Versioning in SharePoint

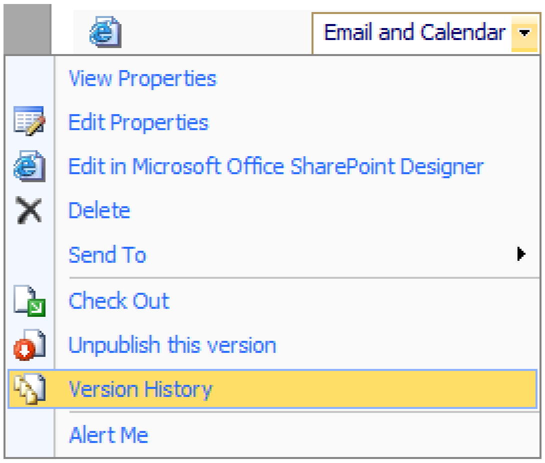
Included in this tutorial:

Introduction to versioning

Versioning enables you to store, track, and restore items in a list or as files in a library as they are changed. When versions are tracked for lists or libraries, revisions to the items or files and their properties are stored. This enables you to better manage content as it is revised and even to restore a previous version (for example, if you make a mistake in the current version). Versioning is especially helpful when several people work together on a project, or when information goes through several stages of development and review.

Versioning is available for list items in all default list types—including calendars, issue tracking lists, and custom lists—and for all file types that can be stored in libraries, including Web Part Pages.

To view the version history of a file or folder, hover over the name of the item with your mouse and select **Version History** from the drop-down menu. A list of versions will be displayed.



You can use versioning to do the following:

- **Record a version history**: When versioning is enabled, you can see when an item or file was changed and who changed it. You can also see when properties (i.e. information about the file) were changed. For example, if someone changes the due date of a list item, that information appears in the version history. For files, you also see comments that people include about their changes.

- **Restore a previous version as your current version**: If you make a mistake in a current version or need to restore part of a document that you deleted, you can easily replace your current version with a previous version. Your current version then becomes part of the version history.
- **View a previous version**: You can view a previous version—for example, to refer to a previous guideline—without overwriting your current version. For .aspx files, you can view only details about the changes that were made to the files, and not the actual pages that the files create.

Libraries can track both major versions (i.e. a new section was added to a document) and minor versions (i.e. a spelling error was corrected). Lists can track only major versions. Lists and libraries can also limit the number of versions that people can store.

To enable versioning, you must have permission to design a list or library.

When versions are created

When versioning is enabled, versions are created in the following situations:

1. When a list item or file is first created or when a file is uploaded.
   - **Note**: If file checkout is required, you must first check in the file in order to create its first version.
2. When a file is uploaded that has the same name as an existing file and the **Add as a new version to existing files** checkbox is selected.
3. When the properties of a list item or file are changed.
4. When a file is opened, edited, and saved. A version is created when you first click **Save**. This version is updated with the latest changes that you make to the file before closing it.
   - **Note**: A version is not created every time that you or another user clicks **Save**, because this would create too many versions.
5. When a file is checked out, changed, and then checked back in.
   - **Note**: If you or another user discards the checked-out version, no version is created.

You can choose to delete a single version of a file (for example, if you know that you made a mistake in that version) which removes that version from the version history. However, if you delete the actual file, all of its versions are deleted with it. By default, when you delete a version, the version is sent to the Recycle Bin, where it can be recovered until it is permanently deleted. Your organization may handle deletions differently, however.

IMPORTANT If your library limits the number of versions that it stores, the oldest versions are permanently deleted when the limit is reached. They are not sent to the Recycle Bin.

Working with major and minor versions

Depending on the needs of your organization, your library may be set up with simple versioning, which tracks only major versions, or it may track both major and minor versions. If people in your group don't often work on several revisions, your organization may only need simple versioning. If many people work on files together and usually create several versions, your organization may want to track both major and minor versions.

Providing two types of versions can help your team to better manage its content. People who work with the content can better understand the current status of a file. For example, a major version is usually one that is ready for a larger group to see and review, whereas a minor version is a draft that someone is still working on.

Tracking both kinds of versions also helps to make the version history more meaningful. A major version is more likely to represent a milestone in the file's development, such as when a file is submitted for review or distributed to others. A minor version is typically used as a routine increment, such as a version that you save or check in while you are still writing the content, or a version in which you correct some minor errors. When you want to view the version history of a file, the major versions may help you to identify the stages of the file's development and make the history easier to browse through.

When major and minor versions are tracked, a version is stored by default as a minor version, unless you designate the version as a major version. When you save a file and close it, the version is tracked as a minor version. You must first publish the file in order for it to become a major version. You can publish the file by using drop-down commands in a library. In some programs that are compatible with Microsoft SharePoint, you can also use commands in the program. By default, each major version can have up to 511 drafts (minor versions), but the site administrator or owner can further limit the number of versions.

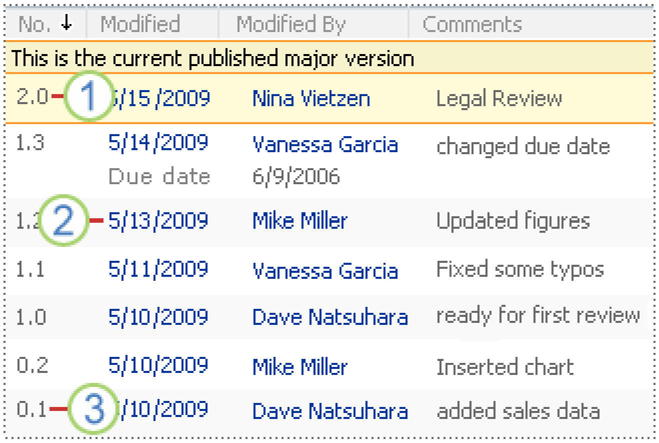
If you have permission to delete versions, you can overwrite a minor version with another minor version. For example, you may want to overwrite a version if you know that the previous version contains an error and you don't need to keep it. If you publish a major version and then realize that you made a mistake, you can turn the version into a minor version again by unpublishing it.

If you check out files before working on them, you can designate which type of version you are checking in. You do not have to publish a file if you designate it as a major version when you check it in.

Version numbering

Versions are numbered as you create them. In a list or library with simple versioning enabled, version 1 is the first version that you create or upload, and subsequent versions are numbered 2, 3, 4, etc.

When you track major and minor versions, the major versions are whole numbers, and the minor versions are decimals. For example, 0.1 is the first minor version of a file, 1.3 is the third minor version of a file that was published once, and 2.0 is the second major version of a published file.



No. ↓	Modified	Modified By	Comments
This is the current published major version			
2.0	5/15/2009	Nina Vietzen	Legal Review
1.3	5/14/2009 Due date	Vanessa Garcia 6/9/2006	changed due date
1.2	5/13/2009	Mike Miller	Updated figures
1.1	5/11/2009	Vanessa Garcia	Fixed some typos
1.0	5/10/2009	Dave Natsuhara	ready for first review
0.2	5/10/2009	Mike Miller	Inserted chart
0.1	5/10/2009	Dave Natsuhara	added sales data

1 The current published major version is highlighted, and the version number is a whole number.

2 A version is created when properties or metadata changes.

3 The first version of a file is always minor version number 0.1.

In a list or library, you can display a **Version** column that displays the version number of files or list items, which can be helpful if your team frequently revises information. Find links to more information about working with views in the **See Also** section.

How versioning works with content approval

Major and minor versioning integrates with content approval for lists and libraries.

When content approval is required, a list item or file remains in a draft or pending state until it is approved or rejected by someone who has approval permission. If the item or file is approved, it is assigned an **Approved** status in the list or library, and it is displayed to anyone with permission to view the list or library. If the item or file is rejected, it remains in a pending state and is visible only to the people with permission to view drafts.

By default, you must first publish a major version of a file before it can be approved. Minor versions are considered drafts that are still being developed, so they don't appear as pending items that are waiting for approval. By default, a pending item or file is visible only to its creator and to the people with permission to approve items, but you can specify whether other groups of users can view the item or file.

When you enable versioning in a library that requires content approval, you can also add a **workflow** (if you or someone in your organization has created one). A workflow controls how your files move through business processes, such as review or approval. You can use a workflow to manage the approval process when major versions are checked in.

For example, a travel agency might use a document library to manage files. While team members develop a new sales proposal, they track minor versions of the file. If they make a mistake in one version, they can restore it to a previous version. When they finish the proposal, they can create a major version and then publish it for approval by their legal department and their manager. When the file is approved, other employees in the company can view the file.

When content approval is required, the people who have permission to read content but who do not have permission to see draft items will see the last approved or major version of the list item or file. If major and minor versions are tracked in a library and no one has published a major version yet, the file will not be visible for the people who do not have permission to see draft items.

How versioning works with file checkout

Checking out files makes the most of versioning. When you check out a file, a version is created only when you check the file back in, which allows you to specifically designate when a version is created. When checkout is not required, a version is created when you first save a file, and then this version is updated when you close it. If you open and save the file again, another version is created. Depending on the situation, you might not intend for multiple versions to be created (for example, if you have to close a file to attend a meeting before you finish making changes to the file).

When checkout is required, you cannot add a file, change a file, or change a file’s properties without first checking out the file. When you check in the file, you are prompted to provide comments about the changes that you made, which helps to create a more meaningful version history.