Migration Guide

SSRS to .NET Core: Your 3 Migration Options Compared

Your .NET 10 migration is blocked by a reporting control that Microsoft never brought to modern .NET — and its cloud replacement is licensed per user, every month. If you're running RDLC reports with ReportViewer, you've hit a wall: no .NET Core support, no Linux, no Linux containers. This guide examines your three realistic options.

The ReportViewer Problem

The Microsoft ReportViewer control has fundamental limitations that block modern .NET adoption:

Platform Constraints

Requires Windows

Requires .NET Framework 4.x

No .NET 8, 9, or 10 support

Cannot run in Linux containers

On Kubernetes, only on Windows nodes

Infrastructure Costs

Windows Server licensing

SQL Server licensing (for full SSRS)

Dedicated Windows infrastructure

In an increasingly Linux-first world

Technical Debt

No new version of SSRS after 2022

SSRS 2022 is supported until January 2033

Keeps the application on .NET Framework

Keeps its servers on Windows

Microsoft's direction is stated: no new version of SSRS after 2022, and Power BI Report Server as the on-premises report server from SQL Server 2025. Both are report servers that run on Windows and query the data themselves; neither is a library that renders an .rdlc from your application's own data.

1 Stay on SSRS (Status Quo)

The simplest option is to change nothing. This only makes sense if your reports aren't business-critical, you have no plans to move to modern .NET, and your application has a defined end-of-life.

The upside is zero migration effort. The downside is accumulating technical debt, no path to Linux containers, and the end of SSRS 2022's support in January 2033.

Verdict: Acceptable for legacy applications with a sunset date. Not viable for anything under active development.

2 Power BI Paginated Reports

Microsoft's cloud path uses the same RDL format: an .rdlc becomes an .rdl whose datasets query sources Power BI can reach. This makes sense if you're already invested in the Microsoft 365/Azure ecosystem, have a small number of report consumers, and running reports in Microsoft's cloud is acceptable.

Pricing Model

Paginated reports are part of Power BI Pro, at **USD 14 per user per month** (USD 168 a year, Microsoft's price on 1 October 2026). Everyone who shares or views a report needs a licence, unless the workspace runs on a Microsoft Fabric capacity (F64 or higher), which is priced by capacity.

Team Size	Monthly Cost	Annual Cost
10 users	$140	**$1,680**
25 users	$350	**$4,200**
50 users	$700	**$8,400**
100 users	$1,400	**$16,800**
300+ users	$4,200+	**$50,400+**

Pros

Microsoft supported
Familiar RDLC/RDL format
Integrated with Power BI ecosystem
Cloud-hosted (no infrastructure to manage)
Email subscriptions built in

Cons

Per-user pricing scales linearly

Reports run in Microsoft's cloud; on-premises data is reached through a gateway

Requires internet connectivity

Reports query their own sources: data your application computes has to be exposed to them

A subscription for as long as anyone reads a report

Verdict: Suitable for organizations with small user bases who are already committed to Microsoft's cloud ecosystem. The cost grows with every reader.

3 Self-Hosted Cross-Platform Library

The third option is replacing ReportViewer with a cross-platform .NET library that renders RDLC reports natively on any platform. This makes sense when you need Linux, Docker, or Kubernetes support, you're migrating to .NET 10 (or to .NET 8 or 9), data must stay within your infrastructure, and you want predictable non-per-user pricing.

The Hidden Cost: Most Libraries Convert Your Reports

Most third-party reporting libraries (FastReport, Stimulsoft, DevExpress, Telerik) have **their own report formats**. Some import RDLC, but a conversion is not a migration. FastReport's guide says scripts, formulas and expressions are not converted and that "full conversion is impossible"; DevExpress's importer does not convert charts, subreports or lists; Telerik has no RDLC importer. For an organization with 50+ RDLC files built over years, that's weeks or months of rework — and the risk of introducing bugs in battle-tested reports.

Evaluation Checklist

- **RDLC compatibility** — Can you drop in existing .rdlc files without modification?
- **Cross-platform support** — Does it run on Linux, macOS, and Windows?
- **No GDI+** — Does it depend on System.Drawing, which .NET supports on Windows only, or on libgdiplus?
- **Current .NET** — Does it run on .NET 10 today, and has it followed each new version?
- **Export formats** — Does it support PDF, Excel, Word, and image output?
- **Licensing model** — Per-developer, per-server, or per-company?

Self-Hosted Options Compared

Capability	SSRS2	MS ReportViewer	FastReport	Stimulsoft
RDLC Compatibility	Runs them as is	Native	Converted; expressions are not	Converted to its format
Linux & Docker	Yes	No	Yes	Yes
.NET 8/9/10	Yes	No	Yes	Yes
Runs existing .rdlc files unchanged	Yes	Yes	No	No

Verdict: Best option for organizations migrating to modern .NET, containerizing applications, or with large user bases where per-user pricing is prohibitive.

A Migration Plan

An example plan for a mid-market company with 80 RDLC reports built over years, migrating from Windows Server / .NET Framework 4.8 to Kubernetes:

Phase	Duration	Activities
Evaluation	1-2 weeks	Test 10-15 critical reports, compare output to existing ReportViewer
Pilot	2-4 weeks	Migrate one module to production, validate with real users
Full Migration	4-8 weeks	Remaining reports, CI/CD pipeline integration, performance benchmarking
Parallel Run	2-4 weeks	Side-by-side validation, output comparison, final sign-off

Key Validation Steps

Report inventory — Document all reports, their complexity, and usage frequency

Compatibility testing — Render each report and compare output to existing ReportViewer output

Performance benchmarking — Measure render times for high-volume reports

Export validation — Verify PDF, Excel, and Word exports match expected output

Integration testing — Test parameter passing, data source binding, and subreports

Common Issues to Check

Fonts — SSRS2 bundles the widths of Arial, Times New Roman, Courier New, Calibri, Segoe UI, Cambria, Tahoma, Verdana, Trebuchet MS, Georgia, Consolas, Aptos and Aptos Narrow; register other fonts' files with `ReportFonts.AddFontFile`

Image paths — May need adjustment for new file system

Expression compatibility — Test complex expressions thoroughly

Subreport references — Verify paths resolve correctly

5-Year Cost Comparison

For a company with 75 report users:

Solution	Year 1	Year 2-5 (each)	5-Year Total
Stay on SSRS	$0*	$0*	**$0***
Power BI Pro (75 users)	$12,600	$12,600	**$63,000**
SSRS2	$799	$799	**$3,995**

* Excludes Windows Server licensing, SQL Server licensing, and opportunity cost of not modernizing.

Key Observations

Power BI Pro costs scale linearly with users — add 10 users, add $1,680 a year

An SSRS2 licence does not scale with user count, so the gap widens as a team grows

Report volume and render count carry no cost at all — there is no metering

The SSRS2 price is public and the same for every company: pricing

SSRS2 Licensing

One licence per company, USD 799 a year, with every feature included — every export format, every chart type, every supported platform. No per-user seats, no per-report metering and no phone-home: the licence key is validated offline, and it keeps working forever with every version released while the licence runs.

Evaluate the whole engine first: install the package and run your own RDLC files; output carries a trial watermark until a key is in place. Buy online and the key arrives by email within a minute.

Pricing and licence

Quick Start: 5 Lines of Code

```
using SSRS2.Reporting.NETCore;

var report = new LocalReport();
report.ReportPath = "Invoice.rdlc";
report.DataSources.Add(new ReportDataSource("InvoiceData", invoiceList));
byte[] pdf = report.Render("PDF");
File.WriteAllBytes("invoice.pdf", pdf);
// Works on Linux, macOS, Windows, Docker — same code everywhere
```

Install: `dotnet add package SSRS2.NETCore`

Docker Deployment

```
FROM mcr.microsoft.com/dotnet/aspnet:10.0 AS base
RUN apt-get update && apt-get install -y --no-install-recommends
libfontconfig1 && rm -rf /var/lib/apt/lists/*

FROM mcr.microsoft.com/dotnet/sdk:10.0 AS build
COPY . /src
RUN dotnet publish /src -c Release -o /app

FROM base
COPY --from=build /app .
ENTRYPOINT ["dotnet", "MyReportApp.dll"]
# RDLC reports render natively on Linux — no GDI+, no libgdiplus
```

Frequently Asked Questions

Can I run RDLC reports on Linux?

Yes. SSRS2 draws reports with its own cross-platform engine, so RDLC reports run natively on Linux, macOS, Docker, and Kubernetes without GDI+ or libgdiplus dependencies. Install via NuGet: `dotnet add package SSRS2.NETCore`

Does SSRS2 require rewriting existing RDLC reports?

No. SSRS2 runs your existing .rdlc files without modification. FastReport, Stimulsoft and DevExpress convert RDLC reports into their own formats, with gaps their own documentation lists (FastReport does not convert expressions, DevExpress does not convert charts or subreports), and Telerik has no RDLC importer. SSRS2 is a drop-in replacement for the LocalReport of Microsoft ReportViewer.

How much does Power BI Paginated Reports cost?

Paginated reports are part of Power BI Pro, at USD 14 per user per month (USD 168 a year), and everyone who views a report needs a licence unless the workspace runs on a Fabric capacity (F64 or higher). For 75 users, that is USD 12,600 a year.

What's the migration timeline from SSRS to .NET Core?

Plan for 10 to 18 weeks for an estate of about 80 reports: 1-2 weeks evaluation (testing critical reports), 2-4 weeks pilot (migrating one module), 4-8 weeks full migration (remaining reports and CI/CD), and 2-4 weeks parallel run for validation.

Does SSRS2 need GDI+ or libgdiplus?

No. SSRS2's native drawing library comes with the package, so there is no GDI+ and no libgdiplus. On Linux it needs fontconfig, one apt-get or apk line in a

Dockerfile. It runs on Windows, macOS and Linux.

How does SSRS2 licensing compare to Power BI?

SSRS2 is licensed per company rather than per user, so the cost does not grow as more people read reports. Power BI Pro costs USD 12,600 a year for 75 users; an SSRS2 licence, USD 799 a year, covers unlimited developers, applications and report renders. Pricing.

Conclusion

SSRS and the ReportViewer control served the .NET ecosystem well for two decades. But the platform has moved on. You have three paths forward:

1. **Stay on SSRS** — Acceptable for legacy applications with a defined sunset
2. **Power BI Paginated Reports** — Works if you're small, cloud-first, and Microsoft-committed
3. **Self-hosted library** — The practical choice for modern .NET applications at scale

The good news: **you don't have to rewrite your reports**. Your RDLC files are portable. The right migration tool lets you keep your existing reports and run them anywhere.

This guide is also available on Medium: SSRS to .NET Core: Your 3 Migration Options Compared

Ready to Migrate?

Try SSRS2 with your own RDLC files: open one in the online designer and see it render on Linux, or generate the sample invoice in the live demo.

Try Live Demo | Buy a licence → | Read Documentation