# FitBPO / Crunch Reports

## Design, Installation, Setup and Deployment

**Development and documentation**
**by**
**Randy Hoover**
**(for Global BI Solutions)**

**1st draft**
**04/15/2011**

## The Reports

There are 4 separate reports discussed in this documentation – below are the contents of each:

1) MAS Reports
    a. Region
    b. RDS
    c. Club
    d. Member Advisors
    e. Ranking
    f. MAS Corp
2) PPR Trainer
    a. Sorted by Revenue ranking
    b. Sorted by Club
3) PPR PMT
    a. DepartmentCode = 845
4) PPR APMT
    a. DepartmentCode = 848

## Report Design

Each of these reports were designed using MS SQL Server 2008 Business Intelligence Development Studio. The report files extensions are .rdl with an xml code file that can be modified as pure ascii text. Below is a list of each of the files that comprise this report package:

1) MASReport.rdl
    a. DataSource: Data Source=SSRST-01;Initial Catalog=MassDataWarehouse
    b. DataSets:
        i. DataSet1: dbo.spMarketSalesReportMAS
        ii. DataSet2: dbo.spMarketSalesReportRDS
        iii. DataSet3: dbo.spMarketSalesReportCLUB
        iv. DataSet4: dbo.spMarketSalesReportMEMADV
        v. DataSet5: dbo.spMarketSalesReportRANK
        vi. DataSet6: dbo.spMarketSalesReportCORP
2) PPRTrainer.rdl
    a. DataSource: Data Source=SSRST-01;Initial Catalog=MassDataWarehouse
    b. DataSet1: dbo.spMarketSalesReportPPR
    c. DataSet1: dbo.spMarketSalesReportPPR (Sorted by Club)
3) PPRPMTTrainer.rdl
    a. DataSource: Data Source=SSRST-01;Initial Catalog=MassDataWarehouse
    b. DataSet1: dbo.spMarketSalesReportPPRPTM (Parameter for DepartmentCode)

All reports share the same MassDataWarehouse database and all Stored Procedures have @ReportDate as a basic input parameter. This allows the user to either manually or programmatically initialize each report deployed on the Report Server. The only variance is with the spMarketSalesReportPPRPTM Stored Procedure that also allows a 2nd parameter, @DepartmentCode, for the DepartmentCode as discussed above.

The report associated with the spMarketSalesReportCORP Stored Procedure is not 100% completed or approved. There are actually 3 parts to this report page:

1) PD (not included)
2) B2B/CEP (not included)
3) Totals (included, but correct business rules and database mapping was never fully given to the developer)

## Stored Procedures

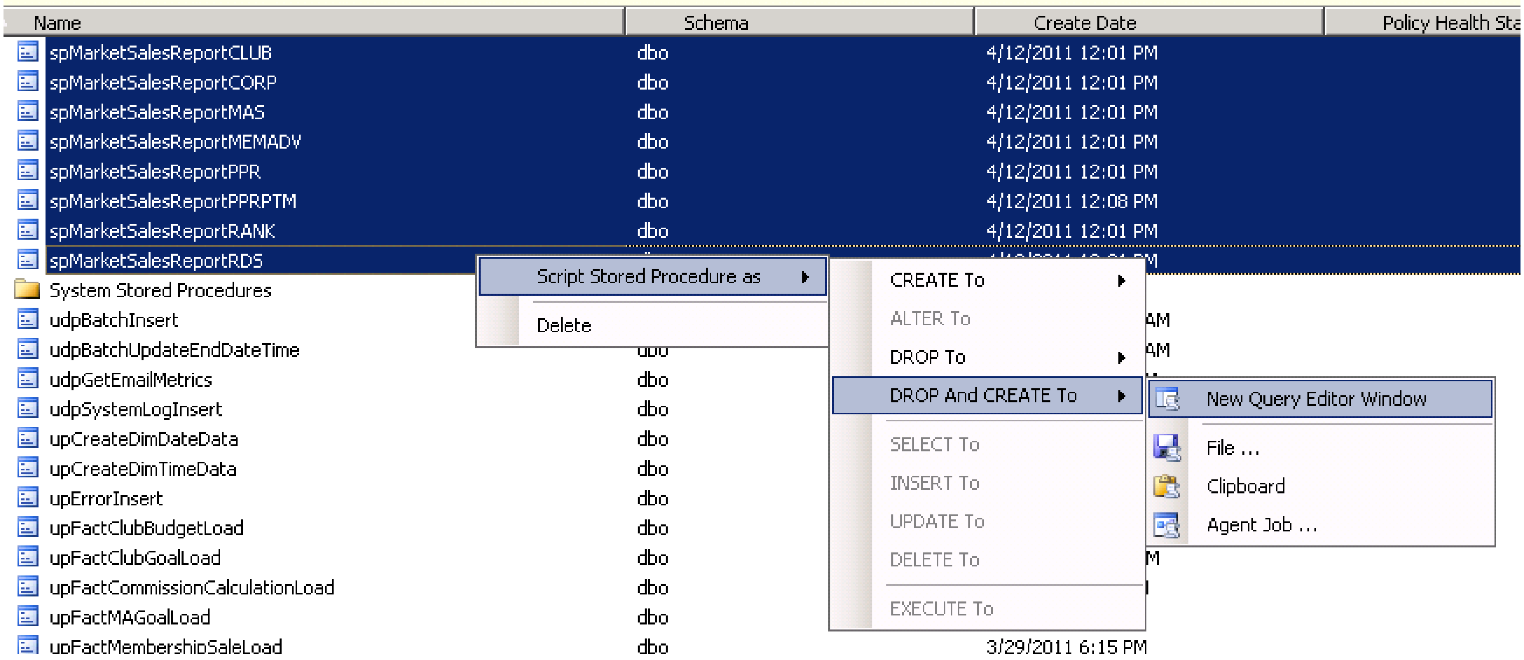
Below is a list of all Stored Procedures, stored in the MassDataWarehouse database. These are required for all reports to function properly:

1) dbo.spMarketSalesReportMAS
2) dbo.spMarketSalesReportRDS
3) dbo.spMarketSalesReportCLUB
4) dbo.spMarketSalesReportMEMADV
5) dbo.spMarketSalesReportRANK
6) dbo.spMarketSalesReportCORP
7) dbo.spMarketSalesReportPPR
8) dbo.spMarketSalesReportPPRPTM

All of these procedures have been deployed and updated to the existing MassDataWarehouse database. The only change made to this database was a table, dbo.ReportingRegions, was added for report sorting purposes.

Creating a script file for these procedures (for future updating or deployment on new server):

1) Go to the MS SQL Server 2008 Management Studio, on the SSRST-01 server.
2) Open the Databases node.
3) Select the MassDataWarehouse node and open it.
4) Open the Programmability node - select and highlight the Stored Procedures node.
5) From the view menu select Object Explorer Details.
6) Select and highlight the 8 Stored Procedures (as shown below in Fig.1).
7) Right click and select as shown in Fig.1.
8) The user can either write to a file or to a new query window.
9) Using the code from the query window or the file.
   a. Open a new query window for the Server and required database.
   b. Execute the script by either hitting Alt-x or the ! Execute button above the query window **NOTE: The functions should be executed prior to running the Stored Procedures.**



| Name | Schema | Create Date | Policy Health Sta |
|---|---|---|---|
| spMarketSalesReportCLUB | dbo | 4/12/2011 12:01 PM | |
| spMarketSalesReportCORP | dbo | 4/12/2011 12:01 PM | |
| spMarketSalesReportMAS | dbo | 4/12/2011 12:01 PM | |
| spMarketSalesReportMEMADV | dbo | 4/12/2011 12:01 PM | |
| spMarketSalesReportPPR | dbo | 4/12/2011 12:01 PM | |
| spMarketSalesReportPPRPTM | dbo | 4/12/2011 12:08 PM | |
| spMarketSalesReportRANK | dbo | 4/12/2011 12:01 PM | |
| spMarketSalesReportRDS | | 4/12/2011 12:01 PM | |
| System Stored Procedures | | | |
| udpBatchInsert | | AM | |
| udpBatchUpdateEndDateTime | dbo | AM | |
| udpGetEmailMetrics | dbo | | |
| udpSystemLogInsert | dbo | | |
| upCreateDimDateData | dbo | | |
| upCreateDimTimeData | dbo | | |
| upErrorInsert | dbo | | |
| upFactClubBudgetLoad | dbo | | |
| upFactClubGoalLoad | dbo | M | |
| upFactCommissionCalculationLoad | dbo | | |
| upFactMAGoalLoad | dbo | | |
| upFactMembershipSaleLoad | dbo | 3/29/2011 6:15 PM | |

**Fig.1**

All Stored Procedures listed above have comments throughout their body of code and a header describing its function. The prefixes of all stored procs, by this developer, are: sp<StoredProcName>. Each Stored Procedure has a line, in the comment header, for ease of execution.

## User Defined Functions

The following is a list of the functions written specifically for the Stored Procedures that are executed for the reports (they are used in various locations and will not be explained individually):

1) dbo.ufnDateOnly
2) dbo.ufnFirstDayOfCurrentMonth
3) dbo.ufnFirstDayOfNextMonth
4) dbo.ufnLastDayOfCurrentMonth
5) dbo.ufnLastDayOfPreviousMonth
6) dbo.ufnReturnClubNameFromClubNumber
7) dbo.ufnReturnDateShippedDateID
8) dbo.ufnReturnDistinctNumberOfPaymentMonths
9) dbo.ufnReturnDistinctNumberOfPaymentMonthsPTM
10) dbo.ufnReturnDivideByPTSalesCashAPOSDollar
11) dbo.ufnReturnDotFit
12) dbo.ufnReturnEmployeeNameFromID
13) dbo.ufnReturnEmployeeTenure
14) dbo.ufnReturnEnrollmentFees
15) dbo.ufnReturnEnrollmentNMD
16) dbo.ufnReturnExerspy
17) dbo.ufnReturnGoalReportDateID
18) dbo.ufnReturnMASTotals
19) dbo.ufnReturnNMU
20) dbo.ufnReturnPTCummIndex
21) dbo.ufnReturnPTProjection
22) dbo.ufnReturnPTProjectionPTM
23) dbo.ufnReturnRegionCash
24) dbo.ufnReturnRegionNewSales
25) dbo.ufnReturnRegionPOS
26) dbo.ufnReturnRegionRenewalSales
27) dbo.ufnReturnReportDate
28) dbo.ufnReturnReportDateFromDateID
29) dbo.ufnReturnReportDateID
30) dbo.ufnReturnRVPT
31) dbo.ufnReturnSalesCummIndex
32) dbo.ufnReturnSeniorityDate
33) dbo.ufnReturnSessionsAmount
34) dbo.ufnReturnSessionsAmountPTM
35) dbo.ufnReturnSessionsUsed
36) dbo.ufnReturnSessionsUsedPTM
37) dbo.ufnReturnShortTermProrate
38) dbo.ufnReturnSuppsMTD
39) dbo.ufnReturnSuppsProjection
40) dbo.ufnReturnTotalRevenue
41) dbo.ufnReturnTrainerLevel
42) dbo.ufnReturnTwelveMonthAverage
43) dbo.ufnReturnTwelveMonthAveragePTM
44) dbo.ufnTrim

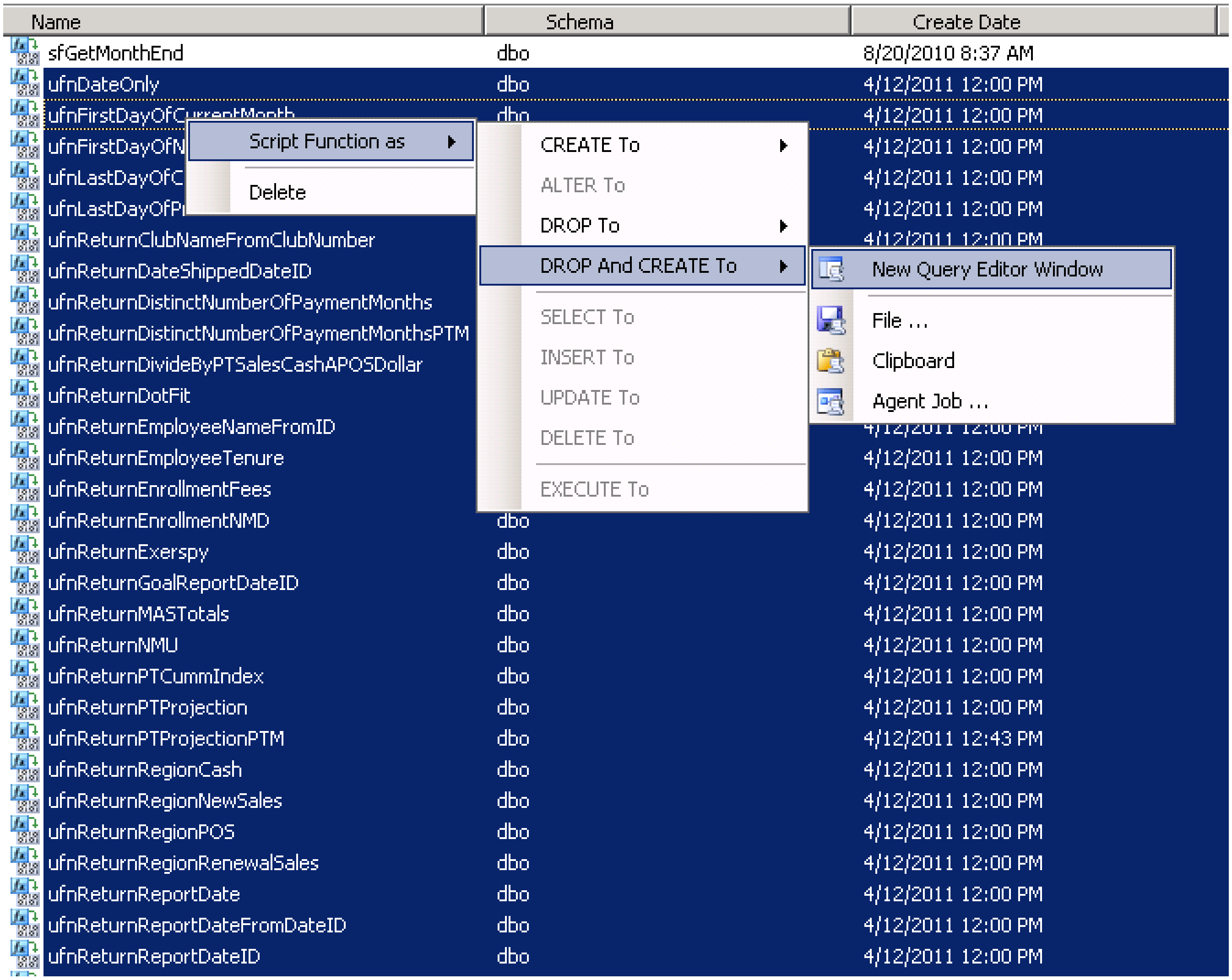
All Functions return a value and if the value is a number it is rounded as per request of the client. If the value equates to a null, it is returned as a “0” if it is numeric, except where it would be used as a devisor or denominator (to avoid divide by zero error).

All of these functions have been deployed and updated to the existing MassDataWarehouse database on the SSRST-01 SQL Server.

All User Defined Functions listed above have comments throughout their body of code and a header describing its function. The prefixes of all functions, by this developer, are: ufn<FunctionName>. Each Function has a line, in the comment header, for ease of execution.

Creating a script file for these functions (for future updating or deployment on new server):

1) Go to the MS SQL Server 2008 Management Studio, on the SSRST-01 server.
2) Open the Databases node.
3) Select the MassDataWarehouse node and open it.
4) Open the Programmability node - select and highlight the Functions node and click on Scalar-valued Functions.
5) From the view menu select Object Explorer Details.
6) Select and highlight the 44 Functions (as shown below in Fig.2).
7) Right click and select as shown in Fig.2.
8) The user can either write to a file or to a new query window.
9) Using the code from the query window or the file
    a. Open a new query window for the Server and required database.
    b. Execute the script by either hitting Alt-x or the ! Execute button above the query window. **NOTE: The functions should be executed prior to running the Stored Procedures.**



| Name | Schema | Create Date |
|---|---|---|
| sfGetMonthEnd | dbo | 8/20/2010 8:37 AM |
| ufnDateOnly | dbo | 4/12/2011 12:00 PM |
| ufnFirstDayOfC | dbo | 4/12/2011 12:00 PM |
| ufnFirstDayOfN | | 4/12/2011 12:00 PM |
| ufnLastDayOfC | | 4/12/2011 12:00 PM |
| ufnLastDayOfPr | | 4/12/2011 12:00 PM |
| ufnReturnClubNameFromClubNumber | | 4/12/2011 12:00 PM |
| ufnReturnDateShippedDateID | | |
| ufnReturnDistinctNumberOfPaymentMonths | | |
| ufnReturnDistinctNumberOfPaymentMonthsPTM | | |
| ufnReturnDivideByPTSalesCashAPOSDollar | | |
| ufnReturnDotFit | | |
| ufnReturnEmployeeNameFromID | | 4/12/2011 12:00 PM |
| ufnReturnEmployeeTenure | | 4/12/2011 12:00 PM |
| ufnReturnEnrollmentFees | | 4/12/2011 12:00 PM |
| ufnReturnEnrollmentNMD | dbo | 4/12/2011 12:00 PM |
| ufnReturnExerspy | dbo | 4/12/2011 12:00 PM |
| ufnReturnGoalReportDateID | dbo | 4/12/2011 12:00 PM |
| ufnReturnMASTotals | dbo | 4/12/2011 12:00 PM |
| ufnReturnNMU | dbo | 4/12/2011 12:00 PM |
| ufnReturnPTCummIndex | dbo | 4/12/2011 12:00 PM |
| ufnReturnPTProjection | dbo | 4/12/2011 12:00 PM |
| ufnReturnPTProjectionPTM | dbo | 4/12/2011 12:43 PM |
| ufnReturnRegionCash | dbo | 4/12/2011 12:00 PM |
| ufnReturnRegionNewSales | dbo | 4/12/2011 12:00 PM |
| ufnReturnRegionPOS | dbo | 4/12/2011 12:00 PM |
| ufnReturnRegionRenewalSales | dbo | 4/12/2011 12:00 PM |
| ufnReturnReportDate | dbo | 4/12/2011 12:00 PM |
| ufnReturnReportDateFromDateID | dbo | 4/12/2011 12:00 PM |
| ufnReturnReportDateID | dbo | 4/12/2011 12:00 PM |

**Fig.2**

## Deployment of all report files

There are 3 separate folders located on the C: drive of the SSRST-01 server that store the report projects. Double clicking on the .sln file of the report will take the user to the using MS SQL Server 2008 Business Intelligence Development Studio with that project ready to be viewed or edited.

1) MASReport
2) PPRTrainerReport
3) PPRPMTTrainerReport

From the Solution explorer right click on the Solution name (MASReport in this example) and select Deploy as shown in Fig.3 below.

All reports are configured for a the server as shown below in Fig.4

All reports are configured for a the correct dataset as shown below in Fig.5

Once the deployment is completed and successful, the report will be accessible on the Report Server. See Fig.6 for details.

**NOTE:** The server requires IIS and SMTP to be configured to allow for subscription by Email. Here are the details of how to setup a Data Driven Subscription:

1) To allow for manual and programmatic initialization of these reports on the Report Server a Data Driven Subscription should be setup.
2) Right click on the report required for subscription and go to Manage (see Fig.7 below).
3) Select the Subscriptions tab and select New Data Driven Subscription
4) When prompted for a Query add this line to allow for a Report Date for the previous day (see Fig.8):
    a. SELECT DATEADD(d,-1,dbo.ufnDateOnly(GETDATE())) AS ReportDate
5) This document will end here for the user to experiment with all the parameters for the subscription.

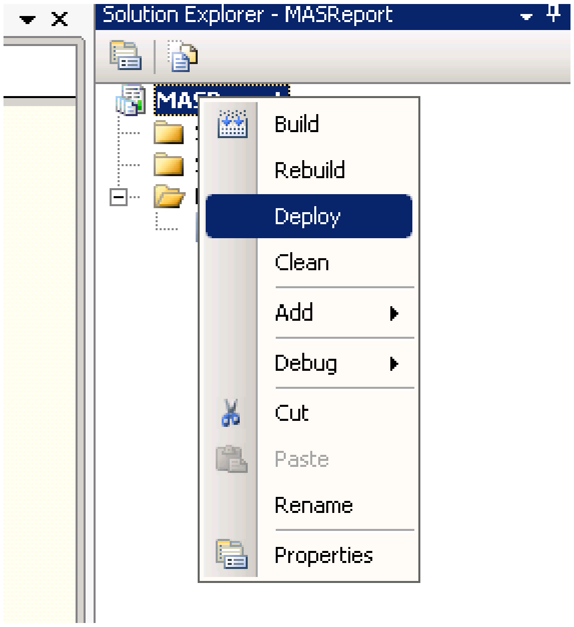
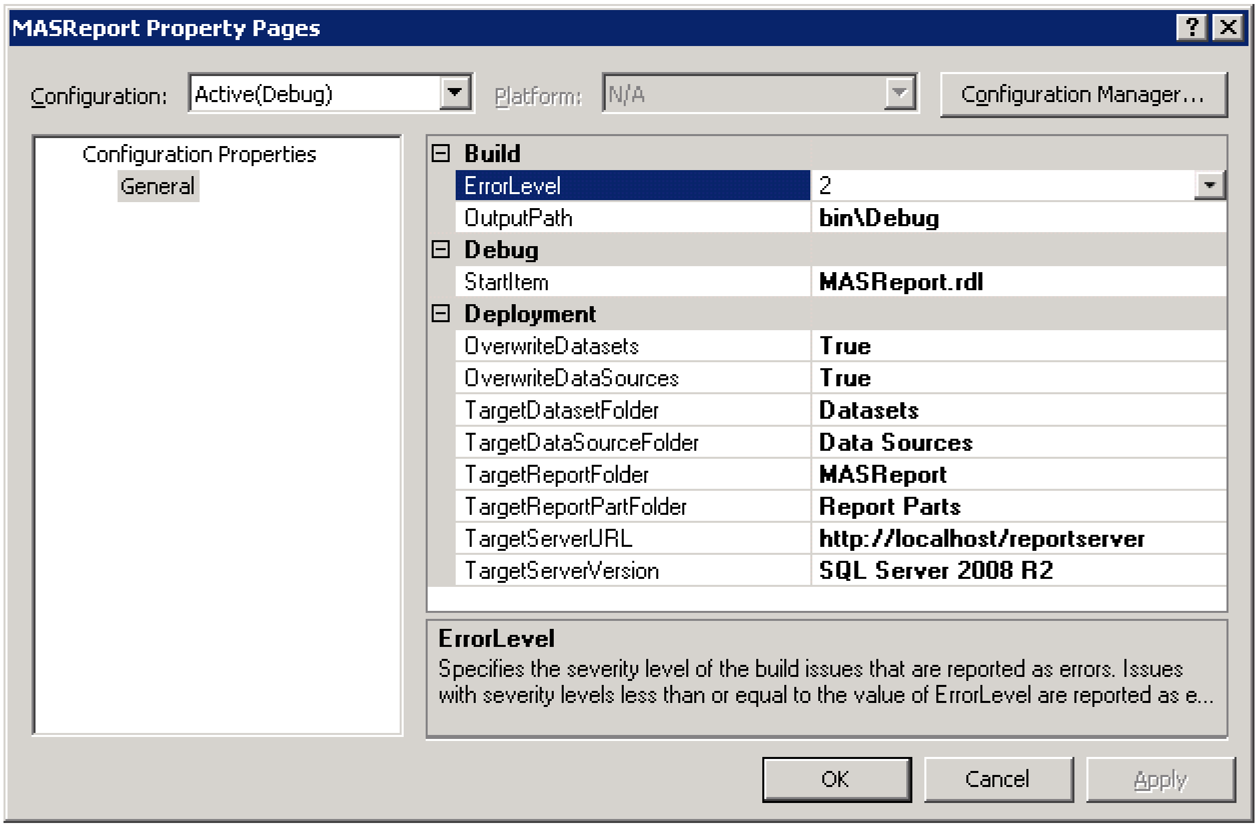


**Fig.3**



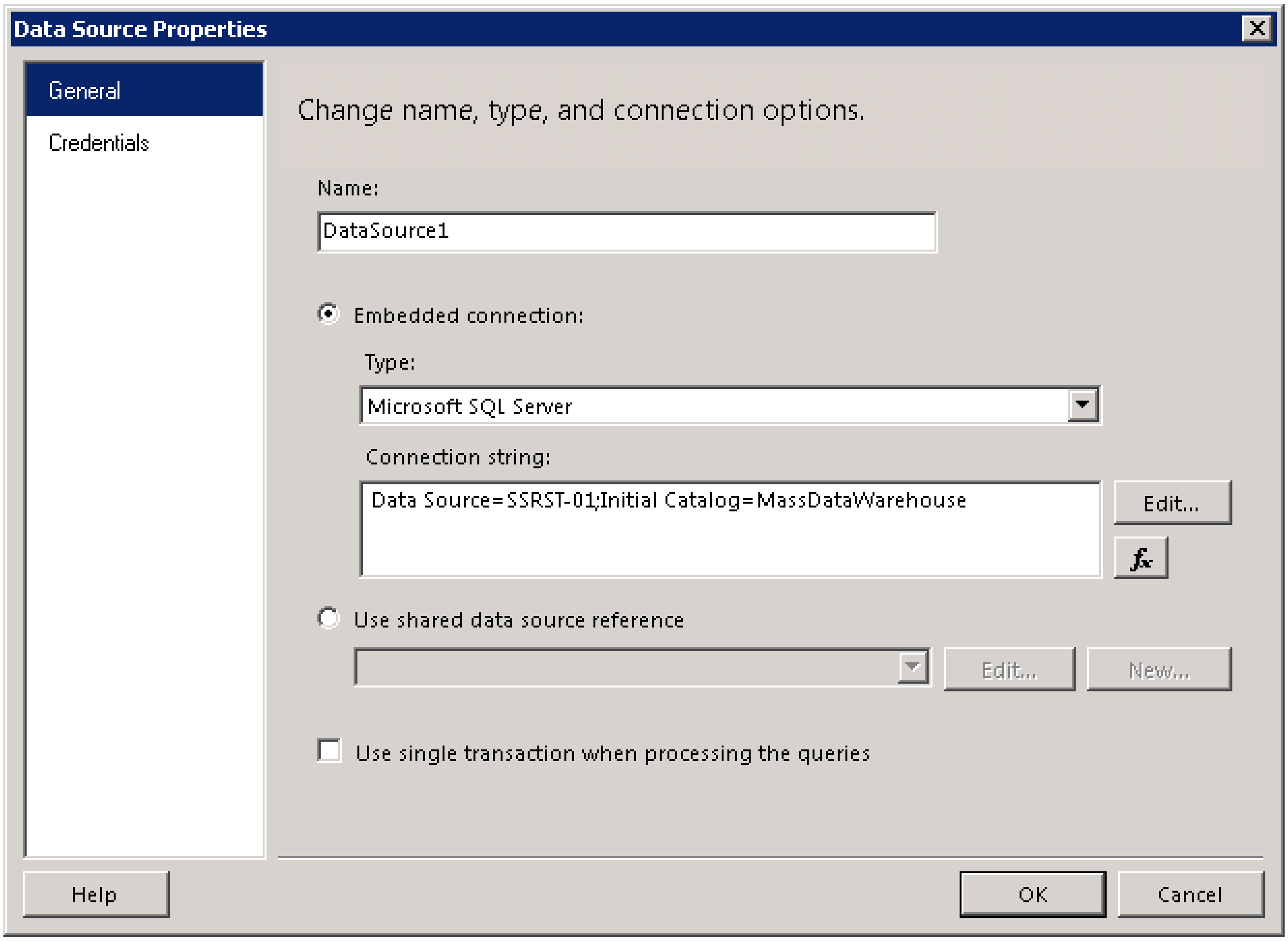
Configuration: Active(Debug)

Platform: N/A

| | |
|---|---|
| **Build** | |
| ErrorLevel | 2 |
| OutputPath | **bin\Debug** |
| **Debug** | |
| StartItem | **MASReport.rdl** |
| **Deployment** | |
| OverwriteDatasets | **True** |
| OverwriteDataSources | **True** |
| TargetDatasetFolder | **Datasets** |
| TargetDataSourceFolder | **Data Sources** |
| TargetReportFolder | **MASReport** |
| TargetReportPartFolder | **Report Parts** |
| TargetServerURL | **http://localhost/reportserver** |
| TargetServerVersion | **SQL Server 2008 R2** |

**ErrorLevel**
Specifies the severity level of the build issues that are reported as errors. Issues with severity levels less than or equal to the value of ErrorLevel are reported as e...

**Fig.4**



Change name, type, and connection options.

Name:
DataSource1

● Embedded connection:

Type:
Microsoft SQL Server

Connection string:
Data Source=SSRST-01;Initial Catalog=MassDataWarehouse

○ Use shared data source reference

□ Use single transaction when processing the queries

**Fig.5**

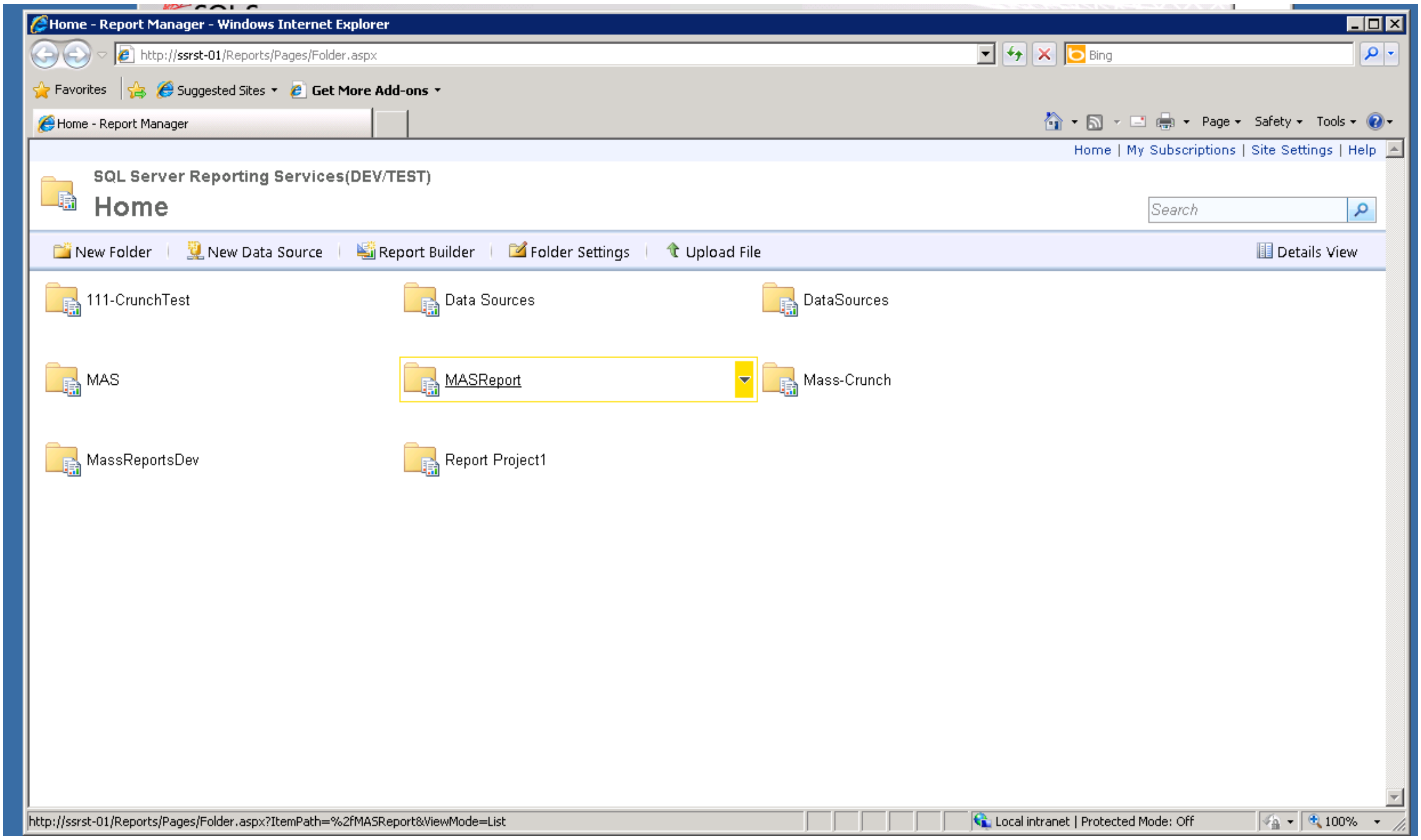

Home - Report Manager - Windows Internet Explorer
http://ssrst-01/Reports/Pages/Folder.aspx
Bing
Favorites
Suggested Sites
Get More Add-ons
Home - Report Manager
Page
Safety
Tools
Home | My Subscriptions | Site Settings | Help
SQL Server Reporting Services(DEV/TEST)
Home
Search
New Folder
New Data Source
Report Builder
Folder Settings
Upload File
Details View
111-CrunchTest
Data Sources
DataSources
MAS
MASReport
Mass-Crunch
MassReportsDev
Report Project1
http://ssrst-01/Reports/Pages/Folder.aspx?ItemPath=%2fMASReport&ViewMode=List
Local intranet | Protected Mode: Off
100%

Fig.6

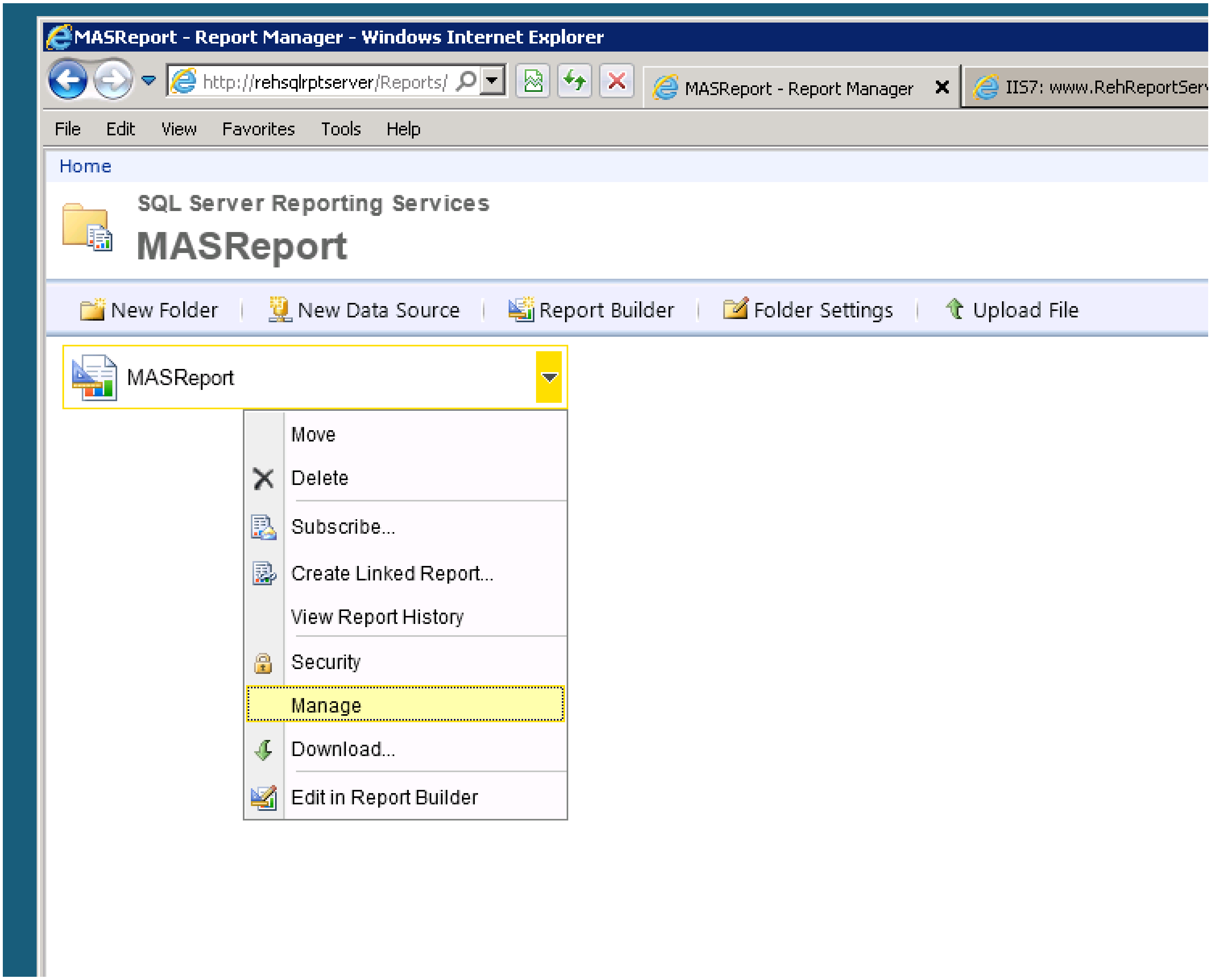
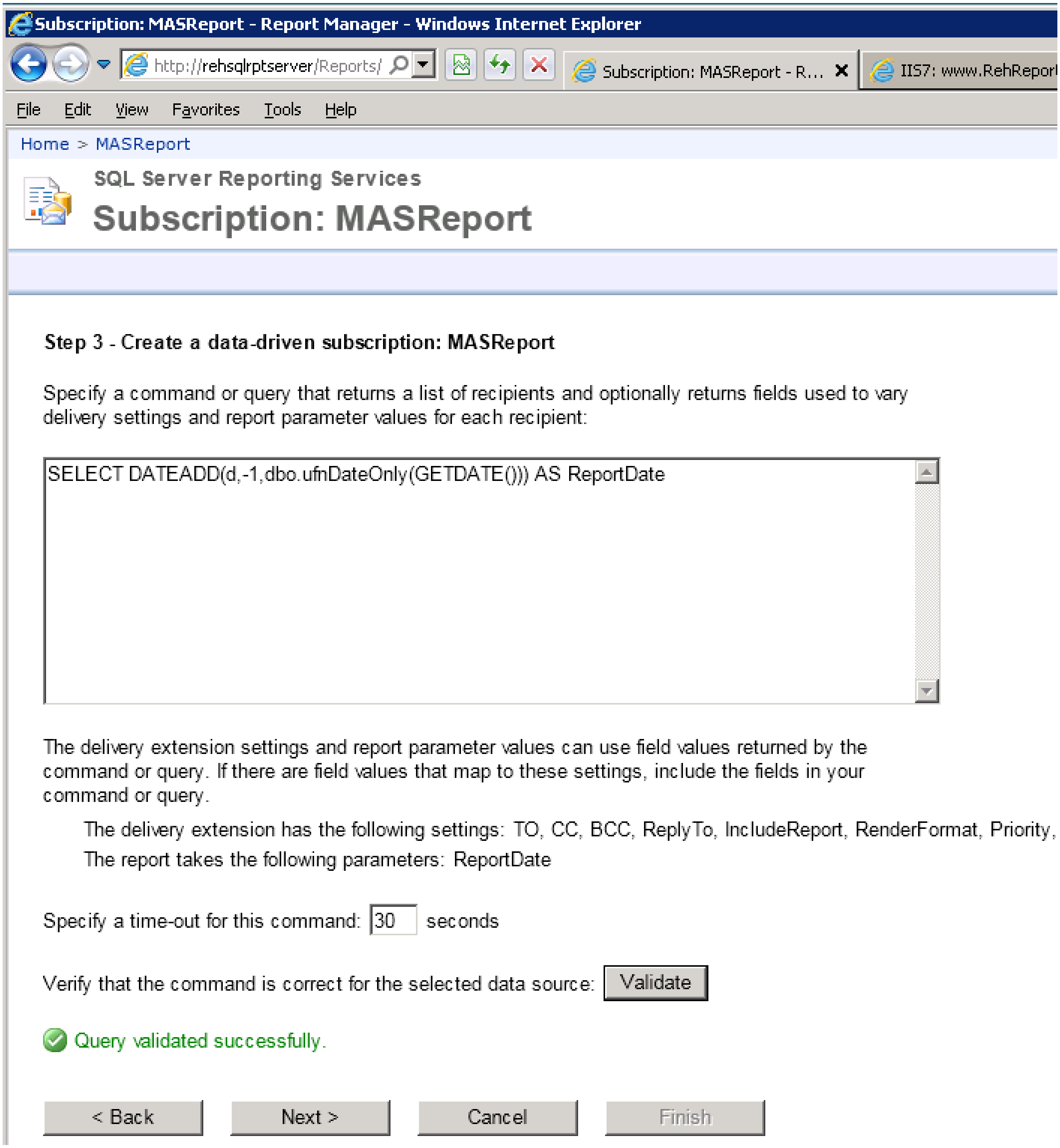

MASReport - Report Manager - Windows Internet Explorer
http://rehsqlrptserver/Reports/
MASReport - Report Manager
IIS7: www.RehReportServ
File
Edit
View
Favorites
Tools
Help
Home
SQL Server Reporting Services
MASReport
New Folder
New Data Source
Report Builder
Folder Settings
Upload File
MASReport
Move
Delete
Subscribe...
Create Linked Report...
View Report History
Security
Manage
Download...
Edit in Report Builder

**Fig.7**


Subscription: MASReport - Report Manager - Windows Internet Explorer
http://rehsqlrptserver/Reports/
Subscription: MASReport - R...
IIS7: www.RehReport
File
Edit
View
Favorites
Tools
Help
Home > MASReport
< Back
Next >
Cancel
Finish

SQL Server Reporting Services

## Subscription: MASReport

**Step 3 - Create a data-driven subscription: MASReport**

Specify a command or query that returns a list of recipients and optionally returns fields used to vary delivery settings and report parameter values for each recipient:

```
SELECT DATEADD(d,-1,dbo.ufnDateOnly(GETDATE())) AS ReportDate
```

The delivery extension settings and report parameter values can use field values returned by the command or query. If there are field values that map to these settings, include the fields in your command or query.

The delivery extension has the following settings: TO, CC, BCC, ReplyTo, IncludeReport, RenderFormat, Priority,

The report takes the following parameters: ReportDate

Specify a time-out for this command: 30 seconds

Verify that the command is correct for the selected data source: Validate

✓ Query validated successfully.

**Fig.8**