

## TOC

- Guidelines brief** ..... **2**
- Style Guide** ..... **2**
  - General Style Guidelines ..... 3
    - Pronouns ..... 3
    - Capitalization ..... 3
    - Punctuation..... 3
    - Spelling ..... 3
    - Links ..... 4
    - Terminology guidelines (Glossary) ..... 4
    - Web.link Extension ..... 4
  - Text formatting ..... 4
    - Headings ..... 4
    - Special formatting (UI elements)..... 5
    - Syntax highlighting ..... 5
    - References ..... 5
  - Layout ..... 5
    - Templates ..... 6

---

## Guidelines brief

This pages provide guidelines for using and contributing to the IPFN Wiki.

Don't worry, it is very easy to revert to a previous revision.

You do not need to become an expert in our Wiki. You can simply add or update content.

Please be familiar with our Style Guide before adding or updating content. In particular:

NEVER:

- Add extra whitespace above or below elements ("empty lines") by inserting forced line breaks.
- Indent ("jump in") paragraphs - paragraph styles are formatted globally by templates
- Add horizontal rules ("lines")
- Format lines as headers by making the whole line bold or italic, instead of using the built-in header tags.
- Skip a header level because the next one looks better.

Generally, you should NOT format text very much. Do not alter:

- fonts (except for folder/filenames in Courier new)
- font sizes (use Heading Tags)
- text colours (unless absolutely necessary)
- add text background colours
- indent text (other than in lists)

These styles will be set in templates so any change to the template style will require manual editing of all your formatting.

You may, and should, use:

- Headings for sections
- bulleted or numbered lists (never number lists manually)
- Bold and Italics correctly (see Style Guide)

## Style Guide

This Style Guide outlines the way text should be written and formatted for the IPFN Wiki.

For guidelines not covered in this style guide, refer to [Read Me First! A Style Guide for the Computer Industry](#). For software documentation, it's a good standard to use. If you can't find the answer on this page or in the *Read Me First! style guide*, refer to the *Chicago Manual of Style* ([book](#) or [online](#)).

1.

## General Style Guidelines

Write clearly and succinctly while providing the most accurate explanations possible.

Use active voice and present tense, i.e. "The boamanual displays" NOT "... was displayed on the card manual screen"

### Pronouns

Try to use second person pronouns, i.e. 'you' rather than 'he/she/they'. This creates a more personal, conversational style, as if you were giving instructions to your friend or family member and avoids the gender issue.

You may also use commands, which imply the second person pronoun, such as 'Press Save'

### Capitalization

For the title of a page, AND first level Headings (H1 tags) use title-style capitalization, which means that the first, last, and all other meaningful words are capitalized.

For sub-headings within a page, use sentence-style capitalization, capitalizing only the first word and proper names.

Do not capitalize *internet*, *email*, or *web*.

Capitalize the first word in all list items, whether the list item is punctuated or not.

For all user interface elements and keyboard key names, such as Shift, Ctrl, and Alt, use the exact same capitalization as the display. For example, Menu option Switch View

### Punctuation

Use serial commas in a list, meaning that a list with three or more items has a comma preceding the last item in the list.

Avoid the use of the ampersand (&); use *and* instead.

When list items form a complete sentence, use a period (or other terminating punctuation) at the end of each list item. When unordered (bulleted) list items do not form complete sentences, do not use punctuation at the end of each list item.

### Spelling

If not enabled, please enable spell check (proofread writing) on the Edit Menu. Usually the button with ABC and check mark.

The IPFN Wiki uses English US spell check, so thus, standard American English spellings.

Always spell check any document you create or edit.

Tip: MindTouch keeps changing the default Spell Check, and none of them are as good as browser spell checkers. If you have enabled 'Check Spelling' in your Firefox browser options/preferences you can use it while editing wiki pages:

1. Ctrl+right click - for browser context menu
2. Select Check Spelling from context menu
3. Ctrl+right click on any misspelled word

Works in Firefox so far.

## Links

Many links are automatically placed in pages, but you can also create inline links. When it works well in context, match the link text with the page title that you are linking to. In most cases you should include the path to the Wiki section.

## Terminology guidelines (Glossary)

Acronyms and technical terms should always be linked to a glossary term in the IPFN Wiki, or to another website page, like Wikipedia. Even if you think the Acronym is well know, it may not be known to all new users.

Note: the IPFN Wiki Glossary contains only terms unique to IPFN, or those used in a unique way. All other terms should be linked to Wikipedia or other websites.

## Web.link Extension

One of the best ways to add a link for acronyms or technical terms, is to use the web.link extension. The advantage of this extension is it allows you to display a title/text for the link on mouse hover.

Example: [HDD](#) The code for this is:

```
{{ web.link{uri: "http://en.wikipedia.org/wiki/Hard_disk_drive", text:
"HDD", title: "Hard Disk Drive",
target: "_blank"} }}
```

However, you can easily use the web.link extension:

1. Select Extensions button on the Edit Menu.
2. Select Built in and navigate almost to the end of the list > web.link.
3. Link: add the hyperlink.
4. Text: link contents; can be text, an image, or another document (default: link uri).
5. Title: the caption which displays on mouse hover.
6. Target: be sure to add `_blank` in the Target field so the link will open in a new tab/window.

In future, we hope to add the web.link extension to the Insert Menu to make it more accessible.

## Text formatting

- **Bold:** Labels of GUI elements, for example icon titles, menu items, button labels.
- *Italic:* Accentuated text, for example to highlight differences or possibly confusing terms. Only use for up to 4 words at a time.
- **Code Inline Style:** Short texts and commands to be typed by the user, directories and filenames.
- **Formatted Style:** Larger amounts of text to be entered by the user, or contents of text-files, XML files, code etc.
- ~~Strike through:~~ Text that is no longer valid or current, but for some reason is still needed. For example to strike out completed items in a progress report.

All other text formatting should be avoided! Most formatting and layout will be handled by templates.

## Headings

For the first heading of any edited page, use an H1 style and Title case.

Use sentence-style capitalization for all other headings, regardless of level.

## Special formatting (UI elements)

Italicize *menu names*, and use bold for button names.

Capitalize all User Interface (UI) elements exactly as they are shown in the interface: e.g. Switch View

Window titles need no special formatting.

## Syntax highlighting

Using syntax highlight for xml or C# code (or others) is possible by using the inbuilt syntax highlight extensions.

Syntax highlighting is a feature of some text editors that display text—especially source code—in different colors and fonts according to the category of terms

The easiest way to format text using the Syntax extension is to use the Transform menu:

1. Create a paragraph using the "formatted" style.
2. Paste or write the code you wish to format or select existing text.
3. Select the "formatted" style.
4. Select the appropriate "syntax" transform from the toolbar menu. The Transformations button is usually the last one either first or second row.



Transformations.png

Note: Syntax highlighting only displays when browsing a page, not in Editor or Revision history.

## References

The following style should be used when using references to:

1. Section > Subsection, Category > Subcategory  
Should be separated by arrows (arrow key), as above, not slashes (Section\Subsection) which is used for folders notation. Always leave spaces around the arrows so the terms can be searched, e.g.: Configuration > General > Skin - when linking, link the whole reference, not just part of it.
2. Folder paths/names - should include the backslash and be entered in Courier New font - e.g. : \folder, \windows\programdata, etc. Common system folders used by IPFN, which vary between Operating Systems, should be referenced as: [user files]\thumbs, [user files]\skins etc. using Courier New font.

Please add other styles for references as you find them.

## Layout

NEVER:

- Add extra whitespace above or below elements ("empty lines") by inserting forced line breaks.
- Indent ("jump in") paragraphs.
- Add horizontal rules ("lines").
- Format lines as headers by making the whole line bold or italic, instead of using the built-in header tags.
- Skip a header level because the next one looks better.

## Templates

When you first add a document, you may choose one of the approved templates for the IPFN Wiki.

The templates ensure standards and consistency in the Wiki layout. There are a few important reasons to use templates:

1. Layout actions performed in the wiki code, have to be (manually) repeated on each and every wiki page. While layout code in the stylesheets only has to be defined once to automatically be applied to each page.
2. Not every contributor will know about all style guidelines, so different contributors will have different perceptions of what is the "best" layout. As a result, even on a single page, several different layout styles would be applied. This causes very chaotic and unprofessional looking pages. We don't have the resources to check every page after it's been edited to apply all the indenting dots and hard linebreaks that may have been forgotten or added in wrong places.
3. What looks good in your browser, may not look good in another. While Internet Explorer and Firefox will display most layout (almost) the same, the layout will look very different when browsed by phone browsers, show in the Documentation plugin, or published as printed documentation. Stylesheets can easily be coded to display optimized on many different outputs.
4. A centralized layout would also allow us to adjust the whole site to a new design, which would require lots more work when the layout is hardcoded in the pages.
5. And last but not least, stylesheets offer far superior layout possibilities compared to what is possible with wiki-code.