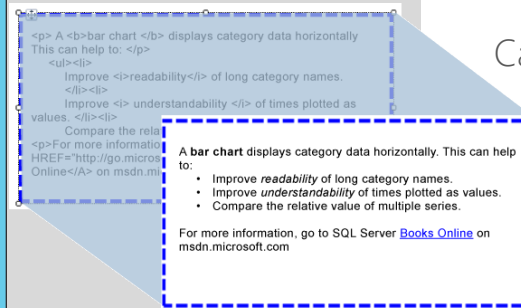
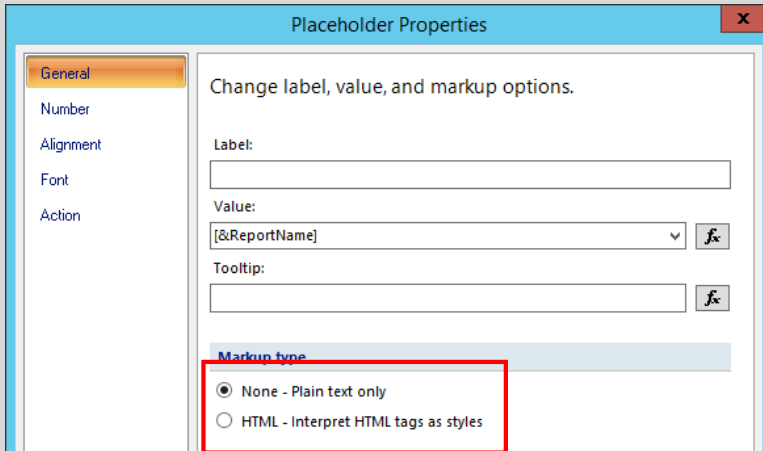
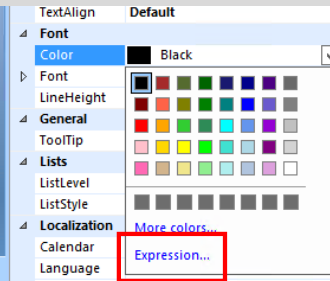


CalendarQuarter	MonthName	um(SalesAmt)	Sum(OrderQty)
Total		um(SalesAmt)	um(OrderQty)
		um(SalesAmt)	um(OrderQty)

or is **confidential** and should only be used in accordance to corporate policy

Page [PageNumber]

Column Groups



# Text Boxes

Detailed reference for formatting textboxes see [Technet](#)

Single-value expression

Range of text

Multiple expression placeholders

Range of text can be individually formatted

Can contain [HTML tags](#)

Group Properties

Set additional group properties.

Recursive parent

Filters

Document map

[EnglishProductName]

Help OK Cancel

Row Groups

Column Groups

ComponentMetadata

# Document Maps

Detailed reference see [Technet](#)

Views Zoom

Navigation

Setup Layout

Print

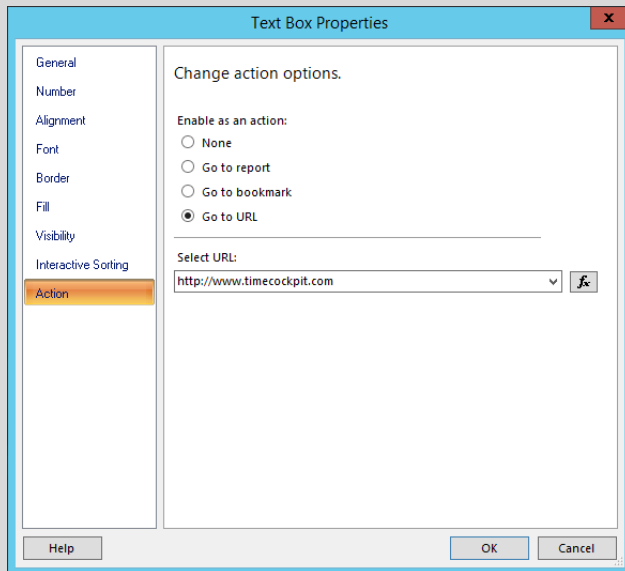
Untitled

- All-Purpose Bike Stand
- AWC Logo Cap
- Bike Wash - Dissolver
- Classic Vest, L
- Classic Vest, M
- Classic Vest, S
- Fender Set - Mountain
- Half-Finger Gloves, L
- Half-Finger Gloves, M
- Half-Finger Gloves, S

Hitch Rack - 4-Bike 39,360.00€

Adams	240.00€
Alexander	360.00€
Alonso	240.00€
Alvarez	240.00€
Anand	240.00€
Andersen	240.00€
Anderson	240.00€





Other	
Bookmark	Summary
CustomProperties	

# Links

Drill-through reports

Navigate to a different report

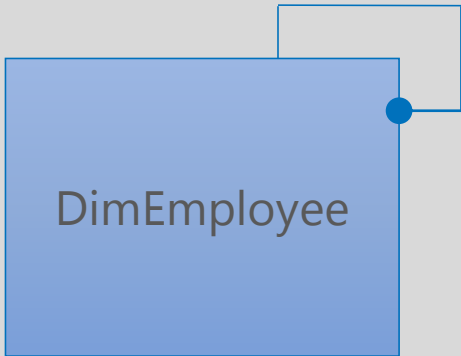
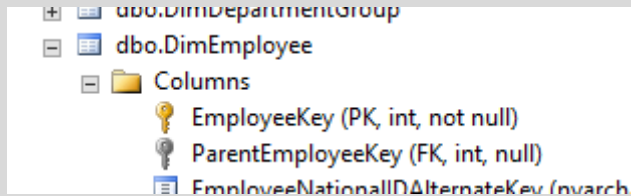
Aka [Drill-Through Reports](#)

Parameters for filtering can be passed

Navigate to bookmarks

Navigate to websites

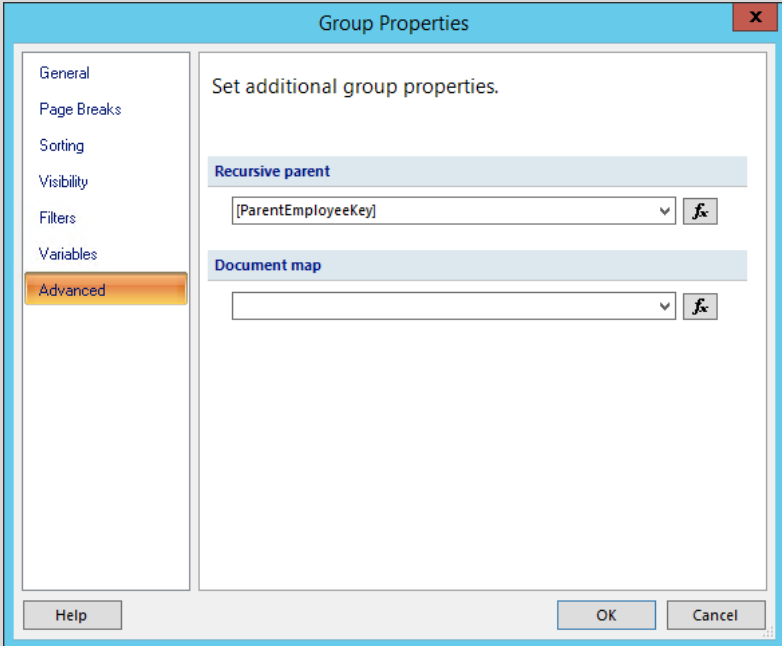
Tip: You can build URL using an expression



# Recursive Parents

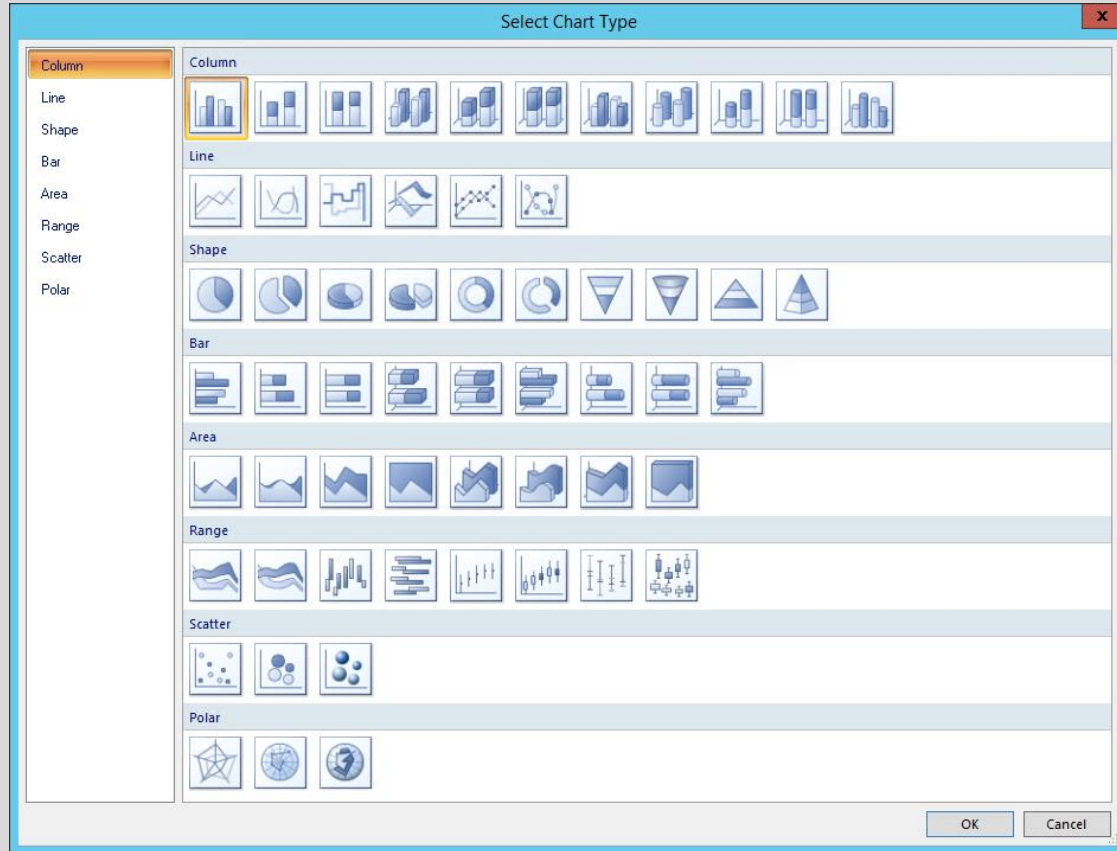
Tip: Use *Level()* function

Detailed reference see [Technet](#)

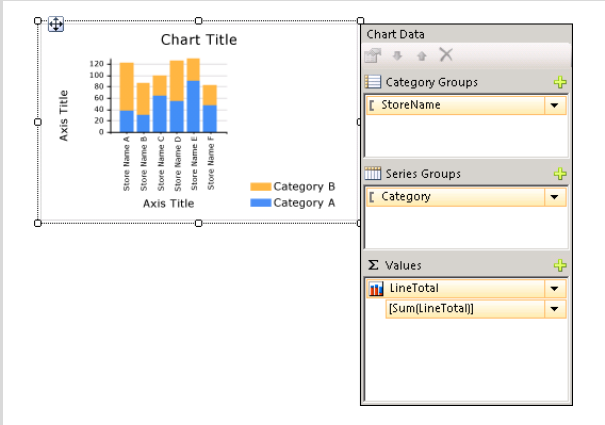
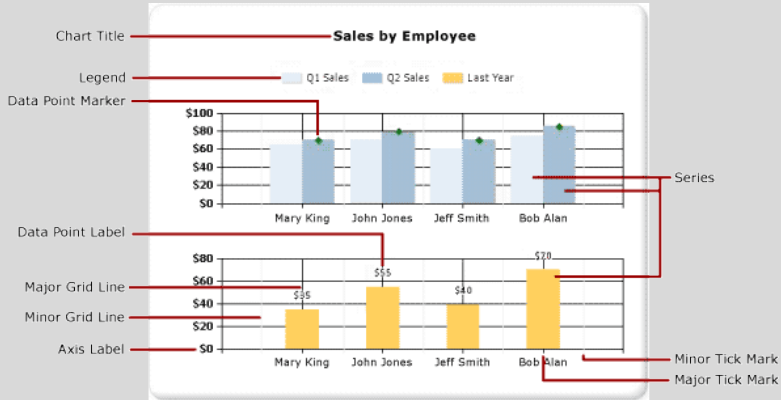


# Charts

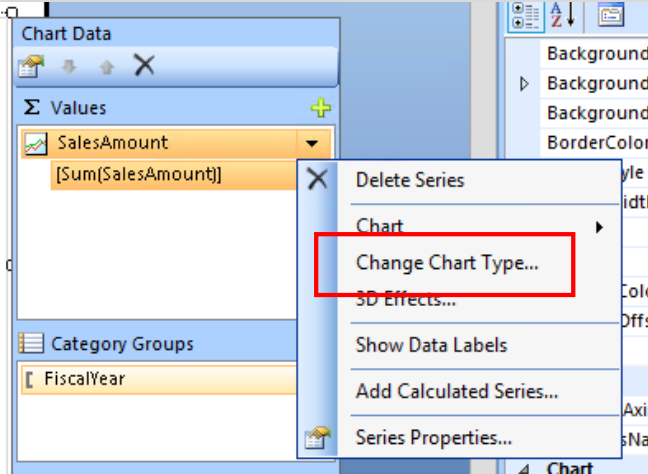
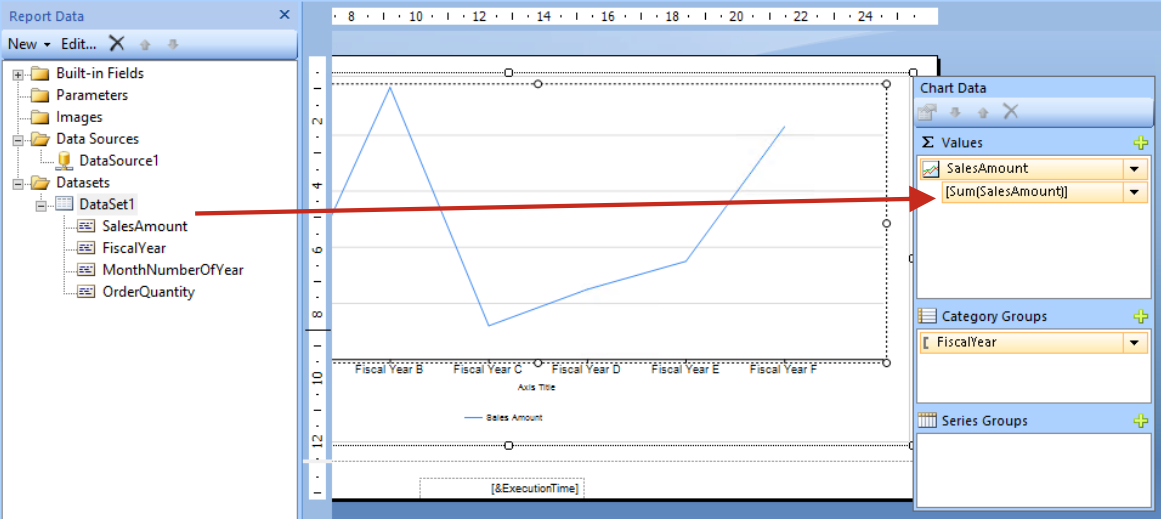
# Chart Types



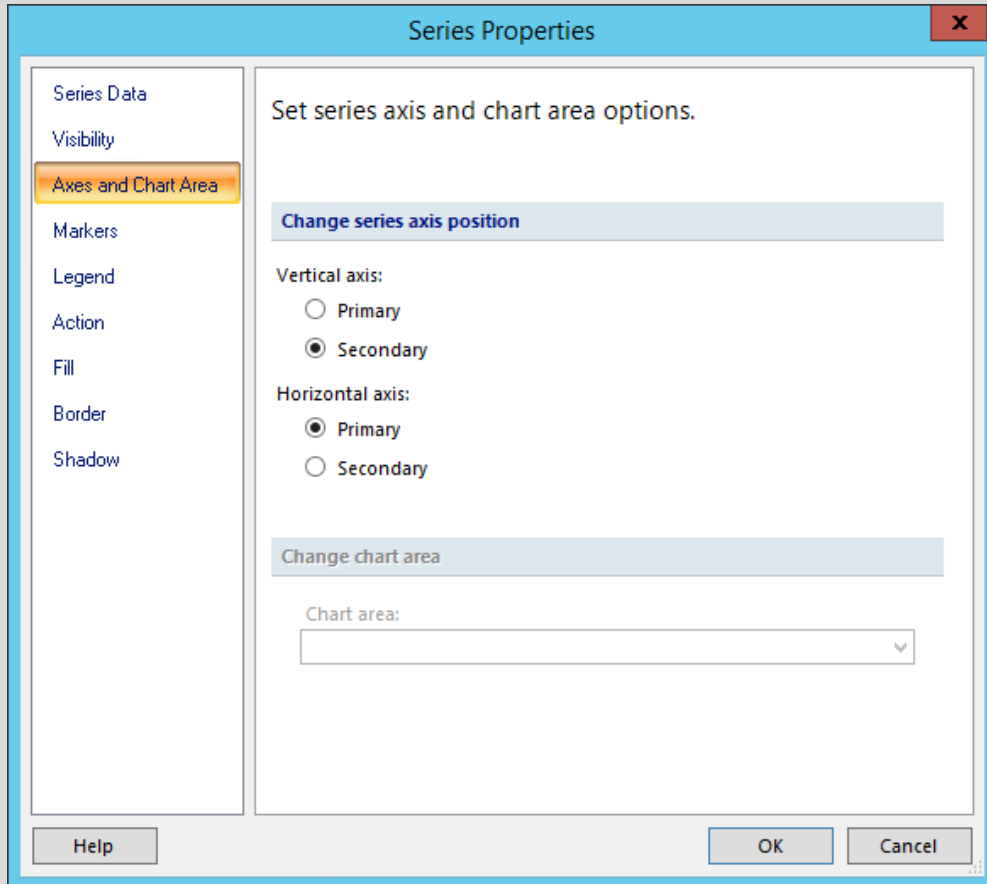
# Defining Chart



# Defining Chart

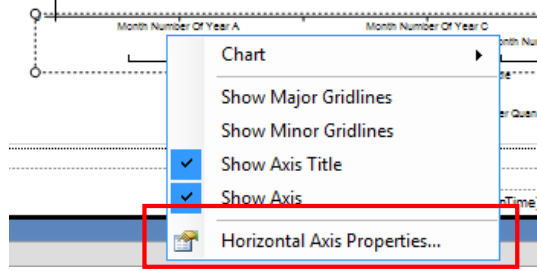


# Chart Axis



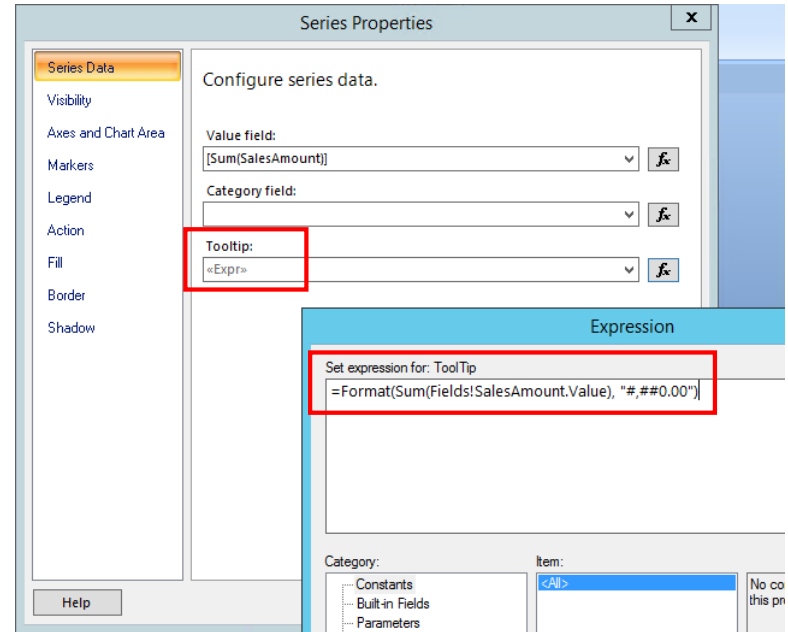
# Chart Tips & Tricks

## ► Configure axis



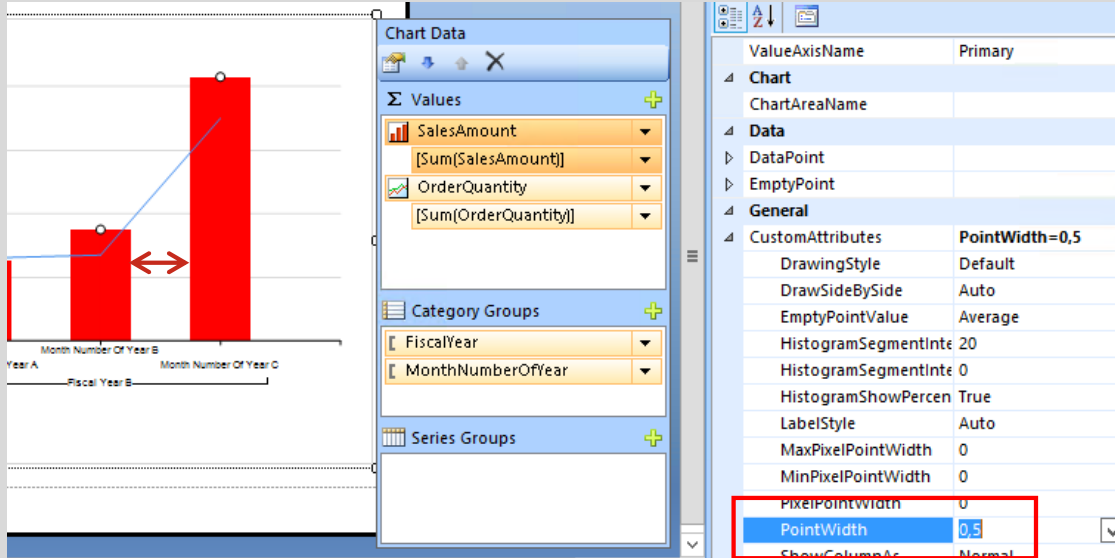
## ► Use *Color* property to change the color of a data series

## ► Add tooltips to data values





# Chart Tips & Tricks



Change distance between chart series

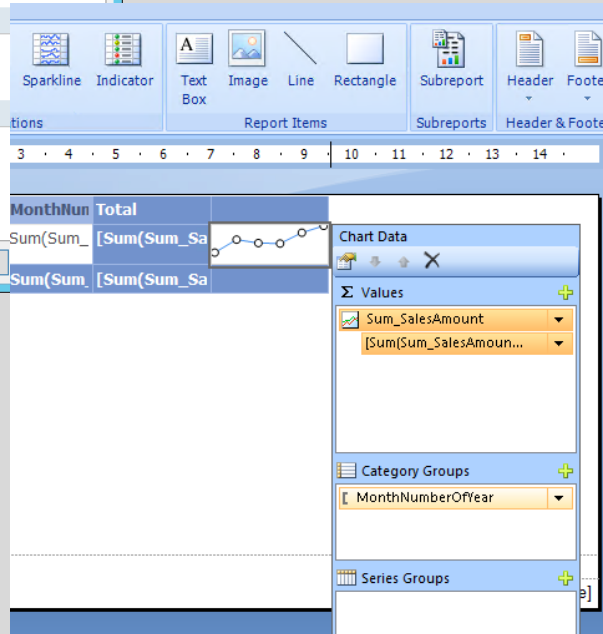
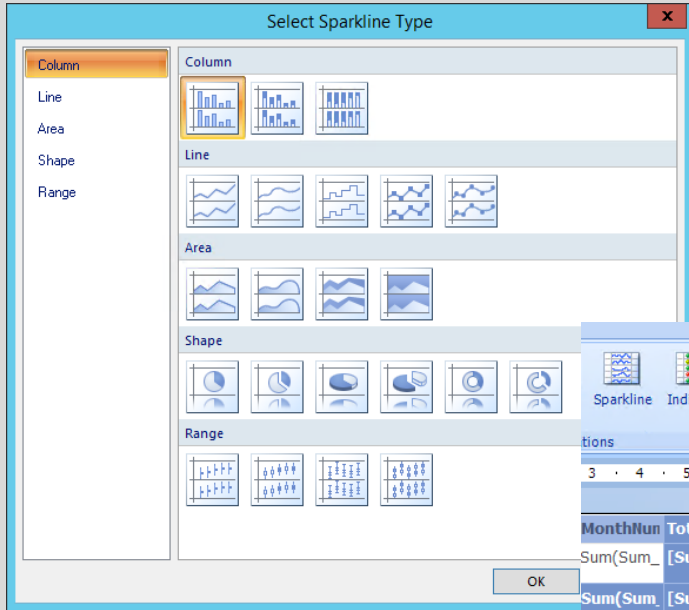
Additional tips & tricks see [Formatting a Chart in Technet](#)

# Sparklines

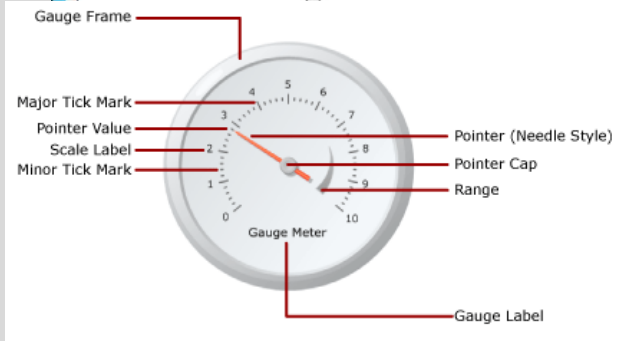
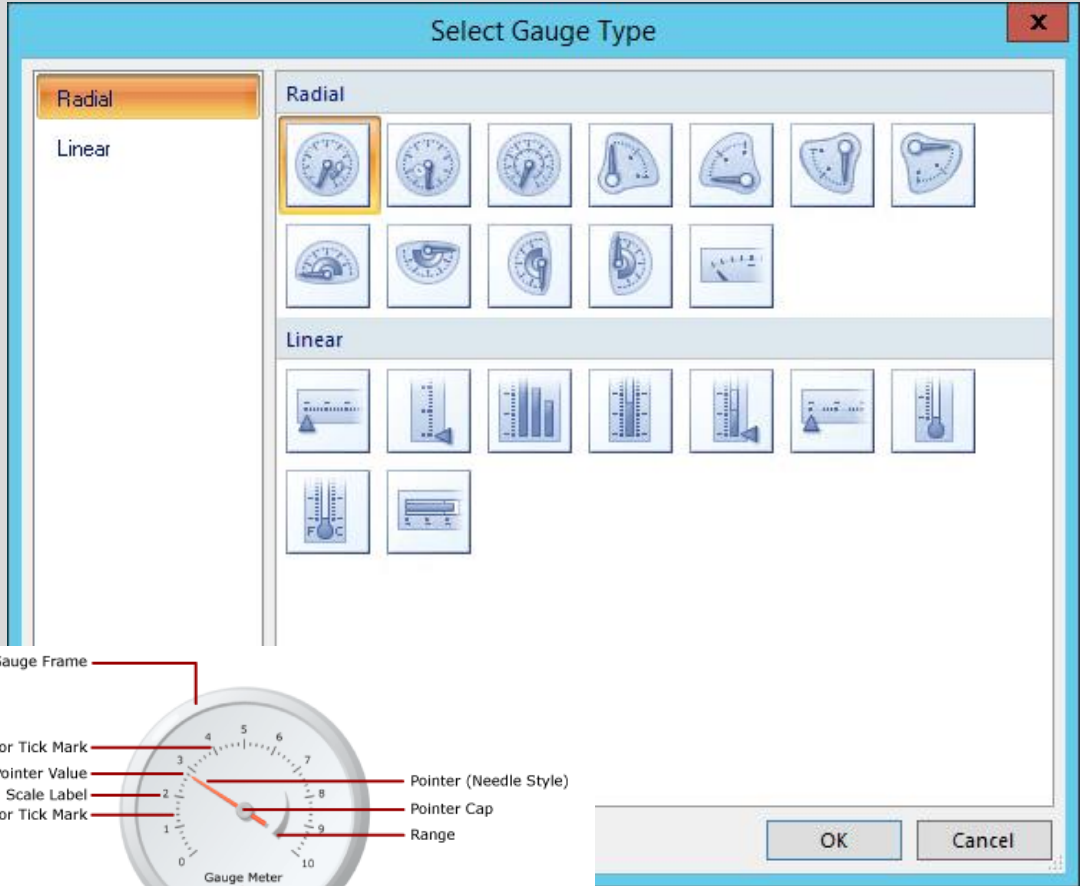
Chart in a nested data region

Detailed reference see

[Technet](#)



# Gauges



# Data Visualization

Indicators and Gauges combined

# Demo

```
with AggregatedResellerSales as (  
    select frs.EmployeeKey,  
           dd.CalendarYear,  
           dd.CalendarQuarter,  
           sum(frs.SalesAmount) as TotalSalesAmount  
    from dbo.FactResellerSales frs  
         inner join dbo.DimDate dd on  
             frs.OrderDateKey = dd.DateKey  
    group by frs.EmployeeKey,  
             dd.CalendarYear,  
             dd.CalendarQuarter  
)  
select ars.EmployeeKey,  
       ars.CalendarYear,  
       ars.CalendarQuarter,  
       ars.TotalSalesAmount,  
       fsq.SalesAmountQuota,  
       ars.TotalSalesAmount / fsq.SalesAmountQuota as Ratio  
from AggregatedResellerSales ars  
     full join dbo.FactSalesQuota fsq on  
         ars.EmployeeKey = fsq.EmployeeKey  
         and ars.CalendarYear = fsq.CalendarYear  
         and ars.CalendarQuarter = fsq.CalendarQuarter
```

# Gauge Demo

Sample Query

	<<Expr>>	Total
Calendar Year	Quota %	Quota %
[EmployeeKey]	<<Expr>>	<<Expr>>
[CalendarYea	<<Expr>>	<<Expr>>
Total	<<Expr>>	<<Expr>>

= Sum(Fields!TotalSalesAmount.Value) /  
Sum(Fields!SalesAmountQuota.Value)



	Quarter 1	Quarter 2	Quarter 3	Quarter 4	Total
Calendar Year	Quota %	Quota %	Quota %	Quota %	Quota %
272	88%	79%	83%	86%	84%
281	87%	87%	82%	78%	83%
2005	NaN	NaN	83%	80%	81%
2006	81%	81%	87%	74%	81%
2007	94%	85%	78%	82%	83%
2008	86%	93%	NaN	NaN	90%
282	91%	91%	83%	88%	88%
283	84%	81%	82%	83%	83%
284	85%	75%	85%	87%	83%
285	83%	87%	80%	87%	84%
286	85%	85%	141%	78%	94%
287	85%	80%	81%	85%	82%
288	91%	87%	79%	80%	83%
289	118%	92%	81%	92%	93%
290	86%	87%	79%	85%	84%
291	88%	80%	77%	83%	81%
292	87%	85%	79%	77%	81%
293	82%	89%	76%	86%	84%
294	76%	82%	85%	85%	84%
295	92%	78%	67%	84%	78%
296	84%	91%	71%	93%	84%
Total	88%	84%	82%	83%	84%



# Gauge Demo

Indicators and Gauge combined

# Expressions

SSRS Expression Language

# Expression Types

## Simple Expressions

- ▶ Reference to a single item in a built-in collection (e.g. field)

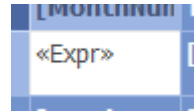


Set expression for: Value

=Fields!EnglishProductName.Value

## Complex Expressions

- ▶ Contains references to multiple items, operators, functions, etc.



Set expression for: Value

=Sum(If(Fields!EnglishProductName.Value="Bike",Fields!Sum\_SalesAmount.Value,0))



# Expression Reference

- ▶ Data Types

<http://technet.microsoft.com/en-us/library/dd255246.aspx>

- ▶ Built-in functions

[System.Math](#)

[System.Convert](#)

[VB Runtime Library](#)

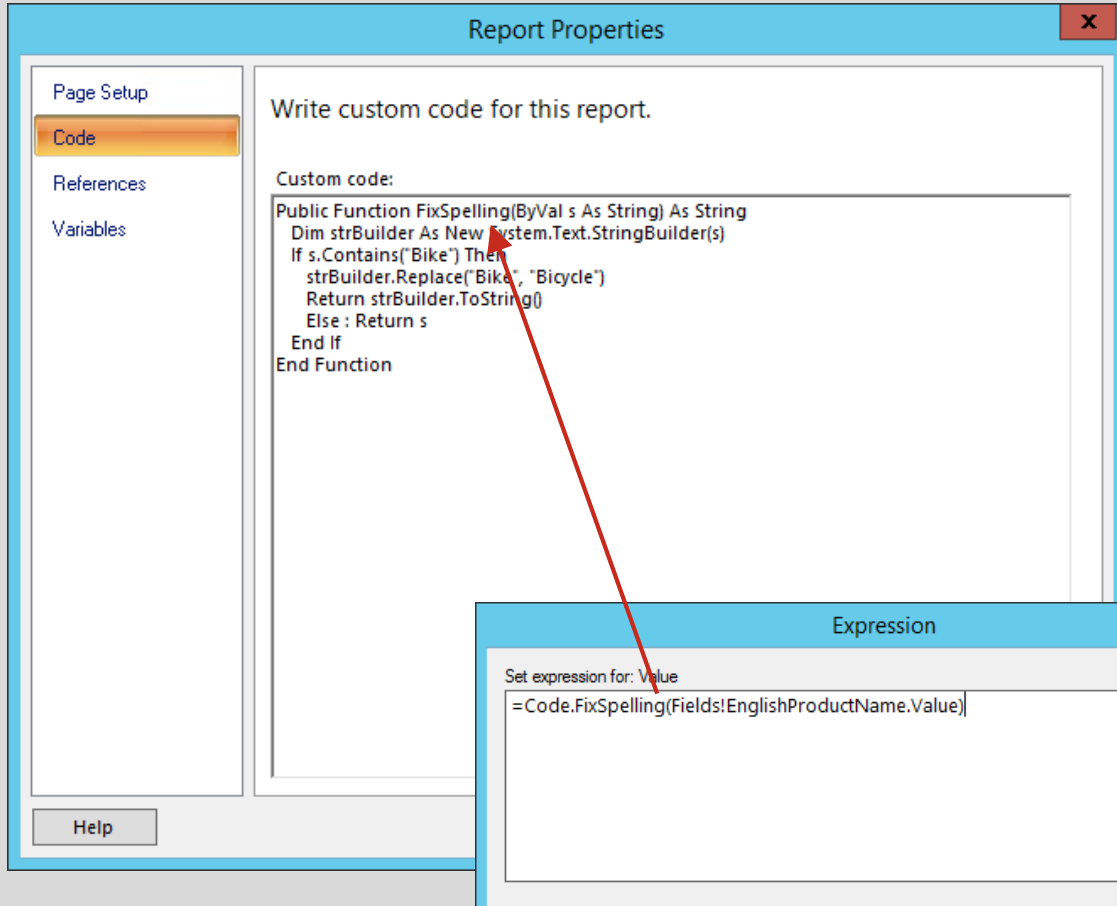
- ▶ Built-in aggregate functions

<http://technet.microsoft.com/en-us/library/dd255275.aspx>

- ▶ Built-in collections

<http://technet.microsoft.com/en-us/library/dd255235.aspx>

# Add Custom Code



The image shows two overlapping windows from a reporting application. The top window is titled "Report Properties" and has a sidebar with "Code" selected. The main area contains a text box with the following VB code:

```
Write custom code for this report.

Custom code:

Public Function FixSpelling(ByVal s As String) As String
    Dim strBuilder As New System.Text.StringBuilder(s)
    If s.Contains("Bike") Then
        strBuilder.Replace("Bike", "Bicycle")
    Return strBuilder.ToString()
    Else : Return s
    End If
End Function
```

A red arrow points from the code in the "Expression" dialog box below to the "FixSpelling" function in the "Report Properties" window. The "Expression" dialog box is titled "Expression" and contains the text "Set expression for: Value" and the expression "=Code.FixSpelling(Fields!EnglishProductName.Value)".

Use VB to embed custom code  
Report properties/Code

Detailed reference see [Technet](#)

Alternative: Custom  
Assemblies

Detailed reference see [Technet](#)

# Expression Language Tips & Tricks

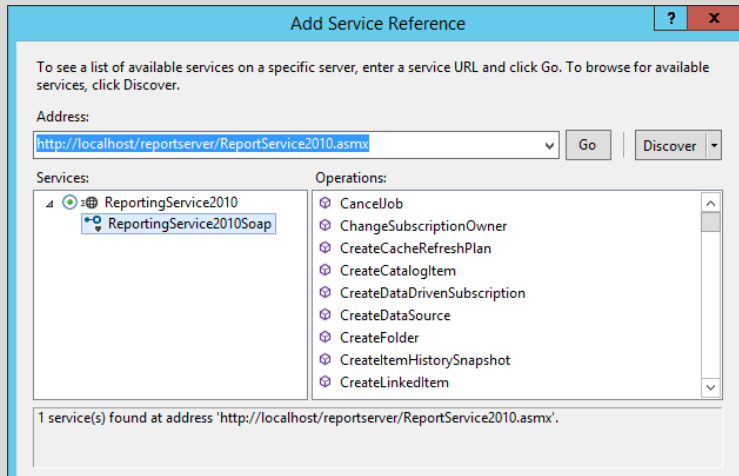
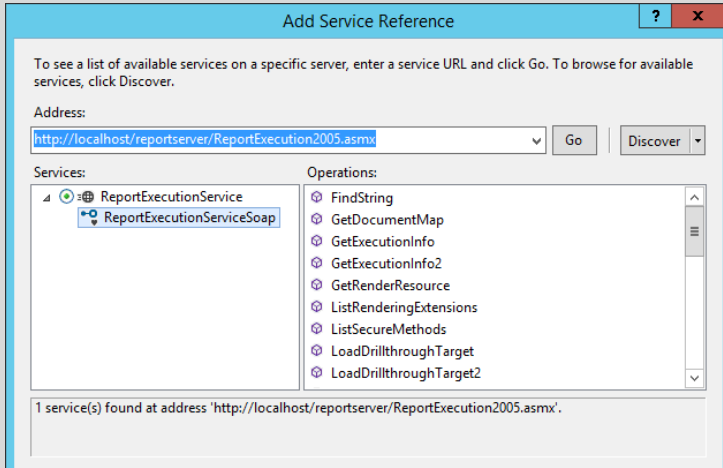
- ▶ Use `vbCrLf` to add line break in string constant
- ▶ Use `Join` function to concat multiple values (e.g. multi-valued filter, result of `LookupSet`)
- ▶ Use `Lookup` and `LookupSet` to combine multiple data sets  
See also [Blog post](#)
- ▶ Use `Switch` instead of `if` to combine multiple conditions
- ▶ `First` and `Last` are useful in page header/footer  
Use `First(ReportItems("LastName").Value)` to get first value on page
- ▶ Use `RowNumber` to get row number  
Also useful to control page breaks or zebra striping

# Programming SSRS

# Web Services

## Add Service References in Visual Studio

Detailed reference see [Technet](#)



# Demo

## Web Services

Rendering a report using  
web services

```

// Prepare report parameter.
REService.ParameterValue[] parameters = GetReportExecutionParameters();

// make sure the report either has parameters that are set or has no parameters.
if ((_reportHasParameters && parameters.Length != 0) || !_reportHasParameters) {
    // Load the report, set the parameters and then render.
    _rsExec.LoadReport(reportItem.Path, historyID);
    _rsExec.SetExecutionParameters(parameters, "en-us");
    result = _rsExec.Render(selectedFormat.Name,
        devInfo,
        out extension,
        out encoding,
        out mimeType,
        out warnings,
        out streamIDs);

    // Make sure there is an output path then output the file to the file system.
    if (txtOutputFolder.Text != "") {
        string fullOutputPath = txtOutputFolder.Text + "\\\"
            + reportItem.Name + selectedFormat.Extension;
        FileStream stream = File.Create(fullOutputPath, result.Length);
        stream.Write(result, 0, result.Length);
        stream.Close();
        MessageBox.Show("Report Rendered to: " + fullOutputPath);
    }
    else {
        MessageBox.Show("Choose a folder first");
    }
}
else {
    MessageBox.Show("No parameters, click Get Parameters button
        first and then set values.");
}
}

```

# Web Services

## Rendering a report using web services

Report Viewer

Demo



# Report Viewer

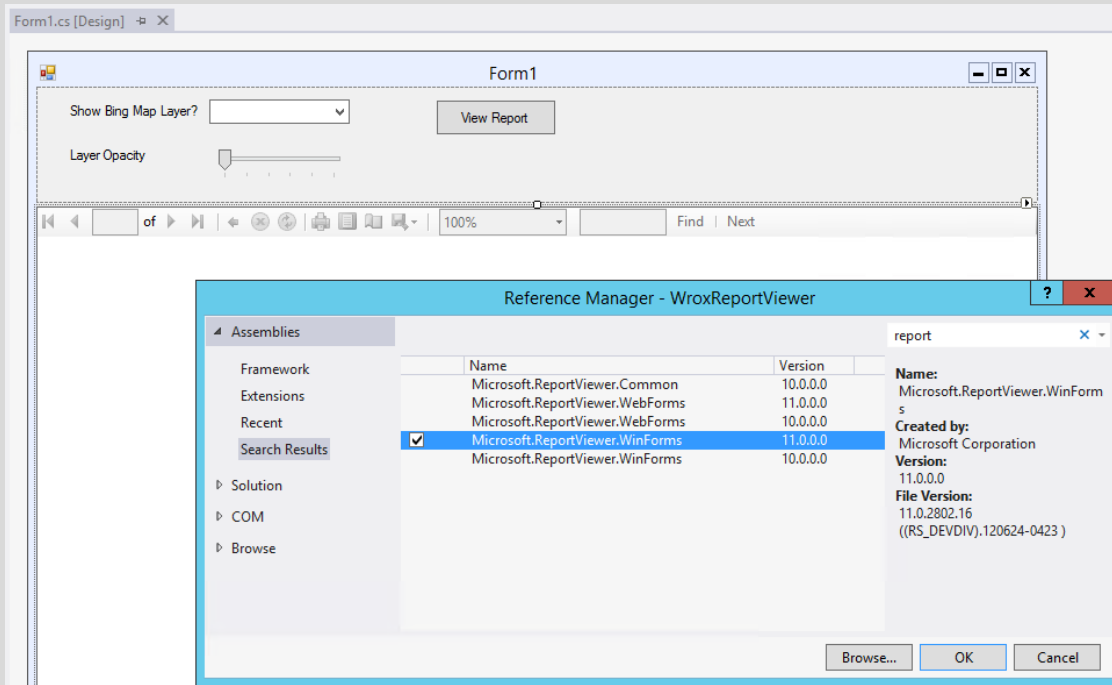
WinForms

[Local](#) vs. [remote](#) processing mode

Report Viewer overview see [Technet](#)

Detailed reference see [Technet](#)

Samples and Walkthroughs see [Technet](#)



## Trainingsunterlagen

# Q&A

Thank your for coming!



## Rainer Stropek

software architects gmbh

Mail  
Web  
Twitter

[rainer@timecockpit.com](mailto:rainer@timecockpit.com)  
<http://www.timecockpit.com>  
[@rstropek](https://twitter.com/rstropek)



**time cockpit**  
Saves the day.



**time cockpit** is the leading time tracking solution for knowledge workers. Graphical time tracking calendar, automatic tracking of your work using signal trackers, high level of extensibility and customizability, full support to work offline, and SaaS deployment model make it the optimal choice especially in the IT consulting business.

Try **time cockpit** for free and without any risk. You can get your trial account at <http://www.timecockpit.com>. After the trial period you can use **time cockpit** for only 0,20€ per user and day without a minimal subscription time and without a minimal number of users.



**time cockpit** ist die führende Projektzeiterfassung für Knowledge Worker. Grafischer Zeitbuchungskalender, automatische Tätigkeitsaufzeichnung über Signal Tracker, umfassende Erweiterbarkeit und Anpassbarkeit, volle Offlinefähigkeit und einfachste Verwendung durch SaaS machen es zur Optimalen Lösung auch speziell im IT-Umfeld.

Probieren Sie **time cockpit** kostenlos und ohne Risiko einfach aus. Einen Testzugang erhalten Sie unter <http://www.timecockpit.com>. Danach nutzen Sie **time cockpit** um nur 0,20€ pro Benutzer und Tag ohne Mindestdauer und ohne Mindestbenutzeranzahl.