I C E M I T
1st International Scientific Conference on Economy, Management and Information Technologies – ICEMIT 2023

Automatic conversion of server reports into client reports using SQL Server Reporting Services (SSRS) technology

Krešo Barić[a], Đurđica Vukić[a*], Denis Jelaš[a]

[a] Aspira University of Applied Sciences, Croatia

Article info

Review paper

DOI:
https://doi.org/10.46793/ICEMIT23.301B

UDC/ UDK:
3.077.7:004.4`242

Abstract

Automatic conversion of server reports into client reports enables more efficient and simpler distribution of reports to end users. This process facilitates data visualization and access to updated information. Client reports can be used directly in web browsers and other applications without the need for special clients or installing additional software or tools on local computers, as well as to increase the security of the computers on which the program is installed. This paper presents the procedure of automatic conversion and implementation, an overview of the applied SSRS (SKL Server Reporting Services) technology and the techniques used.
***Keywords*:** *automatic conversion, server reports, client reports, SQL Server Reporting Services (SSRS)*

1. Introduction

Reports are key documents in the business world, business applications and serve to convey information, analyze problems, and provide recommended course of action. Their purpose is to concisely and effectively communicate relevant information about a particular problem or situation.

There are many different formats and types of reports used in the business sector. A report can be any prepared data generated from a database or any other type of data source. There are several types that are evident, such as sales reports, summaries at the end of each quarter, trend analyzes and comparisons. Reports are used in all aspects of business, and almost every business process requires printing, presenting, viewing, or using some other technique to communicate data to business executives, workers, service providers, consumers, analysts, and others for a number of purposes.

SSRS is a SQL Server technology that enables organizations to create, manage, and distribute reports. It integrates with the database and offers various methods of data visualization and presentation, including tables, graphs, matrices and diagrams. SSRS supports interactive reports and various formats such as PDF, Excel, Word, and HTML, making it easy to share and distribute within an organization ("Rendering Extensions Overview", 2023).

Automatic conversion of server reports to client reports in SSRS technology is essential in a business environment because it enables fast, accurate and efficient transformation of data from server reports into a format suitable for client analysis. This process improves efficiency by reducing employees' repetitive tasks and allowing them to focus on other business activities. It also ensures greater data accuracy as manual data transfer can lead to errors and inaccuracies. SSRS technology reduces the risk of human error, ensuring data consistency and integrity. Additionally, automatic conversion enables better visualization of data, with different graphical elements, tables and graphs, facilitating informed business decisions. Furthermore, the automatic conversion of server reports into client reports contributes to the overall productivity and competitiveness of the organization, enabling faster access to updated information, better planning, performance monitoring and recognition of business opportunities.

The aim of this paper is to provide an overview of the use of SSRS for the creation of client reports and to present the process of automatically converting server reports into client reports. The following will first present an overview of

*Corresponding author
E-mail address: durdica.vukic@aspira.hr

SSRS technology and the importance of automatically converting server reports into client reports in SSRS technology. The process of automatic conversion and implementation, the techniques and tools used, and final considerations and conclusions will be presented.

2. The importance of automatically converting server reports into client reports in SSRS technology

Automatic conversion of server reports into client reports in SSRS technology is extremely important in a business environment. This technique enables fast, accurate and efficient transformation of data from server reports into reports suitable for display and analysis on the client side.

One of the key reasons for automatic conversion lies in improving the efficiency of business processes. Manually generating client reports from server data can be a time-consuming and complex task, requiring time and additional resources. Automating this process frees employees from repetitive tasks and allows them to focus on other business activities.

In addition, automatic conversion ensures greater data accuracy. Manually transferring and reshaping information from server reports can lead to errors and inaccuracies. By using SSRS technology for automatic conversion, the risk of human error is reduced and data consistency and integrity are ensured.

Also, automatic conversion enables better data visualization. SSRS technology provides various options for formatting and displaying reports on the client side. Using various graphic elements, tables and graphs, data can be presented in a clear and intuitive way, facilitating the making of informed business decisions.

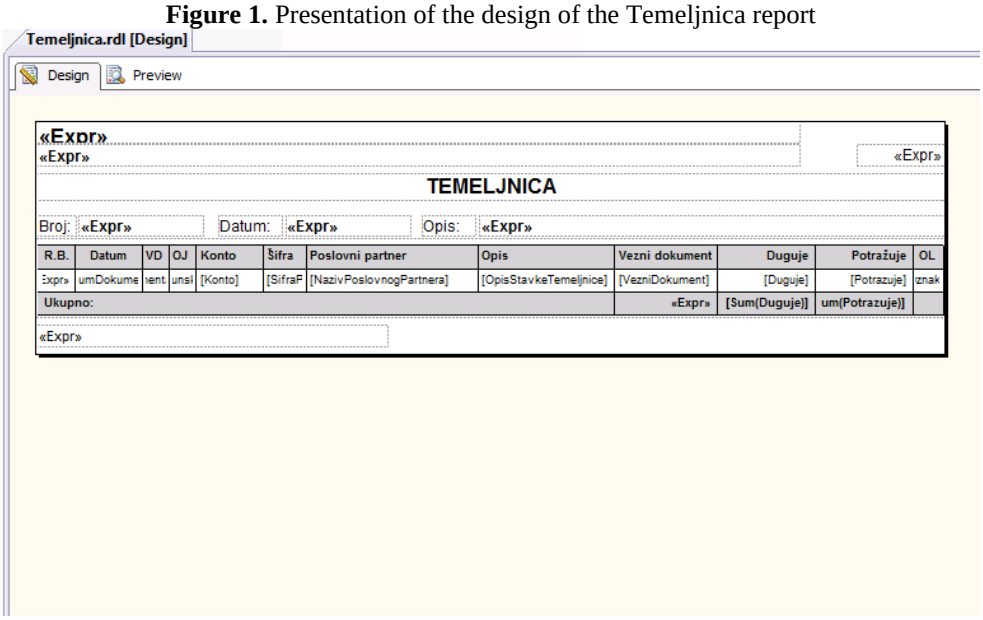
In addition to all of the above, the automatic conversion of server reports into client reports in SSRS technology contributes to improving the overall productivity and competitiveness of the organization. Faster access to updated and relevant information enables better planning, performance monitoring and identification of business opportunities.

3. Overview of SSRS technology

In 2003, Microsoft introduced SQL Server Reporting Services (SSRS) as an add-on to SQL Server 2000, which caused excitement among software developers. Originally planned for release with SQL Server 2005, it was deployed in January 2004 to migrate existing reports. SSRS introduced essential features and extensibility through its web service, allowing additional features to be added programmatically. SSRS 2008 has become a key component in SQL Server, becoming the reporting development environment of choice for developers and designers, especially those with Visual Studio and VB.NET. New features enable user feedback and show how to design professional reports, applications and solutions built on Microsoft's business intelligence initiatives (Lachey, 2008).

Designing and creating reports using information from SQL Server databases or other data sources is the main purpose of SSRS. Using a drag-and-drop interface, developers can easily design dynamic reports that are visually compelling (Figure 1).

Figure 1. Presentation of the design of the Temeljnica report



Source: Authors´ work

Some of the key features and capabilities of SSRS include (Landrum etc al., 2008):

- a standard reporting platform based on the Report Definition Language (RDL), which is an XML schema that dictates the common structure of all SSRS reports. This allows reports to be generated from any third-party application that supports the RDL schema.
- SSRS offers additional subscription service features, report caching, report history, and report execution scheduling.
- SSRS, as a web-based solution, can be deployed on different platforms
- SSRS offers tools such as Report Builder and SQL Server Data Tools for creating reports, offering features such as layout design, data source connection, parameter configuration, and expression-based formatting.
- SSRS supports multiple data sources, allowing users to connect and retrieve data to generate reports, increasing flexibility in data integration.
- SSRS simplifies the implementation and management of reports through its Report Manager web portal, allowing users to publish, organize and schedule reports, while administrators can define security settings and access permissions.
- SSRS supports a number of report formats, including PDF, Excel, Word, and HTML, allowing users to schedule automated report delivery via email or file sharing and customize report delivery.
- For deeper data insights, SSRS offers interactive capabilities including sorting and filtering based on parameters.

SQL Server Reporting Services 2008 has undergone significant changes compared to the version of SQL Server 2000. The latest version, SSRS 2008, is an independent reporting server web service, installed as a Windows service similar to Database Engine and Analysis Services. Key changes include scalability, memory management and a new Tablix design feature. The Tablix feature combines table and matrix elements, including a list control, allowing for more flexibility at design time. In addition, Microsoft SQL Server 2008 Report Builder 2.0 was released as part of the SQL Server 2008 Feature Pack, offering a complete design surface for designing all elements of the report definition language, as well as usability enhancements such as Zoom and Snap. Report Builder 2.0 supports the full capabilities of SQL Server Reporting Services, including flexible data layout, data visualization, and richly formatted text capabilities (Krishnaswamy, 2009).

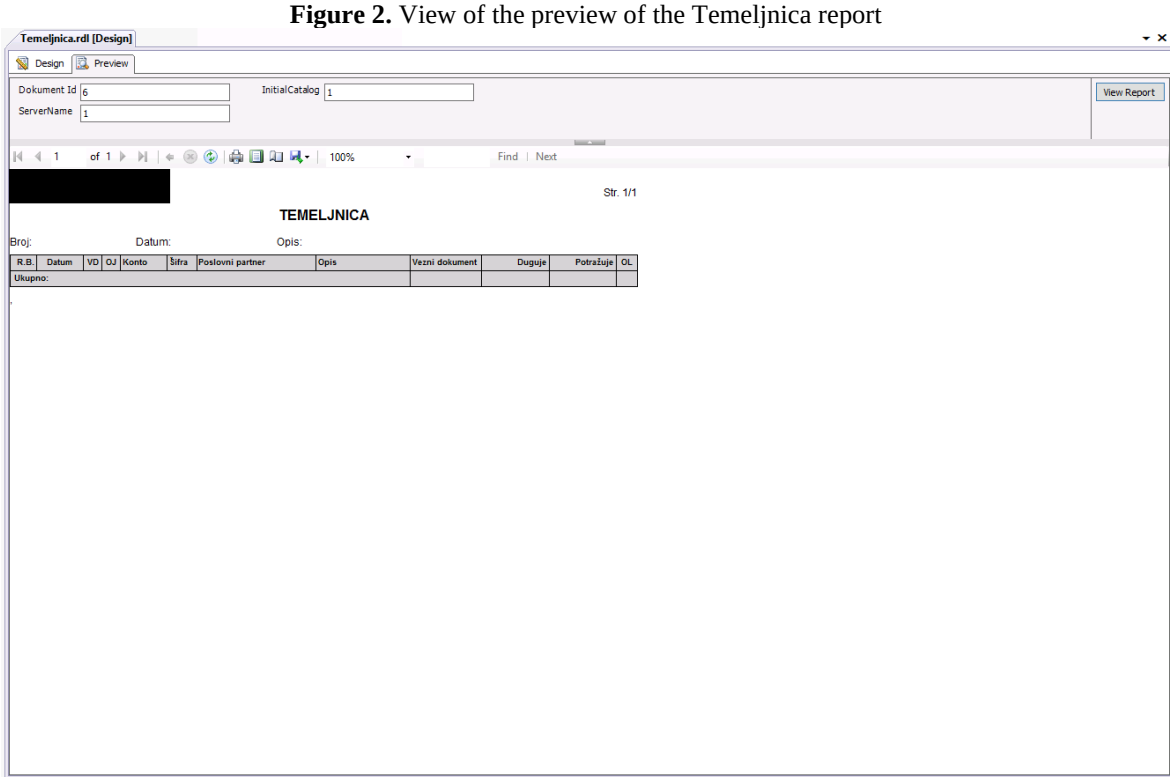
The advantages of migration to SSRS technology are manifested through several factors (Landrum etc al., 2008):

- The standard platform for SSRS reports is VS.NET, which is used by development teams for standard reports. This platform eliminates the need for additional development software, allowing customers to develop custom reports without purchasing additional software. Microsoft's Business Intelligence Development Studio (BIDS) is also available in SQL Server 2008, based on the next release of Visual Studio. SSRS is an integral part of SQL Server 2008 and is available in different editions, from Express to Enterprise.
- SSRS is web-enabled, allowing a single report to be available to a variety of clients, including a browser and custom Windows Forms. Reports are primarily accessed via HTTP or HTTP Secure, allowing users to view reports from any location with access to the SSRS web server. This eliminates the need for heavy client applications to locally install reports.
- SSRS provides a .NET Web Service as a front end, allowing programmatic access to extend report delivery outside of the browser. .NET developers can build custom report rendering applications, allowing them to control the look and feel of the report viewer.
- SSRS subscriptions offer the ability to deliver reports via email or file sharing, out-of-hours processing, providing significant benefits to both businesses and clients.

The need for automatic conversion of server reports into client reports

In this paper, the emphasis is on the difference between client and server reports in SSRS technology through the analysis of their advantages and disadvantages. To create both types of reports, tools are available for layout design, the way to pass parameters, and the creation of data sets that will be displayed.

Figure 2. View of the preview of the Temeljnica report



Source: Authors´ work

Server reports have one significant advantage, the preview (Figure 2.) option that enables the immediate display of data and the appearance of the report, which greatly facilitates the report development process. However, the main disadvantage of server reports is the need to install SSRS tools on the client side, i.e., with each user, in order to display the reports. In addition, the process of transferring new reports to clients is almost manual and difficult to automate, and access to SSRS today is increasingly difficult for security reasons, as access to this tool is difficult, and sometimes completely impossible, because the business application is programmed to use it exclusively renders server reports unusable.

On the other hand, client reports are delivered to the user as secure text files along with the application installation and can be called up and displayed whenever needed. However, one of the main disadvantages of client reports is the impossibility of viewing the report without the application in which it is launched, which means that it is necessary to launch the application every time and get to the place in the application where the report can be launched, and data preparation is difficult. This can lead to a significant loss of time, especially when displaying complicated and visually demanding reports. Also, preparing data for client reports can be problematic because the reports cannot be seen before they are displayed in the application, which means that the report is created without a preview.

Following on from the above, the automatic conversion of server reports into client reports can be considered a more efficient and practical solution for displaying reports in applications. Furthermore, the report can be reviewed at the same time as the data for the report is being prepared, which saves time and facilitates the process of creating the report.

4. Automatic conversion methodology

Used techniques and tools for creating server reports

In order to take advantage of the best of both technologies, we can combine server-side and client-side reporting. In particular, we will use the server mode to develop reports, while we will switch to client reports for application use.
There is a simple way to achieve this. After the server report is finished, it is saved in XML format that describes everything displayed on the report, including controls, details about their positions, and all parameters and queries that retrieve data from the database. Such a server report file has the extension. rdl.

Since the file is saved in XML format, the XSLT record format enables easy analysis of the structure (parsing) of the file (Turley and Bruckner, 2010).

Furthermore, the file can be programmatically decomposed into meaningful units and programmatically read the content contained in it. After that, units related to report parameters and queries that retrieve data are programmatically extracted. This data is then used to automatically and dynamically switch the way client reports work.

Client reports should prepare the data and serve it for display. Given that all queries are read from XML, they are dynamically created, executed, and retrieve data that is then sent to the client report for display. In this way, the best of both methods of reporting is combined and a more efficient and flexible way of working is achieved.

Automatic conversion of server reports into client reports in SQL Server Reporting technology Services (SSRS) involves the use of various techniques and tools. These techniques and tools simplify the conversion process and increase the efficiency and effectiveness of the entire workflow. In the practical part of the work, the following were used ("SSRS Report Builder" and "SQL Server Management Studio", 2023):

- SSRS Report Builder (a tool for creating server reports and viewing them)
- SQL Server Management Studio (SSMS) (a tool for preparing complex queries for data retrieval).

Description of the automatic conversion procedure

The automatic conversion procedure plays a key role in the transformation of server reports into client reports in SQL Server Reporting technology Services (SSRS). This procedure consists of a number of processes that ensure that data is seamlessly and efficiently converted from the server side into a format suitable for presentation and analysis on the client side:

1. Changing the extension of the document: in this step, the extension of the server reports is changed. rdl and in. rdlc - client reports extension.
2. Parsing : in this step, the building elements of the report are recognized and mapped with the intended classes Auxiliary classes: DataSet, Field, Query, QueryParameter, Report, ReportParameter, SerializableBase (Golisch , 2021) needs to be created and have in applied application to dynamically load and parse xml files of the server report and took the parts that are necessary to automatically run the same report as the client report.
3. Data extraction and parameterization: Data extraction from the database according to mapped data from classes and defined parameters

The key mechanism of this step brings together various helper classes to perform the main task of loading a server report that was created on the server side, then executing and displaying it on the client side as if it were a client report. Therefore, this functionality represents the most important part of the final product, and it is crucial to ensure that the mechanism is reliable and efficient in its work.

The program code of the main class that performs the conversion is available at the link (Golisch, 2021), and the important methods and steps are explained in detail below:

Constructor: ClientReport (string ReportName, System.Collections.Hashtable parameters, string ReportPath, string DBConnectionString, ReportViewer reportViewer). The name of the report (ReportName), parameters, physical location, path to the report (ReportPath), connection data to the database (DBConnectionString), and the component that will display the report (reportViewer) are passed to the constructor .
public void ShowReport (). The main method we call to load the xml file of the server report and call all other methods that will parse that XML, read and execute queries on the database, load and pass the parameters in the report and fill them with the parameter values passed in the constructor, and call methods to display all loaded data in the form of a client report.
LoadReportDefinitionFile (reportViewer.LocalReport, reportFullPath). Loads the server report xml file into the application.
Parameters should be prepared for each database query: ds.AssignParameters (ReportParameters);
For each executed query, the data is saved in a table whose data will be displayed. System.Data.DataTable tbl = ds.GetDataTable (DBConnectionString);
The data is loaded into a visual component that provides an overview of the data. reportViewer.LocalReport.DataSources.Add (rds);
reportViewer.RefreshReport (); this method displays the report on the screen. If the report is complex, it can call for the creation and display of sub-reports that are generated in an identical way and merged with the main report.

4. Data transformation: the process takes place automatically in such a way that the data retrieved from the database is arranged according to the structure defined by the design of the report;
5. Formatting and style is defined by the design of the server report
6. Testing and validation: testing and validation are performed by visual inspection of the report
7. Publication and distribution of reports: the process of publication and distribution takes place simultaneously with the distribution of the application that uses them.

Automatic conversion of server reports into client reports using SQL Server Reporting technology Services (SSRS) offers a powerful and efficient solution for companies that want to simplify their reporting processes. By following the steps and code examples provided, organizations can transform server-side data into a format optimized for client-side presentation and analysis (ReportsConverter). This process saves time and resources, ensures accurate delivery of information to end users in real time, and improves visualization capabilities. Automatic conversion of server reports into client reports provides users with valuable insights for better strategic planning and informed actions. This technology-driven approach solidifies SSRS as a key component in the modern data ecosystem, improving efficiency and competitive advantage.

5. Conclusion

Business applications often generate large amounts of data, and effective reporting systems are critical for decision-making and gaining business insights. An automated conversion process automates the transformation of raw data held on the server into a user-friendly format that can be presented and analyzed on the client side, often through the user's web browser.

Reports are key documents in business that enable the presentation of relevant information and facilitate the analysis of a specific problem or situation and the provision of a recommended course of action. The process of automatically converting a server report into a client report affects the productivity and competitiveness of every organization.

The use of SSRS technology enables taking advantage of client and server reports in terms of design and preview, whereby the tools for creating server reports automatically execute them and send them to clients with the application. This approach ensures accurate real-time information delivery, time savings, performance monitoring and improved visualization.

References

Golisch, T. (2021). *Running a RDL/RDLC (SQL Report) in ASP.NET without SSRS*. Code Project. https://www.codeproject.com/Articles/607382/Running-a-RDL-RDLC-SQL-Report-in-ASP-NET-without-S

Krishnaswamy, J. (2009). *Learning SQL Server 2008 Reporting Services*. Packt Pub.

Lachev, T. (2008). *Applied Microsoft SQL Server 2008 Reporting Services*. Prologika Press.

Landrum, R., McGehee, S., & Voytek III, W.J. (2008). *Pro SQL Server 2008 Reporting Services Rendering extensions overview*. Microsoft. https://learn.microsoft.com/en-us/sql/reporting-services/extensions/rendering-extension/rendering-extensions-overview?view=sql-server-ver16

Microsoft (n.d.). *SQL Server Management Studio (SSMS)*. Microsoft. https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

Microsoft. (n.d.). *SSRS Report Builder*. Microsoft. https://www.microsoft.com/en-US/download/details.aspx?id=53613

Turley, P., & Bruckner, R.M. (2010). *Microsoft SQL Server reporting services recipes: for designing expert reports*. Wrox Press Ltd.