



Executive BI Reporting Using Microsoft Reporting Services

White Paper

Published: September 19, 2006

Produced by:

twentysix New York
a business solutions provider

62 West 45th Street
New York, NY 10036

www.26ny.com

Introduction

Traditionally, to provide effective user access to Business Intelligence information stored in Microsoft Analysis Services (OLAP Cubes), one had to either purchase a cube browser from third-party software vendors, such as ProClarity and Panorama, or develop custom software using ADO MD.

Neither solution is cost effective when in-depth analysis is not a requirement as the first involves heavy licensing fees and the second involves significant custom development.

Microsoft Reporting Services provides an easy and cost effective way for business users to access the BI information stored in Microsoft Analysis Services Cubes with minimal development efforts.

Solution summaries

There are a few different approaches that use Microsoft Reporting Services to build BI-enabled reports:

1. Single drillable report.
2. Non-drillable summary and drillable detail reports.
3. Reports that are pre-processed in HTML format.
4. Reports that are pre-processed in XML format.

Each of these approaches create web-based reports that use the MDX statements to retrieve data from OLAP cubes and the Matrix control with drill-down capability to display the information.

Solution 1 – Single Drillable Report

This is the simplest approach: a one-size-fits-all report is created with the needed hierarchies and measures.

Pros:

- Minimal development effort
- Minimal maintenance effort

Cons:

- Performance: real-time rendering of an OLAP based report using Microsoft Reporting Services takes time (seconds)
- Drill-down limitation: when placing the report on a page with other components, limited screen space is available for drill-down
- Security: accessing Microsoft Reporting Services Reports directly might require the enabling of Anonymous Access of the Report Server website for non-domain users
- Limited internet browser support: Microsoft Reporting Services Reports with Matrix control might not display properly with the Firefox browser

Solution 2 – Non-Drillable Summary and Drillable Detail Reports

To resolve some of the issues associated with the first solution, the single report is broken down into two reports: one non-drillable summary report and one drillable detail report, which opens in a separate window after clicking on the summary report.

Pros:

- Performance improvement: especially for executives that only require summary level information
- Drill-down improvement: with the detail report opening in its own browser window, it is no longer shared with other components on a page

Cons:

- Performance: despite the improvement, real-time rendering of the summary report still takes time, which might not be acceptable for a home page
- Security: same issue as solution 1
- Limited internet browser support: same issue as Solution 1

Solution 3 – Pre-process Reports into HTML format

To resolve the remaining issues of Solution 2, an alternate approach can be used.

Microsoft Reporting Services exposes a web service (reportservice.asmx) that provides methods that allow software developers to export the reports in pre-defined formats, such as XML, HTML, Excel, and so on.

With this solution, a small .Net WinForm application is created to call the Reporting Services Web Service to generate HTML files from the Summary and Detail reports. The business users are then directed to the generated HTML files to view the reports. The application is run as the final step of the ETL process to guarantee that the content of the HTML files get refreshed each time the OLAP cubes are processed.

Pros:

- Performance improvement gains are shown across the board
- Security improvement: since the generated HTML files could be placed on a separate web server, access to the Reporting Services website can be limited if required

Cons:

- CSS conflicts: the generated HTML files have build-in CSS style sheets that might cause conflicts with the page CSS style sheets if the report is to share space with other page components
- Limited Internet browser support: same issue as Solutions 1 and 2

Solution 4 – Pre-process Reports into XML format

To resolve the remaining issues presented with Solution 3, the .Net application is modified to generate XML files from the Reporting Services reports. The resulting XML files are displayed to the business users using XML style sheets.

In addition to resolving all of the issues presented in the other three solutions, this solution also provides a couple of unique features and some considerable advantages:

- Allows changes to the report display without having to recompile the reports, which is done simply by modifying the XML style sheets
- XML files, as industrial standard, can be used to deliver report data to other internal or external systems

Both Solutions 3 and 4 can also be used to generate snapshot reports for automatic email delivery to business users without accessing the reporting environment.

Conclusion

Microsoft Reporting Services provides a nice platform to deliver BI information to less demanding business users and provides a nice complement to the more powerful and more expensive BI tools, such as ProClarity and Panorama.

Useful link: <http://msdn.microsoft.com/SQL/2000/learn/bi/reporting/default.aspx>

twentysix
NEW YORK