

І.І. Сугоняк, д.т.н., доц.
О.В. Праздніков, ст. викладач

Державний університет «Житомирська політехніка»

Інтеграція WinForms та служб звітності MS SQL Server

У статті розглянуто приклад автоматизації роботи зі звітами MS SQL-сервера на базі додатку WinForms з використанням бібліотеки Microsoft.ReportingServices.ReportViewerControl. Показано способи інтеграції звітів в додаток робочого стола Windows та автоматизації заповнення параметрів звіту. Розглянуто способи обробки даних, якими оперує звіт. Розглянутий експорт звіту в pdf – формат та шляхи перекладу внутрішніх команд звіту для програмної обробки.

Ключові слова: сервер звітів; служба звітності; автоматичне створення звіту; управління параметрами звіту; робота з даними у звіті.

Актуальність теми. Актуальність теми визначається збільшеною кількістю інтеграцій систем управління виробничими процесами, в яких сучасні операційні середовища та інтелектуальні рішення повинні гармоніювати і співпрацювати з системами управління даними, що базуються на рішеннях попереднього століття.

Робота зі звітами – одна з не дуже шанованих операцій у розробці систем управління даними, оскільки пов'язана з прямим впровадженням неочікуваних бажань замовника і іноді включає в себе реалізацію коду, що свідомо не відповідає сучасним нормам (наприклад, багато звітів включають повторювані шматки неоптимального коду, який залишився від попередників).

На даний момент, коли операційні середовища за останні 10 років пройшли великий шлях розвитку, багато баз даних все ще працюють на старих системах управління (наприклад, MS SQL Server 2008 R2). І при спробі встановити такі системи управління даними на нове операційне середовище відбуваються всілякі неприємні речі, пов'язані із сумісністю. Якщо власне СУБД можна без проблем встановити з більш сучасною версією, то різні додаткові опції, які були встановлені з попередньою СУБД у нових версіях або не працюють під управлінням сучасної операційної системи, або їх встановлення та налагодження потребує додаткових ресурсів.

Ситуація, коли MS Report Server неадекватно встановлено на два чи три сервери компанії, може, звичайно ж, вирішитися за допомогою щасливого інженера та бубна, але що, якщо його треба переналаштувати для сотні-двох нових безкоштовних серверів MS SQL? Або ж компанія взагалі вирішила перейти на PostgreSQL або My SQL, – власне таблиці з даними перенести Вам допоможуть усілякі утиліти імпорту-експорту даних, функції та процедури – настільки старі, що підтримують старі стандарти ANSI SQL і готові до безболісного перенесення. Але наявність різного типу звітів, ім'я яким легіон, і призначення багатьох з них покрито мороком? Залишити тільки ті, які потрібні зараз – так через місяць чи рік компанії знадобиться один із тих звітів, які залишили на полиці та розробнику доведеться переносити дані на старий сервер та виконувати їх на старій операційній системі.

Найбільш правильним був би переведення всіх звітів на нові рейки, але це, крім усього іншого, вимагає додаткових фінансових вкладень і багато частин коду, зрозумілих і прийнятних років двадцять тому, які зараз виглядають дуже нетривіально, а розробників, які створили ці шедеври – інших вже немає, а ті далеко.

У такій ситуації найбільш оптимальне рішення – використання бібліотеки простору імен ReportingServices компанії Microsoft.

Ми можемо взяти з серверів MS Report Server поточні напрацювання звітів компанії, перенести їх на локальний комп'ютер і виконувати звіти, підключаючись до необхідних віддалених серверів даних. Причому в такому випадку нам не дуже важливо, яка СУБД встановлена на сервері, аби вона підтримувала деякі стандартні опції ANSI SQL.

Метою статті є розгляд прикладу розробки додатку WinFormsі наявних середовищ даних для подальшого використання при генерації звітності, використовуючи наявні файли звітів MS Report Server.

Отже, для початку нам знадобиться проєкт WinForms, в який ми додаємо Nuget – пакет Microsoft.ReportingServices.ReportViewerControl.WinForms (рис. 1). Якщо цей пакет брати з останньою версією, при експорті звітів у pdf на комп'ютері з Windows 11 періодично виникають проблеми і тому автор рекомендує використовувати версію 150.900.148

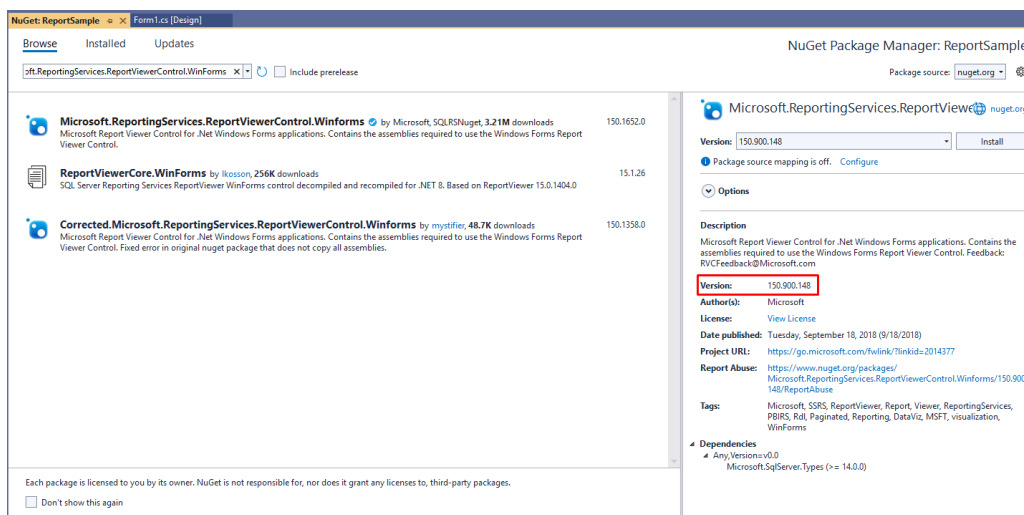


Рис. 1. Додавання пакета в проєкт

Після встановлення цього пакета у нас з'явиться новий елемент керування – ReportViewer, яким ми користуватимемося при перенесенні серверних звітів на локальний комп'ютер. Слід врахувати, що ReportViewer має два режими – ServerReport, для використання якого нам знадобиться MS Report Server, встановлений на деякому пристрої, та LocalReport, який може працювати з локальними файлами і який ми будемо використовувати для роботи зі звітом без публікації на сервері (рис. 2):

```
using Microsoft.Reporting.WinForms;

namespace ReportSample
{
    3 references
    public partial class MainForm : Form
    {
        0 references
        private void InitializeReportViewer()
        {
            // Ініціалізація компонента ReportViewer
            this.reportViewer1 = new Microsoft.Reporting.WinForms.ReportViewer();
            this.reportViewer1.Dock = DockStyle.Fill;
            this.Controls.Add(this.reportViewer1);
            // Вкажіть шлях до RDL-файлу
            reportViewer1.LocalReport.ReportPath = @"C:\Reports\SalesReport.rdl";
            // Додайте джерело даних (за потребою)
            var data = GetSalesData(); // Ваш метод для отримання даних
            ReportDataSource dataSource = new ReportDataSource("SalesDataSet", data);
            reportViewer1.LocalReport.DataSources.Add(dataSource);
            // Оновіть звіт
            reportViewer1.RefreshReport();
        }
    }
}
```

Рис. 2. Лістинг роботи зі звітом без публікації на сервері

За потреби ми можемо встановити параметри звіту (рис. 3):

```
0 references
private void AddReportParameters()
{
    var parameters = new List<ReportParameter>
    {
        new ReportParameter("StartDate", "2025-01-01"),
        new ReportParameter("EndDate", "2025-12-31")
    };

    reportViewer1.LocalReport.SetParameters(parameters);
    reportViewer1.RefreshReport();
}
```

Рис. 3. Лістинг встановлення параметрів звіту

Ну і головне, ми можемо, після того, як звіт згенерований, зберегти його на диск для подальшої передачі замовнику (рис. 4):

```
public static string ExportReport(ReportViewer reportViewer, string path, Store store)
{
    Warning[] warnings;
    string[] streamids;
    string mimeType;
    string encoding;
    string filenameExtension;

    ReportParameterInfoCollection pInfo = reportViewer.LocalReport.GetParameters();
    byte[] bytes;
    if (reportViewer.ProcessingMode == ProcessingMode.Local)
    {
        bytes = reportViewer.LocalReport.Render("PDF", null, out mimeType,
            out encoding, out filenameExtension, out streamids, out warnings);
    }
    else
    {
        bytes = reportViewer.ServerReport.Render("PDF", null, out mimeType,
            out encoding, out filenameExtension, out streamids, out warnings);
    }

    string filename = Path.Combine(path, CreateFileName(store, reportViewer.LocalReport));
    using (FileStream fs = new FileStream(filename, FileMode.Create))
    { fs.Write(bytes, 0, bytes.Length); }
    return filename;
}
```

Рис. 4. Лістинг реалізації експорту у файл

Загальний механізм роботи програми представлено на рисунку 5:

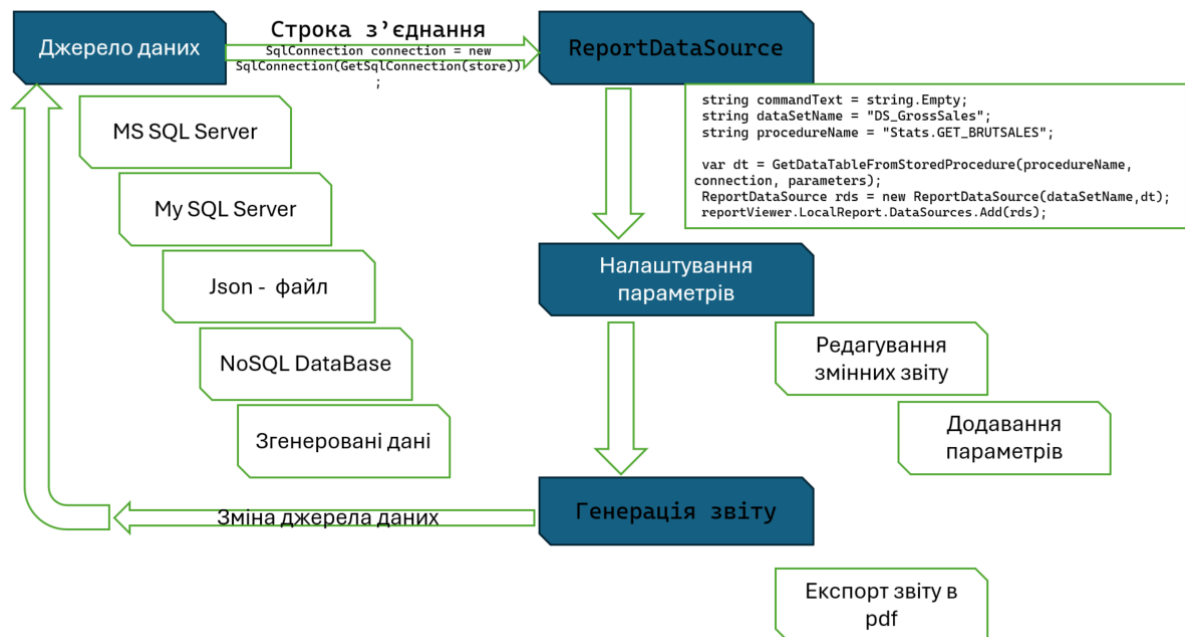


Рис. 5. Загальний механізм роботи додатка

При роботі з файлом звіту (*.rdl) ми можемо переконаватися, що це файл у форматі XML, тому ми можемо працювати з ним як з текстовим документом. Ця опція допоможе нам при локалізації та перекладі тих частин звіту, які не обробляються автоматично. Також можлива автоматизована обробка макета звіту засобами редагування XML-документа.

Приклади обробки внутрішніх даних звіту наведені в таблиці 1.

Таблиця 1

Обробка внутрішньої логіки звіту

Тип даних	Метод обробки
Отримання даних із внутрішніх DataSet	Встановлення значень вручну або програмними засобами
<pre><Textbox Name="textbox7"> <CanGrow>true</CanGrow> <KeepTogether>true</KeepTogether> <Paragraphs> <Paragraph> <TextRuns> <TextRun> <Value>=Code.GetLabel(Parameters!Labels,"100000")</Value> <Style> <FontFamily>Tahoma</FontFamily> <FontSize>12pt</FontSize> <FontWeight>Bold</FontWeight> <Color>SteelBlue</Color> </Style> </TextRun> </TextRuns> </Paragraph> </Paragraphs> </rd:DefaultName>textbox1</rd:DefaultName></pre>	<pre><Textbox Name="textbox7"> <CanGrow>true</CanGrow> <KeepTogether>true</KeepTogether> <Paragraphs> <Paragraph> <TextRuns> <TextRun> <Value>I. CA par DPT</Value> <Style> <FontFamily>Tahoma</FontFamily> <FontSize>12pt</FontSize> <FontWeight>Bold</FontWeight> <Color>SteelBlue</Color> </Style> </TextRun> </TextRuns> </Paragraph> </Paragraphs></pre>
Обробка загальних обчислень	Здійснюється внутрішніми засобами системи
<pre><ReportParameter Name="StartWorkingDay"> <DataType>DateTime</DataType> <DefaultValue> <Values> <Value>=DateAdd("d",-33,Today())</Value> </Values> </DefaultValue> <Prompt>Start Working Day</Prompt> <UsedInQuery>True</UsedInQuery> </ReportParameter></pre>	

На завершення можна зазначити, що застосування ReportViewer в роботі з локальними файлами може значно прискорити роботу з розподіленими середовищами даних – кінцевому користувачеві немає необхідності переходити від одного сервера до іншого, запускати та налаштовувати різні програми роботи зі звітами, встановлювати з'єднання з кожним сервером даних і очікувати виконання. Тепер він може створити список рядків підключення і перебрати ці підключення створити і зберегти звіт для подальшого використання.

При роботі з 5–10 різними джерелами даних це не є досить вагомою причиною, а при використанні сотні магазинів ритейлу або страхових відділень, об'єднаних тільки загальним форматом передачі даних – локалізація та автоматизація звітів вивільняє досить велику кількість чоловічого та машинного ресурсу.

Список використаної літератури:

1. Використання елементу керування WinForms ReportViewer [Електронний ресурс]. – Режим доступу : <https://learn.microsoft.com/en-us/sql/reporting-services/application-integration/using-the-winforms-reportviewer-control?view=sql-server-ver17>.
2. Виклик звіту SSRS з Windows-додатком на C# [Електронний ресурс]. – Режим доступу : <https://stackoverflow.com/questions/28400897/invoking-ssrs-report-from-c-sharp-windows-application>.
3. Інтеграція коду .NET та служб звітності SQL Server [Електронний ресурс]. – Режим доступу : <https://www.codemag.com/article/0701061/Integrating-.NET-Code-and-SQL-Server-Reporting-Services>.
4. Додавання служб звітності SQL Server до Visual Studio 2022 [Електронний ресурс]. – Режим доступу : <https://developercommunity.visualstudio.com/t/Add-SQL-Server-Reporting-Services-to-Vis/1574249>.
5. Використання RDL-файлу локально в .NET для друку звітів [Електронний ресурс]. – Режим доступу : <https://www.sqlservercentral.com/forums/topic/using-an-rdl-file-locally-in-net-to-print-reports>.

References:

1. Vykorystannia elementu keruvannia WinForms ReportViewer, [Online], available at: <https://learn.microsoft.com/en-us/sql/reporting-services/application-integration/using-the-winforms-reportviewer-control?view=sql-server-ver17>
2. Vyklyk zvitu SSRS z Windows-dodatkom na C#, [Online], available at: <https://stackoverflow.com/questions/28400897/invoking-ssrs-report-from-c-sharp-windows-application>

3. Intehratsiia kodu .NET ta sluzhb zvitnosti SQL Server, [Online], available at: <https://www.codemag.com/article/0701061/Integrating-.NET-Code-and-SQL-Server-Reporting-Services>
4. Dodavannia sluzhb zvitnosti SQL Server do Visual Studio 2022, [Online], available at: <https://developercommunity.visualstudio.com/t/Add-SQL-Server-Reporting-Services-to-Vis/1574249>.
5. Vykorystannia RDL-failu lokalno v .NET dlia druku zvitiv, [Online], available at: <https://www.sqlservercentral.com/forums/topic/using-an-rdl-file-locally-in-net-to-print-reports>

Сугоняк Інна Іванівна – кандидат технічних наук, доцент, завідувач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-0484-4839>.

Наукові інтереси:

- системний аналіз та теорія оптимальних рішень;
- проєктування сховищ даних;
- інтелектуальний аналіз даних.

E-mail: org_sii@gmail.com.

Праздніков Олександр Вікторович – старший викладач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0009-0005-6784-080X>.

Наукові інтереси:

- інтелектуальний аналіз даних;
- автоматизація технологічних процесів;
- комп'ютерно-інтегровані технології, штучний інтелект.

E-mail: kkn_pov@ztu.edu.ua.

Suhoniak I.I., Prazdnikov O.V.

Integration WinForms and MS SQL Server Reporting Services

The article considers an example of automating work with MS SQL server reports based on a WinForms application using the Microsoft.ReportingServices.ReportViewerControl library. It shows ways to integrate reports into a Windows desktop application and automate filling in report parameters. It considers methods of processing data that the report operates on. It considers exporting a report to pdf format and ways to translate internal report commands for program processing.

Keywords: report server; reporting service; automatic report generation; report parameter management; working with data in a report.

Стаття надійшла до редакції 01.04.2025.