How to Modernise a Classic ASP Application

11th May 2026

In the Classic ASP era, we built all varieties of web applications with VBScript, ADO and a remarkable level of confidence. Everything lived together in one .asp file - the HTML, the SQL, the business rules, and occasionally a comment reminding future developers to “be careful”.

It wasn’t always elegant, but it got the job done - and some of those systems are still loyally running today, long after their creators expected them to retire.

While these applications still function, they are fragile, difficult to maintain, and do not meet today’s security and compliance standards.

The good news is that there is a solution.

> *You do not need to risk everything with a complete rewrite.*
>
> *You can update your system safely, step by step, without any downtime.*

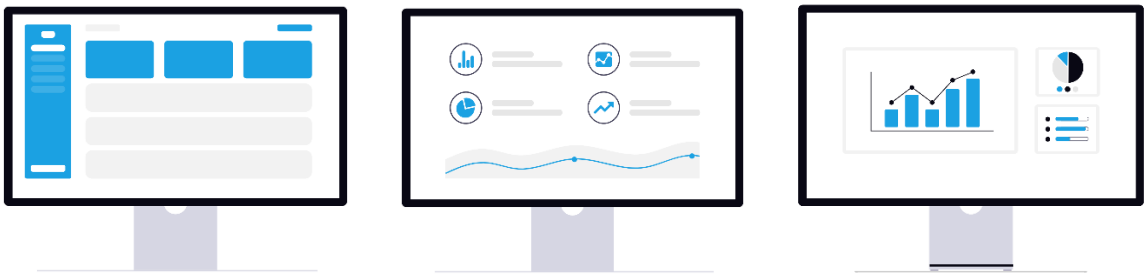
In this article, we’ll walk through the low-risk approach we use to modernise classic ASP applications for UK organisations.

A Safe, Modern Approach: Stabilise, Extract, Replace, and Retire

Modernising a classic ASP system is not about rewriting everything from scratch.

It focuses on making gradual improvements.

Here is a 9-step approach that avoids disruption and delivers value early.



1. Stabilise the Existing System

Before changing anything, we:

- Audit all ASP pages, COM components, and VBScript logic
- Document database usage and dependencies
- Identify high-risk areas (uploads, payments, email, authentication)
- Add monitoring and logging to understand how your applications are being used

This gives us a clear picture of your system and helps you decide what needs attention first.

---

2. Introduce a Modern ASP.NET Core Application Alongside the Old One

This is a key step in the process.

Using IIS URL Rewrite or a reverse proxy, we run:

- Classic ASP for legacy pages
- ASP.NET Core for new pages and APIs

Both coexist under the same domain. This lets you modernise one page at a time instead of everything at once.

---

3. Modernise the Database Layer

Most classic ASP applications rely on:

- Inline SQL
- ADO
- Business logic inside stored procedures

We introduce EF Core or Dapper for fast, secure data access with proper parameterisation, providing a stable foundation for your new application.

---

4. Extract Business Logic Out of ASP Pages

Classic ASP combines UI, logic, and SQL in a single file.

We extract logic into:

- .NET APIs
- Shared libraries
- Background services

Each time you extract logic, the legacy system becomes smaller and less complex.

---

5. Rebuild the UI in ASP.NET Core

We migrate in order of risk:

1. Simple pages
2. CRUD screens
3. Complicated business rules and workflows
4. Reporting and integrations

Each new page replaces the old one via routing rules. Users will not notice the change as the transition happens.

---

6. Replace or Wrap COM Components

Many classic ASP systems depend on:

- VB6 DLLs
- ADO COM objects
- Custom encryption/authentication components

We replace them with modern .NET equivalents or temporarily wrap them to keep the system stable.

---

7. Add Modern Security

We introduce:

- Proper validation
- HTTPS and HSTS
- ASP.NET Core Identity or Azure AD for Single Sign-On
- Secure authentication flows

This updates your system to comply with modern security standards.

---

8. Add Automated Tests

Legacy systems rarely have tests.

We add:

- Unit tests
- Integration tests
- UI tests (Playwright)

This helps make sure the migration is safe and predictable.

---

9. Retire the Classic ASP System

Once all pages and logic are migrated, we archive the old code and decommission the legacy IIS application.

At this point, your system is fully modern, secure, and easy to maintain.

Why This Approach Works

This approach works because it avoids the biggest risk:

A full rewrite that takes years, overruns budgets, and never launches.

Incremental modernisation delivers:

- Early wins
- Lower risk
- Continuous value
- Zero downtime
- Predictable costs

We use this approach for businesses that depend on legacy systems and need to avoid disruption.

Need Help Modernising Your Classic ASP System?

We're a UK team of senior .NET developers specialising in:

- Classic ASP migrations
- VB6/COM modernisation
- ASP.NET Core development
- SQL Server optimisation
- Long-term application support and maintenance

If you would like a free review of your legacy system, we are happy to help.

Scope360 Solutions Ltd

Software Development | Web Applications | Database Solutions | UK Team

Web https://www.scope360.co.uk

Email support@scope360.co.uk