SVF Upgrade Guide & Checklist

SVF Current version: 15.3.0
Umbraco Latest Version: 16.2.0

This document provides a step-by-step process for upgrading Umbraco CMS to the latest version (e.g., **16.2.0**).
It covers **planning, preparation, execution, and validation** phases to ensure a smooth upgrade without business disruption.

1. Create New Repository

Instead of performing a direct upgrade in the existing repository, first create a new repository, for example, **Umbraco-Upgrade-16.2.0**.

2. Need to Check

2.1 Technical / Security Perspective

1. **Security Patches**
   - Older Umbraco versions often contain vulnerabilities that have been fixed in later releases.
   - Staying updated ensures protection against known exploits (e.g., SQL injection, authentication bypass, XSS).
2. **Framework & Dependency Updates**
   - Umbraco now runs on ASP.NET Core (.NET 8+ for v14+).
   - Modern runtime means fewer dependency risks and long-term Microsoft support.
   - Legacy .NET Framework versions are no longer supported or patched.
3. **Performance & Scalability**
   - New versions have optimized caching, faster backoffice UI, and improved content delivery.
   - This reduces server load and improves resilience under heavy traffic.
4. **Compatibility & Support**
   - Only current and LTS versions receive fixes. Running an outdated version means no official support.
   - Integrations (payment, CRM, APIs) often drop support for older Umbraco/.NET.

2.2 User / Business Perspective

1. **Editor Experience**
   - New backoffice UI is more intuitive, with modern design and faster responsiveness.
   - Richer editing tools: block editor, content apps, and preview improvements.
   - Reduces training time for content editors.
2. **New Features**
   - Media management improvements (focal point cropping, bulk upload).
   - Workflow and publishing enhancements (scheduled publishing, approval flows).
   - Headless API options (Umbraco Heartcore) for omnichannel delivery.
3. **Reliability & Future Proofing**
   - Better stability and fewer crashes in the editor.
   - Easier upgrades in the future since Umbraco follows semantic versioning.
   - Confidence that your site won’t suddenly break due to unsupported software.
4. **Compliance**

    - Staying current helps with GDPR and industry compliance since old software may be considered a risk.

3. Upgrade Path Considerations

Direct upgrade path, but requires checking for **breaking changes** (notifications, caching, package dependencies

4. Pre-Upgrade Checklist

Planning

- Identify current Umbraco version.
- Review target version requirements (e.g., Umbraco 16.2.0 requires **.NET 8 SDK**).
- Check release notes for **breaking changes**:
    - Umbraco Release Notes

◆ **Environment & Dependencies**

- Ensure local dev environment matches target .NET version.
- Verify **NuGet package compatibility** (uSync, Forms, Examine, third-party packages).
- Audit custom code (SurfaceControllers, IHttpHandlers, Events, Notifications).

◆ **Backup & Safety**

- Full **database backup**.
- Backup /media folder.
- Snapshot staging/production environments.

5. Upgrade Execution Steps

Step 1: Update .NET & Framework

As for dot.net project framework is 9.0, there is no change in framework

Step 2: Update Umbraco Packages

```
dotnet add package Umbraco.Cms.Web.Common
dotnet add package Umbraco.Cms.Core
dotnet add package Umbraco. CMS
dotnet add package Umbraco.Cms.PublishedCache.HybridCache
dotnet add package uSync
```

Step 3: Handle Code Changes

- Replace ContentPublishedNotification with ContentPublishingNotification for modifiable content events.
- Remove direct **SaveAndPublish** inside publishing events (can cause recursion).
- Update any legacy APIs replaced in v16 (e.g., caching, media, notifications).
- Migrate deprecated configuration from Startup.cs → Program.cs if needed.

Step 4: Database Upgrade

If you want to upgrade the database, you can run the below command to upgrade the database. But svf project is handled auto upgrade the database.

```
dotnet ef database update
```

```
dotnet add package Microsoft.EntityFrameworkCore.Design
dotnet add package Microsoft.EntityFrameworkCore.SqlServer
```

Then run the site; Umbraco will apply its schema upgrades automatically.
As in database, one new table is created as below
umbracoLongRunningOperation

6. Upgrade Checklists

1. Deploy all published files
2. Replaced the backup live media in the wwwroot folder.
3. Replace the appsettings.json file with the backup from the live app or update the SQL connection string.

7. Post-Upgrade Validation

◆ **Functional Checks**

- Verify backoffice login & dashboard.
- Test content publish/unpublish, media upload, member login.
- Check custom workflows, scheduled tasks, notifications.

◆ **Data Integrity**

- Validate media links, URLs, redirects.
- Confirm multilingual setup if applicable.
- Check Examine indexes rebuild properly.

◆ **Package Compatibility**

- Validate uSync export/import.
- Test Forms, Deploy, or any 3rd-party packages.

◆ **Logs & Monitoring**

- Monitor umbraco.log for errors/warnings.
- Enable application monitoring (Application Insights, ELK, Seq, etc.).

8. Rollback Strategy

- Restore DB & media backups if upgrade fails.
- Roll back to previous release branch in source control.
- Document root cause before reattempting.

9. Final Sign-Off

- Client UAT completed.
- QA approval received.
- Deploy to production with release notes.

SVF Upgrade Guide & Checklist

Table UmbracoLongRunningOperation is using as below

Umbraco utilizes long-running operations to manage tasks that require a significant amount of time to complete, preventing the main application thread from being blocked and ensuring a smooth user experience. This is particularly relevant in the context of Umbraco Cloud and its management portal.

Purpose of Long-Running Operations in Umbraco:

1. Preventing UI Freezing:
When a task takes a long time (e.g., deploying changes, syncing data, complex imports), executing it directly on the main thread would cause the Umbraco backoffice or Cloud portal to become unresponsive. Long-running operations offload these tasks to background processes.

2. Providing Real-time Progress Updates:
The Umbraco Cloud portal, for instance, uses long-running activity dialogs to display the progress of these background tasks, offering users visibility into the status of deployments, infrastructure changes, and other operations.

3. Managing Complex Deployments and Infrastructure Changes:
Operations like deploying new versions of a website, applying updates, or configuring server infrastructure can be time-consuming. Long-running operations enable these processes to occur without user interruption.

4. Handling Data-Intensive Tasks:
Importing large datasets, performing extensive content migrations, or running complex reporting queries can also be managed as long-running operations to avoid performance bottlenecks.

5. Facilitating Asynchronous Workflows:
By deferring these tasks, Umbraco can maintain a responsive interface while the background processes handle the heavy lifting, leading to a more efficient and user-friendly experience, especially in a cloud environment.

SVF Upgrade Guide & Checklist

New features in 16.2.0

Based on the official release notes and related information, here are the key changes and updates included in Umbraco 16.2.0.

New features and improvements

- Property-level permissions: This feature, introduced in Umbraco 16 and refined in subsequent updates, gives administrators granular control over which user groups can read or write to specific properties on a document type. In 16.2.0, fixes and refinements were made to ensure stability and correct functionality related to these permissions.
- Improved content structuring: Umbraco 16 brought the ability to move individual properties or entire property groups between tabs within the Document Type editor. This offers more flexibility when organizing content models and is a notable quality-of-life improvement.
- Backoffice UI improvements: Updates to the user interface include improvements to the Media Picker, allowing more dropped and picked files, and the Tiptap Rich Text Editor, including bug fixes for embedded media spacing and reduced loading layout shifts.
- Enhanced functionality for Umbraco Commerce: If you use Umbraco Commerce, 16.2.0 includes several fixes, such as addressing an issue with multiple textstrings as a property editor and fixing an issue with date formatting related to server culture differences.
- Better logging for Docker: The update makes logging more friendly for Docker environments, which is useful for developers working with containerized applications.
- Optimized performance: The update includes various optimizations, such as improving how documents and media are seeded by looking up from the database in batches.

Technical updates

- Dependencies: The Tiptap Rich Text Editor was upgraded to the latest version (v2.x).
- Configuration options: A new configuration option, UseStrictDomainMatching, was added to provide more control over content routing.

Bug fixes

As a minor release, 16.2.0 also includes various bug fixes and stability improvements, with details available in the official GitHub release logs and download pages.

Current Version: **15.3.0**
Release date: Thursday, March 20, 2025

Upgrade New Version: **16.2.0**
Release date: Thursday, September 4, 2025

Coming New Version: 16.3.0
Release date: Thursday, October 16, 2025

***