

# Time to Go with the MVC Flow!



By the end of 2020, Kentico will drop Portal Engine and use pure MVC web development. Additionally, support for Kentico projects built on Portal Engine will cease by the end of 2022.

## Why Is Kentico Moving Toward MVC?

Portal Engine	MVC
✗ The ASP.NET Web Forms framework is slowly becoming obsolete	✓ The MVC development model is future ready
✗ It is harder to find developer talent willing to use the Web Forms framework	✓ Easier hiring for developer talent as MVC is a widely used modern development model
✗ Harder maintenance and upgrades	✓ Easier maintenance and upgrades
✗ Slower performance	✓ Faster performance
✓ Drag-and-drop interface	✓ Page Builder's Drag-and-drop, Widget-based interface with superior UX
✓ Hundreds of default web parts natively available out-of-the-box	✗ MVC Widgets need to be created from scratch
✓ Web parts often need to be customized to fit particular projects' needs	✓ Developed MVC Widgets are perfectly tailored for a particular project
✗ Widget selections cannot be natively limited to specific editable areas	✓ MVC Widget selections can be easily limited to specific editable areas

### Why Is the ASP.NET Web Forms Framework Becoming Obsolete?

In 2016, Microsoft introduced the [ASP.NET Core framework](#). The framework is free, open-source, cloud-ready, lightweight, and high performing with support for cross-platform development. It opens a vast range of possibilities for developing high performing websites, regardless of the platform. The catch? It doesn't support Web Forms anymore and focuses purely on MVC.

### In What Other Ways Is MVC Better Than Web Forms?

Aside from the ease of hiring developer talent, the futureproof framework, and clear separation of frontend versus backend application development, MVC

is also significantly more performant. The frontend MVC application only loads the necessary content and allows websites to achieve much faster page loading times, resulting in better responsiveness.

### Is MVC Going to Last?

ASP.NET MVC is not new hype. [It has been around since 2009](#) and is not just a framework; but also a new way of developing. Its Model-View-Controller approach makes code much easier to develop, maintain, and debug. It is built for the future!

### I Bet Switching to MVC Must Be Expensive, Right?

Initially, widgets will need to be created

from scratch. However, in typical projects, there are only a few widgets used regularly. So, even though there will be some preliminary development needed, it will pay off immensely in the long term, as the widgets tailor-built for the project will fit the requirements perfectly and will need minimum ongoing maintenance, reducing expenses and increasing ROI.

### Where Can I Find Other Frequently Asked Questions About MVC in Kentico?

Feel free to visit Kentico 12 MVC [FAQ page](#) for more answers!

## Quick Links to Make Transitioning to MVC Easier

[MVC Transition Guide](#) | [K12 MVC Documentation](#) | [MVC Blog Posts](#)

## Upgrade to Kentico 12 and Enjoy All These Features!

### Kentico 9

- Enhanced Marketing Automation
- Web Farm Support Optimized for Cloud Deployments
- Email Marketing Simplified
- Continuous Integration
- ASP.NET MVC Support
- Integrated Campaign Management



### Kentico 10

- Responsive Images Management
- Contact Management Redesign
- Enhanced Continuous Integration Support
- Campaigns Redesign
- Roslyn Compiler – Enhanced First Page Load Time
- Power BI Integration
- Support for SQL Server 2016



### Kentico 11

- GDPR and Data Protection Support
- Advanced Email Marketing
- Faceted Search
- Azure Search
- Gift cards
- Modular E-commerce Framework
- Campaign and Email Insights
- Macro Signature Identities



### Kentico 12

- MVC Page Builder
- MVC Widgets
- MVC Form Builder
- MVC Content Personalization
- Smart Forms
- E-commerce Business API

### Service Pack

- MVC A/B Testing
- MVC Landing Pages