



<a href="#">Table</a>   <a href="#">Images</a>   <a href="#">Bibliographie</a>   <a href="#">Glossary</a>   <a href="#">Index</a>   <a href="#">Faceted index</a>								
<a href="#">WIT Manual</a>	<a href="#">News</a>	<a href="#">Foreword</a>	<a href="#">License</a>	<a href="#">Tutorials</a>	<a href="#">Reference</a>	<a href="#">CSS 1</a>	<a href="#">FAQ</a>	

Tree :

News  
Foreword  
License  
Tutorials  
Reference  
CSS 1  
FAQ

Faceted  
index, by  
topics :

Menu  
"Normal"  
view (10)  
(Main tree  
view) (1)  
Display (4)  
Edit (8)  
File (45)  
Help (7)  
Ideas (52)  
Import/Export  
(11)  
Keywords (6)  
Publications  
(19)  
Tasks (1)  
Templates  
(72)  
Windows (1)

Feature  
Not in  
freeware  
edition (105)  
Professional  
ed. only (33)

Catégorie  
V5 (45)  
V5.01 (21)  
V5.10 (15)  
V5.20 (30)

How to insert these  
widgets?

## WIT Manual

Welcome in WIT user's manual.

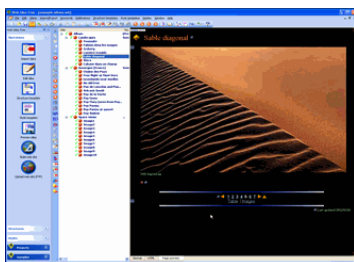
You'll be able to quickly :

- manage, categorize** and **index** all your **personal information** and **knowledge**.
- capture** **information** from the **Web**
- create** **electronic books** with full **bibliography notes**.
- create** **bibliographies, glossaries** and **Indexes** with full **vocabulary control**.
- organize** your **tasks**, and **follow** their **progress**.
- build** and **maintain** **10, 100 or 1000 page web sites**,
- create** efficient and professional looking **Intranets**.

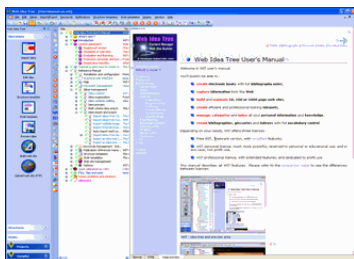
Depending on your needs, WIT offers three licenses :

- WIT personal license, with powerful features, reserved to personal or educational use, and in any case, non profit use,
- WIT professional license, with even more automatic features, and dedicated to profit/commercial use.
- Free WIT, freeware version, with **simplified** features, and up to the version 5.02,

This manual describes all WIT features. Please refer to the [comparison table](#) to see the differences between licenses.



WIT : making an album



WIT : idea tree and preview area

## 1. News

### 1.1. Version 5.20

#### Evolutions

WIT version 5.20 introduces a **new format** for \*.wit documents (it's an extension of the previous format) and takes in charge the conversion.

New Structural elements : **widgets** mimicking blog behavior, automatically built and updated by WIT

- Custom field **Table of Contents Widgets**, showing mini table of contents of any custom field values and pointing to faceted index (like in this page : see the two samples on left side).
- Task field **Table of Contents Widgets**, showing mini table of contents of any task field values and pointing to faceted index.
- Task **progress status Widget**, showing statistics about task progress (10 values) and computed at build time.
- Horizontal **Drop Down Menu** widget
- Vertical **Simple Tree Menu** widget



#### 🔗 Ideas management

- Custom fields can be edited with a WYSIWYG editor for more complex content (links, images).
- Custom fields can now have multiple values per field, with separate entries and links in the faceted index.
- Tasks can be added to the faceted index according to their progress status (late, not started, due today...).

#### 🔗 Publications management

- 15 of the 40 publication fields have now multiple values.

#### 🔗 Style templates

- New CSS styles for :
  - Custom field control
  - Facet names (index)
  - Custom field TOC widgets
  - Task field TOC widgets
  - Task progress status widget

#### 🔗 Graphical User Interface

- Miscellaneous improvements in design and ergonomics (additional keyboard shortcuts, new behaviors).
- New button in the Projects toolbar to change the WIT Project Dir and update the list of projects.

#### 🔗 Miscellaneous

- Search and replace criterion and options are kept in memory for further use.
- Idea search can now be performed either in the whole document or in the local current sub tree.
- Search /replace scope is now either current idea, current tree or whole document.
- Dates are displayed according to ISO 8601 yyyy-mm-dd, both readable and sortable.
- At web site build time, the wit-icons and docs have now subfolders that are structured according to the original places of image and document files.

#### Fixed Bugs

##### 🔗 Major

- Fixed an error when copying resources (images, documents) from a network path (beginning by \\).

##### 🔗 Minor

- Restablishing full compatibility with Microsoft Windows 98.
- WIT <@ /> variable was not interpreted as a bullet, in page headers and footers.
- While building bibliography with 2 grouping criteria, items were not always properly regrouped.
- While building bibliography with grouping criteria, grouping criteria must also be repeated in the bibliographic notice.
- Fix of a loss of just build thumbnails when switching to HTML view, then on Normal view.
- The "paste" button in the mini toolbar of the "preview" view works again.
- The Wysiwyg editor keyboard shortcuts Ctrl + Alt + F and B apply the current foreground and background colors instead of displaying the color choice dialog box.

## 🔗 1.2. Version 5.10 updated

#### Evolutions

##### 🔗 Data import / export

- Mozilla, Firefox and Opera favorite links (bookmarks) can be imported as new ideas.
- Ideas can be exported to CSV files from the idea selector, with filtering search criteria.

##### 🔗 Ideas management

- Custom fields number grows from 10 to 32 and extends faceted classification and indexing capabilities.
- Custom fields styles can be individually controlled in the style template.
- Nice print of current idea, current tree or whole tree, in a few mouse clicks.
- Automatic naming of ideas at creation time, using their title.
- 'Space' character is allowed in idea name (ID).
- New created ideas inherit from sibling idea attributes, then from parent.
- Unified image management, whatever their origin.

##### 🔗 WYSIWYG editing

- A selection containing text can be [pasted, with or without formatting](#) in the WYSIWYG editor.
- [Image fragments](#) (for instance copied from PaintShopPro) can be pasted in the WYSIWYG editor.
- [Indentation & exdentation](#) with TAB and Shift + TAB keys.
- [New keyboard shortcuts](#) for common text formats.
- WYSIWYG editing is now available for idea titles, images and documents legends, and custom fields.
- The ShowGlyphs option is now persistent.

#### Graphical User Interface

- Idea property dialog is now resizable, with its internal tabs.

#### Miscellaneous

- Searched text is highlighted in the "Preview view" in and in the WYSIWYG editor, with distinct colors per criterion.

### Fixed Bugs

#### Major

- Links cannot be any more established accross paragraphs or tables (recommendation of W3C).
- Replacement of accidental \ characters by / in some links.
- Fix of an infinite loop when Ctrl + Shift + E key was repeatedly pressed while WIT was building indexes.
- Fix of a problem while copying text + images between ideas.

#### Minor

- In the keywords manager, keyword counter was not updated after keyword deletion.
- Fix a compatibility problem with some Japanese characters in the HTML code editor.
- Images were not properly captured when pasting from Mozilla/Firefox browsers.
- Including by default the Microsoft DhTMLed.msi component in WIT package for Microsoft Windows Vista.
- Deleting useless line breaks in the CSS rules editor.

## 1.3. Versions 5.0, 5.01

---

WIT versions 5 & 5.0x bring even more power and ease to the organization of your ideas and personal information :

- by adding new features such as task management,
- by automating capture of structured information coming from :
  - the Internet (Web pages content),
  - other softwares (XML, CSV)
  - or simply, from your computer (folders),
- by taking in account the numerous remarks and suggestions made by WIT users (thanks you all !), especially :
  - wished features,
  - graphical user interface ergonomoy.

Please also read the note about the [evolution of WIT document format](#).

#### Graphical user interface

- New "look & feel" with fashion Windows themes.
- The [idea property dialog](#) becomes non modal : several dialog boxes can be simultaneously opened, which simplifies the idea management (and enables drag and drop of idea elements).
- New report grid for the [search and selection of ideas](#) and for [publications management](#). This new grid report lets you :
  - choose the information to display,
  - sort and group ideas by their attributes values,
  - directly edit the idea attributes in its grid cells.
  - It also shows whether a search criteria (filter) is active or not.
- Many new possibilities to drag and drop files, images, documents and ideas between :
  - the idea tree and the idea property dialogs,
  - the file explorer and the idea tree,
  - the file explorer ant the idea property dialogs,
  - between the property dialogs themselves,
  - between the idea property dialogs and the new idea report and selection grid.

#### WIT documents

- Any WIT document is now fully portable ([use of relative path](#)) when its resources are located in the WIT document folder or subfolders.

#### Structure and style templates improved

- [CSS elements inheritance](#) better shown in the style templates.
- [New elements](#) for better style control.
- New [CSS tutorials](#).
- New ready-to use CSS style templates and structure templates.
- Name of previously loaded [structure](#) and [style](#) templates is shown.

#### Idea management

- New idea attributes :
  - [tasks](#)
  - [meta information](#) used in the generated web pages
  - statistics attached to every idea.
- These new attributes are used for :
  - web page creation,
  - in the [structure](#) and [style](#) templates,
  - they empower [WIT intelligent variables](#),
  - they contribute to [faceted indexes](#) (custom fields as well as task fields).
- New Tasks menu in the main window to follow task progress and manage them.
- [Idea IDs and file names](#) now can contain space characters.
- [Highlight of search](#) results in the WYSIWYG editor and the preview view.

#### Idea content editing

- WIT now properly manages Javascript fragments when [pasting](#) data from Web pages :
  - keeps them when fragments are [pasted into the "Page Preview" view](#),
  - cleans them when fragments are [pasted into the WYSIWYG editor](#) ("Normal" view).
- [Keyboard shortcuts](#) are now available in the [WYSIWYG editor](#) for most text editing features.

- Links can be easily cleaned (dedicated function in the [WYSIWYG editor](#))
- Selective cleaning features (professional license) :
  - Removal of style and class attributes
  - Removal of Microsoft Word specific tags
  - All in one removal of text formatting, style attributes and MS Word specific tags
  - Removal of all HTML tags besides paragraphs and line breaks.
- Automatic syntax correction and indentation of HTML code.

#### Data import

- [Folder import](#) : WIT creates ideas with files found in a folder, recreates folder trees with ideas, and can recursively create [photo albums](#).
- [CSV import](#) : It's now possible to specify idea identifiers and hierarchy, instead of just doing "flat" imports.
- [XML import](#) : WIT imports and exports its own idea trees from and to XML.
- Import of structured documents coming from other softwares : WIT can now import data from [TreePad™](#).

#### Publication management

- The [publication manager](#) now uses the same report grid as idea selector : displayed fields can be selected, ordered, sorted and grouped. You can directly edit the publication attributes in the grid cells.

#### Web site generation

- Removal of <meta name="generator" value="WebIdeaTree"> for the professional license.
- Removal of generated <!-- Designed and built with WebIdeaTree --> for the professional license.

#### Global

- Replacement of <WORKDIR> variable by \_WORKDIR\_ and <DATADIR> by C:\Program Files\Web Idea Tree\Data
- Space character is now allowed in page naming.

#### Fixed bugs

Major fixes :

- Focus problem with the left toolbar is solved : no more "blocking" of WIT.
- Fix of an infinite loop while importing an XML idea tree including internal identifiers already used in the current document.
- Fix of a WIT crash when dragging / dropping ideas in the main tree view (only with MS Internet Explorer 7 installed).
- In the evaluation of WIT, the choice of the type of evaluated license was not persistent.
- When launched for the first time from a new Windows user account, WIT would fail : fixed.
- WIT did not respond while trying to import images from Outlook Express messages. Fixed.
- Memory overlap eventually leading to a WIT crash.
- Correct processing (url encoded) of some paths including special characters.
- Potential loss of modifications done in the HTML view, when the idea property dialog is used after.
- After automatic image thumbnails insertion, associated HTML code would be lost when activating the HTML view.
- WIT works again with Windows 95, 98, 98SE and ME.
- Better management and portability of images and documents when the \*.wit file is moved.

Minor fixes :

- Idea text editing : you can now cut or copy text containing links to images and local documents, and paste it in another idea without losing the links. (Note : this works properly only with images and documents that have been included with the WIT version 5).
- Improvement of WIT ability to import images that are contained in fragments of pages that are copied from the Internet (no more "void" links to "wit-imported-images" folder).
- Applet Java (used in some web sites) : the data file (tdm.dat) is now named "tdm.txt" to solve some authorization problems with Microsoft IIS HTTP server.
- The "Preview" folder in your user profile (My Documents) is now cleaned when the last WIT instance terminates.
- Disambiguation of some keyboards shortcuts in the menus.
- "Normal" and "HTML" tabs remained disabled while clicking on some internal WIT links.
- The "Clean HTML" feature has been removed from the main WIT menu (available in the WYSIWYG editor only).
- The chosen Windows Theme for the graphical user interface is now persistent.
- Path of inserted images in the WYSIWYG editor was absolute. Now relative.
- On some computers, the idea text edition field was not properly refreshed when changing it.
- Some images, embedded in pasted HTML fragments were not properly imported due to "\" and '/' characters in their paths.
- Inconsistency between the left toolbar presence and the "Display" menu showing or hiding it.
- No more size limitation in WIT user macros. Was 200 characters (bug), 50 000 now.
- Thumbnail building happened to be incorrect or partial at document load, whereas complete at website build time. Fixed.
- The main "Table" icon in the Table menu (WYSIWYG editor) is effective again.
- More accurate internal criteria to display or not the faceted index (was sometimes displayed whereas there were no values to show).
- When editing an URL, the initially chosen protocol was not shown : fixed.
- When inserting an URL to a local file, the "file://" protocol is now prepended : necessary for Mozilla Firefox browser.
- All line breaks are now coded as CRLF in generated files.
- Fix of a problem on some ignored images when clipboard is auto-pasted.
- Fix of a problem on some ignored images after being resized in the WYSIWYG editor.
- Improvement of preview view computation performance (optimisation of 'extra' subfolder copy).
- Void lines removed in the CSS editor.

## 1.4. Versions 4.20, 4.21

#### Graphical User Interface, and ergonomy

- A new language is available :
  - **dutch** : many thanks to Henricus Roes (KimSystems)
- File and folder selection dialog has now the Microsoft Office 2000 style (with recent files, desktop, computer, network... icons), and keeps the history of latest selected folders..
- Most of GUI objects use now the Windows XP look and feel.
- Improvements of left toolbar (icons, sort of projects, colors...).
- Refresh of main view with F5 key.
- Tooltips on most of list and combo list items, particularly when the text is too long to be displayed.
- New dialog of progress of lengthy operations, with details of current steps (web site building, index building, keyword management...).

#### Support of XML format for import / export of idea trees.

#### WIT freeware version

- WIT is now available in a free version, with [simplified features](#).

#### Better access to idea structure elements from the "Preview" view.

- [Help tags](#) placed in the current idea preview page : these tags give direct access to idea structure elements, to their parameters, or to

- functions that control the structure template itself.
- [Mini toolbar](#) enabling simple navigation (previously viewed page, next page, root idea), page printing, selection copy and direct clipboard paste.
- [Direct paste of information in the "Preview" view](#). This information is automatically placed into the most appropriate structure element of current idea.

#### Enhancement of collection of information for idea content

- Improvement of collection of information for ideas, by automatically pasting different kinds of objects (formatted texts, files, images, links...) in the most appropriate idea structure elements.

#### Faceted idea classification : a new usage for [custom fields](#)

- The 10 idea custom fields allow each idea to be added as an entry in a new index (faceted index). Each custom field acts as a classification facet.
- A new faceted index is built with the web site (option in the structure template).
- Direct access to the names of custom fields from the "Custom fields" tab of idea properties dialog.

#### Web site building

- [Conflicts](#) (images or local documents with a same name but located in different folders) are automatically resolved by WIT, with no alteration of names of source resources files.
- Idem for [thumbnails](#).
- [10 new WIT variables](#) matching individual custom fields values can be used in idea text. They are substituted by the custom fields values at web site build time.

#### Fixed bugs

- Major :
  - Crash in automatic save when the WIT document was opened using network neighborhood instead of named logical drive.
  - On some computers, installation program launched two WIT instances, and WIT crashed.
- Minor :
  - Menus translation incomplete (error in multilingual behavior)
  - Hardcoded "Save As" message.
  - All options of WIT document management are now saved in the WIT document instead of registry : FTP parameters, target folders for web site build.

## 1.5. Version 4.10

---

#### File management

- WIT keeps all open files paths in memory, for each managed file type. (For instance, place of the last linked image, of last WIT document, of the last structure template that has been used...)

#### Information capture, import / export

- Capture of site web fragments : you can [copy and paste images and text](#) from the Web browser
- Automatic import of images contained in HTML files
- [Mass import of HTML files](#) (without confirmation for each file)
- When [auto import](#) function is enabled and "Page preview" view is displayed, WIT creates new ideas for every Web content that is copied into the Windows Clipboard, and keeps the original formatting.
- When importing ideas from CSV files, a [third separator](#) lets specify titles for imported images and documents.

#### Ideas management

- When creating idea associations from a "A" idea to a selection of ideas, you can now create [reciprocal associations](#) from each selected idea to the "A" idea.

#### Keywords and index management

- While [collecting keywords used by an idea tree](#) to add them to keyword list, the occurrence number is displayed, and you can also sort words by occurrence or by length.
- Added some [cross references in the keyword manager](#) : numbers and lists of ideas using every keyword are shown.

#### Publications management

- Publication display (in publication manager) has been improved : sort criteria is kept in memory and reused.

#### Structure templates management

- [New left and right side panels](#) are optionally built in generated pages (similar to page headers and footers).
- Table of content title and table of images title are now independent.

#### Style templates management

- [Import / export of CSS style sheets](#), which eases the partial or total transfer from one style template content to another one.

#### Miscellaneous changes

- Faster startup of WIT when the project directory has many WIT documents.
- [Installation of WIT is now possible if you are not administrator of your PC](#).
- WIT does not write any more in the "Program Files" folder, so Windows "guest" users can use it.
- WIT can be used from several Windows user accounts on the same machine : user settings and project folder are stored into the user profile.
- When an idea is not [eligible for table of contents](#), this does not prevent any more local thumbnail building from one of its parent ideas. However, thumbnail generation in the table of images still depends on idea eligibility for table of contents.

#### Fixed bugs

Major bugs:

- Network paths are now taken in account (type : \\Share\Folder\File)
- Infinite loop while selecting local URLs like file://
- Creating thumbnails from freshly inserted images is possible again.
- Thumbnails : changes were lost when they were built with the "Page preview" view enabled, and when switching to the "Normal" or "HTML" view.
- It's now easy to insert relative links into the WYSIWYG editor.
- Multimedia documents are correctly embedded in the HTML code again (.../docs was missing in the links).
- When "Page preview" tab was enabled, keyword shortcuts such as Ctrl + X were redirected to the tree view (cutting the current idea). They

are now ignored.

- Fatal exception making WIT quit without notice, when encoding UNICODE text to UTF-8 when special and not recognized characters (coming from Microsoft Word) were encountered. These unknown characters are now replaced by a blank.

Minor bugs :

- Idea selector : ideas can now be selected while typing their first title characters.
- CSS styles are correctly interpreted by the browsers in the keyword index (Unicode version)
- Merge of table cells in the WYSIWYG editor is possible again.
- Field order inconsistencies fixed in some dialogs.
- Elimination of duplicates when importing publications (this made the document size increase) : imported publication is kept.
- Elimination of duplicates while importing keywords.
- When building the bibliography, the progress indicator could reach more than 100% : fixed.

## 1.6. Version 4.0

---

### Graphical User Interface and ergonomics

- More dockable toolbars, remembering their positions in the main Window.
- New dedicated toolbar for the tree view window.
- New toolbar and new quick menus in the WYSIWYG editor
- New design of fashion icons
- New Windows XP-like menus
- A new language is available :
  - **Swedish** (base and Unicode versions) : many thanks to Mats Arvandal.

### File management

- WIT documents are **locked** while editing to prevent multiple and concurrent access.

### Idea management

- **Idea header and footer** can now be edited with the same **WYSIWYG editor** as the text.
- **Idea associations** (links), when inserted into the idea text, now use a portable form (<a href="@idea-id">), better interpreted and maintained by WIT.
- When **importing CSV files**, multiple fields can be concatenated in a single idea file.
- A confirmation is now needed to **sort ideas by title** (tree view window).

### Image management

- Images are now managed in the same way, whatever directly inserted in idea text or in general properties.
- Images can be dragged and dropped into idea text or idea properties dialog box.
- Selection of multiple images when inserting them in idea text or properties.

### Document management : [links to local document files](#), [links to URLs](#)

- Documents links are now managed in the same way, whatever directly inserted in idea text or in general idea properties.
- Documents files can be dragged and dropped into idea text or idea properties dialog box.
- Multimedia documents can be dragged and dropped (and embedded when recognized) into idea text (WYSIWYG editor and idea properties dialog box).
- Multiple selection of documents when inserting them in idea text or properties.
- Multiple \*.url files can be dragged and dropped into idea text or properties.
- Current selection is kept as title when inserting a single link in the WYSIWYG editor. When inserting multiple links, selection is replaced by link names.
- New URL selector dialog, with predefined internet protocols.

### Bibliography management

- WIT now generates a **real bibliography index** with a three level sort, and keyword management for publications.
  - First level sort generates a primary header when second level sort is enabled.
  - Second level sort generates a secondary header when third level sort is enabled.
- **Publication fields** have been regrouped and renumbered for better use and identification (with no side effect on already existing WIT documents).
- The **publication manager** now displays all bibliographic fields and can use any field to sort publications : this is useful for error detection.
- Improvements in the **publication editor** (accessed through the publication manager) :
  - WYSIWYG editor for multiline fields
  - Image and URL chooser
  - Comments on current field are displayed when clicking on the field header.
- When **importing CSV files**, multiple values can be concatenated in a single publication field.

### Style templates management

- User defined styles are shown in color
- Added a new CSS style 'table' to overcome a bug in MSIE with font sizes.
- When inserting a CSS style in idea text through the WYSIWYG editor, last inserted style is kept in memory and shown in the menu.
- WIT automatically updates your style templates with new elements when new WIT versions introduce new CSS elements.

### Web site building

- New **option** : images and documents can be left to their original positions (not copied, but linked), so web site co-workers can modify them without rebuilding the whole site (useful in collaborative groups).
- WIT saves last target folders for every document.

### Minor changes

- The "Expert mode" is enabled by default
- WIT **options dialog** box has now an "Ok" button.
- **Conflict detection and resolution** (different images and documents with same names) was improved.
- A confirmation is now needed to **import MS Internet Explorer® favorite links**.

### Bug fixes

Major bugs fixes :

- Index. Words beginning by ä, ü, ö, etc... are now accepted. Generalized to all scriptings.
- Fixed a random crash in index building.
- Fixed an infinite loop while exporting ideas with non unique IDs.
- After a change made in HTML tab on idea text, switching on preview tab and then on the normal (WYSIWYG) tab caused the loss of changes. Fixed.

Minor bugs fixes :

- Idea tree export did not work while the WYSIWYG editor was active.
- Infinite refresh loop in the preview if the page contained <iframe> tags.
- "an unnamed file contains an invalid path" : solved.
- Remaining hard coded text labels are now translatable (many thanks to Mats!).
- No need to add an additional space when inserting a single link on current selection.
- When last document reload option was active, a double click on a WIT document opened both files.
- Default width and height of Java applet are now 100%.
- When running multiple WIT instances, the preview directory was cleaned while quitting the first instance.
- Canceling an operation in the WIT Options dialog does not display any message.
- Broken links in the "Tip of the day" dialog fixed.
- Added missing # in color specifications of foreground and background of top and left frames at build time.
- Preview failed with Mozilla failed when pathnames included spaces.

## 1.7. Version 3.2


---

 Ideas management, text editing :

- Sort ideas by titles
- Embed [multimedia](#) files in your pages
- New text [editing functions](#) : formatting, cleaning, applying CSS styles
- Insert [folding texts](#) (Javascript)
- New special [WIT variables](#) : <@day />, <@month />, <@year />
- More control over [management of child ideas](#) in generated pages (idea properties, "Display" tab)
- [Autopaste](#) of every copy in the clipboard

 Graphic User Interface :

- [Help tags](#) : quick access to most useful page objects.
- Two new languages are available :
  - [Italian](#) (all versions) : many thanks to Francesco Dell'Orso
  - [Polish](#) (Unicode version) : many thanks to Pawel Wawrzyszko
- Better guidance between [structural and style elements](#)

 Bibliographies :

- [Cross references](#) to ideas using a publication reference in the bibliography.

 Web site building

- More control over [Java applet](#) behavior
- [Index rebuild](#) on demand

## 2. Foreword

---

WIT is a powerful Web site creation tool, WIT organizes your ideas and publishes them in a few mouse clicks.

Web site creation is **easy** and **methodical** : no HTML skills required, but expert users can keep full control over their pages.

WIT makes you **efficient**, and you **stay focused on what you mean** : organize your ideas in a tree, add some text, images or links to documents, a style and structure template, and instantly build the whole site.

WIT manages all technical details for you, and ensures consistency of all links, even when you move your information. You don't have to maintain the generated code : just modify your ideas, then rebuild the Web site.

You **save time** : build hundreds of pages in seconds.




You also may insert dynamic behaviors (Javascript, PHP, ASP...), and create new styles (CSS style sheets and code generation rules).

WIT produces stand alone Web sites : browse them locally, as electronic books, or upload them with FTP on Intranet or Internet servers.









WIT has a tutorial and a full contextual on-line help, all built by the software.

Registered users have free upgrades and a technical support from the author (by e-mail).

### PROFESSIONAL USE :

-  Enterprise documentation systems
-  Business repositories
-  Intranet sites
-  Knowledge management
-  Marketing or technical presentations
-  Hypertext helps
-  Data compilations for R&D.

### PERSONAL USE :

-  Electronic books
-  Memoirs
-  Glossaries and bibliographies
-  Personal notes management
-  Personal Web sites
-  Photo albums
-  Collections, inventories
-  Genealogic trees

**This user manual, has been fully created, organized and generated with WIT, just as the contextual help of this software.**

See also :

► [Definition : CSS](#)

## 2.1. WIT main features

---

Here are the main WIT features :

**Editing :**

- 🔗 **Graphical organization** of ideas,
- 🔗 **Instant preview** of Web pages, and intuitive **site browsing**,
- 🔗 **External data import** (formatted texts, images, delimited data lists(records), database tables (ODBC)),
- 🔗 **WYSIWYG** text editing,
- 🔗 **Multi document** and multi views editing.
- 🔗 **Unlimited** document and Web site **size**.

**Capture and agregation of Information :**

- 🔗 **Capture of formattted text** (coming from Internet pages and numerous softwares such as Word or Excel),
- 🔗 **Complation** and references to **external documents** and URLs,
- 🔗 **Integration of Images** (locally or on the Internet), and **multimedia contents**(sound, video and image),
- 🔗 **Automated Import** of HTML documents, images and folders,
- 🔗 **Auto Import** of any copy made in the Windows clipboard.

**Electronic document management :**

- 🔗 **Bibliographic references** handling,
- 🔗 Automatic **glossary** and **tables of contents** builder,
- 🔗 Automatic **table of Image thumbnails** builder,
- 🔗 Assisted build of rational **Indexes** from **keyword** lists,
- 🔗 **Hypertext references** are automatically built and kept **consistent**,
- 🔗 **Error detection** (missing documents, cross references broken),
- 🔗 **Reports** on used resources (images, documents, URLs), for cross referencing.
- 🔗 **Faceted classifications**

**Automation and publication :**

- 🔗 **Automatic generation** of consistent and full browsable Web sites with navigation menus,
- 🔗 **Pre-defined** structure and style **templates** which apply to whole site,
- 🔗 Precise **control over structure and style** of generated Web pages,
- 🔗 Full **contextual help**, **FAQ** and **tutorial** built with WIT,

**Web Integration :**

- 🔗 Direct editing of **raw HTML** code (expert mode),
- 🔗 Support of **foreign character encodings** or **right to left** scripting languages, such as Arabic, Hebrew, Chinese, Japanese, Russian, etc...
- 🔗 Support of all HTML tags : **forms** and **tables**, etc...
- 🔗 Support of **client side scripts** (Javascript...), or **server side scripts** (PHP, ASP, etc...).
- 🔗 **Fast generated HTML code**, well formed, compact and XHTML + CSS compliant,
- 🔗 **Meta tags** are built from keywords and titles for an efficient indexing by **Internet search engines**,
- 🔗 Automatic **FTP upload** of Web sites (only modified files).

## ● 2.2. WIT philosophy

---

WIT was designed to easily manage idea hierarchies, enhance them with textual descriptions, images or references to other resources (ideas, HTML documents, Web sites...), in order to stay focused on content.

WIT is a Web site generator. It is not an HTML editor, even if it finally generates HTML pages. Structure and style of these pages are separately handled, by applying predefined structure and style templates.

WIT ensures that such a generated Web site will have a coherent and uniform aspect. It frees the user from technical concerns (HTML syntax, hypertext links coherence, title numbering, automatic calculations of indexes, tables of contents, ...).

## ● 2.3. WIT usage principles

---

Your Web site is above all made up of content, its form comes in a second time.

WIT is, in a first time, used as an idea organizer (or data manager). You define then the layout and style of idea rendering, by using predefined templates, or, for advanced users (expert mode), by creating new templates. At any time, a complete Web site can be generated from currently being organized ideas. This site can be locally visited, and when tested, be directly deployed on Internet.

Since the Web site is entirely produced by WIT, there is no need to retouch generated code : all subsequent updates are made in a cycle : idea organization and editing, automatic generation, test and deployment.

The tutorial illustrates this working method.

See also :

- ▶ [A quick tour to create a Web site](#)

### ● 2.3.1. Web site features

---

More precisely, WIT is based on an internal representation template of electronic document, made up of structured elements allowing a *total control* of final rendering.

This rendering deals with **structure** and **style** of elements making up the Web site.

A Web site is generally made up of a **home page**, using or not some **frames**, and, in this case, one of these frames can be used as a quick access way to site content (site map, table of contents, ...).

Beyond the home page, the site is made up of the **useful pages**.

Each **page** complies to a common structure (HTML header (*meta tags*), background, page header (general title...), page footer, navigation bar...).

Each **page** hosts one, several or all current document **Ideas**.

Each **Idea** is made up of **elements** which display, order and individual structure can be modified (title, idea text, keywords, image lists, like to other ideas, bibliographic references, etc...).

Last but not least, the Web site includes WIT automatically calculated elements : table of contents, title numbering, keyword indexes, glossaries, list of bibliographic references.

In addition, to each **element of structure** is associated a **style** of presentation (color, font, size, margins, etc...) defined according cascading style sheets syntax (CSS), such as the W3C recommends.



See also :

- ▶ [Definition : CSS](#)
- ▶ [Definition : W3C](#)

### 2.3.2. Structure templates

---

All **elements of structure** of Web site (frames, pages, navigation bars, ideas, tables and index) are grouped in templates which can be separately loaded, modified or saved.

So, predefined templates of structure allow to modify in a few mouse clicks, the organization and presentation of a full Web site, whatever its page number or its complexity.

Numerous options are given (expert mode) to precisely control templates of structure, and an immediate preview of current page allows to evaluate effects of modifications regarding all the Web site.

### 2.3.3. Style templates

---

To each element of structure of the Web site (frames, pages, navigation bar, ideas, tables and index) is associated a **presentation style** allowing to control final appearance : color, font, size, margins, etc..., according to syntax of cascading style sheets (CSS).

These styles are grouped within templates which can be separately loaded, modified or saved.

So, predefined style templates allow to modify in few mouse clicks the appearance and page layout of a complete Web site, whatever its page number or complexity.

Numerous options are given (expert mode) to precisely control templates of style, and an immediate preview of current page allows to evaluate effect of modifications regarding all the Web site.

## 2.4. How to begin with WIT ?

---

WIT use is both simple and intuitive to organize your ideas and publish them quickly.

However, some complexity may appear when you wish to go further, for instance :

- customize exactly to your will the appearance of generated pages, both in their structure and style,
- include automatisms (script programming),
- generate PHP or ASP pages instead of HTML,
- go further with content richness and idea inter connectivity : indexing, faceted classification, bibliography, etc...
- automatically import already structured information,
- etc...

So I advise you to :

- do the [tutorial](#), where you'll discover WIT essential features,
- consult the [FAQ : How to begin ?](#) and above all : [Method to use WIT](#)
- then, depending on your needs, refer to this user's manual (use the contextuale F1 key at any time).

Have a good work with WIT !

Christophe Guibert

**WebIdeaTree**  
Information manager and Web site generator.

WIT organizes and publishes your personal information

## 3. License

---

Web Idea Tree (WIT) 5.10 Copyright © Christophe Guibert 2001-2007, *All rights reserved.*

**License**

Please carefully read the following terms and conditions before using this software. Using this software indicates your acceptance of this license agreement.

### 3.1. Registered version

One registered copy of WIT may be used on a single computer by a one person at a time.

You may not alter this software in any way, including changing or removing any messages or windows.

### 3.2. Disclaimer of warranty

This software and the accompanying files are sold **"AS IS"** and without warranties as to performance or merchantability or any other warranties whether expressed or implied.

In no event shall the author be liable for damage of any kind, loss of data, loss of profits, business interruption or other pecuniary loss arising directly or indirectly. Basically, you use it at YOUR OWN RISK.

### 3.3. Evaluation and licensing

WIT is shareware, it is not free.

You may install this program to test and evaluate for 30 days, after that time you must either acquire a personal or professional license of this program or delete it from your computer hard drive.

The 30 day evaluation version is fully functional. After 30 days, save and web site building are disabled.

Credit card secured and crypted ordering is available, as well as fax or cheque ordering, as described in the registration section of *"Help / Registration"* menu. You receive a licensed version of WIT by Email (registration code) as soon as your payment is accepted.

### 3.4. Personal, professional, and freeware licenses updated!

#### Personal license

The **personal** license applies to Web Idea Tree (WIT) usage, and to redistribution of productions which are realized with WIT :

- WIT **personal** version is strictly reserved for **private, personal and non profit use**. Use of WIT in educational context (schools, universities) is allowed, always in a non profit purpose.
- WIT **personal** version allows you to freely publish and distribute the productions which are realized with WIT (web sites, electronic books...), as long as you do not generate profits with these productions. However, if you sell or make some profit with your productions, you must acquire the WIT professional version. You may not sell any production realized with WIT **personal** version.

Companies and commercial organizations must acquire the professional license.

#### Professional ed. only license

The **professional** license applies to Web Idea Tree (WIT) usage, and to redistribution of productions which are realized with WIT :

- WIT **professional** version allows **commercial or corporate use**.
- WIT **professional** version allows you to freely publish and distribute the productions which are realized with WIT (web sites, electronic books...), whatever you generate profit or not with these productions. You can sell your productions that are realized with WIT professional version.

WIT professional license also adds advanced features (see the table below).

#### Freeware license

The freeware license applies to Web Idea Tree (WIT) usage, and to redistribution of productions which are realized with WIT :

- WIT **freeware** version is strictly reserved for **private, personal and non profit use**. Use of WIT in educational context (schools, universities) is allowed, always in a non profit purpose.
- WIT **freeware** version allows you to freely publish and distribute the productions which are realized with WIT (web sites, electronic books...), as long as you do not generate profits with these productions. However, if you sell or make some profit with your productions, you must acquire the WIT professional version. You may not sell any production realized with WIT **freeware** version.

Features are simplified (see the table below) and this version is **discontinued** since WIT 5.02.

Choose your solution : WIT freeware, personal or professional license

Features	Freeware License (up to 5.02 version)	Personal License	Professional ed. only License
<b>Commercial or corporate use</b>	✗ no	✗ no	✓ yes
<b>You can sell the productions made with WIT</b>	✗ no	✗ no	✓ yes
<b>Personal or educational, non profit use</b>	✓ yes	✓ yes	✓ yes
<b>Royalty free publication and distribution of your productions made with WIT.</b>	✓ yes	✓ yes	✓ yes
<b>Menu : File (WIT documents, and web site management)</b>			
🔗 <b>Reports on resources</b> that are used (images, local documents, internal links) and help to conflict resolution.	✗ no	✓ yes	✓ yes
🔗 <b>"Expert" mode.</b>	✗ no	✓ yes	✓ yes
🔗 <b>Automated generation</b> of web site, with index, bibliography, image table, table of contents, and glossary.	✓ Partially	✓ yes	✓ yes
🔗 <b>Generation of pages with suffixes</b> other than ".html" (.php, .jsp, .asp, ...).	✗ no	✓ yes	✓ yes
🔗 <b>Control of META TAGS "author" and "copyright"</b> to protect your creations (at web site build time).	✗ no	✗ no	✓ yes
🔗 <b>Optimisation</b> of built pages for search engines.	✓ yes	✓ yes	✓ yes
🔗 <b>FTP upload</b> of generated web site.	✗ no	✓ yes	✓ yes

🔗 Pretty print of current idea, current idea tree or complete tree.	✗ no	✗ partially	✓ yes
🔗 The mention "Designed and build with Web Idea Tree" is removed from page footer.	✗ no	✗ no	✓ yes
<b>Menu : Edit</b>			
🔗 Copy/Cut/Paste/Delete ideas or idea trees.	✓ yes	✓ yes	✓ yes
🔗 Find text, globally or in specified idea fields.	✓ yes	✓ yes	✓ yes
🔗 Find/Remplace text, globally or in specified idea fields.	✓ yes	✓ yes	✓ yes
<b>Menu : Ideas</b>			
🔗 Idea management : create child idea, insert before or after current idea, move ideas.	✓ yes	✓ yes	✓ yes
🔗 Sort ideas by titles	✗ no	✓ yes	✓ yes
🔗 Invert child ideas order	✗ no	✓ yes	✓ yes
🔗 Eliminate / clean HTML code in idea text	✗ no	✗ no	✓ yes
🔗 Automatically build thumbnails from images belonging to an idea and its children	✓ yes	✓ yes	✓ yes
🔗 Control names of generated files (IDs of ideas)	✗ no	✗ no	✓ yes
🔗 Copy/Paste custom fields between ideas	✗ no	✗ no	✓ yes
🔗 Internal and external preview of ideas.	✓ yes	✓ yes	✓ yes
<b>Idea properties</b>			
🔗 Edit title	✓ yes	✓ yes	✓ yes
🔗 WYSIWYG (visually) edit text	✓ yes	✓ yes	✓ yes
🔗 Edit HTML code of text	✗ no	✓ yes	✓ yes
🔗 Manage PHP, ASP, JavaScript, etc... scripts	✗ no	✓ yes	✓ yes
🔗 Attribute keywords	✓ yes	✓ yes	✓ yes
🔗 Add images	✓ yes	✓ yes	✓ yes
🔗 Add local documents	✓ yes	✓ yes	✓ yes
🔗 Add Internet links (URLs)	✓ yes	✓ yes	✓ yes
🔗 Add internal links to other ideas (associations)	✓ yes	✓ yes	✓ yes
🔗 Add links to publications (bibliographic references)	✓ yes	✓ yes	✓ yes
🔗 Add task fields	✗ no	✓ yes	✓ yes
🔗 Add custom fields	✓ yes	✓ yes	✓ yes
🔗 Set multiple values for task and custom fields for faceted classification	✗ no	✗ no	✓ yes
🔗 Set local graphical options for an idea	✗ no	✓ yes	✓ yes
🔗 Add meta fields	✗ no	✓ yes	✓ yes
🔗 Add stats fields	✗ no	✓ yes	✓ yes
<b>WYSIWYG editing of idea text</b>			
🔗 Copy/Paste selections of text with images, tables from Word, Excel, Internet Explorer, etc... partially keeping original formatting	✓ yes	✓ yes	✓ yes
🔗 Basic formats for text (bold, underlined, color, font, size...)	✓ yes	✓ yes	✓ yes
🔗 Advanced formats for text using custom and reusable CSS styles	✗ no	✗ no	✓ yes
🔗 Clean basic text formats	✓ yes	✓ yes	✓ yes
🔗 Clean advanced CSS styles (<span>, style="", class="")	✗ no	✗ no	✓ yes
🔗 Clean MS Word specific tags	✗ no	✗ no	✓ yes
🔗 Clean basic and advanced text formats, and MS Word specific tags	✗ no	✗ no	✓ yes
🔗 Clean all HTML tags besides paragraphs and new lines	✗ no	✗ no	✓ yes
🔗 Paragraphs (indent, bullets, lists, alignment)	✓ yes	✓ yes	✓ yes
🔗 Insert and edit Tables	✓ yes	✓ yes	✓ yes
🔗 Insert in the text local links to other ideas, images, local documents, and URLs	✓ yes	✓ yes	✓ yes
🔗 Insert in the text local links to multimedia files	✗ no	✓ yes	✓ yes
🔗 Insert folding text (Javascript)	✗ no	✓ yes	✓ yes
🔗 Insert WIT intelligent macros and variables, computed at build time	✗ no	✗ partially	✓ yes
🔗 Assistance to definition and assignment of keywords	✓ yes	✓ yes	✓ yes
<b>Menu : Import/Export</b>			
🔗 Import ideas from delimited text files (CSV, Excel type)	✓ yes	✓ yes	✓ yes
🔗 Import ideas from a database table with ODBC	✗ no	✗ no	✓ yes
🔗 Import ideas from folders	✗ no	✓ yes	✓ yes
🔗 Import ideas from HTML files with their images, or from text files	✗ no	✓ yes	✓ yes
🔗 Import images from a folder (building albums, to be combined with automatic thumbnail creation)	✓ yes	✓ yes	✓ yes
🔗 Automatic import mode of whatever is being copied into the Clipboard	✗ no	✓ yes	✓ yes
🔗 Import of favorite links of Internet Explorer	✓ yes	✓ yes	✓ yes
🔗 Export of WIT document fragments for workgroups, in XML format	✗ no	✗ no	✓ yes
🔗 Import of WIT document fragments for workgroups, in XML format	✗ no	✗ no	✓ yes
🔗 Import of XML Treepad files	✗ no	✓ yes	✓ yes
🔗 Export ideas into CSV files (filtered, sorted)	✗ no	✗ no	✓ yes
<b>Menu : Keywords</b>			
🔗 Keywords management and control of vocabulary (synonyms, preferred keywords)	✓ yes	✓ yes	✓ yes
🔗 Automatic gathering of used words, alphabetical or occurrence sorting, and help to assignment to ideas using these words	✗ no	✓ yes	✓ yes
🔗 Automatic building of indexes and glossaries	✓ yes	✓ yes	✓ yes
🔗 Import / export of keyword lists	✗ no	✗ no	✓ yes
🔗 Copy/Paste of keywords between ideas	✗ no	✗ no	✓ yes
🔗 Rebuild of all the indexes	✓ yes	✓ yes	✓ yes
<b>Menu : Publications</b>			
🔗 Publication management (creation, modification, sorting, filtering)	✓ yes	✓ yes	✓ yes
🔗 Partial save (with filter) of a publication list	✗ no	✓ yes	✓ yes
🔗 Load or add a list of publications to the current bibliography	✗ no	✓ yes	✓ yes
🔗 Import of publications from a CSV text file (Excel type)	✓ yes	✓ yes	✓ yes
🔗 Export of publications from a CSV text file (Excel type)	✗ no	✓ yes	✓ yes
🔗 Import/Export of publications from tagged text files (analysis of academic bibliographic formats)	✗ no	✓ yes	✓ yes
🔗 Define, save and load academic bibliographic formats parameters	✗ no	✓ yes	✓ yes
🔗 Automatically build bibliographies	✓ yes	✓ yes	✓ yes
🔗 Control and customize bibliographic format	✗ no	✓ yes	✓ yes
<b>Menu : Structure templates</b>			
🔗 Settings of home page and site structure (using or not frames, Java menu applet,...)	✗ no	✓ yes	✓ yes

Settings of page global structure (meta tags, header, footer, left and right panels, separators, navigation bar)	✓ partially	✓ yes	✓ yes
Settings of navigation bar elements	✓ partially	✓ yes	✓ yes
Settings of presence and rendering order of structural elements of ideas, and settings of these elements.	✗ no	✓ yes	✓ yes
Settings of presence and rendering order of structural elements of publications belonging to the bibliography	✗ no	✓ yes	✓ yes
Settings of generated table of contents, table of images, and Java menu.	✗ no	✓ yes	✓ yes
Settings of the index, the bibliography (sorting and grouping), and the glossary.	✗ no	✓ yes	✓ yes
Load pre-defined structure templates	✗ no	✓ yes	✓ yes
Save the current structure template for further reuse in other WIT documents	✗ no	✓ yes	✓ yes
Insert Drop Down Menu Widget	✗ no	✓ yes	✓ yes
Insert Simple Tree Menu Widget	✗ no	✗ partially	✓ yes
Insert Custom field Table of Contents Widgets	✗ no	✗ no	✓ yes
Insert Task field Table of Contents Widgets	✗ no	✗ no	✓ yes
Insert Task progress status Widget	✗ no	✗ no	✓ yes
<b>Menu : Style templates</b>			
Modification of current style template	✗ no	✓ yes	✓ yes
Create custom CSS styles (for the WYSIWYG edition of idea text)	✗ no	✗ no	✓ yes
Import CSS styles into the current template from external CSS style sheets (*.css)	✗ no	✗ no	✓ yes
Export CSS styles from the current template into a standard CSS style sheet (*.css)	✗ no	✗ no	✓ yes
Load pre defined style templates	✗ no	✓ yes	✓ yes
Save the current style template for further reuse in other WIT documents	✗ no	✓ yes	✓ yes
On line contextual help (F1 key)	✓ yes	✓ yes	✓ yes

See also :

- ▶ Import ideas from database tables (ODBC)
- ▶ Insert a special WIT variable **updated**
- ▶ Change idea ID (and web page name)
- ▶ Import an idea tree (XML)
- ▶ Export an idea tree (XML)
- ▶ Define the "author" META TAG
- ▶ Define the "copyright" META TAG
- ▶ Remove HTML format
- ▶ Copy keywords of current idea
- ▶ Paste idea keywords
- ▶ Insert a CSS style from current template

### 3.5. Registration benefits

When registering :

- ▶ you may upgrade **WIT** for free for ALL subsequent update revisions,
- ▶ you get priority support over other users
- ▶ there are no more shareware reminder screens,
- ▶ the message "WIT evaluation version" on generated Web pages is removed,
- ▶ number of generated Web site pages is unlimited after evaluation period,
- ▶ and last but not least, you make possible and support shareware development.

## 4. Tutorials

WIT is now delivered with 3 tutorials :

- The **first one**, included in this manual, teaches you how to easily create a full featured web site with WIT.
- The **second one**, located in a separate \*.wit project, teaches you how to master CSS style elements.
- The **third one**, also in a separate \*.wit project, is a more advanced CSS tutorial, where you'll discover how to use round corners.



See also :

- ▶ Versions 5.0, 5.01

### 4.1. A quick tour to create a Web site

This tutorial is a rapid take in hands of WIT.

Estimated time to complete : 15-30 minutes.

With a step by step creation of a structured document (an example presenting film director Alfred Hitchcock works), main WIT functionalities are described, from initial setting of ideas to final deployment on the Internet.

The result of this tutorial is given in **hitchcock-fr.wit** (French) or **hitchcock-en.wit** (English) files, located in the "*Projects/Tutorial*" directory.

In addition, the reference section of the user's manual elaborates on all the subjects introduced in this tutorial.



Tutorial 1 : create a web site

See also :

- ▶ FAQ : Method to use WIT
- ▶ Tutorials

### 4.1.1. Create and set a document

Let's create a small Web site, presenting a part of film director Alfred Hitchcock works.

#### 4.1.1.1. Create a document

Start WIT, and create a new document : you are asked to save it first (it's a nice idea to save it into a dedicated subfolder).

**Hint :** preferably save the document in a sub-directory of the "projects" directory which was created when installing WIT.

Then give a title to the first idea that is automatically created : type "Alfred Hitchcock".

Since this idea is the first, it is called the document root. It is unique and gives its title to your web site.

We obtain below screen (figure 1) :

- The left side vertical toolbar gives access to most frequently used functions.
- Center screen zone is used to organize idea trees : it's your site outline.
- Right screen zone helps to preview currently selected idea and to modify its main contents.

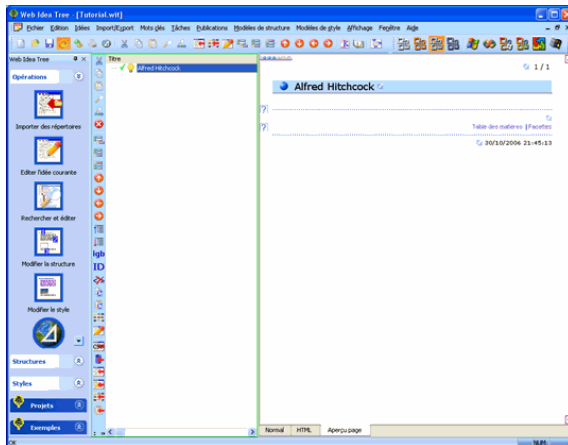


Figure 1 : main screen organization

#### 4.1.1.2. Set document parameters <sup>updated</sup>

Let's customize the document, to adapt it to the subject (present a filmography). This step is optional, but useful in this example.

In addition to predefined fields such as 'title', 'text', 'keywords' or 'associations', characterizing an idea, WIT offers 32 additional and customizable fields : you can define their names.

For this, use the menu : "File/Options", select the "Custom fields" item, and in the dialog box, tick all fields to activate them.

Enter the names below (see figure 2) :

- Title (French)
- Year,
- Duration,
- Character 1,
- Character 2,
- ...
- Character 7.

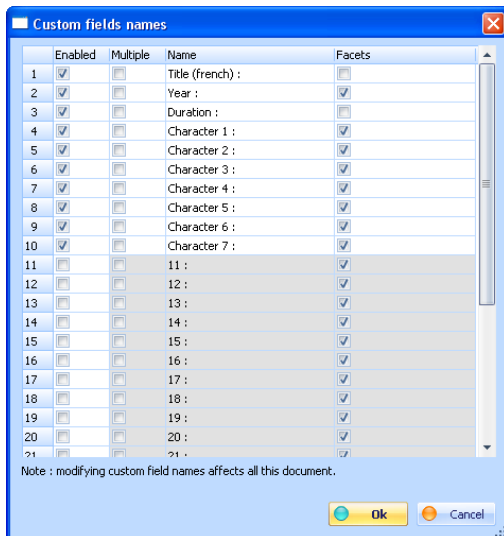


Figure 2 : custom fields

#### 4.1.1.3. Save the document

While pressing Ctrl+S (or using the menu), save the document.

See also :  
[▶ Work directory](#)

### 4.1.2. Outline : organize ideas

It's now time to create the site outline by organizing ideas.

#### 4.1.2.1. Create an idea

In the left view, select the root idea "Alfred Hitchcock".

Press "F2" or "Insert" key, (or use menu *Idea/Create child idea*) to create a new idea.

Give it a title : "Biography" then validate the title dialog box.

The child idea has been created and is previewed in the right part of the screen ("Preview" tab).

#### 4.1.2.2. Preview an idea

Whenever you select an idea in the left tree, the corresponding rendered Web page is displayed in the right screen zone. You may notice that every page has a header and a footer that can be modified (later in this tutorial).

For a complete page rendering with a Web browser (Internet Explorer or Netscape), press "Alt + V" key, or use menu *Ideas/Preview (external browser)*.

If an external browser is used for the first time, WIT asks for its pathname (generally located in a subdirectory of "Program Files" of your system). Choice of default browser can be made through menu *File/Options/Web Browsers*.

**Note :** you should better use Internet Explorer (version 4 or later), rather than Netscape (version 4.5 or later), because, excepting few cases, CSS recommendations are today better implemented in MSIE than in Netscape.

#### 4.1.2.3. Add new ideas

In the tree of the left view, select the "Biography" idea by clicking on its title.

Press "F3" or "Ctrl + N" key to create a same level idea (sister idea), but after this one. Modify its title : type "The beginnings" in the title dialog box.

Press "F3" key again to create another new idea, after last created one. Modify its title in "Film works".

#### 4.1.2.4. Change idea order

Current selected idea can be changed by clicking with the mouse, or by using keyboard up and down arrows.

Ctrl+Up et Ctrl+Down keys change ideas order (when they are children of a same parent idea).

While clicking on an idea icon, and keeping left mouse button pressed, the idea and its children can be dragged and dropped into another idea.

If the Ctrl key is pressed, ideas are copied instead of being just moved.

### 4.1.3. Edit idea contents

An idea has a title, some text, and other fields named "Idea properties".

Select any idea, and press "Alt + Enter".

These properties are shown in a property dialog (figure 3), to allow change of idea contents. The property dialog is made up of several tabs :

- "Idea" : Definition of idea itself (title, text...),
- "Links" : Links to images or other local documents files,
- "Custom fields" : Custom fields,
- "Associations" : Associations to other ideas or books,
- "Display options" : idea specific local display options.

The first tab is the most important.

Jump from tabs to tabs while selecting them or using "Ctrl +Tab" keys.

Please note that the "Custom fields" tab is configured according to the document options you previously defined.

Close the dialog box.

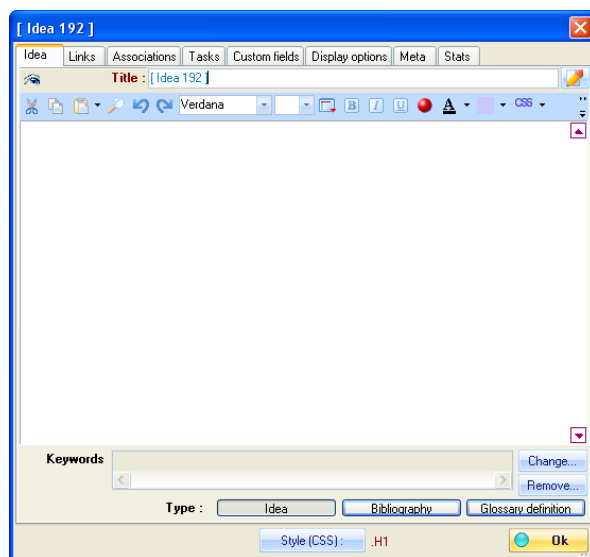


Figure 3 : Idea properties dialog

#### 4.1.3.1. Modify idea properties

Idea properties can be modified from current selection in the main view :

- by double clicking on the idea,
- by pressing "Alt + Enter" key,
- with menu Idea/Properties,
- with the button "Edit idea" in the left side toolbar.

Double click on root idea, to enter text for this idea (see figure 5).

Flows of text can be typed in the WYSIWYG editor, and, even more convenient, be cut and pasted from any text editor or word processor (standard Ctrl+C/Ctrl+V keys).

It is possible, in WIT WYSIWYG editor, to define bold, italic, underlined or colored text : select the text and click on one of the mini toolbar buttons. Similarly, markers (bullets) can be inserted, and substituted by small images at rendering time (use preview function). So simplified HTML tags are inserted to *locally* control text aspect.

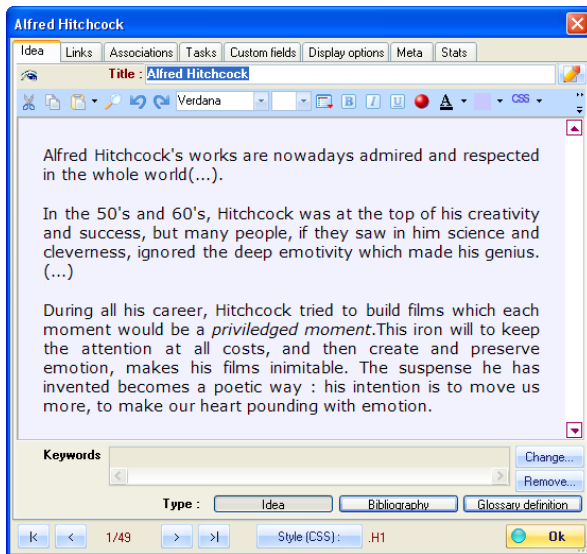


Figure 5 : editing text.

#### 4.1.3.2. Type idea text

Go back to first tab and type a title in replacement of the default title. For example : "Alfred Hitchcock's works".

Validate the property dialog and check that the main view is updated.

Another convenient way to enter text is to use the main view : click on the "Normal" tab in main view, and you can now type some text in a bigger WYSIWYG editor.

The text of ideas in this tutorial is already typed in a file which is provided with WIT (**Projects/Tutorial/Hitchcock/text.txt**) : open it with your favourite text editor, and copy/paste the text in the "Normal" tab (or in the idea properties dialog : same effect).

Click again on the "Preview" tab to have a full page preview.

#### 4.1.4. Import ideas

Let's now automatically fill ideas describing A. Hitchcock's filmography, by using a simple database : a CSV text file with records made of fields and separating characters (delimiters).

The delimiter is a '@' character, and the fields are :

- Title (English)
- Title (French)
- Genre
- Year
- Duration
- Comment
- Character 1
- ...
- Character 7

First, click on the "Film works" idea.

In the "Import/Export" menu select the "Import ideas from CSV file", and, in the dialog box, select the fields as shown in figure 6 : double click on chosen fields (or use "<<" and ">>" buttons), and set field order with "Up" and "Down" buttons.

You can set all these options at once by loading the predefined filter "Tutorial.iif" while using the "Load" button.

Note that 'Genre' field in the data file is mapped to keywords for each idea : it will be used as an index when the Web site is created.

Type separating character '@' and validate.

Chosen format is then shown in a message box. In case of an error, cancel to go back to the previous dialog. Finally, select the data file : **Projects/Tutorial/Hitchcock/films.txt**.

Imported data are transformed into child ideas of current selection : here, all A. Hitchcock's films are placed under the "Film works" idea.



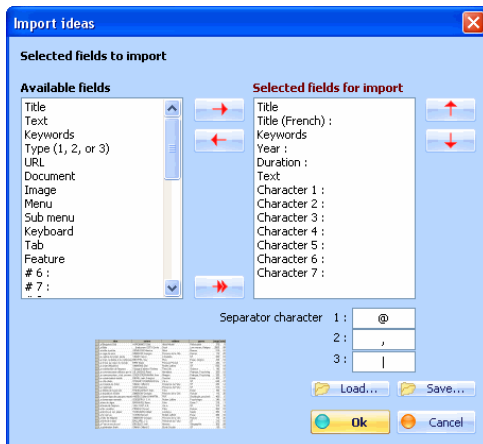


Figure 6 : import ideas.

#### 4.1.5. Add images

Let's add an image to our site first page.

Click on root idea "Alfred Hitchcock" and display its properties by pressing "Enter" key.

Click on "Links" tab in the dialog box. In "Images" section, click on "Add".

In the dialog, type a legend : "Alfred Hitchcock", and select the image *Hitchcok.jpg* in the *Projects/Tutorial/Hitchcock* subdirectory.

Validate the dialog box.

Validate the idea property sheet.

Preview current idea by clicking on the left side toolbar button "Preview idea".

#### 4.1.6. Add bibliography

Add a bibliography : click on the menu "Publications / Add a list" and select the file : *Projects/Tutorial/Hitchcock/Hitchcock.pbl*.

Click on root idea ("Alfred Hitchcock"), display its properties and click on "Associations" tab.

In "Publication" area, add a book by clicking on the "Add" button, and, in subsequent dialog, select François Truffaut's book.

Validate all dialogs.

A bibliography reference has been added to current idea (use preview to check).

#### 4.1.7. Choose structure and style of Web site

We have just entered main information.

Let's choose now the order in which we want to display elements (the structure) and the aspect (the style) of this information, by proceeding from the most general to the most specific details.

**Note :** you can also choose predefined structure and style templates, by selecting them in the left side vertical toolbar.

##### 4.1.7.1. Predefined structures

By clicking on "Structures" tab of the left side toolbar, you can choose in two mouse clicks all structure options of the Web site which will be generated.

Click on the chosen template (for example "Classic"), validate the confirmation message, and check this choice effect in the preview area.

##### 4.1.7.2. Tune structure options (expert mode)

This function is only available in *expert mode* (menu *File/Expert mode*).

Fine tuning of structure options can be made with predefined structures, already loaded, or from a default structure (default options obtained when a new document is created, inherited from the "Basic" template).

##### 4.1.7.2.1. Site structure : using frames ?

First choice in structure options deals with first generated site page : this page can be simple or composed if it uses frames.

Click on the "Structure template" icon in the left side toolbar.

A property sheet is displayed. Set the options as shown in figure 7, in order to use 3 frames.

- Upper frame hosts the title,
- Left frame hosts a Java applet with organizes a table of contents,
- Main frame contains the Web site pages, beginning by home page.

We use here the default home page, but you can substitute a HTML page of your choice, provided it contains a link to the first page containing the root idea (link to *text/index.html*).

For the impatient reader, eager to see a first result, please go to the tutorial section *"Generating and testing Web site"*.

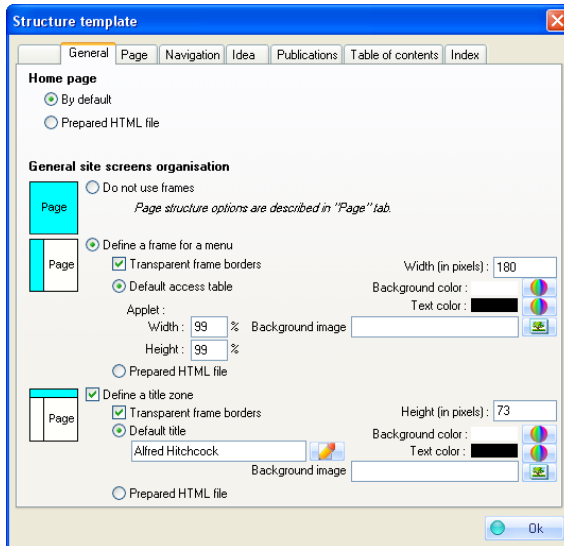


Figure 7 : Web site general options (frames)

See also :

- ▶ [Generate and test Web site](#)
- ▶ [Automatic Web site generation](#)

#### 4.1.7.2.2. Page structure

From the dialog box obtained through icon "Structure template", click on the "Page" tab, and set options as shown in figure 8, and summed up below.

- Options in header are needed for site referencing (HTML meta tags),
- Suppress title on each page, since a frame title is used instead,
- Generate one idea per page,
- Add a separator and a text in the page footer,
- Keep default background options,
- Choose an image as separator (click on the image button and select a ruler image).

Note : < DATADIR > directory is the default WIT data area. Validate dialog box : page options are taken in account in the integrated preview area.

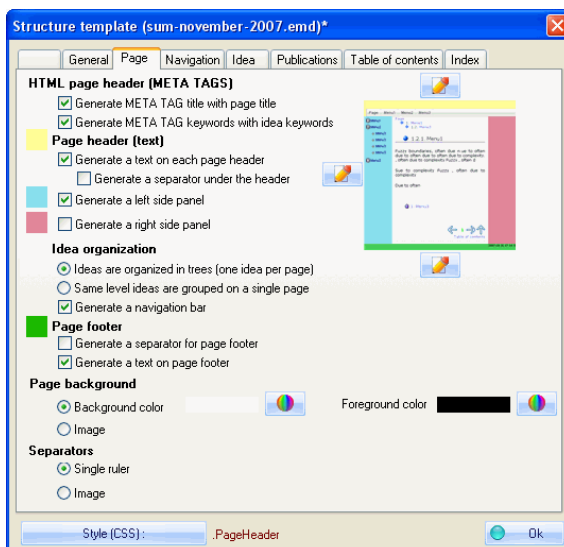


Figure 8 : page structure options

#### 4.1.7.2.3. Navigation bar structure

We'll now replace part of navigation bar texts with images, in the bottom of each page.

From the dialog box raised by the icon "Structure template", click on the "Navigation" tab :

- Activate the options "... arrow image" and select images (buttons "..."), as shown in figure 10.
- Unselect the options "Also display this text".
- Activate the option "Index of keywords".

Validate dialog box, and click on an idea.

In the preview area, the navigation bar should look like figure 11.

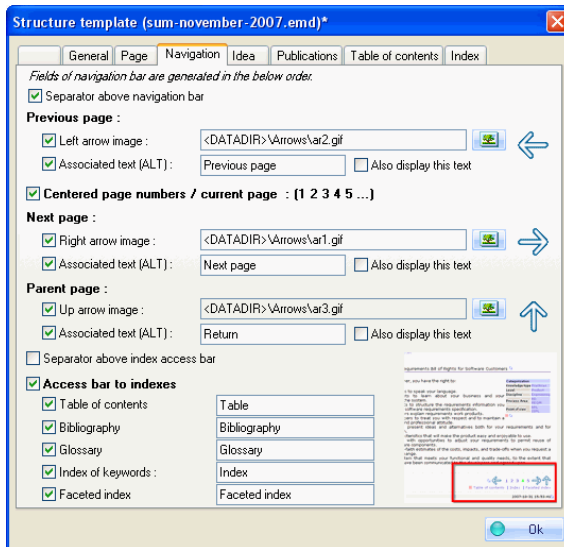


Figure 10 : navigation bar options



Figure 11 : result of navigation bar customization.

#### 4.1.7.2.4. Idea structure

From the dialog box obtained through the icon "Structure template", click on the "Idea" tab.

We'll now choose which idea elements will be displayed and in which order.

We want to display the custom fields that we have defined in document options : select "Custom fields" item in the left list of available fields (see figure 9), and add it to the right side list (with a double click, or with the ">>" button). In the right list, sort the fields so "custom fields" item follows "keywords" item.

We'll now choose the options of each item, by double clicking on them in the right list :

- Title : keep it unchanged,
- Keywords : replace "keywords :" by "Genre : "
- Custom fields : activate the option "Display also field name".
- Text : unchanged.
- Image list : unchanged,
- Links to child ideas : replace "Links to child ideas" with "Contents : ".

Validate the dialog box and click on a film : in the preview area, ideas are displayed according to the chosen structure.

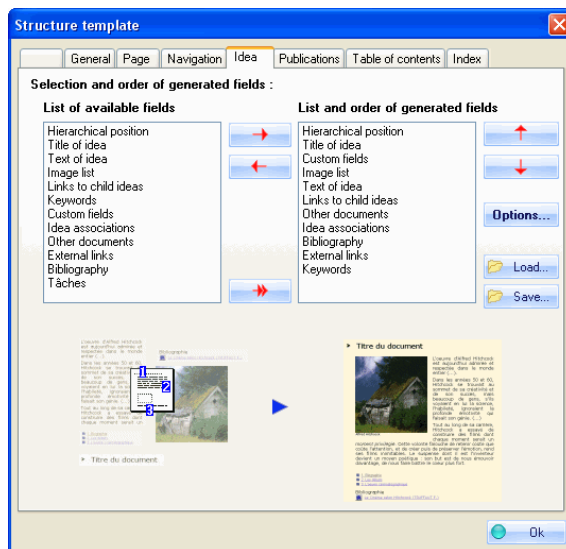


Figure 9 : idea structure options

#### 4.1.7.2.5. Table of contents structure

Figure 12 below shows which table of contents options have been chosen. The main (first) table of contents is included in the site (a HTML page, accessed through the navigation bar).

The second table is available if the left side frame is activated (Java applet).

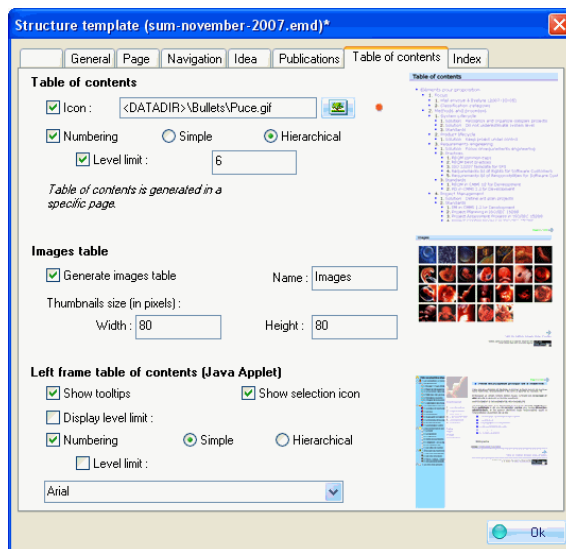


Figure 12 : table of contents options

#### 4.1.7.2.6. Index structure

We just customize here the index generated from keywords (remember : you have used the "genre" of films as keywords).

So this index will help to alphabetically sort the films and group them by genres.

Choose options as shown in figure 13.

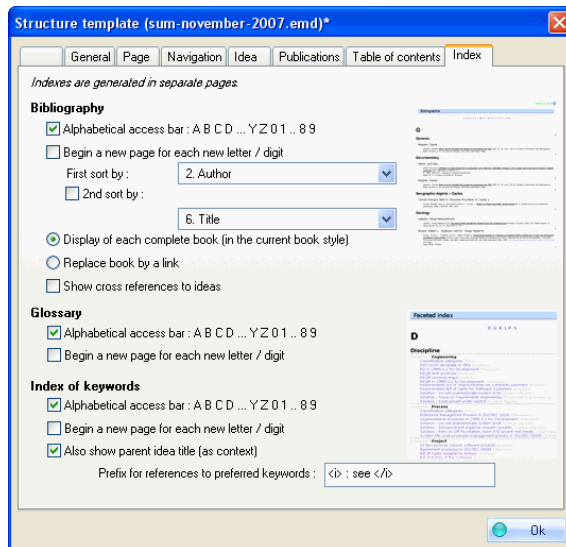


Figure 13 : Indexes options

#### 4.1.7.2.7. Save structure options

If you want to reuse all this structure options, you can save them.

Use the menu *"Structure Templates/Save..."*, and save the current options in the sub directory *Data/Templates/Structures*, under the name *"user-1"*, for example. (You can of course save elsewhere, but this is a method to add new structure templates in the toolbar : see below).

Then, save the document, quit the application, and launch it again.

From now, this new structure options set (template) is displayed in the tab *"Structure Templates"* of the left side vertical toolbar.

Click on this tab to check that *"user-1"* template is available.

#### 4.1.7.3. Predefined styles

By clicking on the tab *"Styles"* of the left side toolbar, you can now, in two mouse clicks, choose all style options of the about to be generated Web site.

Select the root idea.

Click on the chosen template (for example *"Tutorial"*), validate the confirmation message, and watch the result of this choice in the preview area.

Finally, choose the *"Tutoriel"* style.

You can also define all styles in the style template : please refer to the reference manual *"Style templates"* section.

#### 4.1.8. Generate and test Web site

Now, we just have to produce the real Web site pages.

This operation is fully automatic : click on the icon *"Build web site"* in the left side toolbar, or press the *Ctrl + E* key.

Select the folder where you want to build the website. It's a good practice to select a void folder, outside of your WIT projects folder, in order to keep a good separation between your production environment (your source files), and the target folders where you will frequently rebuild your websites.

Then press the button *"Build"* : HTML pages are computed and written (progress bar), the Web browser is launched and displays the first page of your Web site : navigate among this pages, their indexes and tables of contents.

*Note : if no web browser has been configured (in the menu *File/Options/Web browser*), a dialog box asks you to choose between Netscape (or any other browser) and Internet Explorer, and to specify their access path (generally in the Program Files directory of your system disk). You may also use the search button to find this path.*

From now, you can already use your web site :

- Browse it locally on your computer,
- Share it with other users on your local network,
- Burn it on a CD-ROM since it is stand alone,
- Upload it by FTP on a server.

#### 4.1.9. Upload the Web site on an internet server

Now that your Web site has been created and tested, you just have to upload it on an Internet server (HTTP server), in order to give it a world wide audience.

File upload is made with FTP protocol : your Web server should also be a FTP server : most servers are both. Ask your internet service provider (ISP) for

an allocated space, a user account and a password.

An active internet connexion is mandatory to do the file transfer.

To upload your web site, press the button "*Upload web site*" in the left side toolbar. Set your options as follow :

- FTP server : the server name, given by your ISP
- User account : your account name,
- Password : your password,
- Source local directory : the place where the web site was generated (if it was just created, the path is already filled).
- Target directory (distant) : usually void, if you choose to install your web site at your user space root.

Open your Internet connection, and then, press the button "upload".

Some messages and a progress bar show the progress and success of the upload.

When it's finished, type your URL in a web browser.

Here you are : your document has been published on the Internet, and you will easily modify and update it with WIT whenever you want.

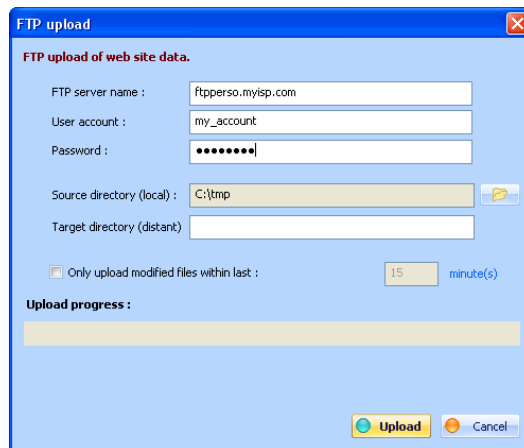


Figure 14 : Web site upload options

## 4.2. CSS tutorial : WIT basics

Welcome in the WIT CSS tutorial : here you'll learn how to configure main WIT style templates features, and create your own designs.

You will obtain the result below.

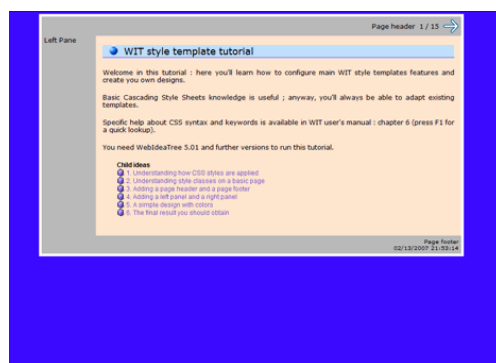
This tutorial is located in :

- My Documents\My Web Idea Tree\Projects\CSS-Tutorials,

and is named :

- CSS-tutorial.wit

Just open the file with WIT and follow the instructions.



Tutorial 2 : WIT CSS basics

See also :  
▶ Tutorials

## 4.3. CSS tutorial : WIT advanced, rounded corners

Welcome in this WIT advanced tutorial on CSS styles.

You'll learn here how to round all those rectangular boxes.

This is done using the excellent [Nifty Corners Cube \(Javascript\)](#), written by Alessandro Fulciniti.

This tutorial starts where the previous one (CSS-tutorial.wit) ended, but all the work is already done : you just have to follow the explanations, then

you'll be able to do your own designs.

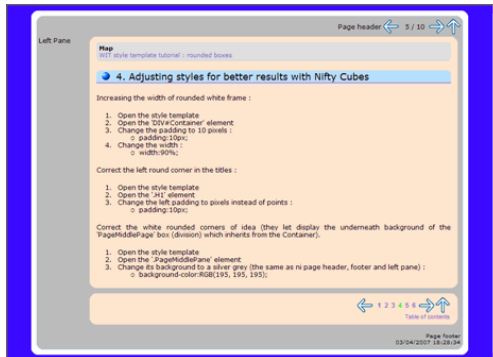
This tutorial is located in :

- My Documents\My Web Idea Tree\Projects\CSS-Tutorials,

and is named :

- CSS-tutorial-2.wit

Just open the file with WIT and follow the instructions.



Tutorial 3 : WIT CSS Advanced

See also :  
► Tutorials

## 4.4. WIT samples

Here are some samples offered with WIT to illustrate how you could use the software. You can view them by using the "Samples" section of the left toolbar.

WIT samples

WIT file name	Short description
3-column web site	Gives the page layout of a typical 3 column web site, using left and right side panels.
Bibliographie-sf	Illustrates an home made (and selective) bibliography of science fiction books.
Bibliography	An academic bibliography.
cmmi-sample	A full featured electronic book. See also <a href="http://chrguibert.free.fr/cmmi12">http://chrguibert.free.fr/cmmi12</a>
Constitution-UE	The french text of the European Union Constitution. 1100 pages with a huge index and a Java menu for hierarchical access to articles. This sample and the CMMI one illustrate WIT capability to import and quick compile information.
Genealogy-sample	A simple genealogy using custom fields and faceted index classification.
les Fleurs du Mal - Charles Baudelaire	An electronic book : French poet Charles Baudelaire works.
roundboxes	A demonstration of rounded boxes, using page header, footer, left and right panes.
Solaris	A fancy web site using background images and animations.
Hints	A simple web site showing hints about WIT page objects.
wit-resources	A simple album showing all WIT graphical resources : bullets, arrows, backgrounds and rulers. This was done with automatic image import.

They are located in the WIT installation folder : C:\Program Files\Web Idea Tree\Data\Models\PredefinedDocs.

Follow this link for [more samples](#).

## 5. Reference

### 5.1. Installation

**Prerequisite :** Internet Explorer (version greater or equal to 4.0) should already be installed. It is required for WIT WYSIWYG editing and preview to work properly.

Installation is done by executing an auto extractible archive :

- WitSetup.exe (Unicode version for Windows NT4, 2000, XP, 2003 and Vista)
- WitSetupW9x.exe (non Unicode version for Windows 95, 98, 98SE, ME, NT4, 2000, XP, 2003, and Vista)

The Unicode version supports all character sets (Russian, Chinese, Thai, etc...), as well as right to left scriptings (Arabic, Hebrew, etc...), and works mainly with UTF-8 encoding.

The non Unicode (base) version supports only ISO-8859-1 charset (Western Europe).

The only option to set is the directory where WIT is installed, usually "Program Files\Web Idea Tree".

- Specific case : If you are not administrator of your machine (which is frequent in a corporation), you can choose any folder outside the Windows "Program Files" folder, where you should have the right to write. In this case, other users on your machine won't have access to WIT.

The install program copies the WIT files in the chosen installation folder, and creates a "My Web Idea Tree" folder in your user profile (inside the "My documents" folder). This new folder is required to store your preferences, the online help system, and, by default, your WIT projects.

If you are administrator of your machine, all users with a Windows account on your machine can also use WIT : WIT will create the "My Web Idea Tree" folder in their profile when run for the first time.

When WIT is run for the first time in a given profile, it asks you for the default language of graphical user interface and on-line help, then automatically builds this help system in the profile.

You can then use the contextual help system at any time : just press the F1 key.

### 5.1.1. File types managed by WIT

WIT has its own document format (\*.wit) and manages many other secondary file formats.

File types that are managed and produced by WIT : specific formats








Names	Extensions	Default paths	Remarks
WIT documents	*.wit	_PROJECTDIR_ (by default : Mydocuments\My Web Idea Tree\Projects)	Main WIT file format, binary format.
Structure templates	*.emd	_INSTALLDIR_\Data\Models\Structures	Binary format (subset of *.wit format, dedicated to document structure)
Style templates	*.csm	_INSTALLDIR_\Data\Models\Styles	Binary format (subset of *.wit format, dedicated to document style)
Idea format (structure)	*.ifm	_INSTALLDIR_\Data\DataFormats	Binary format (subset of *.emd format, dedicated to idea structure)
Publication format (structure)	*.pfm	_INSTALLDIR_\Data\DataFormats	Binary format (subset of *.emd format, dedicated to publication structure)
Idea import format (CSV)	*.iif	_INSTALLDIR_\Data\DataFormats	Binary format : to preconfigure the idea import dialog from CSV files
Publication import format (CSV)	*.bif	_INSTALLDIR_\Data\DataFormats	Binary format : to preconfigure the publication import dialog from CSV files
Folder import format	*.lif	_INSTALLDIR_\Data\DataFormats	Binary format : to preconfigure the folder import dialog
Publication import format (tagged files)	*.tpf	_INSTALLDIR_\Data\DataFormats	Binary format : to preconfigure the publication import dialog from tagged files
Idea selection grid configuration format	*.wrf	_INSTALLDIR_\Data\DataFormats	Binary format : to preconfigure the idea selection grid
Publication lists	*.pbl	_PROJECTDIR_	Binary format : to save publications extracted from WIT documents, and share them with other WIT documents
Keyword lists	*.txt	_PROJECTDIR_	ASCII (or UTF-8) format to import / export keywords as a WIT document. [One keyword per line]
XML idea trees	*.xml	_PROJECTDIR_	XML format to import / export idea trees between WIT documents
Lists analyzable by WIT or a spreadsheet program (CSV)	*.txt	_PROJECTDIR_	ASCII (or UTF-8) format to import / export ideas or publications of a WIT documents in a CSV format file. [One record per line, with delimited fields]
Lists analyzable by WIT (tagged files)	*.txt	_PROJECTDIR_	ASCII (or UTF-8) format to import / export ideas or publications of a WIT documents in a tagged format file. [Multiline records, which fields are identified by tags]

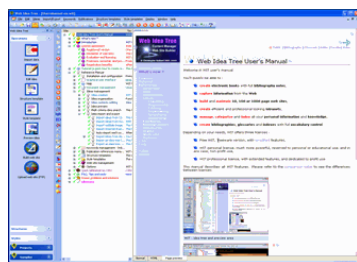
Other file types used by WIT : standard formats

Names	Extensions	Default paths	Remarks
Images (backgrounds)	*.png, *.jpg, *.jpeg, *.gif	_INSTALLDIR_\Data\Backgrounds	Images shipped with WIT
Images (arrows)	*.png, *.jpg, *.jpeg, *.gif	_INSTALLDIR_\Data\Arrows	Images shipped withWIT
Images (bullets)	*.png, *.jpg, *.jpeg, *.gif	_INSTALLDIR_\Data\Bullets	Images shipped withWIT
Images (rulers)	*.png, *.jpg, *.jpeg, *.gif	_INSTALLDIR_\Data\Rulers	Images shipped withWIT
Images (inserted into WIT documents)	*.png, *.jpg, *.jpeg, *.gif	_PROJECTDIR_	Other image formats are not standard (web browsers)
Web pages	*.html, *.php, *.asp, *.jsp, ...	_PROJECTDIR_	ASCII (or UTF-8) format
CSS style sheets	*.css	_PROJECTDIR_	ASCII (or UTF-8) format to import / export / analyze CSS style sheets. [Standard CSS syntax]
External documents (links to these documents)	*.doc, *.xls, *.pdf, *.ppt, *.txt, ...	_PROJECTDIR_	All formats. These files are linked from WIT ideas (HTML links).
Multimedia documents embedded into WIT pages	*.dcr, *.mov, *.mid, *.midi, *.asf, *.asx, *.avi, *.mpg, *.mpeg, *.mp3, *.swf, *.wav, *.wmv, *.ra, *.rm, *.rpm, *.cla, *.class	_PROJECTDIR_	All formats corresponding to standard multimedia plug-ins in Web browsers.
Internet shortcuts (URL)	*.url	-	-

### 5.2. User interface

The graphical user interface can now be displayed in six languages (see the WIT options) :

	English		
	French		
	Italian		Many thanks to Francesco Dell'Orso
	Swedish		Many thanks to Mats Arvendal
	Dutch		Many thanks to Henricus Roes
	Russian	(requires the WIT Unicode version)	Many thanks to Vilis Eglitis
	Polish	(requires the WIT Unicode version)	Many thanks to Pawel Wawrzyszko



Graphical User Interface



### 5.2.1. Main menu

---

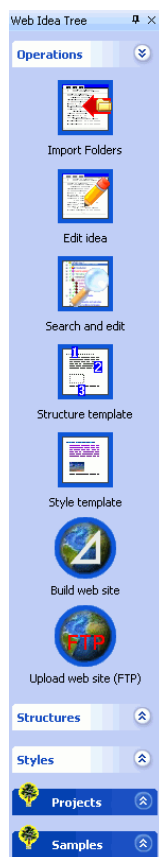
Main menu ( on top of screen) gives access to all WIT functions, as described below.

See also :

- ▶ WIT Documents
- ▶ Options
- ▶ Idea creation
- ▶ Idea organization
- ▶ Idea contents editing
- ▶ Multi criteria idea search
- ▶ Import / Export
- ▶ Keywords
- ▶ Tasks
- ▶ Publications
- ▶ Structure templates
- ▶ Style templates

### 5.2.2. Left side toolbar

---

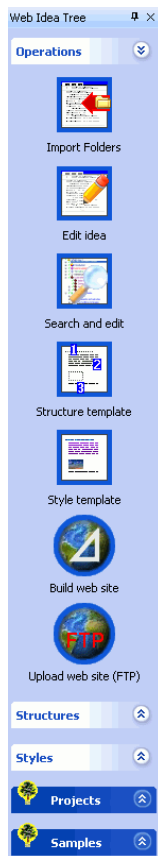


The left side vertical toolbar gives a quick access to most often used functions.

The left side toolbar is made of five sliding tabs.

#### 5.2.2.1. Operations tab

---



This tab (displayed by defaults) helps to :

- Import folders,
- Edit properties of current idea,
- Search and edit ideas,
- Modify web site structure (choice of a predefined template or current template options tuning),
- Modify web site style (choice of a predefined template or current template options tuning),
- Build and testing web site,
- Upload the web site through FTP.

See also :

- ▶ Import ideas from CSV file [updated!](#)
- ▶ Idea contents editing
- ▶ Structure templates
- ▶ Style templates
- ▶ Preview in Web browser
- ▶ Automatic Web site generation
- ▶ Upload on a Web server by FTP

#### • 5.2.2.2. Structures tab

---

This tab, activated while clicking on its title, helps to apply a predefined structure template to the current document.

See also :

- ▶ Structure templates

#### • 5.2.2.3. Styles tab

---

This tab, activated while clicking on its title, helps to apply a predefined style template to the current document.

See also :

- ▶ Style templates

#### • 5.2.2.4. Projects tab [updated!](#)

---

This tab, activated while clicking on its title, gives access to all WIT projects stored in the project directory (\_PROJECTDIR\_), or in one of its subdirectories.

Use the first item button to change the project directory and update the tab.

See also :

- ▶ Project directory
- ▶ Version 5.10 [updated!](#)

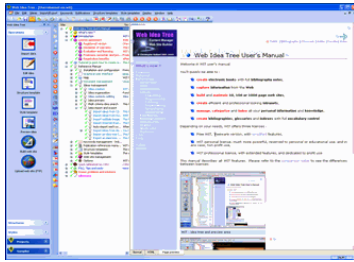
#### • 5.2.2.5. Samples tab

---

This tab, activated while clicking on its title, helps to open predefined sample documents.

### 5.2.3. Main view (idea tree)

Main view helps to create, organize, and interact with ideas.



Main view (middle tree)

#### 5.2.3.1. Tree organization of ideas

Ideas are organized in trees. This helps to sort them and make hierarchies of ideas.

#### 5.2.3.2. Multicolumns and sort criteria

Main view of idea representation has multiple columns.

First column contains a tree, displaying idea titles.

Next columns display other idea attributes (text, type, number of keywords, of associations, etc...).

Displayed columns are defined in the current document options (menu *"File/Options/View columns choice..."*).

Idea display order is by default hierarchy and real position order.

For all columns except the first one, ideas can differently be sorted by clicking on a column title : this column becomes the sort criteria.

A first click on the column title does an ascending sort, a second click does a descending sort. This sort changes the idea presentation but doesn't affect the real idea order : position and hierarchy remain unchanged.

To activate again default display order, click on the first column title.

**Note 1 :** sort is made only among child idea of a same parent.

**Note 2 :** The idea selector now offers better possibilities of idea sorting / grouping and filtering.

See also :

- ▶ Main view columns choice
- ▶ Idea selector **updated!**

#### 5.2.3.3. Selection management

Selection is shown as an inverse video line in the current view.

Keyboard **Left click**  
**Up and Down arrows**

The corresponding idea is the current idea.

Most idea management functions work on current selection ; some work also on child ideas (recursive behavior).

To change selection, just click on another idea, or use "up" and "down" arrows.

#### 5.2.3.4. Display management

Menu **Display**

##### 5.2.3.4.1. Branch expansion/collapse

Menu **Display**  
Keyboard **Ctrl +**  
**Ctrl -**

Branches of idea tree can be expanded or collapsed, either by using menus *"Collapse/Expand"* of the *"Display"* menu, either, by clicking on the '+' and '-' symbols in idea tree, either using '+' and '-' keys.

##### 5.2.3.4.2. Toolbars display

Standard toolbars as well as left side toolbar can be shown/hidden with the *"Display/Toolbar"* menu.

Menu **Display**  
Sub menu **Toolbar**

### 5.2.3.4.3. Status area display

Status area (screen bottom) can be shown / hidden with the menu *"Display/Status bar"*.

Menu **Display**  
Sub menu **Status area**

### 5.2.3.5. Windows management

WIT is multi view (several views can be opened on the same document), and multi-document.

Menu **Windows**

Window handling functions (creation, tile, cascade, selection) can be accessed with standard menu *"Windows"*.

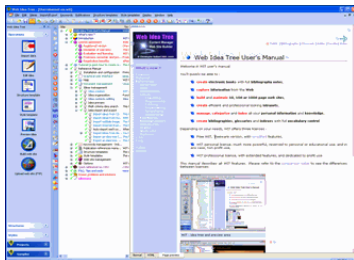
### 5.2.4. Tabbed view (current idea)

The tabbed view area is laid at screen right (its size can be increased / decreased while dragging the separation border between presentation and right view with the mouse ).

It shows the text contents of the current idea selected in the treeview, according to one of these three criteria :

- **Normal** : the idea text only is displayed. It can be directly modified (WYSIWYG editor).
- **HTML** : the idea text is displayed in pure HTML format, and can be modified for a precise control. (Note : this tab is enabled in expert mode only).
- **Page Preview**: the whole page is displayed, as it will appear once the Web site is built.

Selecting another idea in the left tree automatically saves any modification done in the current idea text, and updates the tabbed view with the contents of the new idea.

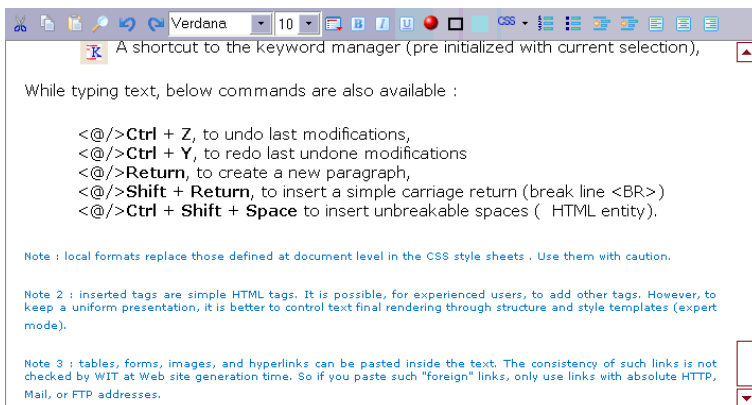


Vue de droite (idée courante)

### 5.2.4.1. Normal view

This view enables current idea text editing.

See the link below to use it.



Normal view (WYSIWYG editor)

See also :  
▶ [Editing idea text : WYSIWYG editor updated!](#)

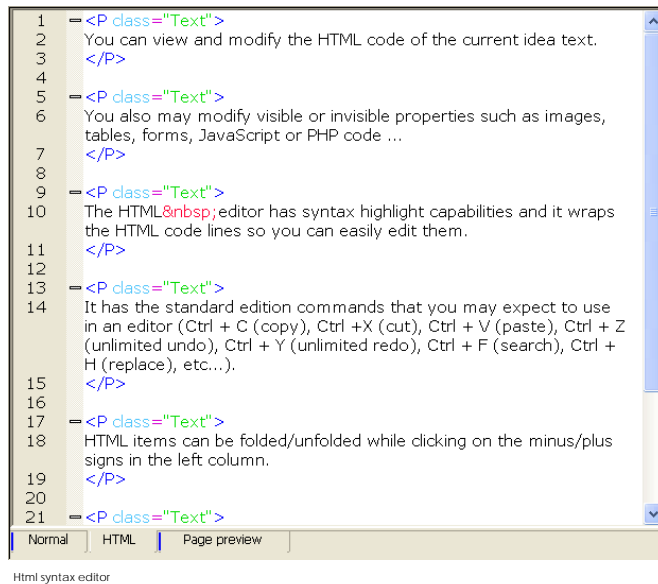
### 5.2.4.2. Pure HTML view

This view is enabled in expert mode only.

Feature **Not in freeware edition**

You can view and modify the HTML code of the current idea text.

Please follow the link below to use it :



Html syntax editor

See also :

▶ [Editing idea text : pure HTML code](#)

### 5.2.4.3. Page preview view

This area is used by the HTML integrated display to give a preview of the final rendering of selected idea.

Keyboard **Ctrl + Q**

It is activated / deactivated with Ctrl + Q key, or with the menu *Ideas/Preview (built-in browser)*.

The preview view is enhanced with **help tags**, giving access to useful WIT features :

This icon displays a useful contextual menu when flown by by the mouse pointer. The menu gives access to specific page objects.

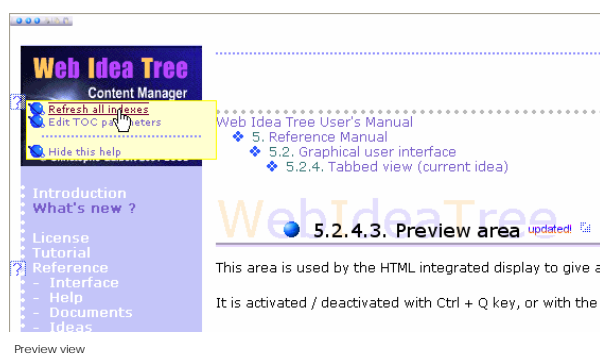
This icon, placed near an idea or page structure element, enables direct editing of its content.

This icon, placed near an idea or page structure element, enables direct editing of its properties.

This icon is changed into when the mouse pointer reaches it :

- Left and right arrows enable navigation in already seen pages (same history as web browsers).
- Up arrow displays the root idea.
- The printer icon prints the current idea page.
- 'Paper' icon copies the current preview view selection into the Windows clipboard.
- 'Clipboard' icon enables **intelligent paste** : the Windows clipboard content is directly pasted into the idea structure elements without any use of the idea property dialogs. The result is immediately displayed.

**Note** : the hypertext links, which are displayed in the this view are correct (refreshed) **only if the user preference "Automatic built of preview pages (preview view)" is enabled.**



Preview view

See also :

▶ [Internal preview](#)  
 ▶ [Intelligent information paste into ideas](#)  
 ▶ [Paste text and images updated!](#)

### 5.2.5. Status area

This area, located at screen bottom, displays state message of WIT ; if preview display area is activated, it displays also the messages of the integrated Web browser. If not activated, current idea path is displayed.

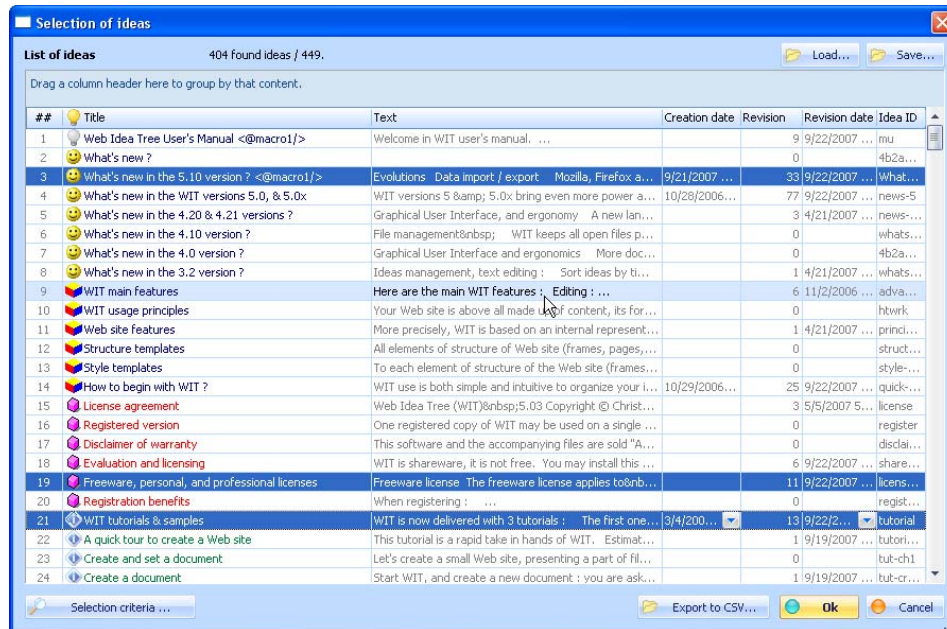
See also :

▶ [Status area display](#)

## 5.2.6. Miscellaneous dialogs

### 5.2.6.1. Idea selector : sorting, grouping, filtering and editing <sup>updated</sup>

This dialog is frequently used, when searching for ideas, and whenever one or several ideas have to be selected (for creating associations, for instance).



Idea selector

See also :

- ▶ Idea selector <sup>updated</sup>
- ▶ Export ideas into a CSV file <sup>updated</sup>
- ▶ Selecting displayed attributes, columns size and order
- ▶ Sorting and grouping
- ▶ Quick idea attributes editing

### 5.2.6.2. Information

This box displays an information or error message, without proposing any choice.

### 5.2.6.3. Confirmation / Question

This dialog asks a question to which a yes/no, Ok/Cancel answer should be given.

### 5.2.6.4. Enter a value

This dialog waits for an alphanumeric value (context dependent).

## 5.3. Help

WIT has a contextual help system, created and deployed with ... WIT.

Menu **Help**

### 5.3.1. Contextual help : F1

At anytime, a contextual help is available by pressing F1 key. Current Web browser is launched and displays help in HTML pages.

Menu **Help**  
Keyboard **F1**

### 5.3.2. User's manual

Access to complete WIT User's Manual is available through menu *"Help / User Manual"*. The whole manual is displayed as an independent local web site in the current web browser.

Menu **Help**  
Sub menu **User's Manual**

### 5.3.3. Help index

Menu [Help](#)  
Sub menu [Index](#)

Help index can be accessed through the menu "*Help / Help Index*" : the user's manual is opened at **Index** page, allowing a navigation based on ideas using a particular keyword.

### 5.3.4. Tutorial

Menu [Help](#)  
Sub menu [Tutoriel](#)

The tutorial is shown by the menu "*Help / Tutorial*" ; the User's manual is opened at **Tutoria** page, allowing a quick and complete mastering of WIT.

### 5.3.5. About ...

Menu [Help](#)  
Sub menu [About...](#)

Menu option "*Help / About...*" displays WIT information.

### 5.3.6. Software registration

Menu [Help](#)  
Sub menu [Registration...](#)

This function helps you to register this copy of WIT, if not already registered, or to display current registration information.

To register, type a name in the field "*Name*" (at least 5 characters), and then click on the link "*Get code registration*".

This link points at an Internet address of secured registration (by check, bank card or fax).

Registration code is displayed when payment has been accepted (please note all registration information : name, and code if you have to reinstall WIT later).

Type the registration code in the "Code" field and validate the dialog.

If the code is correct, a message shows operation success, and the dialog is closed after few seconds.

If the code is not correct, registration can't be performed (check correct spelling of the name and code), and the dialog is closed after few seconds. Please retry with correct data.

## 5.4. WIT Documents

Usual functions of document management are grouped into the "*File*" menu.

Menu [File](#)

A document groups and manages the following elements :

- organization and idea contents,
- structure template,
- style template,
- publication references database.

All this information is stored in a single file (\*.wit suffix), except resources (images, HTML or other documents files) to which the WIT document can refer (links). Information related to structure and style templates, and publication databases can also be saved separately, so it can be used in other WIT documents.

Resources to which a document can refer (links to images or HTML files) are :

- Direct resources (defined in idea properties) : they are automatically taken in account by WIT, and copied at web site creation time.
- Indirect resources (specified in structure or style templates, or inherited from external HTML pages using their own images) : they are taken in account provided they have (manually) been copied in a special subdirectory (named "**extra**") of the current directory of the WIT document.

See also :

- ▶ [Structure templates](#)
- ▶ [Style templates](#)
- ▶ [Publications](#)

### 5.4.1. Create a new document

Menu [File](#)  
Sub menu [New](#)

This function creates a new empty document, with default configuration options (options, structure and style), in the current language of WIT.

**Note 1 :** It's a good idea to immediately save this new document in a dedicated subfolder (preferently in the WIT projects folder).

**Note 2 :** Functions usually used after document creation are : "File/Options" -> "Custom fields", and "Ideas/Create child".

See also :

- ▶ [Document options](#)
- ▶ [Idea creation](#)

### 5.4.2. Open a document

Menu [File](#)  
Sub menu [Open](#)  
Keyboard [Ctrl + O](#)

This functions opens a dialog box allowing selection of WIT documents : files with "\*.wit" suffix.

The chosen document is loaded and displayed in the main view.

WIT is multi-document : you can switch from one opened document to another with the Ctrl + Tab keys combination, or by using "*Windows*" menu to arrange views.

Note : WIT sets a "lock file" when editing a document to prevent concurrent access (by other users on a network for instance). This lock is removed when you close the document. If you open an already locked file, save will be disabled.

See also :

- ▶ Windows management
- ▶ FAQ : My document is locked

#### 5.4.2.1. WIT version 5 - New document format

WIT version 5 introduces a new format for \*.wit documents (it's an extension of the previous format) and takes in charge the conversion.

When loading an older document, an error message may be displayed (incorrect format), then WIT proposes to convert the document into the new version 5 format.

See also :

- ▶ Versions 5.0, 5.01

#### 5.4.2.2. WIT version 5.20 - New document format updated

WIT version 5.20 introduces a new format for \*.wit documents (it's an extension of the previous format) and takes in charge the conversion.

When loading an older document, an error message may be displayed (incorrect format), then WIT proposes to convert the document into the new version 5.20 format.

See also :

- ▶ Version 5.20

#### 5.4.3. Close document

This function closes the current document.

Menu **File**  
Sub menu **Close**

If it was modified, save option is proposed.

#### 5.4.4. Save document

Menu **File**  
Sub menu **Save**  
Keyboard **Ctrl + S**

This function performs document save, with its current settings. If the document has never been saved (new document), a default directory is proposed. Otherwise, saving is done in the current directory, from where the document was loaded.

In order to ease document management, it is better to save them in a subdirectory of current WIT project directory.

See also :

- ▶ Work directory

#### 5.4.5. Save under another name

This function saves the current document by opening a dialog box to choose name and directory of file to save.

Menu **File**  
Sub menu **Save as...**

In order to ease document management, it is better to save them in a subdirectory of current WIT project directory.

See also :

- ▶ Work directory

#### 5.4.6. Print ideas updated

This function performs pretty idea printing, with the web browser help:

Menu **File**  
Sub menu **Print**  
Keyboard **Ctrl + P**  
Feature **Professional ed. only**

- current idea,
- current idea tree ([professional license](#)),
- or the complete idea tree ([professional license](#)),

as an electronic book with controlled page settings ([professional license](#)) :

- You may enable/disable the display of:
  - page header,
  - page footer,
  - page left pane,
  - page right pane.
- [This choice is predefined and can't be modified with the personal license.](#)

Once the selection and page settings parameters are set, WIT builds an HTML page that contains all involved ideas, starts the web browser and proposes to print the page.

You may ask the web browser for a print preview before printing.

As soon as the printing is done, you may close the web browser window.

#### 5.4.7. Quit

This functions closes all opened documents (asking for saving of modified ones), and closes work session with WIT.

Menu **File**  
Sub menu **Quit**



## 5.5. Ideas

Menu [Ideas](#)

### 5.5.1. Idea creation

WIT offers many ways to create ideas :

Menu [Ideas](#)

- manual ways, such as creating individual ideas,
- automatic ways, such as importing data from various sources and creating idea trees.

#### 5.5.1.1. Create a child idea updated

This function creates a child idea, from current selection.

Menu [Ideas](#)  
 Sub menu [Create a child](#)  
 Keyboard [Insert](#)  
 or [Ctrl + I](#)  
 Feature [Professional ed. only](#)

If the "Quick titles" option is enabled (default behavior), a simple dialog is displayed to let you define the title of the new idea.

Otherwise, the idea properties dialog box is shown after creation (figure 4.7), and the title field is filled with a default value calculated from idea id.

Professional ed. only version only :

If expert mode is enabled, and if "Choose title of generated pages" option is enabled, a dialog box asks for file name (figure 4.6) : this name will then be displayed in the "Id" column in the main view.

Otherwise, once you have defined the title, WIT sets the idea ID by reusing your title. This internal ID will be used to name the generated page files.

(If your title contains too specific characters, WIT replaces them by '.' or uses an automatic internal ID. WIT takes care of unicity of internal IDs so you don't have to worry about this).

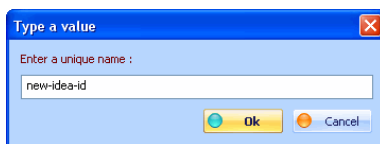


Fig 4.6 : Choice of a file name for currently created idea.

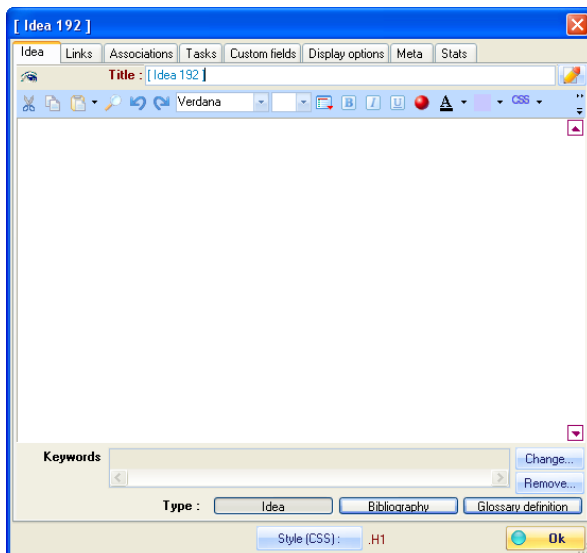


Fig 4.7 : Idea properties

#### 5.5.1.2. Create a sibling idea (before)

Menu [Ideas](#)  
 Sub menu [Insert before](#)  
 Keyboard [F4](#)  
 or [Ctrl + B](#)

This function behaves like "create a child idea", but the created idea is placed at the same level, and **before** the current idea (sibling idea).

#### 5.5.1.3. Create a sibling idea (after)

This function behaves like "create a child idea", but the created idea is placed at the same level, and **after** the current idea (sibling idea).

#### 5.5.1.4. Change idea type

This function changes current idea type, and optionnally, recurses to first level or all current idea children.

Menu **Ideas**  
 Sub menu **Change type**  
 Keyboard **Ctrl + T**  
 Feature **Not in freeware edition**

A dialog box (figure 4.9) displays options :

- idea (default type),
- bibliography note (specialized type with a more compact rendering),
- glossary (specialized type with a more compact rendering, and shown in the general glossary, with indexes).

Type change applies to current idea, to first level children or to all children.

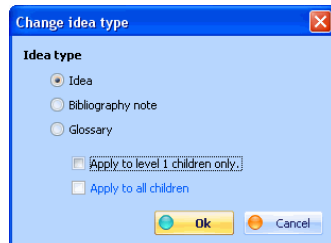


Fig 4.9 : Change idea type

#### 5.5.1.5. Change idea ID (and web page name)

Menu **Ideas**  
 Sub menu **Change idea ID (and web page name)**  
 Keyboard **Ctrl + W**  
 Feature **Professional ed. only**

This option (expert mode) modifies the current idea identifier within WIT *and therefore, the name of corresponding file at web generation time.*

You may so generate files with explicit and easy to remember names.

In the dialog box, type a unique identifier (WIT will check for its syntax and unicity).

WIT also checks the document consistency and updates all internal links.

Note : you may display the ID of ideas in the main left tree view (last column).

Note 2 : this function is available in the WIT professional release only.

Note 3 : some characters are not allowed in file names and in URLs. Here they are.

#### 5.5.1.6. Remove HTML format

This function removes all HTML tags from current idea text, excluding paragraphs.

Menu **Ideas**  
 Sub menu **Remove HTML format**  
 Feature **Professional ed. only**

It is useful to "clean" imported text (from Word or Excel, for instance), when you do not wish to keep its format.

Note : this function is available in the WIT professional release only.

#### 5.5.1.7. Build thumbnails automatically

This function is useful to create photo albums. It applies to current selection (current tree).

Menu **Ideas**  
 Sub menu **Build thumbnails**

All sub tree images are gathered, and a thumbnail is generated for each. All thumbnails are then inserted in the current idea text.

Each thumbnail is a link to the original image.

Notes :

• Using this function also activates the "Generate image table" option in the structure template (check it in expert mode), in the "Table of contents" tab.

• Thumbnail size is also set in the structure template (same tab) as above.

• The thumbnail CSS styles (see in expert mode , the style template) are ".TocImagesGroup" for the whole set of thumbnails and ".TocImage" for each thumbnail.

• The thumbnail list and links are inserted in the current idea text : activate the HTML edit view to change it, and modify the raw HTML code. If you want to refresh the list, delete the HTML code, and automatically rebuild the list of thumbnails.

See also :  
 ► [Import multiple images from folder](#)

### 5.5.2. Idea organization

Functions below handle current idea and its children.

Menu **Ideas**

### 5.5.2.1. Copy an idea

This function operates on current selected idea and all its children.

Copy is multi-document.

Menu **Edit**  
Sub menu **Copy**  
Keyboard **Ctrl + C**

### 5.5.2.2. Cut an idea

This function operates on current selected idea and all its children, and selection is then removed from current document.

Cut operation is multi-document.

Menu **Edit**  
Sub menu **Cut**  
Keyboard **Ctrl + X**

### 5.5.2.3. Paste an idea

This function operates on current selection, which behaves as an insertion point (parent) for pasted ideas.

Previously copied or cut ideas, originating from the current document or from another document, are inserted.

Menu **Edit**  
Sub menu **Paste**  
Keyboard **Ctrl + V**

### 5.5.2.4. Remove an idea

This function operates on current selection : current idea and all its children. Confirmation is required before deleting.

**Note : removing an idea cannot be undone ; it may be more secure to use Cut and Paste functions.**

Another possibility is offered to recover from an imprudent deletion : click on an idea, and press Ctrl+R ("*Repaste destroyed ideas*"), even if cut/copy/paste functions were used meanwhile.

Menu **Edit**  
Sub menu **Remove**  
Keyboard **Del**  
**Ctrl + Del**

### 5.5.2.5. Move an idea before its sibling

This function operates from current selection, and moves upwards the selected idea among its siblings.

If there is no sibling, or if the idea is the first, move has no effect.

Menu **Ideas**  
Sub menu **Up**  
Keyboard **Ctrl + Up arrow**  
**Ctrl + Alt + Up arrow**

### 5.5.2.6. Move an idea after its sibling

This function operates from current selection, and moves downwards the selected idea among its siblings.

If there is no sibling, or if the idea is the last, move has no effect.

Menu **Ideas**  
Sub menu **Down**  
Keyboard **Ctrl + Down arrow**  
**Ctrl + Alt + Down arrow**

### 5.5.2.7. Move an idea right (reparent)

This function changes idea hierarchy.

Current idea (and its children) is moved so as it becomes a child of its previous sibling.

If the previous sibling has children ideas, the currently moved idea is placed in the end of children list.

There will be no move when there is no previous sibling.

Menu **Ideas**  
Sub menu **Move right**  
Keyboard **Ctrl + Right arrow**  
**Ctrl + Alt + Right arrow**

### 5.5.2.8. Move an idea left (reparent)

This function changes idea hierarchy.

Current idea (and its children) is moved so as it becomes a new sibling idea of its current parent. It is moved just after its current parent.

If parent idea is the root idea, no move will be done.

Menu **Ideas**  
Sub menu **Move right**  
Keyboard **Ctrl + Left arrow**  
**Ctrl + Alt + Left arrow**

### 5.5.2.9. Sort ideas by titles

This function sorts children of selected idea by their titles.

Menu **Ideas**  
Sub menu **Sort children by title**  
Feature **Not in freeware edition**

### 5.5.2.10. Invert child ideas order

This function inverts order of children of selected idea.

Menu **Ideas**  
Sub menu **Invert children order**  
Keyboard **Ctrl + Shift + I**  
Feature **Not in freeware edition**

### 5.5.2.11. Move an idea by drag and drop

Keyboard **Maintained left click + move**

Ideas can be moved by drag and drop. To move an idea (and its children), click on its icon, and drag selection into target idea : currently moving idea becomes a child idea or target idea.

**Note :** moving an idea is not directly available between different documents. A copy is done instead. Use Cut/Paste to emulate idea moving (Ctrl+X/Ctrl+V).

You can also drop the idea in the "Associations" tab of an opened property dialog of another idea : in this case, WIT creates an [association](#) of the target idea to the source idea (the one which was just dropped).

### 5.5.2.12. Copy an idea by drag and drop

Keyboard **Ctrl + maintained left click + move**

This operation is similar to move by drag and drop, but a copy is performed.

To do such a copy, press the Ctrl key while dragging.

### 5.5.2.13. Check an idea for table of contents updated

This function hides / shows ideas in several WIT objects when the web site is built :

Menu **(Main tree view)**  
Keyboard **Space bar**  
Click on green check mark in main view

- [Table of Contents](#)
- [Table of Images](#)
- [Java Applet tree menu](#) in the left frame (when selected in the structure template)
- [Drop Down Menu](#) widget
- [Simple Tree Menu](#) widget

This allows a more precise control on these objects, in addition to their specific options.

Ideas are listed in a tree like manner in main view. Before each idea icon, a check mark indicates whether corresponding idea (and its children) should be shown or not in these objects (Figure 4.10).

All ideas are checked by default.

**Note :**

- The indexes (normal and faceted) are unchanged, whatever the state of ideas.

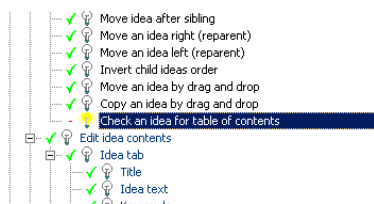


Fig 4.10 : Checking / unchecking Ideas for table of contents

See also :

- ▶ [Table of contents tab](#)
- ▶ [Drop Down Menu widget](#) updated

## 5.5.3. Idea contents editing

Idea contents definition is made with a tabbed dialog box (Figure 4.11) where you can modify all idea properties.

Menu **Ideas**  
Sub menu **Properties**  
Keyboard **Alt + Enter**

- **Idea** tab lists textual attributes of idea (title, description, keywords...),
- **Links** tab helps to establish references to images, local documents, or external Urls,
- **Associations** tab helps to link current idea to other ideas or publication references described in the current document,
- **Custom fields** tab adds attributes which meaning is given when configuring document options,
- **Tasks** tab looks like the previous one, with predefined and task management oriented fields (dates, progress...).
- **Display options** tab defines local display criteria, for a better presentation when manipulating ideas.
- **Meta** tab allows idea local meta data editing,
- and **Stats** tab (statistics) displays idea state information.

If you prefer full screen text editing, you may also use the "Normal" tab of the main view for WYSIWYG idea text editing.

**Note :** when editing the different idea properties, a context button ("Style (CSS):") is displayed to give access to the corresponding CSS item of the style template.

**Note 2 :** when the Expert is disabled, only the **Idea**, **Links** and **Associations** tabs are displayed.

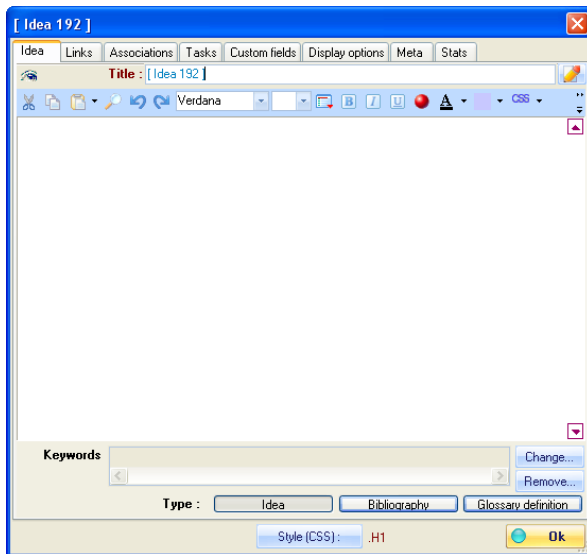


Fig 4.11 : Properties of selected Idea.

See also :

- ▶ Editing idea text : WYSIWYG editor [updated](#)
- ▶ Versions 5.0, 5.01

### 5.5.3.1. Idea tab

This tab contains idea textual information (figure 4.12).

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Idea](#)

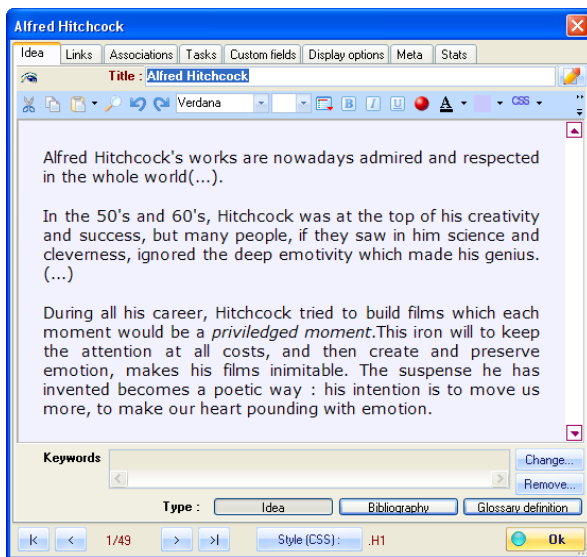


Fig 4.12 : Idea tab.

See also :

- ▶ Editing idea text : WYSIWYG editor [updated](#)

### 5.5.3.1.1. Title

This field contains the Idea title.

This title is used in the Title column in the main view, and in tables of contents.

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Idea](#)

See also :

- ▶ Structure : Title
- ▶ Style class : H1 to H6 (titles from level 1 to 6)
- ▶ Style class : PageHeader

### 5.5.3.1.2. Text

The main area of idea tab is a WYSIWYG HTML text editor (What You See Is What You Get : the text is currently rendered in the same way it will be generated).

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Idea](#)

The same editor is also available from the "Normal" tab of main view.

Please follow the link : [Editing idea text : WYSIWYG editor](#)

See also :

- ▶ Pure HTML view
- ▶ Style class : TextGroup (text paragraphs group)
- ▶ Structure : Text

### 5.5.3.1.3. Keywords

WIT uses keywords to automatically builds an index when generating the Web site. The keywords are associated to ideas and controlled by a keyword manager.

To add keywords, click on the "Change..." button : the keyword manager is displayed. It shows the list of all already defined keywords, and used by ideas.

The "Remove" button clears all keywords that were linked to the current idea (however, they are not removed from global keyword list).

See also :

- ▶ Keywords

### 5.5.3.1.4. Type : idea, bibliography or glossary

An idea can be specialized into bibliographic note or into glossary definition.

At web site creation time, a bibliographic note used by an idea will be transformed into a footer page note rather than into a link.

A glossary definition will also be included in the general glossary (see indexes) at web site creation time.

See also :

- ▶ Style class : NoteGroup (Bibliography note group)
- ▶ Style class : GlossaryGroup (glossary group)

### 5.5.3.2. Links tab

This tab helps to link images, local documents or external references (Uris) to the current idea (figure 4.14).

Images are integrated to the page, documents are separate local files, which will be included in the Web site, and pointed out to by relative (and calculated) links.

External references are links to other web sites (URLs).

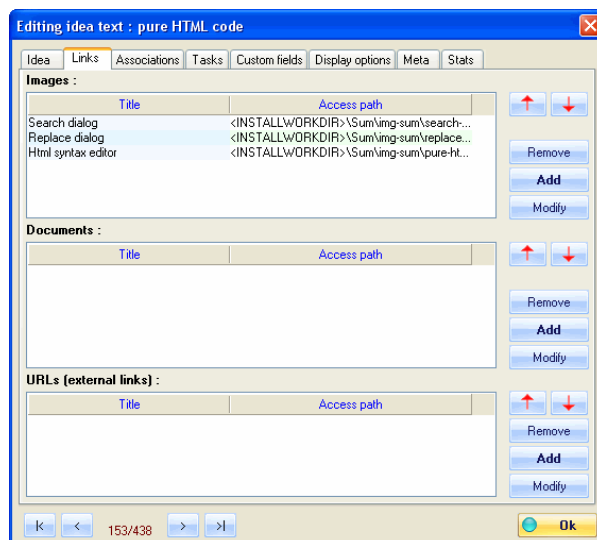


Fig. 4.14 : Images, documents and external references

### 5.5.3.2.1. Images (controlled by WIT)

When idea page is rendered, images are displayed in the same order as they appear in the list, according to the current structure and style templates options. Their files are copied into the target directory wit-icons subfolder.

Each reference to an image is made of a comment and an access path to the image file (png, gif or jpeg).

To add an image, click on the "add" button, and, in the dialog box, type a comment and choose an image file.

Menu Ideas  
Sub menu Properties  
Keyboard Ctrl + Shift + K  
Tab Idea

Menu Ideas  
Sub menu Properties  
Tab Idea  
Feature Not in freeware edition

Menu Ideas  
Sub menu Properties  
Tab Links

Menu Ideas  
Sub menu Properties  
Tab Links

To remove an image reference, select the unwanted reference in the list, and click on the "remove" button. The file is not removed, just the link.

You can select multiple images : in this case, the image file name is appended to the current title.

To modify the image order in the list, select an image reference to move, and use up and down arrows.

You can also add images directly in the idea text, in the WYSIWYG text editor.

Note : you can also drag and drop (single or multiple) images files directly in the property dialog from the file explorer. In this case, WIT uses their file names to create their titles (captions).

Reciprocally, you may drag and drop images from the idea property dialog into the main idea tree : the image reference is then copied in the target idea.

See also :

- ▶ Structure : Images
- ▶ Style class : img (image)
- ▶ Style class : ImageGroup (images group)
- ▶ Insert an image (alternate method) **updated!**
- ▶ FAQ : Side by side images
- ▶ FAQ : Alternate text and images
- ▶ FAQ : Insert image in the middle of text
- ▶ FAQ : Center images **updated!**
- ▶ FAQ : Relatives URLs

### 5.5.3.2.2. Local documents

Menu Ideas  
Sub menu Properties  
Tab Links

A local document is any file that can be linked from your current WIT document, in one or several ideas. The link itself is called a reference.

Each reference to a local document is made of a comment (title) and an access path to its file.

These documents are called local because they are located on your computer (or on the accessible drives of the local network), but they remain stored outside of your current WIT document.

When idea page is rendered, the references are displayed as hypertext links in the same order as shown in the list, and according to the current structure and style templates options. The local documents are copied into the docs subfolder of the web site target directory.

To add a reference, click on the "add" button, and, in the dialog box, type a comment and choose a document file.

You can select multiple documents : in this case, the document file name is appended to the current title.

You can also add documents directly in the idea text, in the WYSIWYG text editor, with drag & drop method.

Tip : You can also drag and drop (single or multiple) document files directly in the property dialog from the file explorer. In this case, WIT uses their file names to create their titles (captions).

You can also drag & drop document references from the idea property dialog into the main idea tree : the reference is then copied into the target idea.

To remove a reference, select the unwanted reference in the list, and click on the "remove" button. The file is not removed, just the link.

To modify the reference order in the list, select the reference to move, and use the up and down arrow buttons.

Note 1 : you can also link HTML files as local documents. These documents may contain images, which should be also be included in the web site (ie be available for WIT). In this case, the best practice is to :

1. create a sub directory with the reserved name extra in the same directory of the current WIT document,
2. place there the images,
3. and change their addresses in the HTML local document with the ../extra relative address.

Note 2 : If you drag and drop a multimedia document, WIT will recognize it and will embed the file in the text of current idea. Otherwise, it will be linked and you will have to click on the link to play the file.

See also :

- ▶ Structure : Links to documents
- ▶ Style class : DocumentsGroup (references to documents group)
- ▶ Automatic Web site generation
- ▶ Insert a link to local documents

### 5.5.3.2.3. URLs

Menu Ideas  
Sub menu Properties  
Tab Links

When the idea page is rendered, URLs are displayed in the same order as they appear in the list, and according to the current structure and style templates options.

Each reference to an URL is made of a comment and a value (the real URL), divided in a protocol (http, ftp, ...) and the real address.



To add an URL, click on the "add" button, and, in the dialog box, type a comment and type the URL value.

You can also add URLs directly in the idea text, in the [WYSIWYG text editor](#), by dragging/dropping Windows internet shortcuts files (\*.url files).

And you can also drag and drop references to URLs from the idea property dialog into the main idea tree. The reference is then copied into the target idea.

To remove an URL, select the unwanted reference in the list, and click on the "remove" button. To modify the URL order in the list, select a reference to move, and use the up and down arrows.

See also :

- ▶ [Structure : External links \(URLs\)](#)
- ▶ [Style class : UrlsGroup \(URLs group\)](#)

### 5.5.3.3. Associations tab

This tab helps to associate other ideas, and bibliographic references (publications) to the current idea (figure 4.16).

These associations are shown a list of links at page rendering time : links to other ideas, links to publications.

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Associations](#)

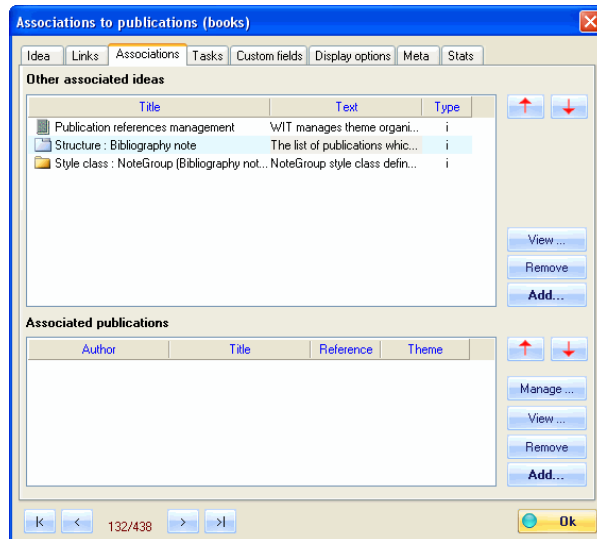


Fig 4.16 : Associations to ideas and books.

#### 5.5.3.3.1. Associations to other ideas

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Associations](#)

To add a new idea association, click on the "Add" button, which opens the idea selection dialog box (figure 4.17). Select one or several ideas (by using Shift and Ctrl click combinations), and validate. Selected ideas are added to the current association list.

Ideas already associated are displayed in the order they were added. To remove an idea, select this idea and click on the "Remove" button. To move an idea, select it and use "Up" and "Down" key arrows.

If the "Create a reciprocal link" option is checked, reciprocal associations to the current idea will also be created in each selected idea.

*Note :* for a more precise idea selection, you may use a selection criterion (see the idea search function).

*Note 2 :* you can also drag & drop ideas in this dialog:

- from the main idea tree (main view),
- from other idea associations, displayed in other idea property dialogs that are already opened.

*Note 3 :* reciprocally, you can drag and drop associations from an opened idea property dialog into the main idea tree : associations are then copied in the target idea.



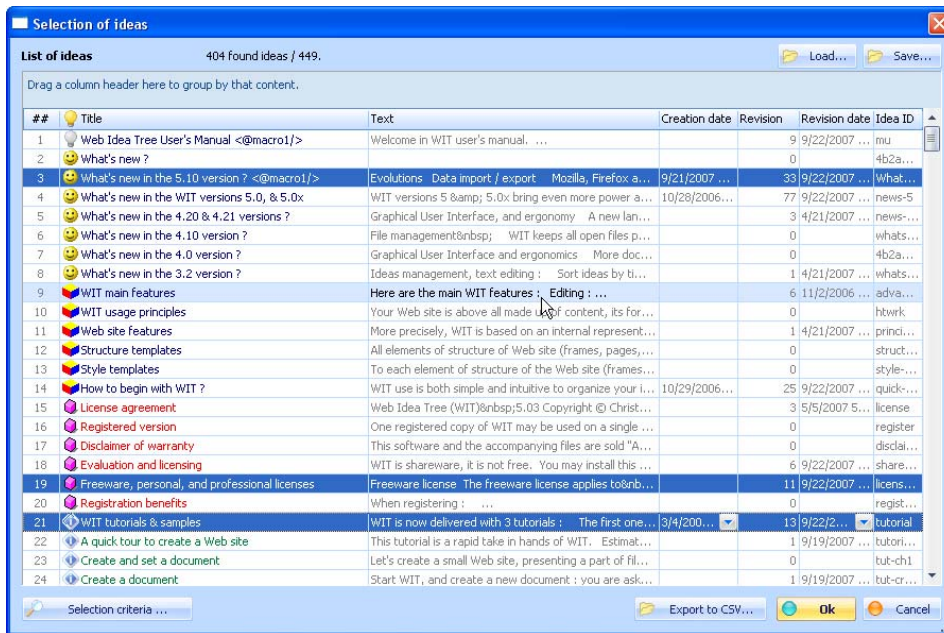


Fig 4.17 : Idea selection

See also :

▶ Multi criteria idea search

### 5.5.3.3.2. Associations to publications (books) updated

Menu Ideas  
Sub menu Properties  
Tab Associations

To add a publication association, click on the "Add" button, which opens the publication selection dialog box, also known as publication manager (figure 4.18). Select a publication and validate. The publication is added to the current publication association list. Already associated publications are displayed in the order they were added.

To remove a publication, select it, and click on the "Remove" button.

To move a publication, select it, and use "Up" and "Down" key arrows.

**Note :** for a more precise publication selection, you may type a theme (see the publication management function).

**Note 2 :** each publication is identified by a unique string, its ID. The link to the publication may be broken if, later, the publication is removed from the bibliography, or if its ID is changed. In this case, the link to the publication would appear in red characters, and would not be generated in the web page (at build time).

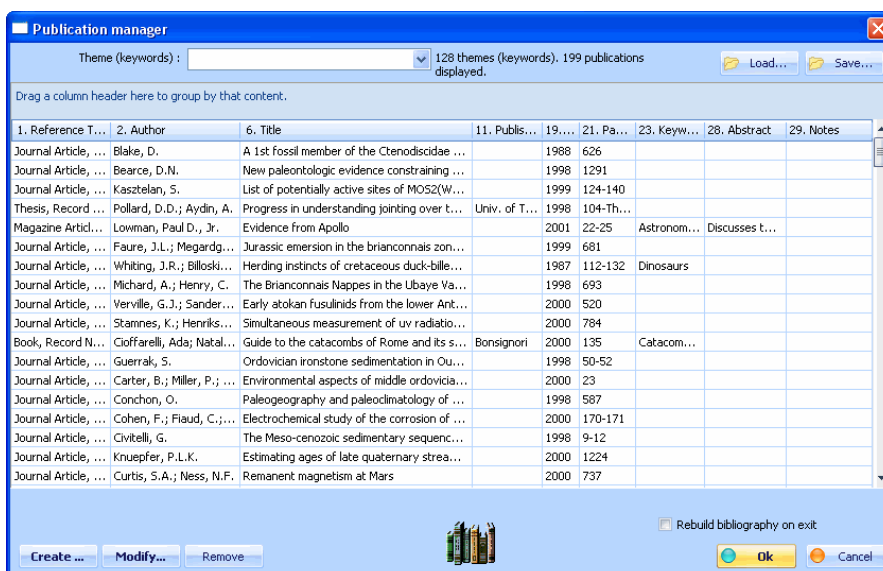


Fig 4.18 : Book selection

See also :

▶ Publications

▶ Structure : Bibliography note

▶ Style class : NoteGroup (Bibliography note group)

### 5.5.3.4. Tasks tab updated

This tab gives access to task oriented information for the current idea.

The names and semantics of such task fields are predefined :

Menu **Ideas**  
 Sub menu **Properties**  
 Tab **Tasks**  
 Feature **Professional ed. only**

- Start date
- Due date
- Completion percentage (from 0 to 100 %)
- Estimated time
- Spent time
- Effective end date
- Status
- Category
- Created by
- Risk / priority
- Assigned to

The 5 last fields are texts, with free content.

To enter a value, and according to the field kind,

- select a date in the calendar (in the menu attached to the field),
- move the progress slider,
- or type some free text in the editable fields or select an already used value in the menu attached to the field.

#### Utility of task fields

The task fields enrich your ideas and add a time dimension which can be used as search / filtering criterion in the "Tasks" menu. You can so build a TODO list, with planned tasks, and control their progress.

The task fields may be used to perform a faceted classification : each field may represent a criterion (a facet) which values take part in a specific index (the faceted index). This is the case for this document in which the "Category" field is used to enlight all additions in WIT version 5.

Each task field can give value to **WIT variables**, that can be reused anywhere in the idea (or anywhere in the code of the web page if it contains scripts).

#### Distinctive behavior

- An idea becomes a task only when a due date or a non null completion percentage is specified.
- Start date is automatically initialized to the current date (if it was not already initialized), as soon as the completion percentage is greater than 0.
- Tasks for which completion is 100 % are considered finished.
- When setting a task completion to 100%, the effective end date is set to the current date.
- Dates (start, due date, end) are shown in days, months and years (no time), according to your Microsoft Windows locale.
- When the dates are used in the faceted index, they are shown as text strings "Year-Month-Day", so they can be properly sorted.

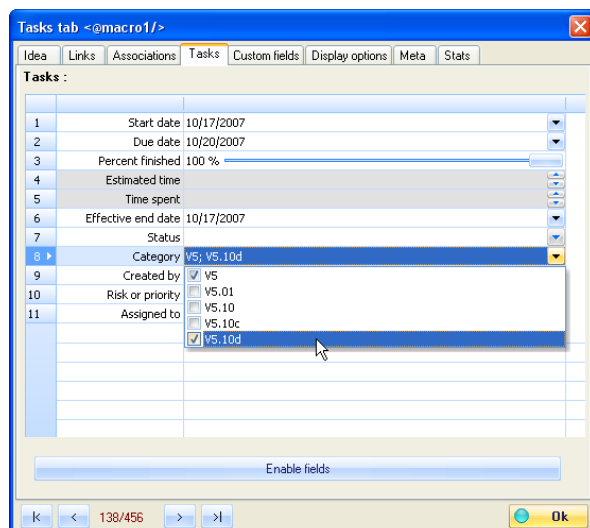
#### Notes (professional licence):

- Text task fields (status, category, created by, risk/priority, assigned to) can accept multiple values (separated by ;) when configured to do so, like here with the "Category" field : see the image and page bottom.
- These values are also links to separate entries in the faceted index if faceted classification was also enabled for the textual task field.
- Synthetic widgets** can be displayed to show summaries of field values and usage by ideas as well as links to faceted index (like in this page left pane).
- Synthetic task progress status widget** can also be displayed to show summaries progress criteria that are met by ideas as links to faceted index.

See the [document options](#) to enable/disable the task fields and make them eligible for faceted classification.

See the [Structure template](#) ('Navigation' tab) to enable/disable the generation of the whole faceted index.

See the [Structure template](#) ('Idea' tab) to control how the task fields are generated in the web pages.



Tasks tab

See also :

- ▶ Task fields options [updated!](#)
- ▶ Task fields Table of Contents widgets [updated!](#)
- ▶ Task progress status widget [updated!](#)
- ▶ Version 5.10 [updated!](#)
- ▶ Personal, professional, and freeware licenses [updated!](#)

### 5.5.3.5. Custom fields tab [updated!](#)

This tab (figure 4.15) helps to give additional information to the current idea, with 32 custom fields.

Menu [Ideas](#)  
 Sub menu [Properties](#)  
 Tab [Custom fields](#)  
 Feature [Professional ed. only](#)

Naming and activating these custom fields for the whole document is available in the document options, or while clicking on the *"Enabling and naming custom fields"* button.

To enter a value, type some text in the edit fields, or select previously defined texts by clicking on the combo box of each field.

You can also use advanced editing features by clicking on the "..." buttons, and insert images and links. Note : images should be small because they can appear in the faceted index !

#### Utility of custom fields

Custom fields can be used to perform faceted classification : each field is a classification criteria (a facet) which values are used as entries in a specialized index. This is the case for this document, where the custom fields 'Menu', 'Tab', 'Keyboard shortcut' and 'Feature' contribute to the faceted index.

Each custom field can give value to [WIT variables](#), that can be reused anywhere in the idea (or anywhere in the code of the web page if it contains scripts).

See [Document options](#) to control fields names and make them eligible for faceted index.

See the [Structure template](#) ('Navigation' tab) to enable/disable the faceted index build.

See the [Structure template](#) ('Idea' tab) to control how the custom fields are generated in the web pages.

#### Note (professional edition) :

- When the "Multiple values" option has been checked for a given field, multiple values (separated by a : delimiter) are accepted.
- If the "Facets" option is also checked, each value is also be linked to a separate entry in the faceted index.
- [Synthetic widgets](#) can be displayed to show summaries of field values and usage by ideas as well as links to faceted index (like in this page left pane).

Sample :

- Look at the ["WYSIWYG editor"](#) page in this manual where multiple values are given for keyboard shortcuts.

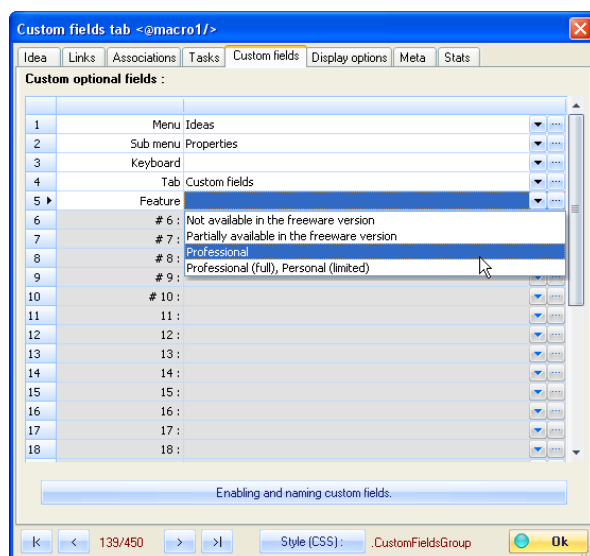


Fig 4.15 : Custom fields

See also :

- ▶ Custom fields options [updated!](#)
- ▶ Navigation tab
- ▶ Version 5.10 [updated!](#)

### 5.5.3.6. Display options tab

This tab (figure 4.19) controls how the idea is shown :

Menu [Ideas](#)  
 Sub menu [Properties](#)  
 Tab [Display options](#)

- In the views where ideas are displayed,
- In the generated Web site, when you wish to *locally* derogate to the rules of structure or style templates.

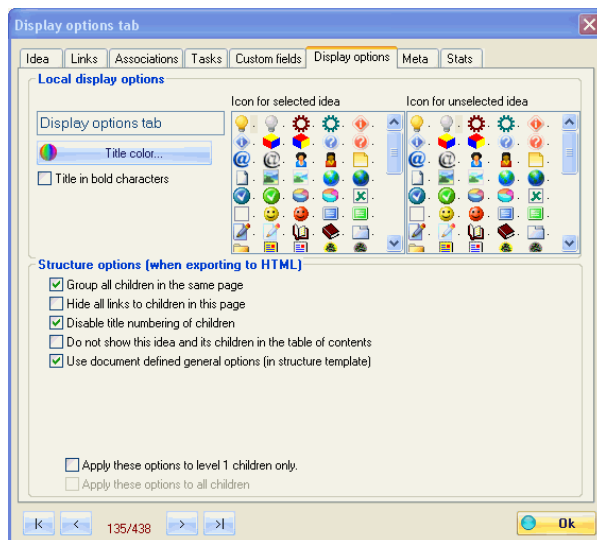


Fig 4.19 : Display options

## ● View display options

Below options affect only the idea display in the views.

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Display options](#)

## ● Title display in main view

Title display in main view can be modified by changing :

- the color,
- the characters (bold or not).

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Display options](#)

## ● Choice of icon to display in main view

Two icons are used when the idea is shown in the main view :

- the first, when the idea is selected,
- the second, when not selected.

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Display options](#)

**Note :** this icon is also used at Web site generation time in the Java applet if the left side frame is used.

## ● Display options in Web site

These options affect the page rendering at Web site build time.

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Display options](#)

They are used to control how child ideas of current idea will be generated :

- All child ideas grouped in a single page (like here),
- Hide all links to child ideas in the page,
- Apply or not of title numbering of child ideas (disabled in this example),
- Decide whether or not child ideas should be displayed in the table of contents.

They are also used to locally specialize background and foreground colors of an idea page (and optionally its children).

Corresponding rules in current structure template are ignored :

- background color,
- background image (tile),
- foreground color.

## ● Display options recursive application

When the property dialog is validated, chosen display options may apply to :

Menu [Ideas](#)  
Sub menu [Properties](#)  
Tab [Display options](#)

- selected idea only (by default)
- selected idea and immediate children (level 1),
- selected idea and all its children (all levels).

## ● 5.5.3.7. Meta tab

This tab adds meta information (description) and optional scripts to the current idea.

Menu **Ideas**  
Sub menu **Properties**  
Tab **Custom fields**

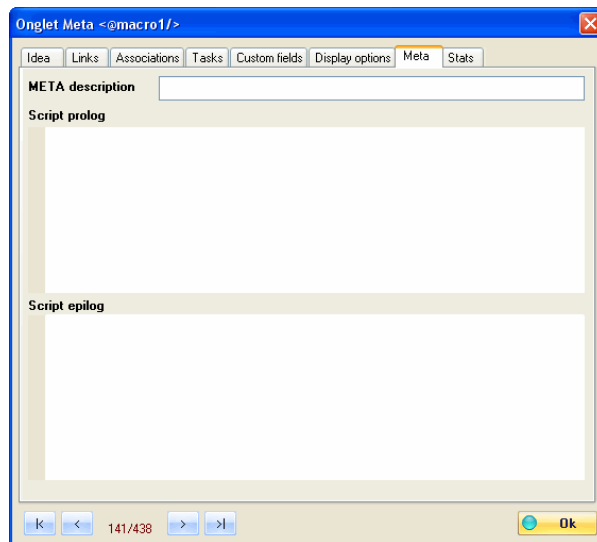
The contents of the *description* field is reused in the code `<meta name="description" content="Description text">` that is generated in the HTML header of pages (meta information placed between the `<head>` and `</head>` HTML tags).

The "Script prolog" field lets you place at the very beginning of the web page any script or script fragment (Javascript, PHP, etc...) that may be useful later in the page : for instance, variable declaration and initialisation, functions that are called later in the page by other scripts.

The "Script epilog" field lets you place at the very end of the web page any script or script fragment (Javascript, PHP, etc...).

Notes :

- All this tab fields are specific to the current idea only. If you wish to insert some scripts that should be shared by all the pages of your web site, use the structure template page header and footer instead.
- The scripts that are inserted in the prolog and epilog fields should not display any text because they are generated outside of any idea structure elements : respectively before the page header and after the page footer.



Meta tab

See also :

- ▶ Versions 5.0, 5.01
- ▶ Personal, professional, and freeware licenses **updated!**
- ▶ Intelligent information paste into ideas

### 5.5.3.8. Stats tab

Menu **Ideas**  
Sub menu **Properties**  
Tab **Stats**

This tab contains read only information that is automatically managed by WIT, regarding the current idea (excepted the idea identifier that can be modified (professional license)).

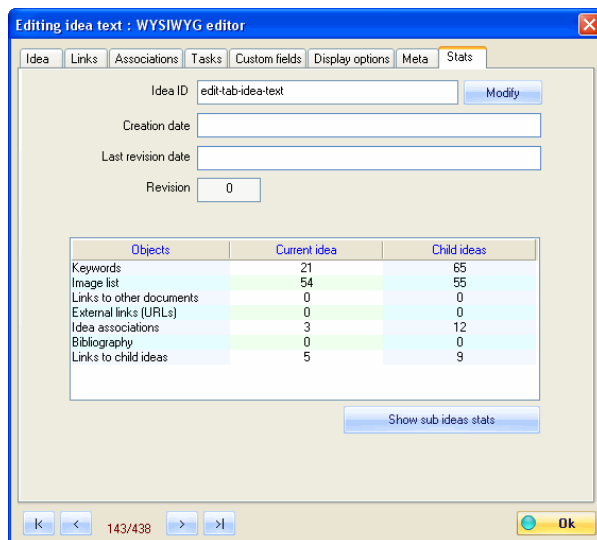
- Idea identifier (internally used by WIT as access key, and also to name the generated web pages)
- Creation date and time (automatic)
- Last modification date and time (automatic)
- Revision (automatically incremented whenever the idea is modified)

These values can also be reused as [WIT variables](#). Outside WIT variable, they do not participate to web pages generation, and are not represented by structural or style elements in the WIT structure and style templates.

Current idea elements counts are also displayed :

- Keywords
- Images
- Documents
- URLs
- Associations
- Bibliography references
- Child ideas.

The "Show sub idea stats" button counts sub ideas elements, with no limitation of depth.



Onglet Stats

See also :

- ▶ Versions 5.0, 5.01
- ▶ Insert a special WIT variable [updated!](#)
- ▶ Personal, professional, and freeware licenses [updated!](#)
- ▶ Change idea ID (and web page name)

### 5.5.3.9. Editing idea text : WYSIWYG editor [updated!](#)

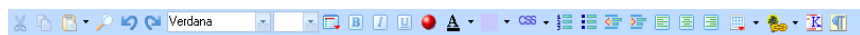
Menu	Ideas
	"Normal" view
Sub menu	Properties
	Ctrl + [1-6LNOP]
	Ctrl + Alt + [A-Z]
Keyboard	Shift + Tab
	Tab
	WYSIWYG editor
Tab	Idea
	"Normal" view (WYSIWYG editor)
Feature	Professional ed. only

Idea text can be modified with the WYSIWYG HTML text editor (What You See Is What You Get : the text is currently rendered in the same way that it will be generated).

Text of any length can be typed without formatting, or copied, cut or pasted (Ctrl C, X, V keys) from any other software (a word processor for example).

Global text format (justification, alignment ...), is defined by the style template parameters (class TextGroup).

However, to perform **local** formatting, WIT offers the following functions (in the toolbar) :



Cut (Ctrl + X)

Copy (Ctrl + C)

Paste (Ctrl + V)

RTF Paste : RTF format is preferred if present in the Windows Clipboard (for instance, when pasting Excel sheet fragments).

Search (Ctrl + F)

Undo (Ctrl + Z)

Redo (Ctrl + Y)


Verdana Font (Arial, Courier, Times, Verdana...)

Font size

Predefined formats


- Normal (Ctrl + N, Ctrl + Shift + N)
- Formatted <PRE> (Ctrl + P, Ctrl + Shift + P)
- Address
- Header 1 <H1> (Ctrl + 1, Ctrl + Shift + 1)
- Header 2 <H2> (Ctrl + 2, Ctrl + Shift + 2)
- Header 3 <H3> (Ctrl + 3, Ctrl + Shift + 3)
- Header 4 <H4> (Ctrl + 4, Ctrl + Shift + 4)
- Header 5 <H5> (Ctrl + 5, Ctrl + Shift + 5)
- Header 6 <H6> (Ctrl + 6, Ctrl + Shift + 6)
- Numbered list <OL> (Ctrl + O, Ctrl + Shift + O)
- Bulleted list <UL> (Ctrl + L, Ctrl + Shift + L)
- Directory list <DIR>

- Menu list <MENU>
- Definition term <DT>
- Definition detail <DD>
- Paragraph

 Bold characters (Ctrl + B)


 Italic characters (Ctrl + I)


 Underlined text (Ctrl + U)

 Graphical bullet : click on the sphere button. A tag is inserted and will be replaced by an *image* when the page is rendered, according to structure template parameters. (Ctrl + Alt + Y)

Text color and formatting : select the text, and choose your option in the menu :


 ▾ Text color (Ctrl + Alt + F)


 ▾ Text background color (Ctrl + Alt + B)

 ▾ Insert a CSS style (with a <span> tag) : the selected text is decorated with the CSS style that you select from the style template dialog. You can create your own CSS styles (prefix their names with a dot (".") in the CSS editor). If the selection includes different text formatings, WIT will apply the CSS style to the parent HTML tag. (Ctrl + Alt + C) (professional licence)


Repeat last CSS style insertion with (Ctrl + Alt + R) (professional licence)

 Clean common text formatings in the current selection (<font>, <strong> or <b>, <em> or <i> tags) (Ctrl + Alt + K)


 Clean advanced text formatings in the current selection (<font> and <span> tags and CSS styles : 'style' and 'class' attributes) (professional licence)

 Clean all links in the current selection (professional licence)


 Clean all MS-Word 2000, 2003 specific tags *in the whole page* (professional licence)

 Clean common and advanced text formatings and MS-Word specific tags *in the whole page* (professional licence)


 Clean all HTML tags *in the whole page* (Paragraphs and line breaks are kept) (professional licence)

 Numbered list (Ctrl + Shift + O)


 Bulleted list (Ctrl + Shift + L)

 Current paragraph indent increase (Tab, Ctrl + Alt + -->)


 Current paragraph indent decrease (Shift + Tab, Ctrl + Alt + <--)


 Left alignment,

 Centered alignment,


 Right alignment,


 ▾ Simple table management (insert table, insert/delete rows, columns and cells) :

 Insert a new table : specify its number of rows and columns, its caption title (Ctrl + Alt + T)


 Insert a new row in the current table

 Insert a new column in the current table


 Insert a new cell in the current table


 Delete current row


 Delete current column

 Delete current cell.

 ▾ WIT controlled internal links insertion :

 Insert a link to an idea (in this document). (Ctrl + Alt + A). Already selected text is kept as WIT creates the link. If no text is selected, WIT inserts the linked idea title. You can insert multiple links to ideas : in this case, WIT enumerates all idea titles. [More info...](#)

 Insert a link to a local document (Ctrl + Alt + D). Already selected text is kept as WIT creates the link. If no text is selected, WIT inserts the linked document name. You can insert multiple links to documents: in this case, WIT enumerates all document names. [More info...](#)


 Insert a link to an external URL (Ctrl + Alt + U): just type the name you want to appear in the text and the URL value. Both can be identical. [More info...](#)

 Insert an image (Ctrl + Alt + I). [More info...](#)

 Insert a WIT variable (Ctrl + Alt + V). [More info...](#)


Multimedia documents : [more info...](#)

 Flash® documents


 Shockwave® documents


 Quick Time® documents

 Media Player® documents

 Midi documents

 Real Audio® documents


 ActiveX® documents

 Plugin documents

 Java Applet documents






 Insert a folding text (**Ctrl + Alt + W**): [more info...](#)

 Display the keyword manager (pre initialized with current selection) (**Ctrl + Shift + K**)

 Toggle glyphs display : paragraphs, break lines, divisions, spans, comments...

↳ Access to the toolbar hidden elements when there is not enough room to display all the icons.

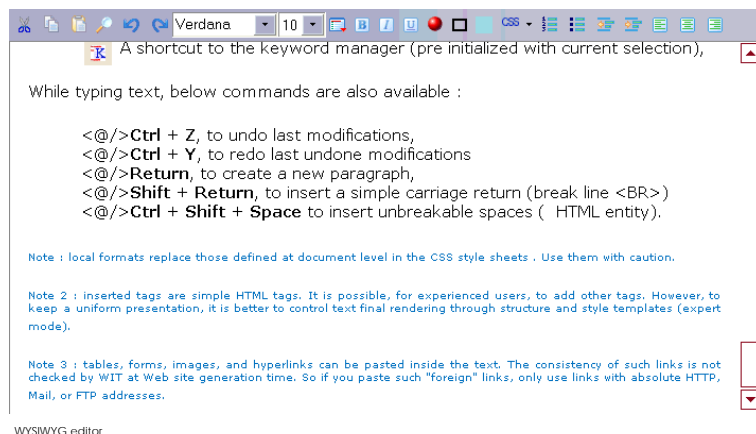
While typing text, below commands are also available :

-  **Ctrl + Z**, to undo last modifications,
-  **Ctrl + Y**, to redo last undone modifications
-  **Return**, to create a new paragraph,
-  **Shift + Return**, to insert a simple carriage return (break line <BR>)
-  **Ctrl + Shift + Space** to insert unbreakable spaces ( HTML entity).

Note : local formats replace those defined at document level in the CSS style sheets . Use them with caution.

Note 2 : inserted tags are simple HTML tags. It is possible, for experienced users, to add other tags. However, to keep a uniform presentation, it is better to control text final rendering through structure and style templates (expert mode).

Note 3 : tables, forms, images, and hyperlinks can be pasted inside the text. The consistency of such links is not checked by WIT at Web site generation time. So if you paste such "foreign" links, only use links with absolute HTTP, Mail, or FTP addresses.



See also :

- ▶ Pure HTML view
- ▶ Style class : TextGroup (text paragraphs group)
- ▶ Structure : Text
- ▶ Versions 5.0, 5.01
- ▶ Header / footer and left/right panels text editing
- ▶ Custom fields tab [updated!](#)

### 5.5.3.9.1. Paste text and images [updated!](#)

The easiest and most efficient method to add content to an idea is to copy this content from another source, and then to paste it in the current idea.

Content source can be :

- a text document,
- any word processor document (for instance : Microsoft Word, Open Office, WordPerfect, etc...),
- a spread sheet (for instance : Microsoft Excel, Open Office),
- a Web page, local or distant.

In all cases, WIT tries to keep the text formatting (styles, tables, images...), according to the Windows Clipboard contents (this depends on the chosen content source).



It is possible that you are not satisfied with the resulting formatting : either "oomuch" or "not enough" formatted.

There are two possible paste methods:

- by default, you perform "standard" paste (Ctrl + V), where WIT first chooses HTML format, then Unicode and Text, depending on their presence in the Windows Clipboard,
- you can also perform the "native" paste made by the WYSIWYG editor (Internet Explorer), which first tries to paste RTF (Rich Text Format) when found in the Windows Clipboard, then HTML. When pasting RTF, eventual images are ignored by WIT.
  - For instance, for a selection that is pasted from Excel :
    - the standard paste produces all the cell contents in a single textual character flow (Excel does not write HTML code in the Clipboard),
    - the "native" RTF paste produces the original cells (in a table), with their contents and formatting.

This information capture process can also be automated, including idea creation, with the [auto import](#) function.

#### Specific case of Images Included in Web pages.

When HTML content is involved, WIT also imports the embedded images in the text that is pasted into the idea :

1. WIT analyzes all links to images included in the HTML code,
2. Then, for each image, it tries to resolve its URL :
  - either by copying the image file from the browser cache (Microsoft Internet Explorer),
  - either by downloading the image (if domain name and link are analyzable),
  - either by copying the image from your hard disk (in the case of a local Web page).
3. In the current WIT document directory, WIT creates a "wit-imported-images" folder, then subfolders corresponding to domains in which it stores the imported images,
4. The paths to these images are added in the current idea properties (so that WIT can access the physical image files while it builds that Web site),
5. The HTML links to images are changed into ``. *(At web site build time, WIT copies the images files from their physical paths (known in the idea properties) into the "wit-icons" subfolder of target directory where the generated web pages are stored).*

If you do not use Microsoft Internet Explorer, this process may require a longer time (image are then downloaded instead of being retrieved from the cache).

Note : some images may not be imported when their URLs are too complex (dynamically evaluated URL, request or Javascript).

#### Specific case of images or images fragments, copied from an image editing software (Paint Shop Pro, Photoshop...)

WIT converts the image or image fragment from the Windows Clipboard into a PNG image file that is placed in the "wit-imported-images" subfolder, and establishes the link to this image.

#### Specific case of Javascript code Included in the pasted fragment.

In the WYSIWYG editor, WIT cleans any Javascript code that may be included in the pasted HTML fragments in order to prevent Javascript errors.

If you wish to keep Javascript code and have WIT automatically gather Javascript dependencies (typically page local variables and functions declarations), you have to paste the fragment in the "Page Preview" view (see [intelligent information paste into ideas](#) ).

See also :  
 ▶ [intelligent information paste into ideas](#)  
 ▶ [Page preview view](#)  
 ▶ [Versions 5.0, 5.01](#)  
 ▶ [Version 5.10 updated!](#)

### 5.5.3.9.2. Insert internal links

Menu Ideas  
 "Normal" view  
 Sub menu Insert links  
 Tab Idea, "Normal" view (WYSIWYG editor)

This is an advanced WIT feature : you can have a more accurate control of your pages, and at the same time, less strict layout than the uniform rules of current structure template.

WIT manages links definition and internal consistency, even if you modify or move your ideas.

First select the link type to insert:

See also :  
 ▶ [Images \(controlled by WIT\)](#)

#### 5.5.3.9.2.1. Insert a link to another idea

Select one or several ideas in the idea selector.

WIT inserts the necessary internal link(s) in your text, at current cursor position.

Menu Ideas  
 "Normal" view  
 Sub menu Insert links  
 Keyboard Ctrl + Alt + A  
 Tab Idea, "Normal" view (WYSIWYG editor)

You can activate the HTML raw editing view (expert mode) to visualize the HTML code that was inserted for each link.

By default, the displayed value for the link is the idea title : you can change it in the WYSIWYG editor.

Please note that if you selected some text and you insert just one link, WIT preserves the selected text (instead of replacing it by the idea title).

Link address is computed by WIT : you won't have to modify it, WIT manages it for you. The address looks like `<a href="@idea-id">`, where "idea-id" is the ID automatically given by WIT to the idea (or chosen by you if you decided to control IDs and [generated file names](#)). WIT will analyze the @idea-id pattern and replace it by the real file name and suffix at build time.

This feature is very similar to idea association mechanism.

See also :  
 ▶ [Idea selector updated!](#)  
 ▶ [Associations to other ideas](#)

### 5.5.3.9.2.2. Insert a link to local documents

Select one or several documents in the file selector.

WIT inserts the necessary internal link(s) in your text, at current cursor position.

Menu Ideas  
"Normal" view  
Sub menu Insert links  
Keyboard Ctrl + Alt + D  
Tab Idea, "Normal" view (WYSIWYG editor)

You can activate the HTML raw editing view (expert mode) to visualize the HTML code that was inserted for each link.

By default, the displayed value for the link is the document name : you can change it in the WYSIWYG editor.

Please note that if you selected some text and you insert just one link, WIT preserves the selected text (instead of replacing it by the document name).

Link address is computed by WIT : you won't have to modify it, WIT manages it for you.

This feature is very similar to local document linking from the idea properties dialog ("Links" tab).

See also :  
▶ Local documents  
▶ Editing idea text : WYSIWYG editor [updated!](#)

### 5.5.3.9.2.3. Insert a link to an external URL

Menu Ideas  
"Normal" view  
Sub menu Insert links  
Keyboard Ctrl + Alt + U  
Tab Idea, "Normal" view (WYSIWYG editor)

In the dialog, give a title to your link (it will be displayed in the human readable text), and type its URL address.

WIT then inserts the necessary HTML code, at current cursor position.

You can activate the HTML raw editor view (expert mode), to visualize the full HTML code.

This feature is similar to external URLs association to idea properties.



Optional : give a title in the first field. If some text is active in the WYSIWYG editor, WIT

will keep it and ignore your title. If you do not provide a title, WIT will show the address in the text.

Choose the URL type in the combo box. Then type the address in the next field. If you selected the URL type "file://", WIT shows the file selector box so you can choose a file on your computer.

See also :  
▶ URLs  
▶ FAQ : URLs

### 5.5.3.9.2.4. Insert an image (alternate method) [updated!](#)

Choose the image to insert in the file selector, at current cursor location.

Menu Ideas  
"Normal" view  
Sub menu Insert links  
Keyboard Ctrl + Alt + I  
Tab Idea, "Normal" view (WYSIWYG editor)

The image is added to current idea (shown in the idea properties with a default title beginning by "-."), while WIT creates the HTML insertion code (see note 2).

Removing an inserted image is easy : just select it and press the delete key. It will also be removed from idea properties as WIT rebuilds the page.

The ".TextImage" CSS style controls how this image is displayed in the text (advanced use).

At build time, WIT will copy the image file to its target location while generating the page.

Note : you can also drag and drop (single or multiple) images files directly in the WYSIWYG editor from the file explorer. If the dropped image is not correctly placed, select it and drag it to its new location (watch the moves the insertion point in the text as you move the image).

Note 2 : the code that WIT inserts in the page looks like:

```

```

At code generation time (either for the "Preview view", either while building the whole web site), WIT analyses ALT attribute value thanks to the @@ and @ markers, and changes the 'initial\_path' (local access to the resource, relative or absolute) into a 'computed\_path' (relative access to the resource, as seen from the generated page code). The 'computed\_path' is placed into the SRC attribute.

Meanwhile, to allow the WYSIWYG editor to properly display images, and absolute access path is mandatory in the SRC attribute: WIT also temporarily does this job.

What's more, an eventual comment (located in the ALT attribute after the @ marker) is reused in the final code.

In practical terms, for an image "image.png" placed in an "Images" subfolder of your current WIT project, you'll get the following HTML code in the HTML view :

```

```

and in the generated web page, you'll get the code:



(the web site generation process has also copied the "image.png" file from your current work directory to the 'wit-icons' subfolder of the target folder that contains all the generated web site).

See also :

- ▶ Images (controlled by WIT)
- ▶ FAQ : Relatives URLs
- ▶ Paste text and images **updated!**
- ▶ Auto-import each copy made in the clipboard

### 5.5.3.9.2.5. Insert a special WIT variable updated!



WIT has a serie of "meta tags" called "WIT variables" : their value is defined by WIT when it builds the pages.





These variables give an effortless enhancement to the document, and can be placed anywhere in :

- the text area,
- the idea title,
- the page header,
- the page footer,
- the left and right panels,
- the custom fields,
- the image titles,
- the document titles,
- the URL titles.

The most efficient places for WIT variables are the page header and footer.

WIT variables are inserted as pseudo XML tags, with the <@variable/> syntax.

Here is the list of WIT 'variable' meanings (an additional space had to be added here after the '@' to prevent WIT from interpreting the tag, otherwise you would get : , , 23:07:55 (time) etc...).

```
<@ date/> : page generation date.
<@ day/> : page generation day.
<@ month/> : page generation month.
<@ year/> : page generation year.
<@ time/> : page generation time.
<@ page/> : page file generated name (useful when placed in page header and used as a literal parameter in a PHP script).
<@ pageNum/> : sequence number of current page in the document,
<@ pages/> : total number of pages in the document,
<@ subideas/> : total number of sub ideas of current idea (useful, for instance, to give the number of articles in a site section),
<@ copyright/> : meta tag "copyright" information (professional license).
<@ author/> : meta tag "author" information (professional license),
<@ /> : bullet used in text : 
<@ prevIdea/> : link to previous idea, shown as an arrow or as text according to current structure template options ("Navigation" tab), like this : 
<@ nextIdea/> : link to next idea, shown as an arrow or as text according to current structure template options ("Navigation" tab), like this : 
<@ parentIdea/> : link to parent idea, shown as an arrow or as text according to current structure template options ("Navigation" tab), like this : .
```

The second WIT variables list below is available with WIT professional ; you can insert structural elements of current idea, with a better and more powerful manner, like here in this manual. These elements are still controlled by the current structure and style templates :

```
<@ navbar/> : the full navigation bar,
<@ text/> : current idea text content,
<@ title/> : current idea title,
<@ titleRaw/> : current idea title (text only, without presentation and style options),
<@ keywords/> : current idea keywords,
<@ children/> : current idea child links,
<@ externalLinks/> : current idea external URLs,
<@ hierarchicalPosition/> : current idea hierarchical position in the document,
<@ internalDocs/> : current idea list of associated ideas,
<@ internalRefs/> : current idea list of associated documents,
<@ bibliography/> : current idea list of books,
<@ imageList/> : current idea list of images,

<@ customFields/> : current idea list of custom fields values.
<@ customField1/> to <@ customField32/> : the raw value of the custom field 1 to 32 (with no specific style)
<@ cfWidgetToc1/> to <@ cfWidgetToc32/> : a block widget presenting a synthetic table of each field values.

<@ startDate/> : start date of task,
<@ dueDate/> : due date of task,
<@ endDate/> : end date of task,
<@ estimatedTime/> : estimated time to perform a task,
<@ spentTime/> : effective time spent to perform a task,
<@ percentComplete/> : task completion percentage,
<@ status/> : task status,
<@ category/> : task category ,
<@ riskPriority/> : risk or priority associated to a task,
<@ assignedTo/> : person to whom a task is assigned,
<@ createdBy/> : task creator,

<@ description/> : META description field,
<@ scriptProlog/> : script prolog field,
<@ scriptEpilog/> : script epilog field,

<@ creationDate/> : idea creation date,
<@ revisionDate/> : idea revision date,
<@ revision/> : idea revision number.
```

Last, variables <@ macro1/> through <@ macro10/> let you define your own macros : for instance, small Javascript scripts. These macros are

defined in the WIT option dialog ("File/Options..." menu).

- See also :
- ▶ Other structural elements : Widgets [updated!](#)
  - ▶ Custom fields Table of Contents widgets [updated!](#)

### 5.5.3.9.3. Insert a CSS style from current template

Menu	Ideas
	"Normal" view
Sub menu	Properties
Keyboard	Ctrl + Alt + C
Tab	Idea, "Normal" view (WYSIWYG editor)
Feature	Professional ed. only

This functions works in the WYSIWYG editor, either in the "Normal" view, either in the "Idea" tab in idea properties dialog.

Select the text you want the style to apply to, before invoking this function. If there is no selection, or if the selection overlap several HTML elements (for instance, paragraphs, different colors, etc...), it will apply to the parent element (for instance, the parent paragraph, up to the whole text).

In the style template dialog which is opened then, select a CSS class which name begins by a dot (example : .KeywordsGroup), and validate : this applies the selected style, enclosing the text with the CSS style specification.

In the example above, if you chose the ".KeywordGroup" CSS style, WIT will enclose the text with : <SPAN class="KeywordGroup"> and </SPAN>.

Note that the leading dot is omitted in the HTML code.

### 5.5.3.9.4. Insert multimedia documents

WIT can embed multimedia documents in your page, such as :

Menu	Ideas
	"Normal" view
Sub menu	Insert links
Feature	Not in freeware edition

- Flash® (\*.swf files)
- Shockwave® (\*.dcr files)
- Quick Time® (\*.mov files)
- Media Player® (\*.asf, \*.asx, \*.avi, \*.mpg, \*.mpeg, \*.mp3, \*.wav, \*.wmf files)
- Midi (\*.mid, \*.midi files)
- Real Audio® (\*.ra, \*.rm, \*.rpm files)
- ActiveX® (\*.\*. files)
- Plugin (\*.\*. files)
- Java applets (\*.class, \*.cla files)

In most cases, you are required to specify the width and height of the document.

WIT adds these documents as documents of the current idea, and manages them at Web site build time : they are added in the current idea list of documents with a "...-" title (see the idea property dialog).

Please note that the rendering of these documents depends on the multimedia plugins that are currently installed in the browser that displays your website.

### 5.5.3.9.5. Insert folding text

Please click on this title...

Menu	Ideas
	"Normal" view
Sub menu	Insert links
Keyboard	Ctrl + Alt + W
Feature	Not in freeware edition

### 5.5.3.10. Editing idea text : pure HTML code

You can view and modify the HTML code of the current idea text.

Tab	"Html" view
Feature	Not in freeware edition

You also may modify visible or invisible properties such as images, tables, forms, JavaScript or PHP code ...

The HTML editor has syntax highlight capabilities and it wraps the HTML code lines so you can easily edit them.

It has the standard editing commands that you may expect to use in an editor (Ctrl + C (copy), Ctrl + X (cut), Ctrl + V (paste), Ctrl + Z (unlimited undo), Ctrl + Y (unlimited redo), Ctrl + F (search), Ctrl + H (replace), etc...).

HTML items can be folded/unfolded while clicking on the minus/plus signs in the left column.

You can also use regular expressions when searching / replacing text.

Please follow the link below for the regexp syntax.



Search dialog



image or screen capture software or when you press the Ctrl + Print Screen key.	current idea images list (5)		
---	------------------------------	--	--

When the left view is active (idea tree), the paste operation inserts the last copied idea tree. (This copy does not affect the clipboard, and is made in a file).

Notes :

(1) Do the paste with the mini toolbar icon.

(2) In the limits that HTML language allows.

(3) WIT tries to locate the images that are used in the copied page by resolving their access path : it downloads and copies them in a "wit-imported-images" subfolder, located in the same folder as the \*.wit project. Images are finally stored in second level subfolders, named according to the domain where they were found (for instance wit-imported-images\www.webideatree.com\tucows-5cows.png). See also : [Paste text and images](#).

(3b) Javascript code is preserved and eventual related Javascript declarations (functions and variables) are also pasted in the idea "scriptProlog" field.

(4) This operation gives the same results as idea properties editing ([Links tab](#)), or as the drag and drop in this tab. Multimedia documents (sounds, vidéos, animations...) are appended to the end of current idea text.

(5) The name of added image is the current idea identifier (ID), followed by a number. WIT creates an image file from the fragment, names it with the image name, and places it into the current "wit-imported-images" \*.wit project subfolder.

See also :

- ▶ [Paste text and images](#) <sup>updated!</sup>
- ▶ [Versions 5.0, 5.01](#)

## 5.5.4. Idea preview

Menu [Ideas](#)

### 5.5.4.1. Internal preview

Idea preview may be internal : just activate the preview area with Ctrl+Q, and select an idea.

Menu [Ideas](#)  
Sub menu [Preview \(Builtin browser\)](#)  
Keyboard [Ctrl + Q](#)

Note : the hypertext links, which are displayed in the this view are correct only if the user preference "Automatic built of preview pages (preview view)" is active.

See also :

- ▶ [Automatic Web site generation](#)
- ▶ [Page preview view](#)

### 5.5.4.2. Preview in Web browser

Menu [Ideas](#)  
Sub menu [Preview \(external browser\)](#)  
Keyboard [Alt + V](#)

Preview can also be done with an external Web browser (defined in the options) : select an idea, and press "Alt + V" key.

Note : the hypertext links, which are displayed in the this view are correct only if the user preference "Automatic built of preview pages (preview view)" is active.

See also :

- ▶ [Default navigator choice](#)
- ▶ [Automatic Web site generation](#)

## 5.5.5. Multi criteria idea search

This function helps to select ideas according to a search criteria.

Menu [Edit](#)  
Sub menu [Search](#)  
Keyboard [Ctrl + F](#)

See also :

- ▶ [Versions 5.0, 5.01](#)

### 5.5.5.1. Selection criteria <sup>updated!</sup>

Selection criteria dialog helps to find a string to search in all fields of all ideas (default behavior).

Menu [Edit](#)  
Sub menu [Search](#)  
Keyboard [Ctrl + F](#)

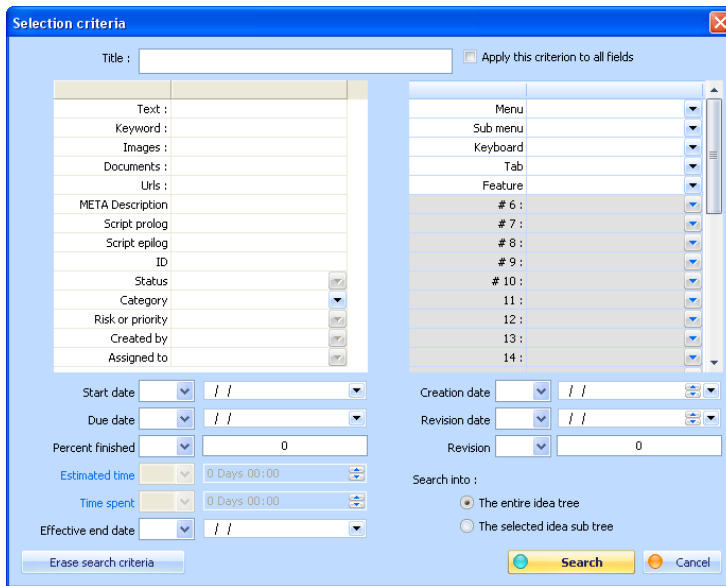
For a more accurate search (figure 4.21), uncheck the box "Apply criteria to all fields", and type values in chosen fields.

Search does a logical AND between all values, field by field. Void fields are ignored.

Search range can be:

- either the current tree branch (under the selected idea)
- either the whole document.

Search results will be highlighted with a bacground color specific to each criterion, both in the WYSIWYG editor and in the Preview view.



The 'Selection criteria' dialog box is used to define search filters. It includes a title field, a checkbox to 'Apply this criterion to all fields', and two main sections of criteria. The left section contains fields for Text, Keyword, Images, Documents, Urls, META Description, Script prolog, Script epilog, ID, Status, Category, Risk or priority, Created by, and Assigned to. The right section contains fields for Menu, Sub menu, Keyboard, Tab, Feature, and a list of numbered items (# 6 to # 14). Below these sections are date and time fields for Start date, Due date, Creation date, Revision date, Percent finished, Estimated time, Time spent, and Effective end date. At the bottom, there are buttons for 'Erase search criteria', 'Search', and 'Cancel'. A 'Search into' section with radio buttons allows selecting 'The entire idea tree' or 'The selected idea sub tree'.

Fig 4.21 : selection criteria

See also :  
[Version 5.10 updated!](#)

### 5.5.5.2. Idea selector [updated!](#)

First, the search function opens idea selector dialog (figure 4.20) which displays ideas matching the current search criteria.

Menu [Edit](#)  
 Sub menu [Search](#)  
 Keyboard [Ctrl + F](#)

To activate, or retrieve the last used search criteria, click on the "Search criteria" button, which opens the criteria dialog box (see the next paragraph).

When this dialog is validated, idea selector filters ideas according to the new search criteria and displays how many ideas were found.

To modify an idea, just double click it.

The idea selector displays the ideas in a grid report, and lets you :

- display and resize all or part of idea attributes,
- sort and group ideas by attributes values.
- quick edit idea attributes without leaving the grid report.

The "Save..." button saves the current report grid configuration.

The "Load..." button restores a previously saved report grid configuration.

The "Export to CSV..." button exports the currently displayed ideas, with respect to the current field display and order, and to the idea sorting.

And the "OK" button closes the selector and automatically sets the main tree view on the last idea that was selected in the report grid (new current idea).

Note : A magnifying glass on the "Selection Criteria" button means that a search criterion is active.

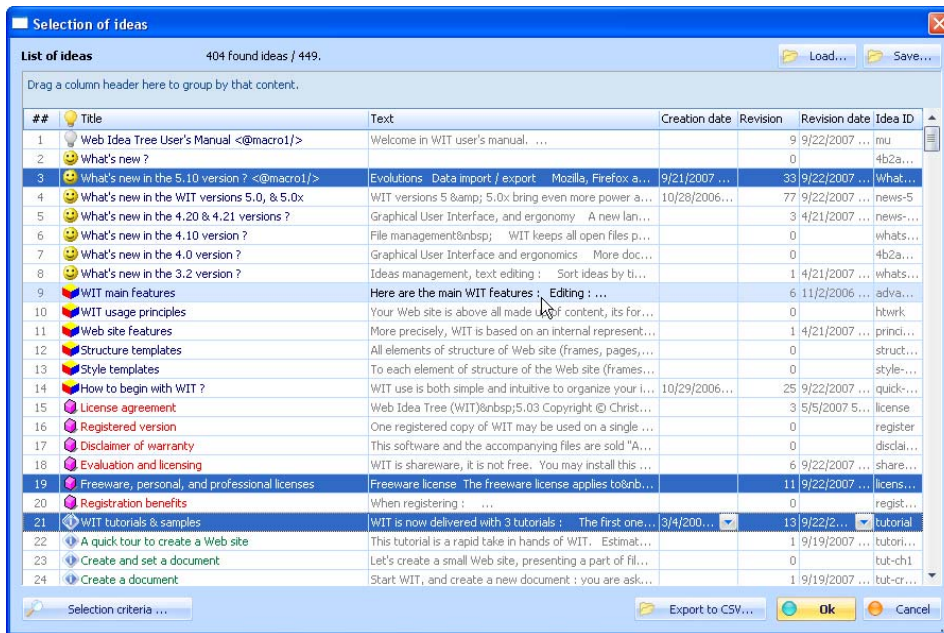
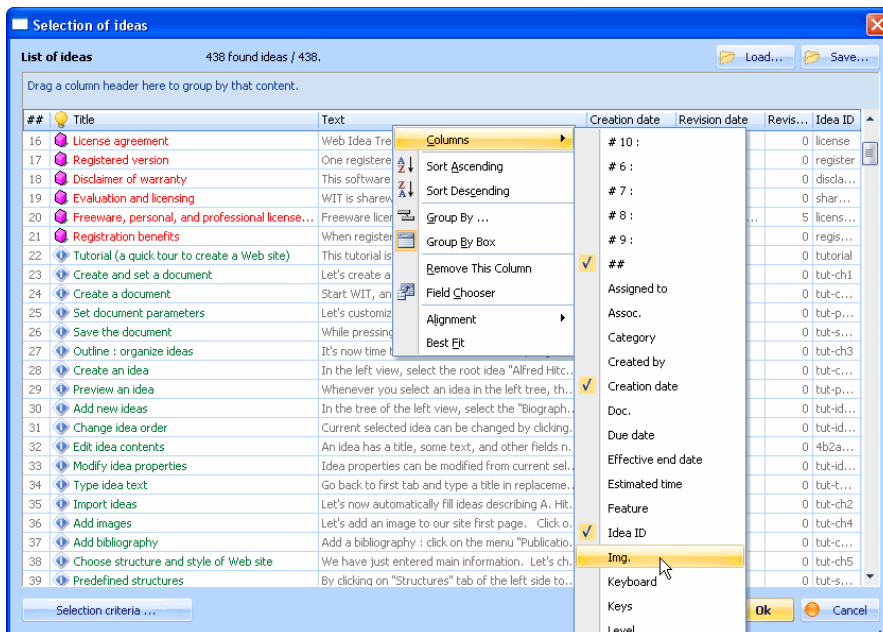


Fig 4.20 : Idea selector

#### 5.5.5.2.1. Selecting displayed attributes, columns size and order

Right click on the list header to choose which idea attributes to display : in the popup menu, select the "Columns" item, then click on every idea attribute you want to display or hide. A final click outside the menu validates your choices.

You can also resize each column width, change column display order by dragging / dropping their headers, and resize the whole dialog : the global configuration of this grid report will be recalled later (unless when the "Cancel" button is pressed).



Choice of columns

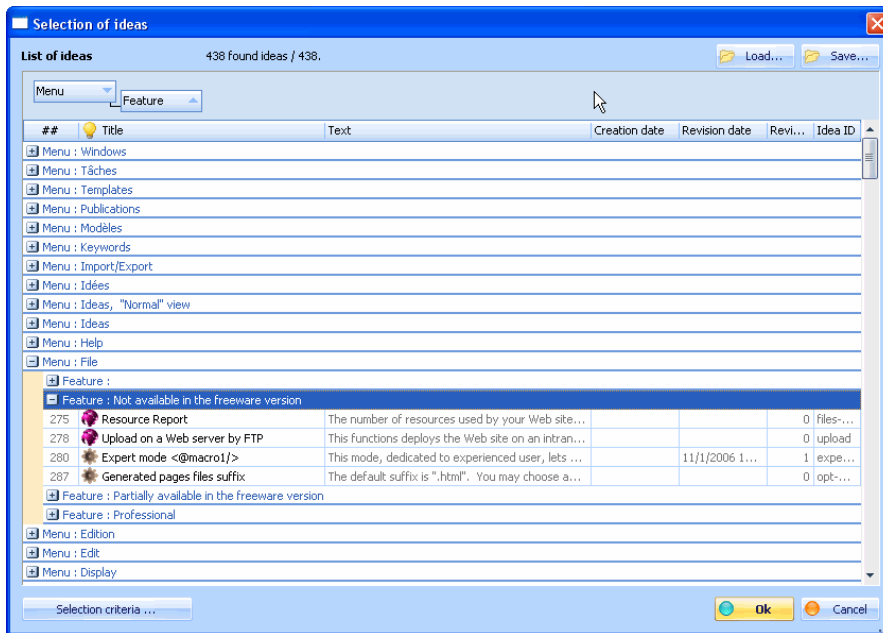
#### 5.5.5.2.2. Sorting and grouping

The whole idea list, can be sorted according to any of displayed columns : click on the column header to have an ascending or descending sort on this column (additional click).

To retrieve the initial idea order (which is the idea tree order of your document), click on the "##" column header.

You can also group the ideas on the values of a particular attribute : select a column header, and drag & drop it into the box area beneath the column headers. You can nest several group criteria.





Grouping Ideas on attributes values

### 5.5.5.2.3. Quick idea attributes editing

You can quickly edit idea attributes without leaving the report grid :

- a single click in a non grayed cell allows direct editing of its contents, and updates the corresponding idea attribute,
- a double click on a line opens the corresponding idea property dialog, which presents the correct tab showing the idea attribute that was designated by the double click in the cell of the report grid.

Grayed cells generally correspond to :

- complex fields such as multiline text of ideas,
- composite fields with object lists, such as associations lists, images, documents, keywords or URLs.

In these cases, editing is done in the idea property dialog box.

### 5.5.6. Search & Replace updated!

This functions searches and replaces a character string by another one, *from the current selected idea*.

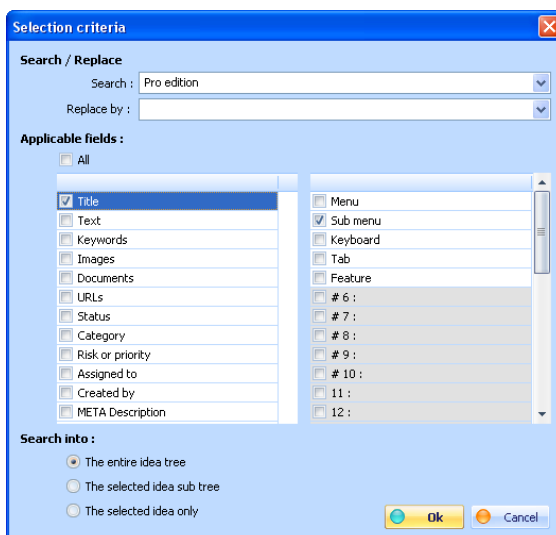
The search / replace operation work on all idea modifiable attributes, or limited to selected ones.

It can also be used on :

- the entire idea tree
- the selected idea sub tree
- the select idea only

Use it with caution, because there's no undo except reverting to an automatic save of your document !.

Menu **Edit**  
Sub menu **Replace...**  
Keyboard **Ctrl + H**



Search / replace

See also :  
 ▶ Version 5.10 [updated!](#)

## 5.6. Import / Export

Menu [Import/Export](#)

### 5.6.1. Import ideas from CSV file [updated!](#)

Menu [Import/Export](#)  
 Sub menu [Import ideas from CSV file](#)

This function helps to create numerous ideas, from a text file containing delimited records (database statement) : character separated values (CSV).

To import ideas, follow the steps below :

- 1. Select or create a parent idea,
- 2. Prepare a data file containing the records (for example, a list of films *Title@French-Title@Type-Keywords@Year@Duration@Critics@Character1@...@Character7*).
- 3. Configure the current document (options), so that custom fields names correspond to record values,
- 4. Invoke import idea function (in *Ideas* menu).
- 5. In the dialog box (figure 4.8), define fields to be used, the order in which they must be read to match record fields order, and set the delimiting character,
- 6. Validate or correct the import format choice. Ideas are imported in the read order, and become children of the current idea.

Expected values for each field :

- **Title** : string (interpreted as idea title),
- **Text** : string (interpreted as idea text),
- **Keyword** : string (interpreted as idea keyword),
- **Type** : integer (1 for idea type, 2 for glossary type, 3 for bibliographic notice type),
- **URL** : string (interpreted both as link displayed value and link address ; if you wish to distinguish between this URL name and its value, see the "Delimiter" paragraph),
- **Image** : string (interpreted as local path to image file, and as image legend (the file name) ; if you wish to distinguish between this image name and its path, see the "Delimiter" paragraph),
- **Document** : string (interpreted as local path to document file, and as link to document name ; if you wish to distinguish between this document name and its path, see the "Delimiter" paragraph),
- **Custom fields 1 to 32** : string
- **Start date** : string, formatted as a date (mm/dd/yy) : interpreted by WIT as task start date. Ignored if incorrect.
- **Due date** : string, formatted as a date (mm/dd/yy) : interpreted by WIT as task due date. Ignored if incorrect.
- **Effective end date** : string, formatted as a date (interpreted ) : interpreted by WIT as task end date. Ignored if incorrect.
- **Estimated time** : integer. Interpreted as duration (number of days).
- **Estimated spent** : integer. Interpreted as duration (number of days).
- **Completion (%)** : integer. Interpreted as task completion percentage (from 0 to 100).
- **Status** : string (interpreted as value for task status).
- **Category** : string (interpreted as value for task category).
- **Risk / Priority** : string (interpreted as value for task risk / priority).
- **Assigned to** : string (interpreted as value for task assignment).
- **Created by** : string (interpreted as value for task author/emitter).
- **Description (META)** : string (interpreted as idea comment inserted into HTML meta tags).
- **Script Prolog** : string (interpreted as script prolog)
- **Script Epilog** : string (interpreted as script epilog)
- **Creation date** : string, formatted as a date - time (mm/dd/yy hh:mm) : interpreted by WIT as idea creation date. Ignored if incorrect.
- **Modification date** : string, formatted as a date - time (mm/dd/yy hh:mm) : interpreted by WIT as idea modification date. Ignored if incorrect.
- **Revision** : integer, interpreted as idea revision index.
- **Idea Identifier** : optional string (interpreted as idea identifier. Replaced by WIT by an automated value if it is not unique or contains special (unauthorized) characters).
- **Idea Level** : optional integer (represents the depth level of idea in the tree : used by WIT to re-create a tree instead of doing a "flat" import).
- **Publications** : string (interpreted as link to a publication). [To be effective, the links are made with the concatenation of the author name, a space character, and the title].

#### Delimiter

Fields are delimited by a separator character (separator 1). A secondary delimiter (separator 2) is used to analyze multiple element fields such as keywords, images, documents, or URLs fields.

For images, documents or URLs fields, a third delimiter allows to distinguish between their names and actual paths. When this third delimiter is not found in the analyzed text, the current field value is affected to both title and path of document / image / URL. When it is found, the value analyzed before the delimiter is the name, and the value analyzed after the delimiter is the path.

Example 1 :

The analyzed field (images) is : "c:\tmp\img1.png, c:\tmp\img2.png"

Two images are recognized :

- image 1 with "img1.png" title and "c:\tmp\img1.png" path
- image 2 with "img2.png" title and "c:\tmp\img2.png" path

Example 2 :

The analyzed field (images) is : "Title of image 1 | c:\tmp\img1.png, Title of image 2 | c:\tmp\img2.png"

Two images are recognized :

- image 1 with "Title of image 1" title and "c:\tmp\img1.png" path
- image 2 with "Title of image 2" title and "c:\tmp\img2.png" path

**Note** : you can specify a same target field several times : in this case, imported values will be concatenated to this field.

**Note** : it may be useful to sort data in the file to import, in order to control final children idea sequence.

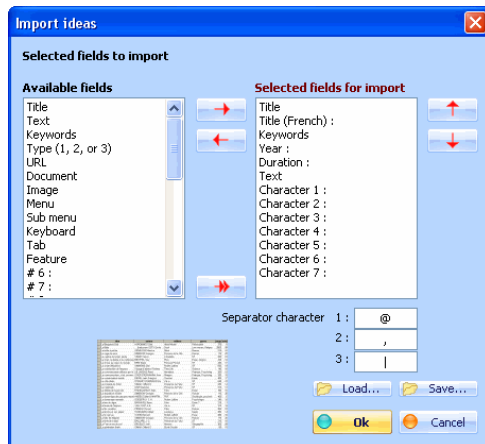


Fig 4.8 : Import ideas from a delimited record file.

### 5.6.2. Import ideas from multiple HTML or text files

With this function, you may import texts from a list of files, selected from a folder.

Menu [Import/Export](#)  
 Sub menu [Import ideas from HTML or text files](#)  
 Feature [Not in freeware edition](#)

Each selected file creates an idea :

- The idea title is the file name (or the < TITLE > tag value if the file is HTML),
- Idea text is the file text. Note : you may keep or eliminate the text format while importing (HTML files).

Operations :

- Click on the "Select files" button to establish the import list.
- Choose the files (use Shift or Ctrl + click if needed to select files).
- Files are imported one after the other. A preview of their contents is displayed : so you may import or skip a particular file (use the dialog buttons).
- The main view is refreshed each time a file is imported.
- The current file name, and the remaining number of files are displayed in the dialog title.
- Check or uncheck the "Keep text format" box, according to your needs.
- Check the "Import all files" option if you don't wish to confirm every file import.

**Important note :** when importing HTML files, you are responsible of their internal consistency regarding references (images and links).

### 5.6.3. Import folders

This function imports folders and their contents, creating ideas with text, images and documents.

The folder import dialog is made of three parts, describing the files suffixes to consider (not mentionned file types are ignored) :

1. File types to import as ideas : WIT renders these files as ideas which text is the file contents.
2. File types to import as images: WIT adds references to theses files in the current idea list of images (the images will be shown in the idea).
3. File types to import as documents : WIT adds references to theses files in the current idea list of local documents.

The "Include sub folders" option recursively adds current folder sub folders.

The "Make image album" option change the import behavior for imported images : WIT creates an idea for each image (wherever the image file suffix is specified), and creates thumbnails of images in every folder containing images.

Use both option to recursively create image albums.

WIT creates at least as many ideas (nodes) as encountered folders.

You can save and restore the import folder parameters, with \*.fif files (*folder import format*).



Import folders

See also :

- ▶ Versions 5.0, 5.01
- ▶ FAQ : Create a photo album

#### 5.6.4. Import multiple images from folder

This function creates a list of child ideas (under the current idea), and each child contains an image selected from a folder.

Menu [Import/Export](#)  
Sub menu [Import images](#)

In the file selector, choose all the images you want to import (GIF, JPEG or PNG formats).

Child ideas are then created, each with the image file name as default title.

See also :

- ▶ Build thumbnails automatically

#### 5.6.5. Import web browser favorites links <sup>updated!</sup>

Menu [Import/Export](#)  
Sub menu [Import Web browser favorite links](#)

This function imports all your Web browser favorite links (or bookmarks), and creates a sub tree under the current idea.

The sub tree has the same structure as your favorite links. Links are added as external URLs. Their titles are the link names.

Supported web browsers are :

- Microsoft Internet Explorer
- Mozilla
- Firefox
- Opera.

See also :

- ▶ URLs

#### 5.6.6. Auto-import each copy made in the clipboard

When this mode is active, each copy made in the clipboard is detected by WIT and pasted into ideas.

Menu [Import/Export](#)  
Sub menu [Auto import from clipboard](#)  
Feature [Not in freeware edition](#)

The behavior depends on the current active view :

🔗 WYSIWYG editor : each copy in the clipboard is pasted as new text paragraph into the editor.

🔗 HTML raw code editor : the raw text is pasted into the editor.

🔗 Preview view : each copy in the clipboard **creates a new idea in WIT**, under the current selected idea. Each new idea contains a link to the source of copy (URL).

Switch off this mode when not used any more.

**Note :** When the WYSIWYG editor or the "Page preview" are enabled, WIT tries to also import images that are included in the selection. It will also try to keep the styles and formatings used in the source (except CSS styles that may be defined outside the selection).

#### 5.6.7. Import ideas from database tables (ODBC)

You may import data from database tables, via ODBC (WIT professional license only).

Menu [Import/Export](#)  
Sub menu [Import ideas from database \(ODBC\)](#)  
Feature [Professional ed. only](#)

Select the idea where imported data will be stored, and activate "Import ideas from database (ODBC)", in the "Ideas" menu.

Importing database tables is a three step operation (successively click on the top button of the dialog) :

1. Connect to a database.
2. Select a table to import.
3. Select the fields to import, and import.

When a table has been selected, you can have a preview of table data.

Ideas are imported in the same order as the table records, and use only selected fields. They are created under the current idea.

### 5.6.7.1. Connect to the database

Feature **Professional ed. only**

Follow standard Windows dialog guidelines to connect to the database : the database must have been installed as an ODBC source, from Windows Configuration Panel.

You may so access to different databases : dBase, MS Accesse, MS SQL Server, MySQL (with a downloaded driver), Oracle, etc...

### 5.6.7.2. Select a table

Feature **Professional ed. only**

Select the table from which you want to import data.

You may iterate the selection until it fits your needs (the contents preview in the main ODBC import dialog will help choosing).

If you want to import data from several tables, you have to create a result table first, from within the database, and then import the result table.

### 5.6.7.3. Select the fields to import

Feature **Professional ed. only**

This dialogs helps you establishing the link between imported data and the way they will be stored into WIT ideas.

1. Select (check/uncheck) the DB fields you want to import.
2. Associate a WIT field to each DB field to import (use the combo boxes).

Using at least WIT "Title" and "Text" fields is a good practice.

If you associate several DB fields to a single WIT field (e.g. WIT "Text" field selected twice), DB data will be concatenated into the WIT field (using a space separating character).

## 5.6.8. Import an idea tree (XML)

This function inserts an idea tree from a \*.xml file.

Menu **Import/Export**  
Sub menu **Import an idea tree (XML)**  
Feature **Not in freeware edition**

File contents are inserted under the current idea, like when pasting the selection.

The XML file must have formerly been produced by WIT, or respect the WIT XML grammar.

### 5.6.8.1. Import XML from Web Idea Tree

Menu **Import/Export**  
Sub menu **Import an idea tree (XML)**  
Feature **Professional ed. only**

This function imports a WIT idea tree, made by a previous WIT idea tree export, or by a manual build, or by a XSLT transformation.

See also :  
► Versions 5.0, 5.01

#### 5.6.8.1.1. WIT XML format (grammar)

The XML grammar of WIT generated files will be described soon.

Feature **Professional ed. only**

### 5.6.8.2. Import XML from TreePad (tm)

Menu **Import/Export**  
Sub menu **Importer une arborescence d'idées (XML)**  
Feature **Not in freeware edition**

This function imports XML data from the excellent TreePad<sup>tm</sup> software and rebuilds an idea tree, similar to the TreePad articles and nodes.

For this, you first need to prepare an XML file from TreePad (export data from TreePad) :

1. In TreePad main view, select the node or article you want to export.
2. Right click on it to show the popup menu and choose "Export subtree / to files"
3. Selection the option : "Export to one file"
4. Choose the option : "One XML file"
5. Choose the options "Treepad XML file" and "Export rich text articles as HTML" for the text export.
6. Let the options checked : "Include articles" and "Include nodes"
7. This produces an XML file that you can then import into WIT.

Treepad is a registered trademark of Freebyte.com.

See also :  
► Versions 5.0, 5.01

## 5.6.9. Export an idea tree (XML)

This function exports the current idea and its children in a \*.xml file.

The file can be reused in other documents (see the import idea tree function).

The file can be reused in other WIT documents (see the import function), it is readable by a human being and can be easily be transformed or processed with standard XML tools.

## 5.6.10. Export ideas into a CSV file updated!

This function exports selected ideas (from current search criteria or all ideas) into a CSV file.

1. First, the search dialog is displayed. You may:
  - o enter a search criterion or not (if you do not wish any filtering),
  - o choose the search range: either the whole document or just the current selected idea sub tree.
2. Then, WIT displays the idea selector with search results, from which you can export to a CSV file.
  - o Note : you can [configure the idea selector](#) to display only the idea fields that you wish to export. WIT will only export the displayed fields, in the order they appear, with the current applied sort criteria.

See also :

► [Personal, professional, and freeware licenses](#) updated!

## 5.7. Keywords

Keywords are used in the index and assigned to ideas.

Menu **Keywords**

### 5.7.1. Keywords updated!

Keywords can be associated to every idea.

Menu **Keywords**  
Sub menu **Keyword manager**  
Keyboard **Ctrl + Shift + K**

The total number of keywords is not limited.

These keywords are then used to build an index when the web site is created, and establish correlations between all ideas using the same keyword.

Keyword management is done at two levels :

- management of the global list of keywords,
- management of current idea keywords.

Keywords must exist in the list prior to be added to ideas.

#### Management of list of keywords.

To **add** new keywords to the list, click on the *"Create"* button, and type a new keyword.

To **remove** a keyword from the list, select it, and click on the *"Remove"* button. If the keyword is already used by ideas, these ideas will be displayed in an error dialog : pressing *"OK"* in the dialog will remove the keyword, and *"Cancel"* will preserve it.

You may also **control** your keywords by defining preferred keywords. A preferred keyword is a more general meaning keyword to which the current keyword refers. For example, you might want all ideas using the *"Apple"* keyword to be listed under the *"Fruits"* keyword. To do this, you have to create a preferred keyword as new keyword : *"Apple @ Fruit"*. So ideas using *"Apple"* keyword will be listed under the *"Fruits"* entry in the index, while the *"Apple"* entry in the index will redirect to *"Fruit"* entry (*"Apple"* : see *"Fruits"*).

To **rename** a key word, proceed as to remove it and you'll be asked for a new keyword. If the old keyword is already used by ideas, these ideas will be displayed in an error dialog : pressing *"OK"* in the dialog will replace the keyword (both in keyword list and ideas), and *"Cancel"* will keep it.

To **import** a list of keywords from an ASCII or UTF-8 text file (one keyword per line), use the *"Import list"* button. (Professional ed. only license)

To **export** the current keyword list to an ASCII or UTF-8 text file, use the *"Export list"* button. (Professional ed. only license)

#### Management of Idea keywords.

To **add** or **remove** a list of defined keywords in an idea, select the keywords you want (you may use Shift and Ctrl click combination to select several words), and click on the *"Add"* or *"Remove"* button.

If the *"Recursive"* box is checked, this action also applies to current idea children.

#### Cross references

If the *"Show usage"* option is checked, the number of ideas using every keyword is shown.

To **list** all ideas using a specific keyword (cross references), select the keyword in the list, then press the *"Cross references"* button.

Inserted keywords are updated in the idea tab, and alphabetically sorted.

**Tip :** to create and easily add keywords from chosen words in the WYSIWYG editor, select a word and show the Keywords Manager : It shows the Keyword creation dialog, pre-filled with the selected word ; just validate twice with 'Return' key (first validation creates the new keyword, second adds it).

**Tip 2 :** a double click in the keyword list adds the selected keyword in the current idea, without closing the dialog.

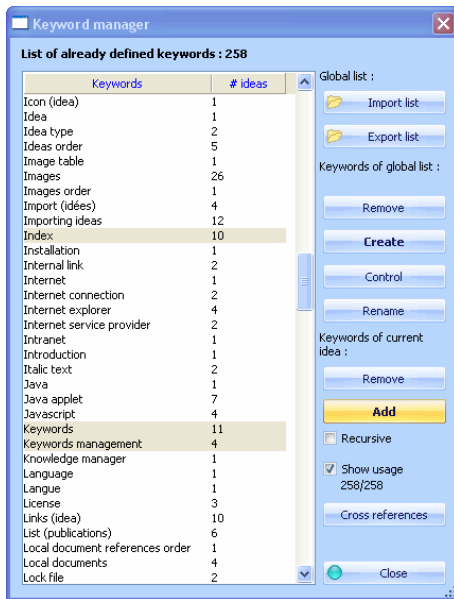


Fig. 4.13 : Keywords manager

See also :

- ▶ [Collect words automatically](#)
- ▶ [Structure : Keywords](#)
- ▶ [Style class : KeywordsGroup \(keywords group\)](#)
- ▶ [Rebuild all indexes](#)

## 5.7.2. Collect words automatically

Working with current selection, this function partly automates the keyword index build :

- Collects all the words used in the current idea and its children (in the "title" and "text" fields).
- Displays a dialog box to select the correct keywords (figure 4.9b) : eliminate all unwanted or irrelevant ones,
- Adds selected keywords to all ideas using them.

Tip : the word collection represents all the used words (several hundreds of different words in a medium sized web site). You should carefully select the words to be used as keywords.

To help you, check the "Show occurrences" box : this displays how often every word appears in the selected idea and its children (this may require some time in large selected trees).

You can also sort the words alphabetically, by occurrence, or by length with a click on the list header : so you will be able to identify and eliminate rare, too frequent, or too short words to be included in the index.

In the word selection dialog, you may :

- eliminate at once several words from the list (multiple selection) : "Remove from list" button,
- add the whole list : "Add all the list" button,
- add only current selection of words (multiple) : "Add current selection" button.

If you check the "Match whole words only" option, only single words corresponding to whole words in a idea will be added as keywords of this idea : for instance, if the word is "euro" and if the current idea contains the "Europe" word, "euro" won't be added as keyword. If the option is not checked, "euro" will be added if "Europe" is encountered.

The "Cancel" button closes the dialog box without adding any keyword.

Menu [Keywords](#)  
 Sub menu [Collect words..](#)  
 Keyboard [Ctrl + K](#)  
 Feature [Not in freeware edition](#)

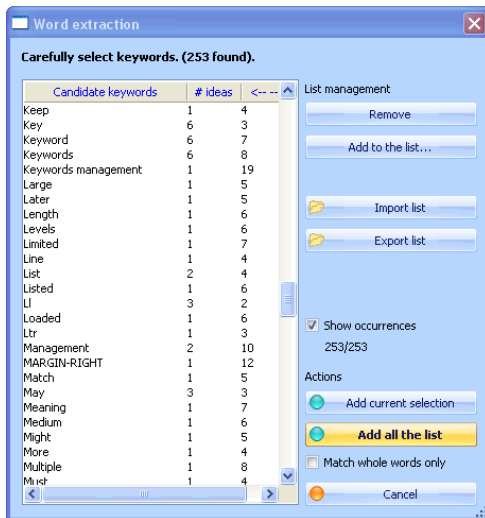


Fig 4.9b. Automatically build a list of keywords

See also :

- ▶ Keywords [updated!](#)
- ▶ Rebuild all indexes

### 5.7.3. Copy keywords of current idea

Menu **Keywords**  
 Sub menu **Copy idea keywords**  
 Keyboard **Ctrl + Shift + C**  
 Feature **Professional ed. only**

This function copies the keywords of current idea in a special internal clipboard, so you may paste (add) them later into another idea.

### 5.7.4. Paste idea keywords

This function adds the previously copied keywords to the current idea.

Already existing keywords are simply ignored.

Menu **Keywords**  
 Sub menu **Paste idea keywords**  
 Keyboard **Ctrl + Shift + V**  
 Feature **Professional ed. only**

## 5.8. Tasks

This menu lists all defined tasks (idea attributes), according to different criteria (cf figure).

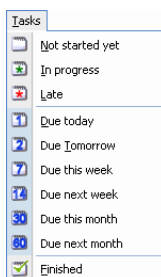
Menu **Tasks**  
 Feature **Not in freeware edition**

Tasks that match the criteria are searched and displayed in the idea selector (this one is shown in task oriented configuration displaying specific task attributes). It's then easy to edit tasks from this report grid.

Search / selection criteria :

- **Not started** : ideas which have a 0% completion, and with a specified due date.
- **In progress** : ideas which have a non null completion, with an expressed start date, but without effective end date.
- **Late** : ideas which have a completion less than 100%, without effective end date, and for which due date is passed.
- **Due today** : unfinished tasks, with no effective end date, and for which due date is today.
- **Due tomorrow** : unfinished tasks, with no effective end date, and for which due date is tomorrow.
- **Due this week** : unfinished tasks, with no effective end date, and for which due date the end of the current calendar week.
- **Due next week** : unfinished tasks, with no effective end date, and for which due date the end of the next calendar week following current week.
- **Due this month** : unfinished tasks, with no effective end date, and for which due date the end of the current calendar month.
- **Due next month** : unfinished tasks, with no effective end date, and for which due date the end of the next calendar month following current month.
- **Finished** : tasks which have a 100% completion, whatever their due date.

Note : late tasks show a red due date in the report grid.



Tasks management menu

See also :

- ▶ Versions 5.0, 5.01
- ▶ Tasks tab [updated!](#)



- ▶ Idea selector [updated!](#)
- ▶ Task progress status widget [updated!](#)

## 5.9. Publications

WIT manages theme organized documentary databases (publications).

Menu [Publications](#)

Publication databases are included in the document, and saved with it.

The publication manager helps to view and modify publications, whereas load, save and import functions help to manage lists of publications, in order to exchange them between documents.

The bibliography index is then automatically built while you generate the web site. You may refer to structure and style templates to control how (contents and style) the bibliography index will be generated.

Note about terminology :

- 'Publication' and 'book' terms are used with the same meaning in WIT software and documentation.
- If the exact term should be 'publication reference' or 'bibliographic reference', 'publication' or 'book' are used instead by misuse of language.
- Bibliography is a list of publication references which are grouped for a special purpose. In WIT software, the bibliography is an index of all the publications used in the WIT document.

See also :

- ▶ Structure templates
- ▶ Navigation tab
- ▶ Publications tab [updated!](#)
- ▶ Style class : BookGroup (publication presentation group)
- ▶ Style class : NoteGroup (Bibliography note group)
- ▶ Style class : BibliographyGroup (Bibliography group)
- ▶ Style class : IndexGroup (keywords index group) [updated!](#)

### 5.9.1. Publications manager

The publications manager (figure 4.22) :

Menu [Publications](#)  
Sub menu [Manage publications ...](#)  
Keyboard [Ctrl + Shift + B](#)

- selects publications by theme,
- views publications ,
- creates publications ,
- modifies publications ,
- or removes them.

All publication fields are displayed in the publications manager : so you can resize the dialog and display it in full screen.

The behavior is the same as the idea selector : you may choose the displayed fields, their order, sort and group by fields, and directly edit cells from within the grid to modify publications.

You can save / restore the display configuration of the publication manager.

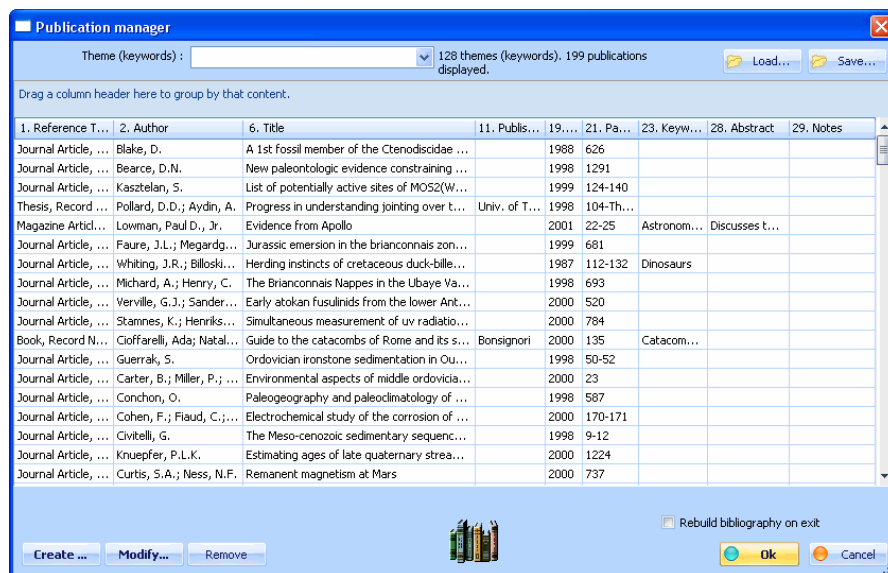


Fig 4.22 : Book manager

See also :

- ▶ Versions 5.0, 5.01
- ▶ Idea selector [updated!](#)

#### 5.9.1.1. Filter publications by theme

Menu [Publications](#)  
Sub menu [Manage publications ...](#)

By clicking on the "Theme (keywords)" box, or by typing characters, the book manager filters publications according to the chosen theme.

The search is not only done in the "Keywords" field, but also in the "Author", "Title" and "Short title" fields.

### 5.9.1.2. Sort publication list

Publication list is sorted by author name (default).

Menu **Publications**  
Sub menu **Manage publications ...**

To sort by any other field, click on the column headers.

### 5.9.1.3. Create a publication

Clicking on the "Create" button opens the publication edit dialog (figure 4.23), where you can type publication information.

Menu **Publications**  
Sub menu **Manage publications ...**

Please note that, depending on the current field, you may be proposed some help to enter your information when you click on some fields :

- Pre defined drop down menus for fields like "Keywords", "Author"..., but you can obviously create new items.
- A WYSIWYG text editor for multiline text for fields like "Notes" and "Abstract",
- A file selection dialog for entering one image pathname in the "Image" field. (The image will be rendered at page or bibliography build time, like other images in your web site).

The screenshot shows a 'Publication' dialog box with a table of fields and values. The fields are numbered 1 through 40. The values are as follows:

Field	Value
1 Reference Type	-- Values --
2 Author	CMMI Product Development Team.
3 Secondary Author	
4 Tertiary Author	-10
5 Subsidiary Author	
6 Title	ARC V1.0, Assessment Requirements for CMMI, Version 1.0 (CMU/SEI-2000-TR-0...
7 Short Title	SEI 00e
8 Alternate Title	
9 Secondary Title	SR148
10 Tertiary Title	
11 Publisher	Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University
12 Place Published	
13 Edition	
14 Original Publication	
15 Reprint Edition	
16 Volume	
17 Number of Volumes	
18 Number	
19 Year	2000
20 Date	
21 Pages	0
22 Section	
23 Keywords	CMM
24 Medium	
25 ISBN/ISSN	
26 Reviewed Item	
27 Label	
28 Abstract	
29 Notes	
30 URL	http://www.sei.cmu.edu/publications/documents/00.reports/00tr011.html
31 Author Address	
32 Image	
33 Caption	
34 Custom 1	
35 Custom 2	
36 Custom 3	
37 Custom 4	
38 Custom 5	
39 Accession Number	
40 Call Number	

At the bottom of the dialog box are 'Ok' and 'Cancel' buttons.

Fig 4.23 : Edit a book

### 5.9.1.3.1. WIT publication fields <sup>updated</sup>

Here are all the publication fields supported by WIT.

Menu **Publications**  
Sub menu **Manage publications ...**

The list below gives their names (default name first, and their eventual aliases or meaning variations) :

1. **Reference Type**, type of publication, book, manuscript, article, report, thesis, communication, artwork, audiovisual material, film, electronic source
2. **(m) Author**, reporter, programmer, writer, artist, cartographer, created by
3. **(m) Secondary Author**, series editor, recipient, director, committee, reporter
4. **(m) Tertiary Author**, series editor, producer
5. **(m) Subsidiary Author**, translator, performers, sponsor, counsel
6. **Title**, name of act, case name
7. **Short Title**, abbreviated title, abbreviated case name
8. **Alternate Title**, reference
9. **Secondary Title**, journal, series title, book title, collection title, magazine, newspaper, conference name, academic department
10. **Tertiary Title**, series title, legislative body
11. **(m) Publisher**, university, institution, producer, distributor, assignee, court
12. **(m) Place Published**, city, conference location, country
13. **Edition**, version, session
14. **Original Publication**, history
15. **Reprint Edition**
16. **Volume**, code volume, code number, reporter volume
17. **Number Of Volumes**, extent of work

18. **Number**, issue, issue number, document number, bill number, public law number
19. **(m) Year**, year of publication, year of conference, year released
20. **Date**, date of creation
21. **Pages**, number of pages, description, code pages, first page
22. **Section**, code section, sections
23. **(m) Keywords**, theme
24. **Medium**, type of work, type of article, type, type of medium
25. **(m) Isbn/Issn**, isbn
26. **Reviewed Item**, review rating
27. **Label**
28. **Abstract**
29. **Notes**
30. **Uri**
31. **(m) Author Address**
32. **Image**
33. **Caption**, Image Caption
34. **(m) Custom 1**, User1
35. **(m) Custom 2**, User2
36. **(m) Custom 3**, User3
37. **(m) Custom 4**, User4
38. **ID**, (string, unique signature of the publication. Can be edited, provided it remains unique)
39. **Accession Number**
40. **Call Number**

Field names marked with a **(m)** in the list below admit multiple values, separated by the ";" character.

Multiple values have two influences :

- on the way information is entered while editing publications (multiple choice drop down menus in the graphical user interface),
- on the way bibliographic indexes are built : if a publication has several values specified in a multiple field, and if that field is used to sort/group the bibliographic index, the publication will appear several times in the index : once per value that was extracted from the multiple field.

### 5.9.1.3.2. Mandatory publication fields updated

Two publication fields are mandatory :

Menu **Publications**  
Sub menu **Manage publications ...**

• The title.

• The ID (proposed by WIT, but user editable),

**Note** : each publication is internally identified by a unique string, its ID. The links made from Ideas to the publication may be broken if, later, the publication is removed from the bibliography or if its ID is changed.

### 5.9.1.4. Modify a publication updated

Menu **Publications**  
Sub menu **Manage publications ...**

This function operates from current selection, in the book manager, by clicking on the *"Modify"* button, or by double clicking on a book.

It opens the publication edit dialog.

**Note** : 'Title' and 'Author' fields are mandatory. Modifying one of them changes the "signature" of the current publication and can break the links that are made from the ideas.

Fields	Values
1 Reference Type	
2 <b>Author</b>	CMMI Product Development Team.
3 Secondary Author	
4 Tertiary Author	-10
5 Subsidiary Author	
6 <b>Title</b>	ARC V1.0, Assessment Requirements for CMMI, Version 1.0 (CMU/SEI-2000-TR-0...
7 Short Title	SEI 00e
8 Alternate Title	
9 Secondary Title	SR148
10 Tertiary Title	
11 Publisher	Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University
12 Place Published	
13 Edition	
14 Original Publication	
15 Reprint Edition	
16 Volume	
17 Number of Volumes	
18 Number	
19 Year	2000
20 Date	
21 Pages	0
22 Section	
23 Keywords	CMM
24 Medium	
25 ISBN/ISSN	
26 Reviewed Item	
27 Label	
28 Abstract	
29 Notes	
30 URL	http://www.sei.cmu.edu/publications/documents/00.reports/00tr011.html
31 Author Address	
32 Image	
33 Caption	
34 Custom 1	
35 Custom 2	
36 Custom 3	
37 Custom 4	
38 Custom 5	
39 Accession Number	
40 Call Number	

Book editor

### 5.9.1.5. Remove a publication.

This function operates from current selection , in the book manager, by clicking on the "Remove" button.

Menu **Publications**  
Sub menu **Manage publications ...**

You can select multiple publications (using the Shift + Left Click or Control + Left Click combinations).

After confirmation, the selected books are removed from the publication list.

### 5.9.2. Open a new list of publications

This function loads a list of publications (\*.pbl files).

Menu **Publications**  
Sub menu **Open a new list**  
Feature **Not in freeware edition**

Caution : this list *replaces* the current publication list (a confirmation is therefore required).

### 5.9.3. Add a list of publications

This function adds a list of publications (\*.pbl file) to the current list.

Menu **Publications**  
Sub menu **Add a list...**  
Feature **Not in freeware edition**

**Note :** if the added list contains already referenced publications (author + title), these publications will be ignored.

### 5.9.4. Save a list of publications

This function saves all or part of current list of publications.

Menu **Publications**  
Sub menu **Save a list...**  
Feature **Not in freeware edition**

A single theme can be saved (type it in the publication manager, and corresponding books will be saved).

#### 5.9.4.1. Save all publications

When the book manager is displayed, no theme is selected.

Menu **Publications**  
Sub menu **Save a list...**  
Feature **Not in freeware edition**

Clicking the "OK" button without selecting a theme saves all the list of publications in a \*.pbl file.

#### 5.9.4.2. Save a publication theme

If a theme is selected (or partially typed), the list of publications is filtered according to this criterion.

Menu **Publications**  
Sub menu **Save a list...**

Clicking on the "OK" button saves the filtered list of publications on a \*.pbl file.

Feature **Not in freeware edition**

So you can save the publications theme by theme, and share them between documents.

## 5.9.5. Import / Export publications

You can import / export publications from most common bibliographic formats, CSV and tagged formats.

Menu **Publications**

### 5.9.5.1. Import a list from a CSV text file.

Menu **Publications**  
Sub menu **Import a list (CSV text)...**

This function helps to import a large number of publications from a CSV text (character delimited) file which format is known : the separator used, and the meaning and order of each field.

Such a file can be produced by a database or a spreadsheet, and eventually formatted in the purpose of its importation by WIT.

To import books, follow the steps below :

1. Prepare a data file containing the records (for example, *title:author:publisher:subject:isbn:edition\_year*).
2. Invoke import books function (in the *Publications* menu).
3. In the dialog box (figure 4.24), define the fields to be used, the order in which they will be read to match the file records, and set the separating character (delimiter).
4. Validate the choice of import format or correct it. Books are then imported.

Note : before importing the data, you may save your current format for a further use. The WIT \_INSTALLDIR\_Data\DataFormats may be a good place to store your import formats.

Note : you can specify a same target field several times : in this case, imported values will be concatenated to the associated field.



Fig 4.24 : Import a list of publications

### 5.9.5.2. Import a list from a tagged text file.

Menu **Publications**  
Sub menu **Import a list (tagged text)...**  
Feature **Not in freeware edition**

This function imports a large number of publications from a tagged text file (multiline records which values are prefixed by field names).

Such a file can be produced by a database or exported by most bibliographic format softwares, and eventually formatted in the purpose of its importation by WIT.

WIT analyzes the data according to rules which give high flexibility :

- WIT will recognize all fields names (and their aliases) mapped to **WIT publication fields**.
- A new book is created when the "Reference type" pattern (or one of its aliases) is encountered, or when a blank line is detected (if the "Blank lines end records" option is checked).
- For other lines :
  - when analyzing "name:value" tagged like lines, if "name" matches with a WIT bibliographic field, "value" is stored in that field,
  - if it does not match (or there's only a value), the "value" is appended in the current WIT field.
- Except for the first "Reference type" line of each record, other lines order is not important.
- The separator between tag and value is made of one or several characters (: by default).

If you specify multiple aliases for a WIT bibliographic field, all corresponding values (if encountered while parsing) will be appended to this field, with an embedded "string separator" that you may specify (if you wish to insert a new line : type <br>, since the final code will be HTML).

The "aI=A" checkboxes indicates for each WIT field whether the tags should be analyzed in a case sensitive or insensitive manner.

For example, the information below is fully analysable :

```
REFERENCE TYPE:
AUTHOR: CMMI Product Development Team.
TITLE: CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, Version 1.02 Continuous Representation (CMU/SEI-2000-TR-031, ESC-TR-2000-096).
URL: http://www.sei.cmu.edu/publications/documents/00.reports/00tr031.html
SHORT TITLE: SEI ood
```

PUBLISHER: Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University  
 YEAR: 2000  
 KEYWORDS: CMM  
 CMMI  
 TERTIARY TITLE: SR147

REFERENCE TYPE:  
 AUTHOR: Juran, J. M.  
 TITLE: Juran on Planning for Quality.  
 SHORT TITLE: Juran 88  
 PUBLISHER: New York, New York: MacMillan  
 YEAR: 1998  
 KEYWORDS: Quality  
 TERTIARY TITLE: SR134

REFERENCE TYPE:  
 AUTHOR: Federal Aviation Administration.  
 TITLE: Integrated Capability Maturity Model, Version 1.0.  
 SHORT TITLE: FAA 97  
 YEAR: 1997  
 KEYWORDS: CMM  
 TERTIARY TITLE: SR109  
 URL: <http://www.faa.gov/aio/ProcessEngr/CMM/index.htm>

You may save or load your tagged import formats, and will find some predefined ones in the \_INSTALLDIR\Data\DataFormats folder.

Tip : for each given tag separator, you can specify multiple import formats in the same filter, since you can map as many tags to WIT fields as you want. For instance, for the WIT Title fields, you could associate the NASA Astrophysic Data System "%T" title, and the Refworks © "T1" title.

Configure the tagged import format : map fields to WIT default bibliographic fields

See also :

► [WIT publication fields updated](#)

Documents :

 [Sample : bibliographic name:value text file.](#)

### 5.9.5.2.1. Customize default WIT analysis when importing bibliographic data

This is for advanced users.

Feature **Not** in freeware edition

WIT offers 40 pre-defined bibliographic fields, and aliases to some of them.

You can define your own aliases (if you need to analyse lines yet ignored by WIT), and map them to WIT fields by appending them to the current alias list of the chosen field.

Please refer the configuration file (located in \_INSTALLDIR\Data\DataFormats, named "name-value-format.txt", and listed below) for more explanations.

# WIT bibliography import format specification for files made of name:value couples (one per line, like below) # Example of recognized format (except the #): # Reference type: Memoir # Author: Smith Bill # Title: The story of ... # Publisher : Phantom # Country : England # Year: 2000 # Keywords : keyword1# keyword2 # keyword3# Notes : my opinion... ## Reference type: Article # Author: Dupond & Dupont # Title: ... ## ... etc # # A new book is detected when analyzing the 'Reference type:' field. # ..... ## Lines beginning with # or void are comments (ignored) ## # Below line is the separator between names and values @SEPARATOR : # Next 40 lines are WIT bibliographic field names and their aliases # You can add your own aliases (use the , (comma) to separate them). # DO NOT change the first name of each line (uppercase). # DO NOT change the order of lines. # Leading/trailing white spaces are ignored. # Comparison between this field names and those of your import files are case insensitive. # While parsing the bibliography, if WIT encounters a line without name (just the value, it will be appended to the last # value of the current field, and separated by a comma. REFERENCE TYPE, publication type, book, manuscript, article, report, thesis, communication, artwork, audiovisual material, film, electronic source AUTHOR, reporter, programmer, writer, artist, cartographer, created by TITLE, name of act, case name SHORT TITLE, abbreviated title, abbreviated case name ALTERNATE TITLE, reference PUBLISHER, university, institution, producer, distributor, assignee, court PLACE PUBLISHED, city, conference location, country EDITION, version, session ORIGINAL PUBLICATION, history REPRINT EDITION

VOLUME, code volume, code number, report volume  
 NUMBER OF VOLUMES, extent of work  
 NUMBER, issue, issue number, document number, bill number, public law number  
 YEAR, year of publication, year of conference, year released  
 DATE, date of creation  
 PAGES, number of pages, description, code pages, first page  
 SECTION, code section, sections  
 KEYWORDS, theme  
 SECONDARY AUTHOR, series editor, recipient, director, committee, reporter  
 SECONDARY TITLE, journal, series title, book title, collection title, magazine, newspaper, conference name, academic department, published source, image source program  
 TERTIARY AUTHOR, series editor, producer  
 TERTIARY TITLE, series title, legislative body  
 SUBSIDIARY AUTHOR, translator, performers, sponsor, counsel  
 MEDIUM, type of work, type of article, type, type of medium, medium  
 ISBN/ISSN, isbn  
 REVIEWED ITEM, review rating  
 LABEL  
 ABSTRACT  
 NOTES  
 URL  
 AUTHOR ADDRESS  
 IMAGE  
 CAPTION, image caption  
 CUSTOM 1, user1  
 CUSTOM 2, user2  
 CUSTOM 3, user3  
 CUSTOM 4, user4  
 CUSTOM 5, user5  
 ACCESSION NUMBER  
 CALL NUMBER

Note : if the file is missing or corrupted, WIT will warn you at startup, and revert to default behavior (as specified here).

### 5.9.5.3. Export a list (CSV text file)

Menu **Publications**  
 Sub menu **Export a list (CSV text)**  
 Feature **Not in freeware edition**

This exports all the publication list, using all WIT 40 fields, and separating them by a '|' character, so you can use the bibliographic data in a spreadsheet or in any bibliographic format software.

### 5.9.5.4. Export a list (tagged text file)

Menu **Publications**  
 Sub menu **Export a list (tagged text)**  
 Feature **Not in freeware edition**

This exports all the publication list, so you can use the bibliographic data in a spreadsheet or in any bibliographic format software.

## 5.10. Structure templates

The web site is generated from the current document, according to a structure template.

Menu **Templates**  
 Sub menu **Structures**

This template is chosen in a few mouse clicks, among predefined templates.

In expert mode, this template can be entirely defined with graphically modifiable properties.

The template is made of imbricated structures which control the order and the display of information related to :

- the site itself (home pages, frames organization...),
- the pages (headers, text bodies, footers...),
- the content of each idea (choice and order of elements to be displayed),
- the navigation bar to move in the site,
- the publications,
- the tables of contents,
- and the indexes.

Whenever you click on an element of the template, WIT displays a contextual button on the bottom of the dialog box : this button helps you to edit to corresponding CSS style element from the style template, if needed.

### 5.10.1. Load a structure template

This function loads a new template from a file (\*.emd suffix) and replaces the current template, after confirmation.

Menu **Templates**  
 Sub menu **Structures/Load**  
 Feature **Not in freeware edition**

There are two access ways :

- from the menu "*Templates/Structure/load...*",
- from the left side toolbar, in the "*Structure Templates*" tab, by directly clicking on a template.

The menu lets loading a template whatever its hard disk location.

The "*Structure Templates*" tab in the left side toolbar only presents predefined templates that are stored in the special "*Data/Templates/Structures*" directory (subdirectory of MyIde@tree installation directory).

Note : in the "*Operations*" tab of left side toolbar, a click on "*Modify structure*" activates the "*Structure Templates*" tab, in normal mode. In expert mode, the same click opens the structure edit dialog box.

Note 2 : The name of the template is displayed in the dialog title. A "\*" is appended when any of its options is changed.

See also :  
 ► Versions 5.0, 5.01

### 5.10.2. Save current structure template (expert mode)

This function is available in expert mode only.

It saves current template parameters in a file (\*.emd suffix).

Menu **Templates**  
Sub menu **Structures/Save**  
Feature **Not in freeware edition**

Note : If the template is saved into the "*Data/Templates/Structures*" directory (located in the WIT installation directory), it is added to the predefined templates list, and will be displayed in the "*Structure Templates*" tab of the left side toolbar the next time WIT is launched.

Note 2 : The name of the template is updated in the dialog title. The eventual "\*\*\*" is removed until any of its options is changed.

### 5.10.3. Modify current structure template (expert mode)

This function is available in expert mode only, and modifies current structure template properties.

Menu **Templates**  
Sub menu **Structures/Modify**

It can be used from the "*Templates/Structures/Modify...*" menu or en when clicking on the "*Modify page structure*" button in the "*Operations*" tab of left side toolbar.

It displays the tabbed property dialog box of structure templates.

#### 5.10.3.1. General options tab (web site)

First choice in information structuration deals with Web site general structure (figure 4.25).

Menu **Templates**  
Sub menu **Structures/Modify**  
Tab **General**  
Feature **Not in freeware edition**

**Home page.**

- WIT proposes a default home page (its default HTML code is located in the *Data/Templates/homePage-en.html* file).
- Any prepared HTML file, produced with a professional HTML editor, can also be used : it should however contain an HTTP link such as **href="text/index.html"**, to access properly to the automatically generated pages.

**General site screens organization.**

- "Do not use frames": the first site page is the home page. Other pages structure is defined in the "*Page*" tab.
- "Define a frame for a rapid access table" (left frame for table of contents, or shortcuts). Frame border may be transparent. Frame width (in pixels) can be changed. A default table of contents (a Java applet) can be used : in this case, its size can be modified, and its properties also (background and foreground colors, background image). A prepared HTML file, produced with a professional HTML editor, can be used instead of the Java applet (you will have to specify your shortcuts).
- "Define a title zone" (using a top frame). Frame border may be transparent. Frame height (in pixels) can be changed. A default title can be specified (in this case, it may be not useful to use a textual page header (general title) on each page). Display properties (background image, background and foreground colors) can be modified. A prepared HTML file, produced with a professional HTML editor, can be used instead of default title.

**Important remark for prepared HTML files.**

These files are produced with external HTML editing tools and may use specific resources (images, applets, data files, etc...).

The best practice is to place them, with their resources, in a special subdir named "*extra*" that you must create in the directory where you save your current document.

They will so be taken in account with their resources, at web site creation time.

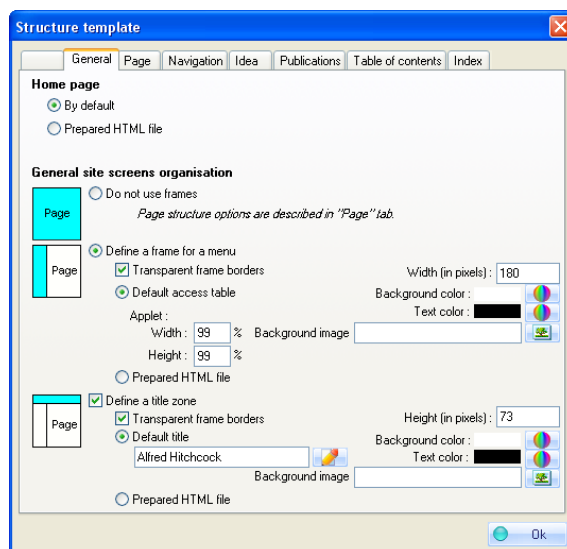


Fig 4.25 : Site general structure options

See also :

- ▶ Check an idea for table of contents [updated!](#)
- ▶ Simple Tree Menu widget [updated!](#)

#### 5.10.3.2. Page tab

This tab controls the structure of each HTML page (figure 4.26).

Menu **Templates**  
Sub menu **Structures/Modify**  
Tab **Page**



#### HTML Page header

This options are related to HTML code instrumentation for navigators and internet search engines (Meta Tags) : title insertion (HTML tag *title* in the header), and keywords (HTML tag meta name="keywords"...). Keywords are those defined in the ideas.

#### Page header.

This option generates a textual header in each HTML page and an optional horizontal ruler. This option may be deactivated if a top frame already contains a title (see the template "General" tab).

Text of page header may be multiline and include HML tags. A window lets you edit the text or the HTML code (see below link).

In the same manner, left side and right side panels can be generated in each web page.

#### Idea organization.

This option controls how ideas are distributed in the pages : a tree structure can be used (one idea per page), or, ideas can be grouped with their children on a same page, from a parametrized hierarchical level.

#### Page footer.

A navigation bar (parametrized in the "Navigation" tab) can be generated. It can be followed by a separator (horizontal ruler), and by a footer text (for example, the date of last update).

Page footer may be multiline and include HML tags. A window lets you edit the text or the HTML code (see below link).

#### Page background.

These options controls the colors or background images. They can be superseded by local disply options (in idea properties), or by style template specifications (specially in the Body style class).

#### Separators.

This options controls how horizontal rulers will be rendered : a simple HTML ruler tag ("hr"), or an image.

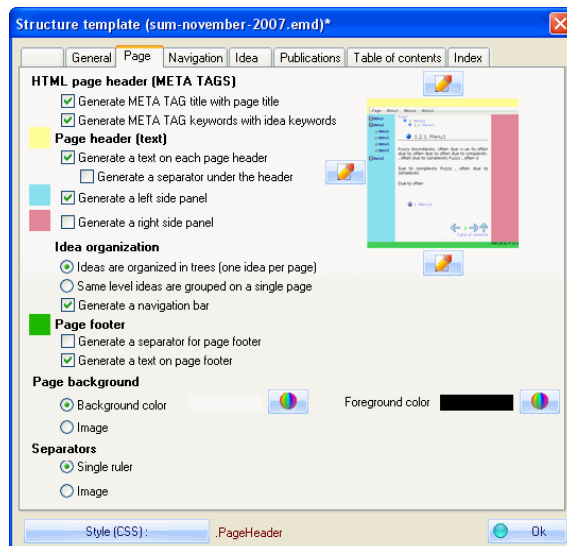


Fig 4.26 : Structure template, page tab

See also :

► [Style class : Body \(document body\)](#)

### 5.10.3.2.1. Header / footer and left/right panels text editing

Header or footer, as well as left or right panel texts can be typed in this window.

Text can be simple and in a single line (most frequent case).

It can also be multiline and formatted thanks to the [WYSIWYG editor](#), and tuned with HTML tags (expert users). In this case, the tags must be well formed (correctly closed and without syntax errors).

WIT lets you insert internal links to its own document pages : click on the "Add internal links" button, and in the idea selector, select one or more ideas and validate : HTML code is inserted and you may modify it : for example, add spaces or new lines to build a menu.

NOTE :

This function is useful if you don't want to use frames (which are less and less used), and have a general menu in each page.

Click on the link below to know how to proceed.

See also :

► [Insert internal links](#)

### 5.10.3.3. Navigation tab

This tab controls the navigation bar properties (figure 4.27). The navigation bar gives access to previous, next, or parent pages of current page. It

Menu [Templates](#)  
Sub menu [Structures/Modify](#)  
Tab [Page](#)  
Feature [Not in freeware edition](#)

Menu [Templates](#)  
Sub menu [Structures/Modify](#)  
Tab [Navigation](#)


may also have links to indexes, and separators.

 Separator above navigation bar.


The separator is an horizontal ruler or an image, as described in the "Page" tab.

 Previous page.


An image may be used to materialize access to previous page (usually a left arrow). A text can be associated to this image : it will be displayed in a yellow rectangle when the mouse pointer is moved over the image. More over, this text can be displayed at the image side.

 Centered page numbers.


Current page number and numbers of sibling pages are displayed as links. When there is more than 20 pages, the number of current page related to all its siblings is shown.

 Next page.


An image may be used to materialize access to next page (usually a right arrow). A text can be associated to this image : it will be displayed in a yellow rectangle when the mouse pointer is moved over the image. More over, this text can be displayed at the image side.

 Parent page.

An image may be used to materialize access to parent page (usually an up arrow). A text can be associated to this image : it will be displayed in a yellow rectangle when the mouse pointer is moved over the image. More over, this text can be displayed at the image side.

 Separator above index access bar.

The separator is an horizontal ruler or an image, as described in the "Page" tab.

 Separator above index access bar.

These options define a second line of optional links in the navigation bar : access to the table of contents, to the bibliography, to the glossary, to the keywords index, and too the faceted index created from custom fields. Each time an option is checked, the link text is displayed : it may be modified.

**Note :** you should activate links to indexes only when you have the corresponding indexes. For example, don't activate the glossary link if you did not define any glossary term.

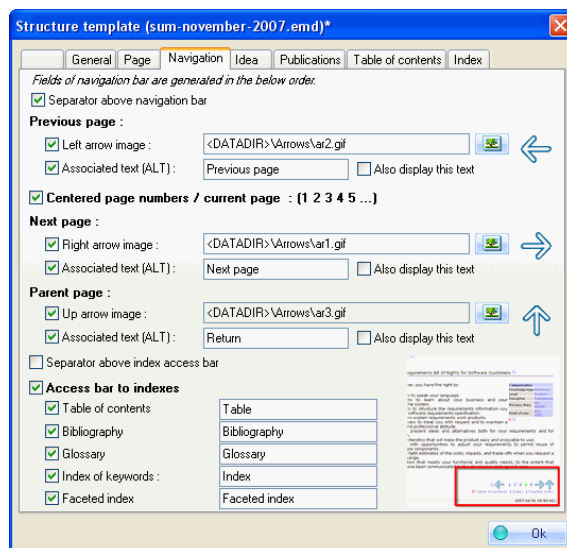


Fig 4.27 : Structure template, Navigation tab

See also :

► Custom fields tab [updated!](#)

### 5.10.3.4. Idea tab

Menu [Templates](#)  
Sub menu [Structures/Modify](#)  
Tab [Idea](#)  
Feature [Not in freeware edition](#)

This tab defines how and in which order information contained in ideas (structures or fields) will be displayed (figure 4.28).

Fields to display are chosen from the left list (in the available fields) : selected fields are added in the right list.

To modify the display order of selected fields, in the right list, select a field and click on the "Up" or "Down" buttons.

Idea elements will be generated in the order shown in the right list.

Each structure has options : to modify them, select the field in the right list, and press the "Options" button (or double click).

You may separately load / save the idea structure parameters in specific files (for instance if you want to retrieve them once you have switched to another template).

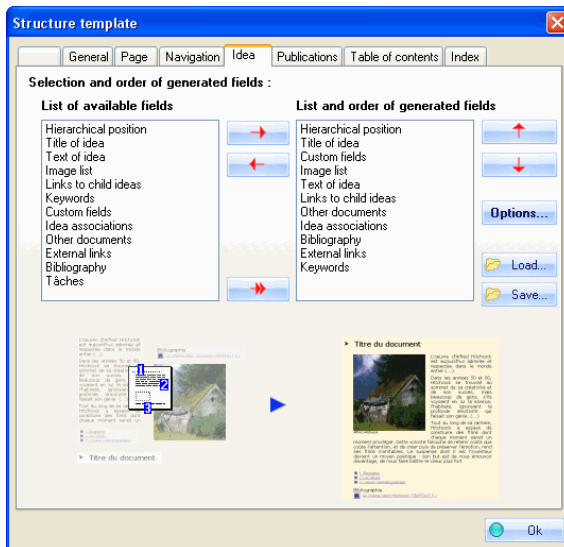


Fig 4.28 : Structure template, idea tab

#### 5.10.3.4.1. Structure : Hierarchical position

Menu [Templates](#)  
 Sub menu [Structures/Modify](#)  
 Tab [Idea](#)  
 Feature [Not in freeware edition](#)

Hierarchical position indicates the place of current idea in the site map : path of parent ideas, from the document root to current idea.

##### Options

- Subtitle first : a subtitle may be displayed before the path of parent ideas (for example : *Map* ).
- Apply title numbering : the title automatic numbering may be used with each element of the path.
- Display once per page : position can be displayed for each idea, or just once per page (useful when pages group ideas of same hierarchical level).
- Small icon first : path elements may use a graphical bullet (select an image).
- Position displayed like a path : idea hierarchy is displayed as a file path with slashes (/), or any other separator that can be specified.
- Position displayed like an indented list : each path element is displayed on a new line, with an indentation depending on the hierarchical level.

See also :

- Style class : HierarchyGroup (web site hierarchical position group)

#### 5.10.3.4.2. Structure : Title

Title is idea title. Title level is idea level. Root idea has 0 level.

Menu [Templates](#)  
 Sub menu [Structures/Modify](#)  
 Tab [Idea](#)  
 Feature [Not in freeware edition](#)

##### Options

- Title numbering. An automatic numbering, up to a modifiable level, can be applied. This numbering can be simple (a single number), or hierarchical (using upper levels numbers, (example : 1.2.2.)).
- Small icon before title. Each title (and its eventual numbers), can have a bullet made with a image (specify the image).

See also :

- Style class : H1 to H6 (titles from level 1 to 6)

#### 5.10.3.4.3. Structure : Text

Text is idea description.

Menu [Templates](#)  
 Sub menu [Structures/Modify](#)  
 Tab [Idea](#)  
 Feature [Not in freeware edition](#)

The only text option is the choice of the bullet image, optionally used in the text : select an image.

See also :

- Style class : TextGroup (text paragraphs group)

#### 5.10.3.4.4. Structure : Images

Menu [Templates](#)  
 Sub menu [Structures/Modify](#)  
 Tab [Idea](#)  
 Feature [Not in freeware edition](#)

Image list displays the images and comments (legends or captions) which were associated to the idea (in the "Links" tab of idea properties).

##### Options

- Position of caption. The legend can be placed beneath or above the image. It can also be ignored.

Note : the caption also appears in a yellow rectangle when the mouse pointer is moved over the image.

See also :  
► Style class : ImageGroup (images group)

#### 5.10.3.4.5. Structure : Links to child ideas

Menu    Templates  
Sub menu Structures/Modify  
Tab    Idea  
Feature Not in freeware edition

Links to child ideas are a major feature of site navigability (if omitted, the only way to explore the site is to use the table of contents, provided it is complete).

##### Options

- ☑ Subtitle first. A subtitle may be displayed before the list of links (by default : *Links to child ideas*).
- ☑ Apply title numbering. Title numbering, with its current options, can be applied.
- ☑ Use an icon before the link. A graphical bullet can be used before each link : select an image.

See also :  
► Style class : ChildrenLinkGroup (links to child ideas group)

#### 5.10.3.4.6. Structure : Keywords

Menu    Templates  
Sub menu Structures/Modify  
Tab    Idea  
Feature Not in freeware edition

Keywords are used both in keyword index construction, and in the hidden HTML code (meta tags) for the internet search engines.

##### Options

- ☑ Presentation as a flat list. The keywords are displayed as an enumeration, words separated by commas.
- ☑ Presentation in two columns. Keywords are displayed in a two column table.
- ☑ Use a subtitle first. A subtitle may be displayed before the enumeration list or the table (by default *Keywords*).

See also :  
► Style class : KeywordsGroup (keywords group)

#### 5.10.3.4.7. Structure : Custom fields

Custom fields are additional characterizations of ideas. They are displayed only when they have non void values.

Menu    Templates  
Sub menu Structures/Modify  
Tab    Idea  
Feature Not in freeware edition

##### Options

- ☑ Display also the field name. Name of each custom field may be displayed before the field value.
- ☑ Subtitle first. A subtitle may be displayed before the set of custom fields and their values (default : *Custom fields*).

See also :  
► Style class : CustomFieldsGroup (custom fields group) **updated!**

#### 5.10.3.4.8. Structure : Tasks

Task fields are additional characterizations of ideas. They are displayed only when they have non void values.

Menu    Templates  
Sub menu Structures/Modify  
Tab    Idea  
Feature Not in freeware edition

##### Options

- ☑ Display also the field name. Name of each custom field may be displayed before the field value.
- ☑ Subtitle first. A subtitle may be displayed before the set of custom fields and their values (default : *Task fields*).

See also :  
► Versions 5.0, 5.01  
► Tasks tab **updated!**  
► Style class : TasksGroup (task fields group)

#### 5.10.3.4.9. Structure : Idea associations

Menu    Templates  
Sub menu Structures/Modify  
Tab    Idea  
Feature Not in freeware edition

The list of idea associations groups all idea associations of the current idea : links to other ideas in the same document. It is another important feature of web site navigability.

##### Options

- ☑ Subtitle first. A subtitle may be displayed before the list of associations (default : *See also ...*).
- ☑ Note label. When the associated idea is a bibliography note (an idea of bibliography type), an additional label may be displayed before the link (default : *Note* :).
- ☑ Glossary label. When the associated idea is a glossary definition (an idea of glossary type), an additional label may be displayed before the link (default : *Definition* :).
- ☑ Use an icon before the link. A graphical bullet may be displayed before each association link. Specify an image.

See also :

► Style class : AssociationsGroup (idea associations group)

#### 5.10.3.4.10. Structure : Links to documents

Links to local documents are grouped into a list of links.

Menu **Templates**  
Sub menu **Structures/Modify**  
Tab **Idea**  
Feature **Not in freeware edition**

##### Options

🔗 Subtitle first. A subtitle may be displayed before the list of links (default : *Other documents :*).

🔗 Use an icon before the link. A graphical bullet may be displayed before each link. Specify an image.

See also :

► Style class : DocumentsGroup (references to documents group)

#### 5.10.3.4.11. Structure : External links (URLs)

External links (external HTML pages, located on the Web) are grouped into a list of links.

Menu **Templates**  
Sub menu **Structures/Modify**  
Tab **Idea**  
Feature **Not in freeware edition**

##### Options

🔗 Subtitle first. A subtitle may be displayed before the list of links (default : *External links (URLs)*).

🔗 Use an icon before the link. A graphical bullet may be displayed before each link. Specify an image.

See also :

► Style class : UrlsGroup (URLs group)

#### 5.10.3.4.12. Structure : Bibliography note

The list of publications which are associated to the current idea is displayed, either as a list of links to the publications, either as a list of the publications themselves. In this later case, the structure of publications is controlled by the "*Books*" tab of the structure template.

Menu **Templates**  
Sub menu **Structures/Modify**  
Tab **Idea**  
Feature **Not in freeware edition**

##### Options

🔗 Subtitle first. A subtitle may be displayed before the list of bibliography notes (default : *Bibliography : (URLs)*).

🔗 Use an icon before the link. A graphical bullet may be displayed before each link or publication. Specify an image.

🔗 Display links to books. The bibliographic list is made of links to books : the publications are displayed in separate pages.

🔗 Display books. To the contrary, the books themselves are displayed according to the options of the "*Books*" tab of the structure template.

See also :

► Style class : NoteGroup (Bibliography note group)

► Style class : BookGroup (publication presentation group)

#### 5.10.3.5. Publications tab updated!

This tab defines how and in which order information contained in publications is displayed.

Menu **Templates**  
Sub menu **Structures/Modify**  
Tab **Idea**  
Feature **Not in freeware edition**

Fields to display are chosen from the left list (in the available fields) : selected fields are added in the right list.

To modify the display order of selected fields, in the right list, select a field and click on the "*Up*" or "*Down*" buttons.

Idea elements will be generated in the order shown in the right list.

Each structure has options : to modify them, select the field in the right list, and press the "*Options*" button (or double click). Publication structure elements are :

1. Reference Type
2. (m) Author
3. (m) Secondary Author
4. (m) Tertiary Author
5. (m) Subsidiary Author
6. Title
7. Short Title
8. Alternate Title
9. Secondary Title
10. Tertiary Title
11. (m) Publisher
12. (m) Place Published
13. Edition
14. Original Publication
15. Reprint Edition
16. Volume
17. Number Of Volumes
18. Number
19. (m) Year
20. Date
21. Pages
22. Section
23. (m) Keywords
24. Medium
25. (m) Isbn/Issn
26. Reviewed Item
27. Label
28. Abstract

29. Notes
30. Url
31. (m) Author Address
32. Image
33. Caption
34. (m) Custom 1
35. (m) Custom 2
36. (m) Custom 3
37. (m) Custom 4
38. ID
39. Accession Number
40. Call Number

Fields marked with (m) are multiple value fields : please refer to [WIT publication fields](#) for more information.

You may separately load / save the publication structure parameters in specific files (for instance if you want to retrieve them once you have switched to another template).

WIT is shipped with few academic format samples : try them.

See also :

- ▶ Mandatory publication fields [updated](#)
- ▶ WIT publication fields [updated](#)
- ▶ Style class : NoteGroup (Bibliography note group)
- ▶ Publications manager
- ▶ Publications
- ▶ Create a publication

#### 5.10.3.5.1. Common options of publication fields [updated](#)

Menu	Templates
Sub menu	Structures/Modify
Tab	Idea
Feature	Not in freeware edition

All the publication structure fields have the same options, intended to produce an homogeneous and concise presentation.

##### Options

- Prefix. A subtitle or a special character may be displayed before the field value (default : the field name).
- Separator for multiple values. For the fields wich admit multiple values (see their [list](#)), the default ";" separator string can be replaced by HTML code to change the presentation : for instance, <br> to break the line.
- Suffix. Usualle a field separator (default : HTML tag < br >), but anything else is possible.

By default, a publication is displayed in a single column, with a line break at the end of each field.

To keep consecutive fields on a same line, you may replace the HTML break tag by a dot or a comma, followed by a space, for all these fields besides the last.

#### 5.10.3.6. Table of contents tab

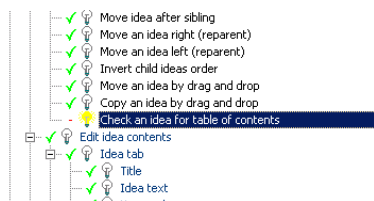
This tab controls the two kinds of table of contents options.

Menu	Templates
Sub menu	Structures/Modify
Tab	Idea
Feature	Not in freeware edition

The first one is generated in a HTML page to which the navigation bar refers.

The second produces a data file to set the left frame Java applet (see the "General" tab of the structure template). Both tables of contents may be identical or differ : they are independent.

Reminder : for a more accurate control of table of contents composition, you may exclude some ideas and their children by unchecking their check boxes in the main view.



##### Main table of contents

- Icon : a graphical bullet may be displayed before titles. Associated image must be specified.
- Numbering : when activated, it may be simple or hierarchical (like titles). Numbering level can also be limited.
- Images table : thumbnails of all images used in the document can be put in a single page (album). Check the option and define thumbnail size. An "Images" entry will be added to navigation bar at Web site build time.

##### Left frame table of contents : Java applet (Ignore If no left frame)

- Show tool tips. Tooltips are shown when the mouse is moved over menu items.
- Show selection icon. An icon is shown on the right of the menu on the last selected item.
- Display level limit. Display level can be limited to a specified hierarchical level.
- Numbering : when activated, it may be simple or hierarchical (like titles). Numbering level can also be limited.
- Font (combo box) : select the font to use in the Java applet menu.

See also :

- ▶ Check an idea for table of contents [updated!](#)
- ▶ Style class : TocGroup (Table of contents group)
- ▶ Build thumbnails automatically

### 5.10.3.7. Index tab

The index tab controls the bibliography, the glossary and the index options.

Menu [Templates](#)  
 Sub menu [Structures/Modify](#)  
 Tab [Idea](#)  
 Feature [Not in freeware edition](#)

#### Bibliography

- ▶ Alphabetical access bar. This option enables an alphabetical list of links corresponding to the first letter of displayed elements in the bibliography (either the author names, either the publication titles). These links are shortcuts.
- ▶ Begin a new page for each new letter (or digit). The bibliography can be grouped in a single page, or use a new page for each new letter or digit (useful for large lists).
- ▶ Display of each complete book. Each book reference is displayed with all its elements, according to the rules defined in the "Books" tab of current structure template.
- ▶ Replace books by a link. Each book reference is replaced by a link, and a specific page is built to describe the book, according to the rules defined in the "Books" tab of current structure template.
- ▶ Display cross references to ideas. For each generated book, a link is generated for each idea containing a reference to the book.
- ▶ Sorting.
  - Bibliography is always sorted (first sort) by any of the bibliographic fields (usually Author, Title or Keyword). Sort is ascending.
  - You can also use a second sort criterion by selecting a second field. For instance, different titles of a same author. In this case, the first sort criterion will appear as a primary header and sorted publications will be grouped under this header.
  - You can also use a third sort criterion by selecting a third field. In this case, the second sort criterion will appear as a secondary header and sorted publications will be grouped under this header.

**Note :** some fields, like "Keywords", "Author", "Secondary Author", "Tertiary Author", "Subsidiary Author", "ISBN", and "URL" may have **multiple values**, delimited either by ";" or ",". When used as **primary sort criterion**, these fields will generate multiple entries in the bibliographic index, and associated publications will be found for each entry.

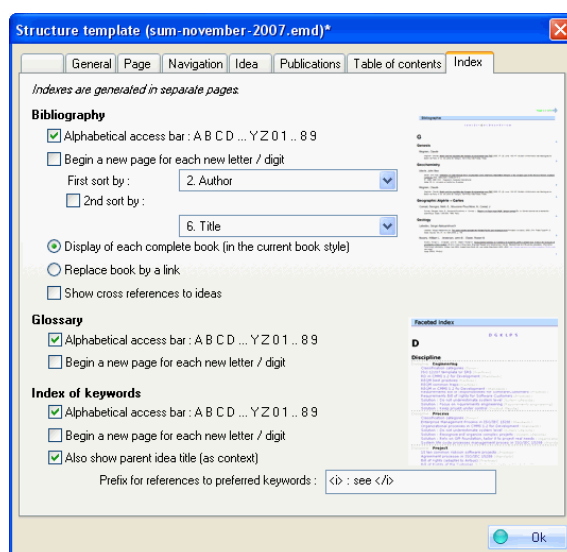
**Tip :** How to create a real bibliographic index ? Select "Keywords" as first sort criterion, "Author" as second, and "Title" as third. Generate the bibliography (Ctrl + Shift + E), and watch the result.

#### Glossary

- ▶ Alphabetical access bar. This option enables an alphabetical list of links corresponding to the first letter of displayed elements in the glossary. These links are shortcuts.
- ▶ Begin a new page for each new letter (or digit). The glossary can be grouped in a single page, or use a new page for each new letter or digit (useful for large lists).

#### Index of keywords

- ▶ Alphabetical access bar. This option enables an alphabetical list of links corresponding to the first letter of displayed elements in the index. These links are shortcuts.
- ▶ Begin a new page for each new letter (or digit). The index can be grouped in a single page, or use a new page for each new letter or digit (useful for large lists).
- ▶ Also show the parent idea title (as context). This is useful to show the context of an idea in the index, especially when different ideas using the same keyword have similar titles. The parent idea title is displayed between parenthesis and with the .IndexKeywordParent CSS style.
- ▶ Prefix for references to preferred keywords. This text is inserted between a keyword and a preferred keyword. Default : <i> : see </i>.



Index tab

See also :

- ▶ Style class : GlossaryGroup (glossary group)
- ▶ Style class : BibliographyGroup (Bibliography group)
- ▶ Style class : IndexGroup (keywords index group) [updated!](#)

## 5.10.4. Other structural elements : Widgets updated

Feature **Not in freeware edition**

As WIT evolves, supplemental structural elements are developed. Widgets mimic the behavior of some elements that are common to blogs : quick access to categories of information, that are dynamically built by WIT as you modify the idea tree.

WIT widgets are introduced in your pages through the use of **WIT variables**.

The picture below shows the use of some widgets in the page header (horizontal drop down menu), and in the page left page (simple tree menu, and custom fields values widgets).



Most of WIT widget need no specific options in the structure template, but have style options in the style template for a better control.

See also :

► [Version 5.10 updated](#)

### 5.10.4.1. Drop Down Menu widget updated

When you insert the **WIT variable** :

- `< @dropDownMenuWidget/ >`

Sub menu **Insert WIT variable**

Keyboard **Ctrl + Alt + V**

Feature **Not in freeware edition**

WIT replaces it at build time by a block containing :

- an horizontal drop down menu, which entries are the three first level ideas of your WIT document (including root idea, shown in the first cell of the menu bar).
- like this :

WIT Manual	News	Foreword	License	Tutorials	Reference	CSS 1	FAQ	
------------	------	----------	---------	-----------	-----------	-------	-----	--

The best places to insert this widget are the page header or footer.

The **presentation styles** of the Drop Down Menu are controlled in the style template.

Notes :

1. Use short titles for the three first levels of ideas, so the menu bar fits on a single line.
2. [Check/uncheck ideas for making them eligible](#) for the menu (as well as for table of contents).
3. When adding new ideas or changing titles, the menu is automatically updated.
4. You can insert only one menu per page (the second one won't show the drop down sub menus).
5. The Javascript for this menu is provided with kind permission of [www.dynamicdrive.com](http://www.dynamicdrive.com) where you can download the original code for Chrome Drop Down Menu (slight adaptations on pathes and CSS styles were necessary for integration into WIT).

See also :

► [Check an idea for table of contents updated](#)

► [Style class : WITChromeMenu updated](#)

### 5.10.4.2. Simple Tree Menu widget updated

When you insert one of the **WIT variables** :

- `< @ treeWidgetLevel1/ >`
- `< @ treeWidgetLevel2/ >`
- `< @ treeWidgetLevel3/ >` ([professional edition only](#))
- `< @ treeWidgetLevel4/ >` ([professional edition only](#))
- `< @ treeWidgetLevelN/ >` ([professional edition only](#))

Sub menu **Insert WIT variable**

Keyboard **Ctrl + Alt + V**

Feature **Not in freeware edition**  
**Professional ed. only**

WIT replaces them at build time by a block containing :

- a tree menu of child ideas, limited to sublevels 1, 2, 3, 4 or N (N = no limit).
- like in this document (left pane).

The best places to insert this widget are the left or right page panes.

If your web browser accepts cookies, the state of the tree is persistent : opened branches remain open as you navigate to new pages.



The [presentation styles](#) of the Simple Tree Menu are controlled in the style template.

Notes :

1. Use short titles for the three first levels of ideas, if you want to avoid text wrapping.
2. [Check/uncheck ideas for making them eligible](#) for the tree menu (as well as for table of contents).
3. When adding new ideas or changing titles, the tree menu is automatically updated.
4. You can insert only one tree menu per page (the second one will be shown with expanded entries and no way to collapse them).
5. The Javascript for this menu is provided with kind permission of [www.dynamicdrive.com](http://www.dynamicdrive.com) where you can download the original code for Simple Tree Menu (slight adaptations on pathes and CSS styles were necessary for integration into WIT).

Possible performance issue on large documents (several hundreds of pages) :

Using deep levels in the tree menu can lead to huge lists of links, slowing the navigation within WIT (since it has to maintain the list), and weighting the generated web pages. To avoid this, you may :

- reduce the depth of your tree,
- uncheck ideas (many of them),
- or use instead the Java Applet menu, which is displayed in a separate frame and is built and loaded once (see the structure template : [generic tab](#)).

See also :

- ▶ [Check an idea for table of contents](#) [updated!](#)
- ▶ [Style class : WITree](#) [updated!](#)

#### 5.10.4.3. Custom fields Table of Contents widgets [updated!](#)

Sub menu [Insert WIT variable](#)  
Keyboard [Ctrl + Alt + V](#)  
Feature [Not in freeware edition](#)  
[Professional ed. only](#)

When you insert one of the [WIT variables](#) <@cfWidgetTocX/> (where X goes from 1 to 32), WIT replaces it at build time by a block containing :

- the custom field X name
- the list of all values that are used by ideas for this field, and the number of ideas using each value.

If the custom field is eligible for faceted index classification, the values are also links that point to the faceted index.

You can see an example of this widget in the present document (left side), where the widget for the custom field "Menu" (#1) and the widget for the custom field "Feature" (#5) were used.

Thus, inserting <@ cfWidgetToc1/> variable places the following block :

```
Menu
"Normal" view (10)
(Main tree view) (1)
Display (4)
Edit (8)
File (45)
Help (7)
Ideas (52)
Import/Export (11)
Keywords (6)
Publications (19)
Tasks (1)
Templates (72)
Windows (1)
```

Inserting <@ cfWidgetToc5/> variable places the following block :

```
Feature
Not in freeware edition (105)
Professional ed. only (33)
```

The widgets are updated whenever you build the web site (Ctrl + E), force the index calculation (Ctrl + Shift + E) or load the document.

The [presentation styles](#) of these widgets are controlled by the style template.

See also :

- ▶ [Style class : CFWidgetToc \(Custom Field Table of Contents Widget\)](#) [updated!](#)
- ▶ [Insert a special WIT variable](#) [updated!](#)
- ▶ [Version 5.10](#) [updated!](#)
- ▶ [Custom fields tab](#) [updated!](#)

#### 5.10.4.4. Task fields Table of Contents widgets [updated!](#)

When you insert one of the [WIT variables](#) :

Sub menu [Insert WIT variable](#)  
Keyboard [Ctrl + Alt + V](#)  
Feature [Not in freeware edition](#)  
[Professional ed. only](#)

- < @tfWidgetStartDate/> ,
- < @tfWidgetDueDate/> ,
- < @tfWidgetPercentFinished/> ,
- < @tfWidgetEstimatedTime/> ,
- < @tfWidgetSpentTime/> ,
- < @tfWidgetEndDate/> ,
- < @tfWidgetStatus/> ,
- < @tfWidgetCategory/> ,
- < @tfWidgetCreatedBy/> ,
- < @tfWidgetRiskPriority/> ,
- < @tfWidgetAssignedTo/> ,

WIT replaces it at build time by a block containing :

- the corresponding task field name (see [idea task attributes](#)),
- and the list of all values that are used by ideas for this field, with the number of ideas using each value.

If the task field is eligible for faceted index classification, the values are also links that point to the faceted index.

Thus, inserting <@ tfWidgetCategory/> variable places the following block :

## Catégorie

V5 (45)  
 V5.01 (21)  
 V5.10 (15)  
 V5.20 (30)

The widgets are updated whenever you build the web site (Ctrl + E), force the index calculation (Ctrl + Shift + E) or load the document.

The [presentation styles](#) of these widgets are controlled by the style template.

## See also :

- ▶ [Style class : TFWidgetToc \(Task Field Table of Contents Widget\)](#) [updated!](#)
- ▶ [Insert a special WIT variable](#) [updated!](#)
- ▶ [Version 5.10](#) [updated!](#)
- ▶ [Tasks tab](#) [updated!](#)
- ▶ [Personal, professional, and freeware licenses](#) [updated!](#)

### 5.10.4.5. Task progress status widget [updated!](#)

When you insert the [WIT variable](#) :

- < @tasksWidgetStatus/> ,

WIT replaces it at build time by a block containing :

- the "Tasks" title (see [idea task attributes](#)),
- and the list of all tasks progress criteria (identical to those shown in the "Tasks" menu), with a count of ideas matching the progress criteria.

If the [task progress criteria](#) are eligible for faceted index classification, they are also links that point to lists of matching ideas in the faceted index.

Thus, inserting <@ tasksWidgetStatus/> variable places the following block :

## Tâches

Non commencées (0)  
 En cours (0)  
 En retard (0)  
 Pour aujourd'hui (0)  
 Pour demain (0)  
 Pour cette semaine (0)  
 Pour la semaine prochaine (0)  
 Pour ce mois-ci (0)  
 Pour le mois prochain (0)  
 Terminées (103)

The widget is updated whenever you build the web site (Ctrl + E), force the index calculation (Ctrl + Shift + E) or load the document.

The [presentation styles](#) of this widget are controlled by the style template.

## See also :

- ▶ [Style class : TFWidgetToc \(Task Field Table of Contents Widget\)](#) [updated!](#)
- ▶ [Insert a special WIT variable](#) [updated!](#)
- ▶ [Version 5.10](#) [updated!](#)
- ▶ [Tasks](#)
- ▶ [Task fields options](#) [updated!](#)
- ▶ [Tasks tab](#) [updated!](#)
- ▶ [Personal, professional, and freeware licenses](#) [updated!](#)

## 5.11. Style templates

The web site is generated from the current document, according to a style template which controls its final look and feel.

This template is chosen in a few mouse clicks, among predefined templates.

In expert mode, this template can be entirely defined with graphically modifiable properties following the style sheets of CSS/CSS2 standard.

The template is made of imbricated elements (classes) in correspondence with the structure template.

**Note :** all user defined CSS elements (non WIT default elements) are displayed on a blue background for easier identification.

### 5.11.1. Load a style template

This function loads a new template from a file (\*.csn suffix) and replaces the current template, after confirmation.

There are two access ways :

- ▶ from the menu "*Templates/Style/load...*",
- ▶ from the left side toolbar, in the "*Style Templates*" tab, by directly clicking on a template.

The menu lets loading a template whatever its hard disk location. The "*Style Templates*" tab in the left side toolbar only presents predefined templates that are stored in the special "*Data/Templates/Style*" directory (subdirectory of WIT installation directory).

**Note :** in the "*Operations*" tab of left side toolbar, a click on "*Modify page style*" activates the "*Style Templates*" tab, in normal mode. In expert mode, the same click opens the structure editing dialog box.

**Note 2 :** The name of the template is displayed in the dialog title. A "\*" is appended when any of its options is changed.

## See also :

- ▶ [Versions 5.0, 5.01](#)

### 5.11.2. Save current style template (expert mode)

This function is available in expert mode only. It saves current template parameters in a file (\*.csn suffix).

Sub menu [Insert WIT variable](#)  
 Keyboard [Ctrl + Alt + V](#)  
 Feature [Not in freeware edition](#)  
[Professional ed. only](#)

Menu [Templates](#)  
 Sub menu [Style](#)  
 Feature [Not in freeware edition](#)

Menu [Templates](#)  
 Sub menu [Style/Load](#)  
 Feature [Not in freeware edition](#)

Menu [Templates](#)

Sub menu **Style/Save**  
Feature **Not in freeware edition**

Note : if the template is saved into the "Data/Templates/Styles" directory (located in the WIT installation directory), it is added to the predefined templates list, and will be displayed in the "Style Templates" tab the next time WIT is launched.

Note 2 : The name of the template is updated in the dialog title. The eventual "" is removed until any of its options is changed.

### 5.11.3. Modify current style template (expert mode)

Menu **Templates**  
Sub menu **Style/Modify**  
Feature **Not in freeware edition**

This function is available in expert mode only, and modifies current structure template properties. It can be used from the "Templates/Structures/Modify..." menu or when clicking on the "Modify page structure" button in the "Operations" tab of left side toolbar. It opens a dialog box with a list of CSS style classes (figure 4.29).

To **export** the current style sheet into a CSS file (\*.css), use the "Export list" button.

To **import** another CSS style sheet, use the "Import list" button. If imported items have the same name as existing ones, the imported values will overwrite the previous ones, so you should have eliminated them from the imported CSS file before importing unless you wish them.

To **modify** an element, select it and click on the "Modify" button, (or double click on the element, on the left column).

This opens another dialog box (figure 4.30), containing a mini text editor which lets you type the CSS specifications of the chosen element.

To **create a new item**, click on the "Add" button : this item will then appear with a blue background in the template, and will be able to be **suppressed** with the "Remove" button.

The "Help on CSS" button opens the user manual at the CSS chapter.

The "Insert a RGB color" inserts a color specification (Red, Green, Blue) from the Windows color selector, and copies the text at the editor insertion point.

Following pages explain how the style template is organized, and how to use CSS specifications.

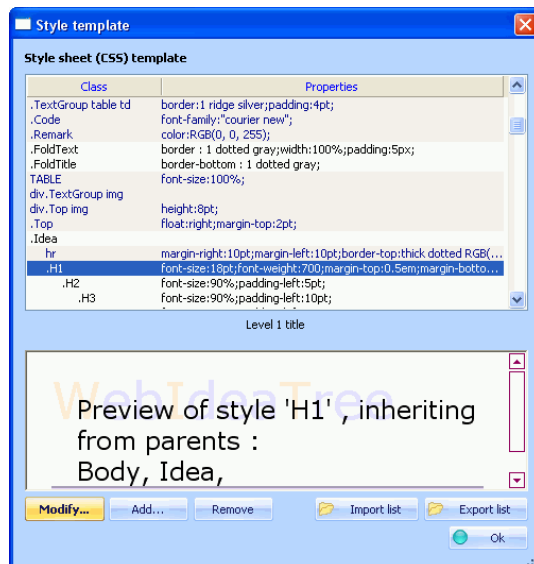


Fig 4.29 : CSS style template

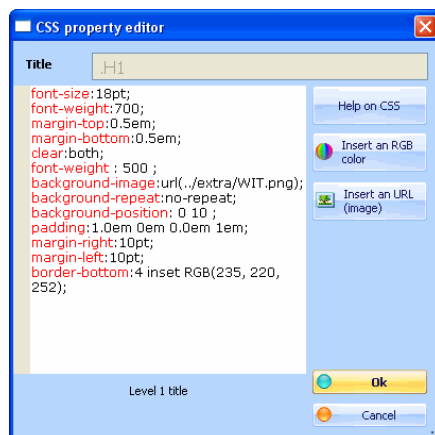


Fig 4.30 : CSS properties editing

See also :  
► Definition : CSS

#### 5.11.3.1. Using cascading style sheets (CSS)

The use of cascading style sheets (CSS) gives an homogeneous presentation of web site pages, with a virtually unlimited control on layout.

The counterpart is the necessity to know a minimum of CSS syntax (see the quick reference on CSS annex). You may learn with samples, by using predefined style templates, and gradually modifying them. By using preview mechanisms and automatically generating web sites, you get immediate effects of the changes to the style template.

CSS properties apply to distinct document elements. This elements are organized in correspondence with elements of structure template : they are named **style classes**. For example, by changing title class style, you control how all document titles that will be displayed, and immediately check the effects thanks to the preview feature.

See also :  
► CSS 1

### 5.11.3.2. Style classes and structure elements

To each structure element (or item) is associated at least one style class.

Menu [Templates](#)  
Sub menu [Style/Modify](#)  
Feature [Not in freeware edition](#)

#### 5.11.3.2.1. Hierarchy and inheritance principles

Style classes may inherit properties from parent classes.

Menu [Templates](#)  
Sub menu [Style/Modify](#)  
Feature [Not in freeware edition](#)

They are displayed in indented lists in the style template (figure 4.31).

First level is associated to general scope classes, such as visited and not yet visited links aspect.

Next levels (with more and more important indentations), correspond to child classes, which may be parent of even more specialized classes, and so on. Sub classes inherit their parent properties : so the most general properties will be defined in the parent classes, and more specialized properties will be defined in the child classes.

Main level 1 style classes are :

- Ideas (Idea class),
- Navigation bar (NavBarGroup class),
- Table of contents (TocGroup class),
- Glossary (GlossaryGroup class),
- Bibliography notes (NoteGroup class),
- Bibliography (BibliographyGroup class),
- Keywords index (IndexGroup class).

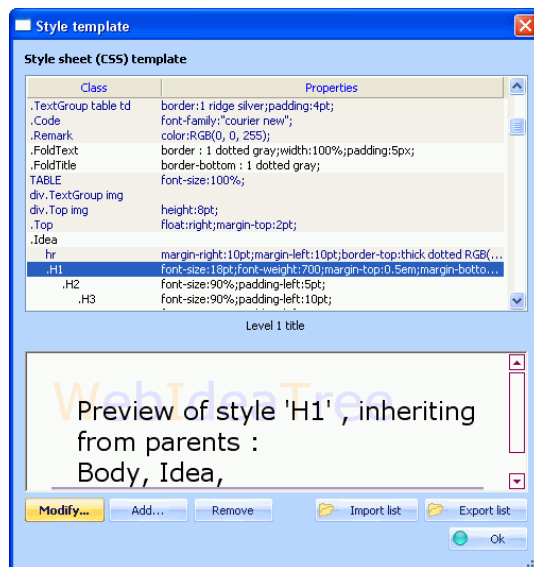


Fig 4.31 : Style classes indentation (inheritage)

See also :  
► Style class : Idea (Idea)  
► Style class : NavBarGroup (navigation bar group)  
► Style class : TocGroup (Table of contents group)  
► Style class : GlossaryGroup (glossary group)  
► Style class : NoteGroup (Bibliography note group)  
► Style class : BibliographyGroup (Bibliography group)  
► Style class : IndexGroup (keywords index group) [updated!](#)

### 5.11.3.3. List of style classes

Next paragraphs give a detailed description of the style classes of the style template.

Menu [Templates](#)  
Sub menu [Style/Modify](#)  
Feature [Not in freeware edition](#)

### 5.11.3.3.1. Style class : Body (document body)

This class is attached to the HTML code BODY tag.

It typically specifies the fonts to use, and except local sub class specification, it applies to all the document.

Example : *font-family: verdana, arial;*

Note : if a foreground or background color is specified in this class, they will have a greater priority than those specified in the structure template ("Idea" tab), or than those locally specified in the idea properties ("Display options" tab).

Menu    Templates  
Sub menu Style/Modify  
Feature   Not in freeware edition

### 5.11.3.3.1.1. Style class : Container

A special HTML division (<DIV id="Container">) is generated in the body to hold all the page content.

This is a placeholder for CSS tweaking, for instance if you'd like to draw a "frame" around your pages (like in this manual).


The scope of this class and child classes below is global to the document.

Menu    Templates  
Sub menu Style/Modify  
Feature   Not in freeware edition


### Style classes : Links

These classes define, for all the document, the link appearance, when they have been visited or not, or when the mouse pointer is moved over them.

Examples :

 class a:active  
*color:rgb(247, 98, 17);*

 class a:link  
*color:rgb(131, 104, 208);*


 class a:visited  
*color:rgb(129, 115, 107);*


Menu    Templates  
Sub menu Style/Modify  
Feature   Not in freeware edition

### Style classes : B (bold), I (italic), U (underlined)

These classes define how, at the document level, the HTML tags B and /B, I and /I, U and /U should be interpreted.

Examples :

 class B  
*font-weight:800;*

 class I  
*font-style:italic;*

 class U  
*text-decoration:underline;*

Menu    Templates  
Sub menu Style/Modify  
Feature   Not in freeware edition

### Style class : table

This class defines the general table presentation and overrides the normal behavior of the HTML tag <TABLE>.

Example :

*font-size:100%;*

This was necessary because the Microsoft Internet Explorer does not inherit correct text size in a table and we had to force it.

Menu    Templates  
Sub menu Style/Modify  
Feature   Not in freeware edition

### Style class : img (image)

Caution : this class controls ALL document images, including bullets and rulers. So you should not specify geometric constraints here (you may use the ImageGroup class and its children for this purpose).

Example :

*vertical-align:middle;*

Menu    Templates  
Sub menu Style/Modify  
Feature   Not in freeware edition

### Style class : Separator (horizontal ruler)

This class controls the style of horizontal rulers. Since the separator is often an image (specified in the structure template), the style may only specify its alignment.

Menu    Templates  
Sub menu Style/Modify  
Feature   Not in freeware edition

**Example :**  
*text-align:center;*

See also :  
 ► Page tab

## ● Style class : PageHeader

This class defines :

- the general title presentation (topmost frame title when activated),
- and the upper page part (when activated).

**Example :**

*text-align:center;*  
*font-size:26pt;*  
*clear:left;*

See also :  
 ► General options tab (web site)  
 ► Page tab

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

## ● Style class : TablePanels

This class is used by a container table used by WIT as a placeholder for structural page left pane, middle pane (this one contains the idea) and right pane elements.

When no side panes (left or right panes) are used, the table is not used too, and .Idea CSS class directly inherits from the div#Container class.

Typical example :

*width:100%;*  
*margin:0px;*

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

## ● Style class : RowPanels

This class is associated to the row of the table used as a placeholder for left panel, right panel and middle panel (idea) cells.

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

## ● Style class : PageLeftPane

This class is associated to the cell of the table used as a placeholder the page left panel, when activated.

**Example :**

*background-color:RGB(213, 255, 255);*  
*width:100px;*  
*vertical-align:top;*

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

## ● Style class : PageRightPane

This class is associated to the cell of the table used as a placeholder the page right panel, when activated.

**Example :**

*background-color:RGB(213, 255, 255);*  
*width:100px;*  
*vertical-align:top;*

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

## ● Style class : PageMiddlePane

This class is associated to the cell of the table used as a placeholder the page middle panel, where the idea is generated.

**Example :**

*vertical-align:top;*  
*margin:0px;*  
*padding:0px;*

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

## ● Style class : Idea (idea)

This class and its children control the presentation of ideas and their components.

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

See also :  
▶ Idea tab

## ● Style class : H1 to H6 (titles from level 1 to 6)

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature   [Not in freeware edition](#)

H1 to H6 classes define the level 1 to 6 title rendering. They are imbricated (inherit from upper level title classes) : for example, properties defined in H2 are known in H3. Level 7 and more titles have level 6 titles style.

### Examples :

🔗 Class H1  
*font-size:20pt;*  
*margin-top:2em;*  
*margin-bottom:1em;*  
*clear:left;*

🔗 Class H2  
*font-size:18pt;*

🔗 Class H3  
*font-size:16pt;*

...

See also :  
▶ Structure : Title

## ● Style class : TextGroup (text paragraphs group)

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature   [Not in freeware edition](#)

This class defines the text group : the set of paragraphs which are the idea description. To each paragraph is associated a sub class : the Text class.

### Example :

🔗 Class TextGroup  
*text-align:justify;*  
*font-size:12pt;*  
*margin-top:1em;*  
*margin-bottom:1em;*  
*margin-left:30pt;*  
*margin-right:30pt;*

🔗 Class Text  
*margin-top:0.5em;*

Text is justified, and aligned between precise margins. And inter paragraph distance is *increased* by half a character.

🔗 Class TextImage  
*float:left;*  
*vertical-align:bottom;*

This property controls the images that are directly inserted into the text. If you want them to be exactly inserted at cursor location, instead of having the text flowing around them, please remove the "float:left;" specification.

See also :  
▶ Structure : Text

## ● Style class : HierarchyGroup (web site hierarchical position group)

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature   [Not in freeware edition](#)

*HierarchyGroup* style class and its children ( *HierarchySubtitle* and *HierarchyPos*) organize the aspect of hierarchical position group.

See also :  
▶ Structure : Hierarchical position

## ● Style class : ChildrenLinkGroup (links to child ideas group)

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature   [Not in freeware edition](#)

*ChildrenLinkGroup* class style and its children ( *ChildrenLinkSubtitle* and *ChildrenLink*) organize the aspect of links to references to child ideas group.

See also :  
▶ Structure : Links to child ideas

## ● Style class : KeywordsGroup (keywords group)

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature   [Not in freeware edition](#)

*KeywordGroup* style class and its child classes ( *KeywordSubtitle* and *Keyword*) control the keyword group style.

See also :  
▶ Structure : Keywords

## ● Style class : ImageGroup (images group)

*ImageGroup* style class controls the style of group of all images associated to an idea.

Each image and its caption (legend) belong to a sub group linked to the *ImageSubgroup* sub class.

This sub class is divided in two parts : associated caption (*ImageText* style class), and the image itself (*Image* style class).

Finally, the "*div.Image img*" class controls geometric constraints off all images associated to ideas (except the bullets and rulers).

For example, to force image width to 200 pixels, you have to type : *width=200*; and the height is computed by the browser.

To force both width and height, type : *width=200; height=150*; (for example).

**Note** : this options helps to normalize images size, but has no effect on their originating files.

See also :

► Structure : Images

Menu    Templates  
Sub menu Style/Modify  
Feature    Not in freeware edition

## ● Style class : CustomFieldsGroup (custom fields group) updated!

*CustomFieldsGroup* style class has 3 main sub classes, controlling the tabular representation of custom fields :

- *CustomFieldsSubtitle* manages associated subtitle style,
- *CustomFieldName* manages custom fields names style,
- and *CustomField* manages custom fields values style.

Two other classes types, CFs and CF1 to CF32, control the display of all fields (CFs class), and the individual display of fields (CF1 to CF32).

**Example :**

🔗 Class CustomFieldsGroup  
*font-size:0.8em;*  
*font-family:arial;*  
*margin-left:30pt;*  
*margin-right:30pt;*  
*color:white;*  
*float:left;*  
*background-color:RGB(74, 128, 193);*

🔗 Class CustomFieldsSubtitle  
*font-weight:800;*

🔗 Class CustomFieldName  
*font-style:italic;*

🔗 Class CustomField  
*font-weight:800;*  
*background-color:RGB(123, 182, 225);*

In such an example, custom fields are stored in a floating table with a blue background, written in white with arial font, with italic letters for field names and bold characters for field values.

See also :

► Structure : Custom fields

► Version 5.10 updated!

Menu    Templates  
Sub menu Style/Modify  
Feature    Not in freeware edition

## ● Style class : CFWidgetToc (Custom Field Table of Contents Widget) updated!

*CFWidgetToc* style class has 2 main sub classes, controlling :

- the title of the block (*CFWidgetTitle*),
- and the list of values (*CFWidgetValues*).

Two *CFWidgetValues* sub-classes (*CFWidgetValue* and *CFWidgetNumber*) control the text value and number of associated ideas.

**Example (this document styles) :**

🔗 Class CFWidgetToc  
*padding-left:5px;*  
*margin:5px;*  
*background-color:RGB(234, 233, 251);*  
*padding-top:5px;*  
*padding-bottom:5px;*  
*font-size:90%;*

🔗 Class CFWidgetTitle

🔗 Class CFWidgetValues  
*font-size:90%;*  
*font-weight:100;*  
*padding-left:10px;*

🔗 Class CFWidgetValue

🔗 Class CFWidgetNb  
*font-size:80%;*

**Note** : the rounded corners are obtained with "NiftyCorners" Javascript script, invoked in the page header; please refer to the Advanced CSS tutorial in this manual.

See also :

► Custom fields Table of Contents widgets updated!

Menu    Templates  
Sub menu Style/Modify  
Feature    Professional ed. only



► Version 5.10 **updated!**

## ● Style class : TasksGroup (task fields group)

Menu    Templates  
Sub menu **Style/Modify**  
Feature    **Not in freeware edition**

*TaskFieldsGroup* style class has 3 sub classes : *TaskFieldsSubtitle* manages associated subtitle style, *TaskFieldName* manages custom fields names style, and *TaskField* manages custom fields values style.

**Example :**

🔗 Class TaskFieldsGroup  
*font-size:0.8em;*  
*font-family:arial;*  
*margin-left:30pt;*  
*margin-right:30pt;*  
*color:white;*  
*float:left;*  
*background-color:RGB(74, 128, 193);*

🔗 Class TaskFieldsSubtitle  
*font-weight:800;*

🔗 Class TaskFieldName  
*font-style:italic;*

🔗 Class TaskField  
*font-weight:800;*  
*background-color:RGB(123, 182, 225);*

In such an example, custom fields are stored in a floating table with a blue background, written in white with arial font, with italic letters for field names and bold characters for field values.

See also :  
► Versions 5.0, 5.01  
► Tasks tab **updated!**

## ● Style class : TFWidgetToc (Task Field Table of Contents Widget) **updated!**

Menu    Templates  
Sub menu **Style/Modify**  
Feature    **Professional ed. only**

*TFWidgetToc* style class has 2 main sub classes, controlling the title of the block (*TFWidgetTitle*), and the list of values (*TFWidgetValues*).

Two *TFWidgetValues* sub-classes (*TFWidgetValue* and *TFWidgetNumber*) control the text value and number of associated ideas.

**Example (this document styles) :**

🔗 Class TFWidgetToc  
*padding-left:5px;*  
*margin:5px;*  
*background-color:RGB(234, 233, 251);*  
*padding-top:5px;*  
*padding-bottom:5px;*  
*font-size:90%;*

🔗 Class TFWidgetTitle

🔗 Class TFWidgetValues  
*font-size:90%;*  
*font-weight:100;*  
*padding-left:10px;*

🔗 Class TFWidgetValue

🔗 Class TFWidgetNb  
*font-size:80%;*

**Note :** the rounded corners are obtained with "NiftyCorners" Javascript script, invoked in the page header; please refer to the Advanced CSS tutorial in this manual.

See also :  
► Version 5.10 **updated!**  
► Task fields Table of Contents widgets **updated!**

## ● Style class : AssociationsGroup (idea associations group)

Menu    Templates  
Sub menu **Style/Modify**  
Feature    **Not in freeware edition**

*AssociationsGroup* style class has two subclasses : *AssociationsSubtitle* manages associated subtitle style, *AssociationLink* manages links style.

See also :  
► Structure : Idea associations

## ● Style class : DocumentsGroup (references to documents group)

Menu    Templates  
Sub menu **Style/Modify**  
Feature    **Not in freeware edition**

*DocumentsGroup* style class has three sub classes : *DocumentsSubtitle* manages associated subtitle style, *DocumentName* manages associated document name style, and *DocumentLink* manages link to associated document style.

See also :

► Structure : Links to documents

## ● Style class : UrlsGroup (URLs group)

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature    [Not in freeware edition](#)

*UrlsGroup* style class has 3 sub classes : *UrlsSubtitle* manages the associated subtitle style, *UrlName* manages the associated site name style, and *UrlLink* manages the style of the link to the associated web site.

See also :

► Structure : External links (URLs)

## ● Style class : NavBarGroup (navigation bar group)

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature    [Not in freeware edition](#)

*NavBarGroup* style class has two sub classes : *NavBarPages* manages the references style to previous and next pages (and has its own sub class : *NavBarNumbers* for references to sibling pages), and *NavBarUp* manages the parent page reference style.

**Example :**

🔗 Class *NavBarGroup*

*font-size:0.8em;*

*font-family:arial;*

*margin-left:30pt;*

*margin-right:30pt;*

*color:green;*

*text-align:center;*

See also :

► Navigation tab

## ● Style class : BookGroup (publication presentation group)

*NoteGroup* style class inherits properties from parent classes using it : *Idea* class or *BibliographyGroup* class.

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature    [Not in freeware edition](#)

A publication is a structured element, with 40 sub classes describing associated styles (in relation with the book elements of the structure template) :

Correspondance between Style CSS  
classes and structure elements

Style template CSS class name	Structure template element
.BookReferenceType	Reference Type
.BookAuthor	Author
.BookTitle	Title
.BookShortTitle	Short Title
.BookAlternateTitle	Alternate Title
.BookPublisher	Publisher
.BookPlacePublished	Place Published
.BookEdition	Edition
.BookOriginalPublication	Original Publication
.BookReprintEdition	Reprint Edition
.BookVolume	Volume
.BookNbVolumes	Number Of Volumes
.BookNumber	Number
.BookYear	Year
.BookDate	Date
.BookPages	Pages
.BookSection	Section
.BookKeywords	Keywords
.BookAuthor2	Secondary Author
.BookTitle2	Secondary Title
.BookAuthor3	Tertiary Author
.BookTitle3	Tertiary Title
.BookSubsidiaryAuthor	Subsidiary Author
.BookTypeOfWork	Type Of Work
.BookIsbn	Isbn/Issn
.BookReviewedItem	Reviewed Item
.BookLabel	Label
.BookAbstract	Abstract
.BookNotes	Notes
.BookUri	Uri
.BookAuthorAddress	Author Address
.BookImage	Image
.BookCaption	Caption
.BookUser1	Custom 1
.BookUser2	Custom 2
.BookUser3	Custom 3
.BookUser4	Custom 4
.BookUser5	ID
.BookAccessionNumber	Accession Number
.BookCallNumber	Call Number

See also :

► Structure : Bibliography note

► Style class : NoteGroup (Bibliography note group)


► Style class : BibliographyGroup (Bibliography group)

## ● Style class : IndexesGroup (index links group)

Menu    Templates  
Sub menu [Style/Modify](#)  
Feature    [Not in freeware edition](#)

*IndexesGroup* style class defines the style of links to indexes : table of contents, bibliography, glossary and keyword index. *IndexToc*, *IndexBibliography*, *IndexGlossary* and *IndexKeywords* sub classes define the style of each index.

#### Example :

 Class *IndexesGroup*  
*font-size:0.8em;*  
*text-align:center;*  
*font-family:arial;*  
*margin-left:30pt;*  
*margin-right:30pt;*  
*color:blue;*

See also :  
 [Navigation tab](#)

### Style class : *TocGroup* (Table of contents group)

*TocGroup* style class controls the style of table of contents.


Menu    Templates  
 Sub menu [Style/Modify](#)  
 Feature    **Not in freeware edition**

It is specialized into the *TocTitle* sub class to manage title style, and into cascading sub classes *TocLevel1* to *TocLevel7* and *TocLevelN*, to handle styles of all subtitle levels.

It also includes *TocImagesGroup* and *TocImage* classes to handle respectively the table of contents set of thumbnails, and each thumbnail.

#### Example :

 Class *TocGroup*  
*font-family:arial;*

 Class *TocTitle*  
*text-align:center;*  
*font-size:20pt;*

 Class *TocLevel1*  
*margin-left:20pt;*

 Class *TocLevel2*  
*margin-left:20pt;*

 Class *TocLevel3*  
*margin-left:20pt;*  
*font-size:0.8em;*

 Class *TocLevel4*  
*margin-left:20pt;*

In this example, all levels of title are 20 point indented relative to upper level. From level 3, title character size is reduced to 80%.

See also :  
 [Table of contents tab](#)

### Style class : *GlossaryGroup* (glossary group)

*GlossaryGroup* style class manages the glossary style.

Menu    Templates  
 Sub menu [Style/Modify](#)  
 Feature    **Not in freeware edition**

It has the following sub classes : *GlossaryTitle* controls title style, *GlossaryAccessBar* manages alphabetical access bar style, *GlossaryLetter* manages big initial letter style, and *GlossaryItem* manages glossary definitions style.


*GlossaryItem* is also specialized into *GlossaryDefinition* sub class, dedicated to definition style.

#### Example :

 Class *GlossaryGroup*  
*font-family:arial;*

 Class *GlossaryTitle*  
*text-align:center;*  
*font-size:20pt;*

 Class *GlossaryAccessBar*  
*text-align:center;*  
*font-weight:800;*

 Class *GlossaryLetter*  
*font-size:20pt;*  
*font-weight:800;*

 Class *GlossaryItem*  
*font-weight:800;*

 Class *GlossaryDefinition*  
*font-weight:800;*

This example shows the presentation that was chosen for this manual.

See also :  
 [Index tab](#)

### Style class : *NoteGroup* (Bibliography note group)

*NoteGroup* style class defines the style of bibliography note.

Menu    Templates  
 Sub menu [Style/Modify](#)

Feature **Not in freeware edition**

It has the following sub classes : *NoteSubtitle* controls the subtitle style, *NoteLink* controls the link style, when the book is replaced by a link, and when the book is displayed in the note, *BookGroup* manages the book style.

See also :

► Structure : Bibliography note

## ● Style class : BibliographyGroup (Bibliography group)

*BibliographyGroup* style class controls the bibliography style.

Menu **Templates**  
Sub menu **Style/Modify**  
Feature **Not in freeware edition**

It has the following subclasses : *BibliographyTitle* manages the style of main bibliography title, *BibliographyAccessBar* manages the style of the alphabetical access bar, *BibliographyLetter* manages the style of initial letters, and *BibliographyLink* manages the style of links to books.

Links to books can be replaced by the description of the publications themselves : the *BookGroup* style class is then used, inheriting properties of *BibliographyGroup*.

Two additional classes control the bibliographic index headers :

### Examples :

🔗 BibliographyPrimarySort :

```
font-size:14pt;
font-weight:800;
margin-top:1em;
```

🔗 BibliographySecondarySort :

```
font-size:11pt;
font-weight:500;
margin-top:1em;
margin-left:10pt;
```

See also :

► Index tab

► Style class : BookGroup (publication presentation group)

## ● Style class : IndexGroup (keywords index group) updated!

*IndexGroup* style class controls the style of keywords index.

Menu **Templates**  
Sub menu **Style/Modify**  
Feature **Not in freeware edition**

It has the following sub classes :

- *Index Title* manages the title style,
- *IndexAccessBar* the alphabetical access bar,
- *IndexLetter* the initial letter,
- *IndexFacet* the facet name (faceted index only),
- and *IndexKeyword* the index values.

*IndexKeyword* is also specialized into

- *IndexFacetName* sub class, as a reminder of facet name (shown before facet value in faceted index),
- *IndexReference* sub class, dedicated to the link to the indexed idea,
- *IndexKeywordParent* sub class, showing the parent idea context.

### Example :

🔗 Class IndexGroup

```
font-family:arial;
```

🔗 Class IndexTitle

```
text-align:center;
font-size:20pt;
```

🔗 Class IndexAccessBar

```
text-align:center;
font-weight:800;
```

🔗 Class IndexLetter

```
font-size:20pt;
font-weight:800;
```

🔗 Class IndexFacet

```
font-size:140%;
font-weight:800;
margin-top:1em;
```

🔗 Class IndexKeyword

```
font-weight:800;
```

🔗 Class IndexFacetName

```
color:silver;
```

🔗 Class IndexReference

```
font-weight:800;
```

🔗 Class IndexKeywordParent

```
font-size:smaller;
color:silver;
```

This example shows the presentation which was chosen for this manual.

See also :  
 ▶ Index tab  
 ▶ Version 5.10 [updated!](#)

## ● Style class : PageFooter (page footer text)

*PageFooter* style class controls the page footer text aspect.

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

### Example :

```
font-size:10pt;
text-align:right;
```

See also :  
 ▶ Page tab

## ● Style class : Copyright

*Copyright* style class controls the copyright text style.

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Not in freeware edition](#)

## ● Style class : WITChromeMenu [updated!](#)

*WITChromeMenu* style class has 13 sub classes, controlling the global appearance of the Drop Down menu, but also its behavior : change with caution.

**Example (this document styles : extract of style sheet. Do not change values shown in red) :**

```
/* WIT wrapping of Chrome CSS Drop Down Menu (courtesy of www.dynamicdrive.com) */
.WITChromeMenu {
/* Insert properties */
float:right;
}

/* Chrome CSS Drop Down menu bar */
.chromestyle {
width: 100%;
font-weight: 100;
font-family:arial;
font-size:100%;
}

/* Chrome CSS Drop Down Menu */
.chromestyle:after {
/*Add margin between menu and rest of content in Firefox*/
content: "";
display: block;
height: 0;
clear: both;
visibility: hidden;
}

/* Chrome CSS Drop Down menu : horizontal bar */
.chromestyle ul {
border: 1px solid #BBB;
width: 100%;
background: url(../extra/chromebg.gif) center center repeat-x; /*THEME CHANGE HERE*/
padding: 4px 0;
margin: 0;
font-size:100%;
text-align: left; /*set value to "left", "center", or "right"*/
}

/* Chrome CSS Drop Down Menu : horizontal bar elements list */
.chromestyle ul li {
display: inline;
}

/* Chrome CSS Drop Down Menu : horizontal bar links */
.chromestyle ul li a {
color: #494949;
padding: 4px 7px;
margin: 0;
text-decoration: none;
border-right: 1px solid #DADADA;
}

/* Chrome CSS Drop Down Menu : horizontal bar links, hovered */
.chromestyle ul li a:hover {
text-decoration: none;
background: url(../extra/chromebg-over.gif) -8px center repeat-x; /*THEME CHANGE HERE*/
}

/* Chrome CSS Drop Down Menu : horizontal bar links, visited */
.chromestyle ul li a:visited {
color: #494949;
}

/* Chrome CSS Drop Down Menu : trick... */
.chromestyle ul li a[rel]:after {
/*HTML to indicate drop down link*/
/*content: " v";
content: "" url(../extra/downimage.gif); /*uncomment this line to use an image instead*/
```

```

}

/* Chrome CSS Drop Down Menu : Dropdown pane */
.dropmenudiv {
border: 1px solid #BBB; /*THEME CHANGE HERE*/
border-bottom-width: 0;
font-size:90%;
line-height:18px;
text-align:left;
background-color: white;
color:grey;
width: 160px;
/* Do not change below */
position:absolute;
top: 0;z-index:100;
visibility: hidden;
filter: progid:DXImageTransform.Microsoft.Shadow(color=#CACACA,direction=135,strength=4); /*Add Shadow in IE. Remove if desired*/
}

/* Chrome CSS Drop Down Menu : dropdown pane links */
.dropmenudiv a {
width: auto;
display: block;
text-indent: 3px;
border-bottom: 1px solid #BBB; /*THEME CHANGE HERE*/
padding: 2px 2px;
text-decoration: none;
font-weight: 100;
color: #666666;
}

/* Chrome CSS Drop Down Menu : IE only hack */
* html .dropmenudiv a {
width: 100%;
}

/* Chrome CSS Drop Down Menu : dropdown link, hovered */
.dropmenudiv a:hover {
background-color: #F0F0F0;
text-decoration: none;
}

/* Chrome CSS Drop Down Menu : dropdown link, visited */
.dropmenudiv a:visited {
color: #666666;
}

```

See also :  
▶ Drop Down Menu widget [updated!](#)

## ● Style class : WITTree [updated!](#)

WITTree style class has 13 sub classes, controlling the global appearance of the Simple Tree menu, but also its behavior : change with caution.

**Example (this document styles : extract of style sheet. Do not change values shown in red) :**

```

/* WIT wrapping of CSS Simple Tree Menu (courtesy of www.dynamicdrive.com) */
.WITTree {
font-family:arial, verdana;
font-size:115%;
font-weight:500;
padding:3px;
background-color:RGB(132, 142, 200);
margin:5px;
margin-top:5px;
}

/* Inner container of WIT Simple Tree Menu */
.WITTreeInner {
margin-left: -38px;
margin-top:0px;
margin-bottom:0px;
padding: 0px;
}

/* Links in WIT menu, active */
a.WITTreeLink:active {
/* Insert properties */
}

/* Links in WIT menu, visited */
a.WITTreeLink:visited {
color:white;
}

/* Links in WIT menu, hover */
a.WITTreeLink:hover {
text-decoration:none;
color:RGB(255, 232, 211);
}

/* Links in WIT menu, link */
a.WITTreeLink:link {
color:white;
}

/* CSS for Simple Tree Menu : Main list */
ul.treeview {
margin-top:0px;
margin-bottom:0px;
}

```

```

}

/* CSS for Simple Tree Menu      */
.treeview ul {
/*CSS for Simple Tree Menu*/
margin: 0px;
padding: 0px;
}

/* Style for LI elements in general (excludes an LI that contains sub lists)      */
.treeview li {
/*Style for LI elements in general (excludes an LI that contains sub lists)*/
background: transparent url(../extra/list.gif) no-repeat 0px 2px;
list-style-type: none;
padding-left: 15px;
margin-bottom: 3px;
}

/* Style for LI that contains sub lists (other ULs).      */
.treeview li.submenu {
/* Style for LI that contains sub lists (other ULs). */
background: transparent url(../extra/closed.gif) no-repeat -9px -6px;
cursor: hand !important;
cursor: pointer !important;
}

/* Style for ULs that are children of LIs (submenu)      */
.treeview li.submenu ul {
/*Style for ULs that are children of LIs (submenu) */
display: none; /*Hide them by default. Don't delete. */
}

/* Style for LIs of ULs that are children of LIs (submenu)      */
.treeview .submenu ul li {
/*Style for LIs of ULs that are children of LIs (submenu) */
cursor: default;
}

```

See also :

► Simple Tree Menu widget [updated](#)

## 5.11.4. Import style sheets

This function creates a standard CSS style sheet (\*.css file) from the current WIT style template.

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Professional ed. only](#)

## 5.11.5. Export style sheet

This functions imports a standard CSS style sheet (\*.css file) into the current WIT style template.

Menu [Templates](#)  
 Sub menu [Style/Modify](#)  
 Feature [Professional ed. only](#)

Then, you just have to use the imported CSS classes in the text of your ideas.

## 5.12. Web sites

Menu [File](#)

### 5.12.1. Resource Report

The number of resources used by your Web site (images, documents, and URLs) may dramatically increase.

Menu [File](#)  
 Sub menu [Resource report...](#)  
 Feature [Not in freeware edition](#)

Resource report give you at a glance the list of all the resources that are used by all document ideas.

It displays a list of three value items :

- 🔗 Type : *Doc* for a document, *Img* for an image, *URL* for an Internet address.
- 🔗 Name : the name of the referenced file (void when type is *URL*).
- 🔗 Path : directory where the referenced resource is stored (or complete URL when type is an *URL*).

Double click on a resource item to see the list of all ideas currently using it. This opens the idea selector (with pre-filled selection criterion with resource name).

#### 5.12.1.1. Resolving resource name conflicts

Feature [Not in freeware edition](#)

Resource files such as documents and images are copied into *docs* or *wit-icons* subfolders at web site build time. Name conflicts may occur when resources coming from different places use the same name : the last copy would overwrite the previous.

WIT resolves itself these conflicts at web site build time: it renames the resources in conflict while copying them : the unique identifier of the idea which uses the resource is added. There is no effect on the resource source files.

If you do not wish this automatic rename, you may manually solve the conflicts : use the resource report, and double click on the resources in conflict ; the list of ideas using this resource is displayed.

How to eliminate a conflict : rename the file on the hard disk, and change the reference in the idea using it.

### 5.12.2. Automatic Web site generation

This function generates all the HTML pages, links and image resources belonging to the Web site.

Menu **File**  
Sub menu **Build and test web site...**  
Keyboard **Ctrl + E**

A dialog box is displayed (figure 4.1). The web site is generated in a free target directory (it should either better be initially empty, or contain results of the previous creation : files are overwritten).

Target directory contains :

- a *text* subdirectory in which generated pages are stored,
- a *wit-icons* subdirectory in which all images used by generated pages are stored,
- a *docs* subdirectory in which all local documents referred to by generated pages are stored,
- the frame and home pages, and the collection of all external resources stored in the source *"extra"* directory (a copy of this *extra* directory is also made).

If WIT is in expert mode, the structure and style templates which control the Web site construction can be adjusted.

Finally, the "Export" button starts the site construction (display of creation steps with progress bars), and then launches the default web browser, on the just created site home page.

If images or resources to which the current document refers are deleted or moved, non fatal errors will be signaled at site construction time : files not found, and then not copied in the target directory.

If resources with same names but coming from different directories are used, this will cause name conflicts. Use the resource report to easily solve these conflicts.

An error dialog box will be opened with all missing file names : double click on errors to open involved idea properties and make corrections.

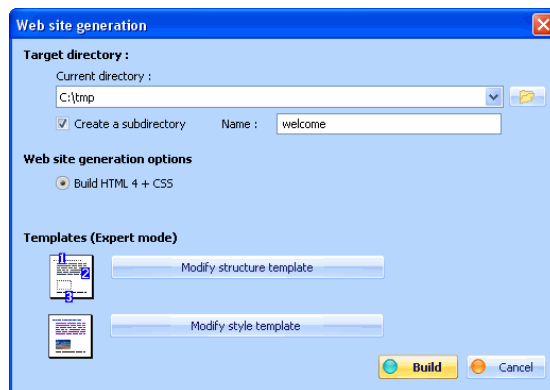


Fig 4.1 : Build Web site(expert)

See also :

- ▶ Expert mode
- ▶ Structure templates
- ▶ Style templates
- ▶ Default navigator choice

### 5.12.3. Rebuild all indexes

This function rebuilds all indexes : table of contents, bibliography, glossary, keyword index. It is useful when, for instance, new keywords have been added to ideas.

It is automatically invoked at web site build time, and when the document is loaded.

See also :

- ▶ Keywords
- ▶ Keywords *updated!*

### 5.12.4. Upload on a Web server by FTP

This functions deploys the Web site on an intranet or internet server, with standard FTP protocol (*file transfer protocol*).

Menu **Keywords**  
Sub menu **Rebuild all indexes**  
Keyboard **Ctrl + Maj + E**

Web server should also be an FTP server. Most Internet Services Providers (ISP) servers are both : a space is allocated, and an account name and password too.

An active Internet connexion is mandatory to perform the file transfer.

Enter following options :

- FTP server: the server name such as given by the ISP (mandatory field).
- User account : your account (mandatory field).
- Password : your password (mandatory field).
- Local source directory : the directory where the site was generated (if it was just created, the directory name is up to date).
- Target distant directory : usually a void name, when the site should be installed at the root of the allocated server space.

Establish Internet connexion (external means), and press then the "Transfer" button. Messages and a progress bar show transfer steps and success.

Some transfer errors may occur.

Menu **File**  
Sub menu **Upload site with FTP...**  
Feature **Not in freeware edition**



They are explained by WIT :

- Internet connection not available, unknown server, bad account, wrong password.
- Read/write rights on distant FTP server.

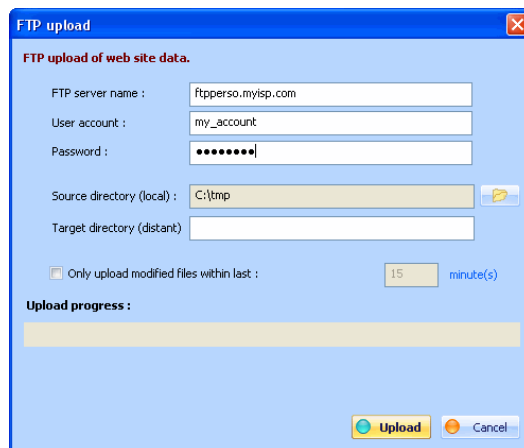


Fig 4.2 : FTP transfer options.

## 5.13. Options

WIT is highly customisable.

Menu **File**  
Sub menu **Options**

Documents handling and software use can be adapted to your options.

Options are grouped in an option tree dialog so that all parameters can be checked at a glance.

For each item, click on left column to get help, on right side to change the item value.

### 5.13.1. Expert mode

This mode, dedicated to experienced user, lets you :

Menu **File**  
Sub menu **Options**  
Feature **Not in freeware edition**

- Access to the pure HTML edit view tab.
- Parametrize, create and save structure templates,
- Parametrize, create and save style templates,
- Control (and specify) names of generated pages from each idea (useful to write a contextual help).
- Access to "Tasks", "Custom fields", "Display", "Meta" and "Stats" tabs in idea properties.

**This mode is enabled by default.**

If not enabled :

- when pressing one of the buttons *Modify page structure*, or *Modify page style* of the leftside toolbar, the tab *Structures* or *Styles*, slides in order to offer predefined templates **loading**.
- Template modification and save options are also disabled by default in the *Templates* menu, and in the web site build and test dialog box. Names of generated pages are automatic by default.

When the expert mode is active, the above functions come to be active. When pressing one of the *Modify page structure*, or *Modify page style* buttons in the leftside toolbar, specific dialogs of structure or style template modification are displayed.

See also :

- ▶ Generated pages names choice
- ▶ Structure templates
- ▶ Style templates
- ▶ Tasks tab **updated!**
- ▶ Custom fields tab **updated!**
- ▶ Display options tab
- ▶ Meta tab
- ▶ Stats tab

### 5.13.2. Document options

These options apply to the current document.

Menu **File**  
Sub menu **Options (document)**

#### 5.13.2.1. Custom fields options **updated!**

Menu **File**  
Sub menu **Options (document)**  
Feature **Professional ed. only**

In addition to predefined fields such as title, text, keywords or associations which characterize an idea, WIT offers 32 more fields which name and use can be customized.

Figure 4.3 shows an example, where all fields have been activated (check boxes), and renamed.

Right checkboxes show whether each custom field contribute or not to the faceted index. Les cases à cocher de droite indiquent la participation ou non de chaque champ personnalisable à l'index à facettes. The index is controlled by the [structure template](#) ('Navigation' tab).

These choices are linked to the current document and are reflected in the different dialog boxes which help to manipulate ideas, search, or establish structure and style templates.

Note ([professional edition](#)) :

- If the "Multiple value" option is checked for a given field, multiple values (separated by a ; delimiter) will be accepted.
- If the "Facets" option is also checked, each value will also be linked to a separate entry in the faceted index.

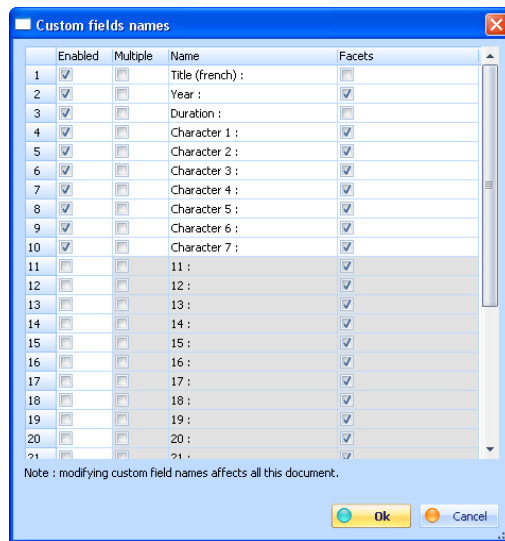


Figure 4.3 : Enabling and naming custom fields

See also :  
▶ [Custom fields tab updated!](#)

### 5.13.2.2. Task fields options [updated!](#)

Menu [File](#)  
Sub menu [Options \(document\)](#)  
Feature [Professional ed. only](#)

In addition to predefined fields such as title, text, keywords or associations which characterize an idea, WIT offers 11 more fields allowing task management.

The figure below shows an example, where some fields have been activated (check boxes), and made eligible for the faceted index.

Right checkboxes show whether each task field contribute or not to the faceted index. The index is controlled by the [structure template](#) ('Navigation' tab).

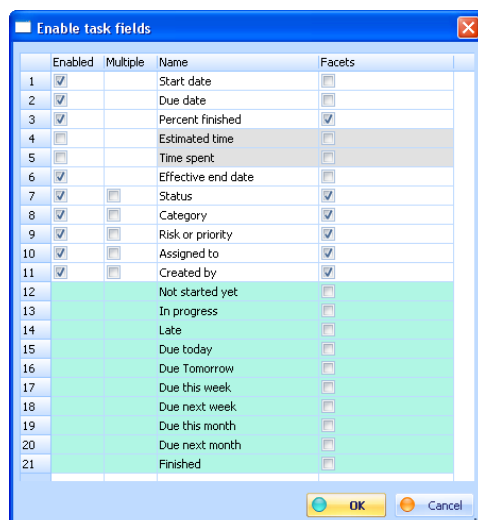
These choices are linked to the current document and are reflected in the different dialog boxes which help to manipulate ideas, search, or establish structure and style templates.

Note 1 :

- The names shown with colored background are not idea task fields, but progress criteria that WIT evaluates dynamically at website build time.
- If faceted classification is enabled for these criteria, WIT adds the lists of matching ideas to the faceted index.
- WIT can also add a synthetic [task progress widget](#) ([professional license](#)) to summarize progress status and link quickly to the faceted index.

Note 2 ([professional edition](#)) :

- If the "Multiple value" option is checked for a given field, multiple values (separated by a ; delimiter) will be accepted.
- If the "Facets" option is also checked, each value will also be linked to a separate entry in the faceted index.



Enabling task fields

See also :

- ▶ Versions 5.0, 5.01
- ▶ Tasks tab [updated!](#)
- ▶ Task progress status widget [updated!](#)
- ▶ Task fields Table of Contents widgets [updated!](#)
- ▶ Version 5.10 [updated!](#)

### 5.13.2.3. Main view columns choice

This function sets columns display in the main view. It opens a dialog box where chosen columns are selected (checked). Menu [File](#)  
Sub menu [Options \(document\)](#)

Column display order is given by the dialog box. Right fields are the custom fields.

For example, in figure 4.4 custom fields belongs to a document managing bug reports of a software in development phase.

When the dialog box is validated, current view is not changed yet, but new views will be affected (either by creating new views, either by saving, closing and reloading the document).

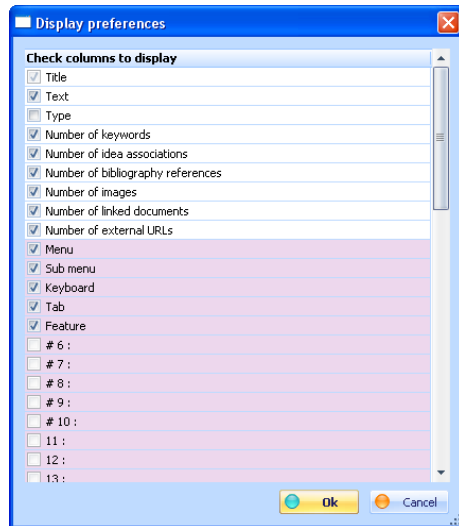


Fig 4.4 : Displayed columns choice in the main view

### 5.13.2.4. Work directory

Each WIT document uses a work directory (by default equal to the [project directory](#)). Menu [File](#)  
Sub menu [Options \(document\)](#)

When creating the documents in a subdirectory of the work directory (for example *MyDocument*), or when referencing images or other files located in subdirs of work directory, pathnames are changed into a relative portable form : *C:\Program Files\Web Idea Tree\Projects\MyDocument*, in order to make them and their internal references independent of their physical location.

You can subsequently move all the work directory without modifying the references to the documents it contains (relative references are kept). You just have to tell WIT where the new work directory is located.

Seemingly, WIT has internal and extensible resources to build structure templates and skeletons of generated web sites. This resources are located in the *\Data* subdirectory of WIT installation directory (example : *C:\Program Files\WIT\Data*). This special directory is named **\_DATADIR** and is not customizable (it depends on installation only).

Important note : when referenced images and files are located in your current WIT document folder or subfolders, pure relative paths are used (without the *C:\Program Files\Web Idea Tree\Projects* prefix). This means that your WIT source environnement (WIT document, referenced images and files) is fully portable on another computer when all the information is located in the WIT document folder and sub folders.

### 5.13.2.5. Generated pages names choice

This function is available in expert mode, when precise name of generated web page is wanted, in correspondence with each idea. Menu [File](#)  
Sub menu [Options \(general\)](#)  
Feature [Professional ed. only](#)

There's no need to activate this function in other cases : MyIdeaTree automatically takes in account references of pages to establish hypertext links and organize files in the site directories.

When activated, this function displays a dialog box (unique name query), when an idea is created, duplicated, or pasted from another WIT document.

**Note :** this option is also a general WIT option : in this case, it is interpreted as a default value for all new documents.

See also :

- ▶ Create a child idea [updated!](#)
- ▶ Create a sibling idea (before)
- ▶ Create a sibling idea (after)
- ▶ Import ideas from CSV file [updated!](#)
- ▶ Paste an idea
- ▶ Copy an idea by drag and drop

### 5.13.2.6. Generated pages files suffix

The default suffix is ".html".

You may choose another suffix, for all WIT generated pages.

Suffix change is mandatory for server side interpreted scripts languages such as PHP, ASP or JSP :

🔗 PHP (Personal Home Pages) : .php suffix,

🔗 ASP (Active Server Pages) : .asp suffix,

🔗 JSP (Java Server Pages) : .jsp suffix.

Note : this function is available in Expert Mode only.

Note : this option is also a general WIT option : in this case, it is interpreted as a default value for all new documents.

Menu **File**  
Sub menu **Options (general)**  
Feature **Not in freeware edition**

### 5.13.2.7. Define the "author" META TAG

When active, this feature lets you define the author of documents and insert the meta tag :

```
< META name="author" content="author name">
```

in all generated pages.

(This option is specific to professional license).

Note : this option is also a general WIT option : in this case, it is interpreted as a default value for all new documents.

Menu **File**  
Sub menu **Options (general)**  
Feature **Professional ed. only**

### 5.13.2.8. Define the "copyright" META TAG

When active, this feature lets you define the copyright of documents and insert the meta tag :

```
< META name="copyright" content="copyright mention">
```

in all generated pages.

(This option is specific to professional license).

Note : this option is also a general WIT option : in this case, it is interpreted as a default value for all new documents.

Menu **File**  
Sub menu **Options (general)**  
Feature **Professional ed. only**

### 5.13.2.9. Keep text fields while loading structure templates

This option is checked by default.

It allows to keep the texts fields you may have changed in the current structure template of your document, even when you load another structure template.

If this option is not checked, current document values are replaced by the values of the last loaded structure template.

Note : this option is also a general WIT option : in this case, it is interpreted as a default value for all new documents.

Menu **File**  
Sub menu **Options (general)**

### 5.13.2.10. Target documents

This options controls the local document links behavior : when such a link is clicked, the document may be opened in a new window (default behavior), or in the current window.

Menu **File**  
Sub menu **Options (general)**

### 5.13.2.11. Target URLs

This options controls the external URLs links behavior : when such a link is clicked, the URL may be opened in a new window (default behavior), or in the current window.

Menu **File**  
Sub menu **Options (general)**

### 5.13.2.12. Copy resources

This options control whether resources (images and local documents) are copied or not when the web site is built.

Default behavior : images and documents are copied into the wit-icons and docs subfolders of the target folder. All the web site files are stored into the target folder and its subfolders so that the web site is portable and does not depend on any other files.

Alternate behavior : images and documents files are not copied, and absolutes links are used to refer to them. This implies that the generated web site should not be moved on another computer unless the resources files are seen the same way (either in a network shared folder or in a copy of the resource tree folders). So web site co-workers can modify them without rebuilding the whole site (useful in collaborative groups).

Menu **File**  
Sub menu **Options (general)**

### 5.13.3. WIT General options

These options apply to all documents.

Sub menu [Options](#)

#### 5.13.3.1. Project directory

The Projects folder allow you to store your WIT documents. By default, it is located in your Windows profile :

Menu [File](#)  
Sub menu [Options \(general\)](#)

*My Documents\My Web Idea Tree\Projects.*

WIT also uses this folder to populate the "Projects" tab of the left vertical toolbar, for a rapid access to all your projects.

See also :  
▶ [Version 5.10 updated!](#)

#### 5.13.3.2. Sound activation

This functions activates or disables sounds played by WIT (when signalling errors, questions, at start time, at end).

Menu [File](#)  
Sub menu [Options \(general\)](#)

#### 5.13.3.3. Default navigator choice

Menu [File](#)  
Sub menu [Options \(general\)](#)

Although WIT generates portable and standard HTML code, using W3C recommended Cascad Style Sheets (CSS), all web browsers may not correctly display some elements of page. The two most compliant navigators are Netscape (4.7) and Microsoft Internet Explorer (from 4.01 release), with a slight advantage for MSIE 5, or better 5.5.

WIT helps to view the rendering of generated pages with any navigator. A dialog box helps to choose the default browser, by specifying its access path. This dialog is also invoked the first time a page is viewed.

Notes :

- usual path to MSIE is generally *C:\Program Files\Internet Explorer\Iexplore.exe* if your system is installed on the C: drive.
- usual path to Netscape is generally *C:\Program Files\Netscape\Program\netscape.exe* is Netscape was installed with default options .

Pressing the search buttons automatically finds the needed programs.

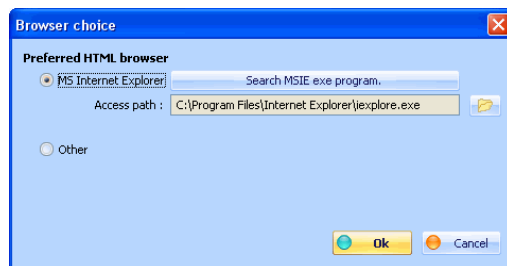


Fig 4.5. Choice of default navigator

#### 5.13.3.4. Language choice

Menu [File](#)  
Sub menu [Options \(general\)](#)

Language choice is currently made between English, French, Italian, Swedish, Polish and Russian (the two latest with the WIT Unicode version), and is effective once WIT is restarted.

Language choice affects graphical interface, on line help system, but also, default structure and style template options, when creating a new document. These options can however be modified later in the document life.

However, to create a document and a web site in a given language, it's better to use WIT in this language. For all other tongues, please configure WIT in English.

Note : when WIT is launched for the first time, a default language must be chosen (dialog box). The choice can be modified later.

#### 5.13.3.5. Character set

Menu [File](#)  
Sub menu [Options \(general\)](#)

This option is available only in the WIT international version (Unicode enabled), for Windows NT4, Windows 2000, Windows XP and Vista.

You may :

- display all languages scriptings, including right to left scriptings such as Arabic or Hebrew,
- specify how Web browsers should interpret character codes in generate web pages.

For example, to write with Russian scripting :

- Configure the keyboard for Cyrillic characters,
- Choose the "Cyrillic (KOI8-R)" or "Cyrillic (KOI8-U)" character set.

The UNICODE coding is the default value, suitable for most languages : keep it if possible.

Trick :

Users of Western Europe, USA, Canada, Australia, New Zealand, English or French speaking, etc... KEEP the "Western Europe" character set (iso-8859-1). HTML pages will be generated with 8 bit instead of 16 bit characters, and will be twice lighter (in size) than in other encodings.

### 5.13.4. Advanced WIT options

These options apply to all documents.

Menu **File**  
Sub menu **Options (advanced)**

#### 5.13.4.1. Reopen last document

Reopens the last document at startup.

Menu **File**  
Sub menu **Options (advanced)**

#### 5.13.4.2. Quick titles

When creating an idea, WIT displays a dialog to just type the idea title (instead of full properties dialog).

Menu **File**  
Sub menu **Options (advanced)**

This option is useful when you create the structure of your document.

#### 5.13.4.3. Linear navigation

Choose this option is to get linear navigation links in your website.

Menu **File**  
Sub menu **Options (advanced)**

When activated, the Next and Previous page arrows allow a linear reading of the site as though it were a book (its structure is "flattened").

Parent page arrows still allows to go up a level in the structure.

#### 5.13.4.4. Titles in navigation bar

The titles of next, previous, and parent page are also displayed in the navigation bar (either in plain text, or as alternate text if images are used : see the structure template options).

Menu **File**  
Sub menu **Options (advanced)**

See also :  
► [Navigation tab](#)

#### 5.13.4.5. Top page links

This option adds links to top of page in the indexes, glossaries and bibliographies built pages.

Menu **File**  
Sub menu **Options (advanced)**

#### 5.13.4.6. Autosave

When activated, WIT periodically auto saves your current document.

Menu **File**  
Sub menu **Options (advanced)**

The auto saved document has the same name as yours, with an autosaved- prefix, and is located in the same folder.

#### 5.13.4.7. Autosave period

This period is given in minutes.

Menu **File**  
Sub menu **Options (advanced)**

#### 5.13.4.8. HTML font size

This option controls the font size of HTML raw editor (expert mode).

Menu **File**  
Sub menu **Options (advanced)**

Give the size in points. 18 points is the default value.

#### 5.13.4.9. Wrap HTML lines

This option will wrap/unwrap all lines in HTML editors.

Menu **File**  
Sub menu **Options (advanced)**  
Keyboard **Ctrl + W**  
Tab **"Html" view**

Tip : You can use the 'Ctrl + W' key to toggle this option when you are in the "Html" tab in the main view.

#### 5.13.4.10. HTML WYSIWYG

This option is enabled by default.

Menu **File**  
Sub menu **Options (advanced)**

Disable it if you have not Microsoft Internet Explorer 5.0 installed on your system.

WIT will work in a simpler mode where you won't have fine editing controls over your HTML text (WYSIWYG editing disabled), but you'll still be able to conceive and generate web sites.

### 5.13.5. User macros updated

You have there 10 fields where you can put anything (usually small HTML or Javascript elements).

Menu [File](#)  
Sub menu [Options \(advanced\)](#)  
Feature [Professional ed. only](#)

If you insert special WIT variables from < @macro1 /> to < @macro10 /> in idea texts, titles, custom fields, etc..., at build time, WIT will replace these occurrences by the values you have specified here.

These values are stored in the Windows registry (default values), and are also saved in the document : document saved values have precedence over registry values.

## 6. CSS 1

CSS1 (Cascade Style Sheet 1) is a norm created in 1996 by the World Wide Web Consortium (W3C).

CSS1 helps to define style sheets which offer a constant and homogeneous HTML page style.

This language is supported by Internet browsers such as Internet Explorer (from 4.0 version) and Netscape (from 4.0x versions).

[Note : if a browser doesn't recognize style sheets, it will ignore unknown directives, with a default page settings : no information should be lost.](#)

Bibliography :

 [DREYFUS2000 : HTML 4 \(DREYFUS Michel\)](#)

### 6.1. Style classes

In WIT, styles are applied with the help of style classes ; the classes are attached to the HTML DIV tags (blocks of text) and HTML SPAN tags (sets of characters on a single paragraph).

These classes are predefined and named by WIT in relation with elements of the site structure template.

A set of rules is associated to each class. A rule is a couple (property:value:). The rules apply to current style class, but can be inherited if HTML blocks (DIV or SPAN) are embedded.

In the WIT style template options dialog, the order and indentation of elements reflects their hierarchy. The most general rules will be specified in the upper level classes, more specialized properties in derived classes.

The top class is named BODY : its properties are known in all other classes, except if these classes locally specify most specific properties ("overwriting" inherited properties).

See also :

► [Structure templates](#)

### 6.2. Syntax of rules

A rule is a set of couples (property : value) separated by ';' :

```
property1 : value1; /* Optional comments, as in C language */ property2 : value2; ...
```

Properties are reserved names.

Example : `color font-family`

Values are predefined names (colors), or integer or float numbers, optionally followed by a unit.

Example : `10mm 2.5em 16% magenta rgb(15%, 26%, 80%)`


### 6.3. Values

#### 6.3.1. Numeric values

They are integer or float numbers : `2 -13 1.0 -4.7` and can be followed (without any space) by a unit name.

##### 6.3.1.1. Absolute units

 mm : millimeter,

 cm : centimeter,

 inch : inch (2,54 cm),


 pixel : pixel of screen,


 pt : point, ie 1/72 inch, or 0.353 mm,

 pica : 12 points, or 4,233 mm.

##### 6.3.1.2. Relatives units

 % : percent of something defined in a parent property (example : margins related to screen width).

 `em` : size (horizontal) of the current font (if the font body is 14 pt, 1em = 14pt). It is the 'm' character width.

 `ex` : height of the current font (height of 'x' character).

### 6.3.2. Keywords

---

Some properties admit keywords with conventional meaning such as : `larger` `smaller` `dotted` `dashed`

### 6.3.3. Colors

---

There are three ways to specify colors.

#### 6.3.3.1. Color name

---

16 predefined names can be used :



-  `aqua` (clear blue)
-  `black`
-  `blue`
-  `fuchsia`
-  `gray`
-  `green`
-  `lime` (clear green)
-  `maroon`
-  `navy` (navy blue)
-  `olive`
-  `purple`
-  `red`
-  `silver`
-  `teal`
-  `white`
-  `yellow`

#### 6.3.3.2. RGB triplet

---

RGB triplet defines a color from its red, green and blue components.



Two possibilities :

-  Intensity value of each part in a 0 - 255 range : example `rgb(123, 191, 45)` is a pale yellow-green.
-  Intensity value of each part in percentage (0% - 100%) : example `rgb(48%, 75%, 18%)`

#### 6.3.3.3. Hexadecimal number

---

Two possibilities :

-  With values in the 0 - 255 range : `#7bbf2d` gives the same RGB specification than `rgb(123, 191, 45)`.
-  With values in the 0 - 15 range : `#8c3` gives nearly the same color as the previous.

### 6.3.4. Url

---

Some values can be files, especially images.

Syntax :

```
url(http://www.my-web.com/file.html) /*absolute address */
url(../extra/image.gif) /* local image (from the "extra" source directory) */
url(../wit-icons/image.gif) /* local image, already used as an illustration by at least one idea. */
```

Note : Urls to images used by WIT must be those which will be effective when the site is produced. So you MUST use relative URLs. Especially, all image URLs are relative and must start by `../` and related image files will be placed in the *"extra"* subdirectory of the current WIT document save directory.

## 6.4. Properties

---

### 6.4.1. Blocks

---


All HTML elements are contained in rectangular blocks to which properties are attached.

A block may be separated from other blocks by *margins* (*margin* property), it can be delimited by a *border* (*border* property), and its contents may be adjusted relatively to this border (*padding* property).

A block can "float" above another (*float* property) or require to be alone on a line (*clear* property).

#### 6.4.1.1. width and height properties

---

 width & height



These properties define block geometry. They accept one value, either absolute (width in pixels, height in points, etc...), either relative (for example two thirds of screen width).

Example :

```
width:66%; /* two thirds of screen width */.
```

### 6.4.1.2. margin property

#### margin




This property applies to all elements and define the margin separating them from page borders or other neighbor elements. It accepts from 1 to 4 values, separated by spaces.

These values should be in the following order : top, right, bottom and left.

Example :

```
margin:15; /* 15 pixel margin on all sides */
margin:1em 2em; /* 1em vertical margin, 2em horizontal margin */
margin:1em 1em 1em 2em; /* all margins of 1em, except left (2em) */
```

Top, right, bottom and left margins can be separately specified with below properties (they accept 1 value) :

-  margin-top
-  margin-right
-  margin-bottom
-  margin-left

### 6.4.1.3. border properties

#### border

This property applies to all elements and separates margin area (*margin*) from internal adjustment area (*padding*). It admits from 1 to 3 values, separated by spaces, and define *thickness*, *style* and *color*. These values apply to all sides : top, right, bottom and left.

Thickness is specified by a value (*number*, or keyword) in the list : *thin* , *medium*, *thick*.

Style is defined by a keyword (cf below, border-style property).

Color is defined by a keyword (one of the 16 predefined colors).





#### border-style

This property defines the border style, from a list of predefined styles (keywords). It accepts from 1 to 4 values, separated by spaces. These values apply in the following order : top, right, bottom, and left. Predefined values are : *none*, *dotted*, *dashed*, *solid*, *double*, *groove*, *ridge*, *inset*, *outset*.

#### border-color

This property defines separately the border color, in a 16 predefined color list (keywords). It accepts from 1 to 4 values, separated by spaces. These values apply in the following order : top, right, bottom, and left.

Top, right, bottom and left borders can be separately specified with below properties (they accept from 1 to 3 values (thickness, style and color)) :

-  border-top
-  border-right
-  border-bottom
-  border-left

See also :





► [Colors](#)

### 6.4.1.4. padding property

#### padding

This property defines the inner adjustment (internal margin) of the element, relatively to the block border (even if the border width is null), and, like *margin*, accepts from 1 to 4 values, separated by spaces. These values apply in the following order : top, right, bottom, and left.

Top, right, bottom and left padding can be separately specified with below properties (they accept 1 value) :

-  padding-top
-  padding-right
-  padding-bottom
-  padding-left

See also :

► [margin property](#)

### 6.4.1.5. float property

#### float

This property makes the current element "float", relatively to the next element, instead of being vertically displayed in sequence.

It accepts the following values (keywords) : *left*, *right*, *none*.

For example, if an image is contained in a floating block (left) and is followed by a text block, the text, instead of beginning after the image, begins to the right side of the image and "flows" around it.

### 6.4.1.6. clear property

---

#### clear

This property specifies which side of a block can't be placed next to the previous floating block (float property).

The effect is that the block is moved downwards, under the clear block. No effect if the previous block is not a floating one.

The clear property accepts one value, in a keyword list : *left*, *right*, *both*, ou *none*.

When *both* is specified, the current block is placed under the biggest floating block.

## 6.4.2. Foreground, background

---

### 6.4.2.1. color property

---

#### color

This property defines foreground color, and accepts a unique value.

See also :  
► Colors

### 6.4.2.2. background-color property

---

#### background-color

This property defines background color, and apply to all element types. It accepts one value : by default the *transparent* keyword, or a color specification.

See also :  
► Colors

### 6.4.2.3. background-image property

---

#### background-image

This properties displays a background image (GIF or JPEG format). It accepts one value, an URL :

```
background-image:url(icons/back.jpg); /* see note 1 below */
background-image:url(../extra/extra-image.jpg); /* see note 2 below */
```

Note 1 : if you use such an url (with *icons* leading directory), the image file (back.jpg) should be used by at least one idea (idea properties : links tab, images section) in order to be correctly copied in the automatically generated *icons* directory of web site.

Note 2 : if you use such an url (referencing to parent directory of current generated html pages directory (text)), you have to store the image file (extra-image.jpg) in the *"extra"* subdirectory of your current document directory.

See also :  
► Uri  
► Background image is not properly displayed (CSS background-image property)

### 6.4.2.4. background-repeat property

---

#### background-repeat

This property tiles the background image (specified by background-image property). It accepts one value in the keyword list : *repeat* (image repeated in both directions (default)), *repeat-x* (X axis repeated image), *repeat-y* (Y axis repeated image), *no-repeat* (no repetition).

### 6.4.2.5. background-attachment property

---

#### background-attachment

This property controls optional background move when the main view is scrolled in the browser. It admits one value in the below keyword list :

*scroll* : default value. Background is linked to other elements and moves with them when the view is scrolled.  
*fixed* : background stays fixed, even if page contents are scrolled.

### 6.4.2.6. background-position property

---

#### background-position

Defines the initial offset of background image. It admits 2 values : x and y positions. This values may be absolute, percent values, or keywords.

For x position, keywords are : *top*, *center*, *bottom*.  
For y position, keywords are : *left*, *center*, *right*.

---

#### 6.4.2.7. background property

---

##### background

This property groups the two previous ones and accepts from 1 to 6 values, in any order (except position, counting for 2 values, where x is placed before y) :

- background color,
- background image,
- x and y position,
- attachment,
- repetition.

Omitted values are replaced by default values.

---

#### 6.4.3. Text

---

##### 6.4.3.1. word-spacing property

---

##### word-spacing

This property controls space between words. It admits 1 numerical value. Attention : it is not correctly implemented yet in web browsers.

##### 6.4.3.2. letter-spacing property

---

##### letter-spacing

This property controls space between characters. It admits 1 numerical value.

Attention : it is not correctly implemented yet by Netscape (4.7).

##### 6.4.3.3. text-decoration property

---

##### text-decoration

This property defines text enhancements without changing font. It admits one value in the below keyword list : *underline*, *overline*, *line-through*, *blink*, *none* (default).

##### 6.4.3.4. text-transform property

---

##### text-transform

This property transform text uppercase and lowercase letters. It admits one value in the below keyword list :

- capitalize* (first letter of each word is uppercase, others lowercase),
- uppercase* (transforms all letters in uppercase),
- lowercase* (transforms all letters in lowercase),
- none* (default : normal display).

##### 6.4.3.5. text-align property

---

##### text-align

This property controls text alignment between borders (more precisely padding areas) of current block. It admits one value in the below keyword list : *left* (default), *right*, *center*, *justify*.

##### 6.4.3.6. text-indent property

---

##### text-indent

This property specifies a first paragraph line indent value (other lines are unchanged). The indentation is a positive or negative value, absolute, or relative to the block width.

##### 6.4.3.7. vertical-align property

---

##### vertical-align

This property controls vertical alignment of an HTML object, relatively to another one (typically an image relatively to text). It accepts one value, in the below keyword list : *baseline* (aligned baselines), *sub*, *super*, *text-top*, *text-bottom*, *middle* (objects vertical middle parts aligned), *top*, *bottom*.

---

#### 6.4.4. Fonts

---

#### 6.4.4.1. font-family property

---

##### font-family

This property specifies fonts to use with text.

Several fonts may be specified : the browser will take the first found.

The expected parameter is either a font name (string), either a list of names (separated with ',').

Example :

```
font-family:verdana, arial; /* specifies Verdana, otherwise Arial */
```

Lower or upper case does not matter. If the font name has spaces, use double quotes ("Times New Roman").

#### 6.4.4.2. font-size property

---

##### font-size

This property sets font size in a block including some text.

Size may be specified as absolute with keywords (*xx-small*, *x-small*, *small*, *medium*, *large*, *x-large*, *xx-large*) or as relative to parent block : *smaller* and *larger*.

Size may also be specified with a numeric value : absolute (pt, pica, mm, cm) or relative (%).

#### 6.4.4.3. font-weight property

---

##### font-weight

This defines the text weight (boldness).

It expects a keyword, either string or numeric : *lighter*, *normal*, *bold*, *bolder*, *100*, *200*, *300*, *400*, *500*, *600*, *700*, *800*, *900*.

#### 6.4.4.4. font-style property

---

##### font-style

This property specifies text style : *normal* (default), *italic*, *oblique*.

#### 6.4.4.5. font-variant property

---

##### font-variant

This property, similar to text-transform, display text in small capitals, ie the first letter in uppercase, and next letters in small uppercase. It admits one value (keyword) : *normal* (default), *small-caps* (small caps display).

Example : **TEXT DISPLAYED IN SMALL CAPS**. This property is not well applied by browsers.

#### 6.4.4.6. line-height property

---

##### line-height

This property controls the spacing between two consecutive liens inside a paragraph.

Height is a numerical value, absolute or relative.

#### 6.4.4.7. font property

---

##### font

This property groups in all font properties in a single rule.

It has 6 values, which order must be : *style*, *variant*, *weight*, *size*, *line-height*, *family*.

### 6.4.5. Lists

---

#### 6.4.5.1. list-style-type property

---

##### list-style-type

This property applies to elements displayed in lists (when the *display* property is *list-item*).

If it is an ordered list, expected value is a keyword :

- *decimal* (default, decimal numbering),
- *lower-roman*,
- *upper-roman*,
- *lower-alpha*,
- *upper-alpha*.
- *none* (order number not shown).

If the list is not ordered (dot list), expected value is a keyword in the below list :

- *disc* (big dot of current color),
- *circle* (the same dot, but void),
- *square* (a little square of current color),
- *none* (dot not shown).

#### 6.4.5.2. list-style-position property

---

 list-style-position

This property controls dot or numbering alignment of elements displayed by lists. A keyword value is expected :

- *outside* (default),
- *inside* .

#### 6.4.5.3. list-style-image property

---

 list-style-image

This property applies to elements displayed in lists (when the *display* property is *list-item*). It replaces the dot or the number placed before the text by an image. Expected value is therefore an URL to an image.

#### 6.4.5.4. list-style property

---

 list-style

This property groups into the three previous properties in a single rule.

### 6.4.6. Other elements

---

#### 6.4.6.1. display property

---

 display

This property applies to all elements and controls how they are displayed. One value (keyword) is expected :

- *block* (default. The element is displayed at the beginning of a new line, in a block),
- *list-item* (the element is displayed as a list item),
- *inline* (the element is displayed on the same line as the previous block),
- *none* (the element is not displayed).

Internet Explorer (versions 4 and 5.0) doesn't recognize *block* and *list-item* keywords.

#### 6.4.6.2. white-space property

---

 white-space

This property applies to all elements containing some text, and controls separators (spaces, tabs, carriage returns). One value is expected (keyword) :

- *normal* (spaces and tabulations are translated into a unique space, and carriage returns by a unique carriage return).
- *pre* (separators are literally interpreted),
- *nowrap* (like *pre*. Moreover, long lines are cut instead of being wrapped).

This property is not recognized by Internet Explorer (5.0).

## 7. FAQ

---

Here is a compendium of most frequently asked questions (FAQ).

For a flat list of all questions, please click on the FAQ keyword (at page bottom).

### 7.1. General

---

#### 7.1.1. FAQ : How to begin ?

---

There are two steps :

- your site design and test.
- its publication on the Internet.

The first step is WIT specific, the second is more general (and not discussed here).

I first suggest that you read the "FAQ : Method to use WIT " page to acquire an efficient and straightforward method to outline your ideas, and well understand web site creation with WIT :

- organise disk space to manage your site data(WIT document WIT, images, etc...)
- organise the site outline by creating, moving and hierarchising the ideas
- add text and images to you ideas
- automatically build the site and test it
- deal with site structure and style by applying templates or creating new ones.

I strongly advise to read and work the tutorial (count between 15 and 30 minutes) : the tutorial is a part of the user's manual (Tutorial (a 15 minute tour to create a Web site)), or available at <http://www.webideatree.com/mu-en/text/tutorial.html>.

Once you are done with the tutorial, you know almost everything.

You also have the "welcome-en.wit" document (open it and modify it with WIT) : you get there fast and simple overlook.

See also :  
▶ FAQ : Method to use WIT

## 7.1.2. FAQ : Method to use WIT

*Abstract* : You must work first on the idea organization, then on their content, the keywords, and finally on presentation structure and style.

At any time during the process, you may generate the web site.

### Web site creation methodology

#### Basic steps

1. First, organize the space where you will work :
  - create a sub directory of the "Projects" WIT directory (for instance) : it will be your current project directory,
  - in this directory, you may create "images" and "documents" subdirs, in which you will store the images and documents you will use in your site.
  - create a new document wit WIT and save it in your project directory.
  - optional : set the work directory C:\Program Files\Web Idea Tree\Projects environment variable to your project folder (see the [document options](#) to do this.)
2. Writing Phase : concentrate on what you mean :
  - create the outline : create / organize your ideas (with WIT),
  - type (or paste) some text in your ideas,
  - read your writing, correct eventual mistakes : you may use the preview window for a comfortable reading.
3. Production phase :
  - choose a structure template (or modify current template (expert mode)). Check that the page elements you want are activated in the template.
  - choose a style template or modify it.
  - generate the web site in a specific production directory, not in your project directory.
  - visualize the site.

#### Optional steps

4. Illustration and external references phase. Associate :
  - images,
  - documents
  - or URLs.
5. Internal references definition phase : create idea associations (internal links).
6. Bibliographic references definition phase :
  - Use the book manager to create books
  - If you have many books, consider importing them with the import function (use F1 help).
  - Sweep ideas and determine those which will have references to books.
  - In idea properties, add references to books.
7. Keyword definition phase (I recommend the manual method, more accurate) :
  - For each idea, display its properties and activate the keyword manager.
  - At the beginning, the keyword list is empty :
    - create the keywords you need
    - think about the consistency of these words (for example, names only, or verbs only)
  - Add pertinent keywords to the current idea.
8. Custom fields definition phase :
  - If you defined custom fields (options menu), you may give them values.

The optional steps can be done, in two manners :

- For each step, sweep all ideas,
- or for each idea, apply all steps by following the idea properties dialog tabs.

#### Other useful WIT functions.

🔍 To help you to know the progress of your work, you can enlarge the treeview (and set it to display more fields in the options menu) : you now have statistics about each idea : keywords number, associations, references numbers, etc.... This is useful to detect where you are in the steps above.

🔍 WIT also produces a report on used resources (images, documents, or URLs) : this function is available in the "File" menu. It displays a list of images, documents, and URLs that are used in your WebSite : you just have to double click on an element to have the list of all ideas that use it

(cross references).

See also :

- ▶ idea creation
- ▶ idea contents editing
- ▶ Title
- ▶ Editing idea text : WYSIWYG editor *updated*
- ▶ Keywords *updated*
- ▶ Structure templates
- ▶ Style templates
- ▶ Automatic Web site generation
- ▶ A quick tour to create a Web site

### 7.1.3. FAQ : Using other language scriptings

---

*How can I use other languages such as Japanese, Russian or Arabic in my pages ?*

1. Use the Unicode WIT add-on (available at [www.webideatree.com](http://www.webideatree.com) for Windows NT 4, 2000 and XP).
2. In your Windows system, activate the keyboard using your character set (for example Russian).
3. In the "File/Options/Character set" menu, choose the language code you want (for example Cyrillic KOI8-R).

Right to left scriptings such as Hebrew or Arabic are correctly managed by WIT.

### 7.1.4. FAQ : Java virtual machine in Windows XP

---

*When I build a Web site with left frame access table (Java Applet), on Windows XP, MS Internet Explorer doesn't load it.*

Yes Microsoft excluded Java virtual machine from Internet Explorer release on Windows XP.

You may download a Java virtual machine at : <http://java.sun.com/getjava/index.html> (5.4 MB).

## 7.2. Documents and web sites

---

### 7.2.1. FAQ : My document is locked

---

WIT always creates a 0 sized lock file when you edit a document : `document.wit.lock` if your document is named `document.wit`. The lock file is located in your document folder.

This is useful to prevent concurrent access by other users, and the lock is removed when the document is closed. A locked document can't be saved to avoid any change on the file.

Sometimes, if the WIT session is killed in any way, the lock remains : you just have to remove the null size `document.wit.lock` file.

### 7.2.2. FAQ : Create a menu to main pages *updated*

---

*I might want to create a framed or tabled navigation menu of the main headings.*

There are several methods, depending on your needs. Use one or several :

- Use the [Drop Down Menu widget](#) in page header.
  - Edit the structure template
  - Choose the "Page" tab
  - Enable and edit the Page Header
  - Insert the WIT variable (menu "Links") : `< @dropDownMenuWidget/>`
- Use the [Simple Tree Menu widget](#) in the page left or right pane.
  - Edit the structure template
  - Choose the "Page" tab
  - Enable and edit the Page Left Pane (for instance)
  - Insert the WIT variable (menu "Links") : `<@treeWidgetLevelX/>`
    - where X is either 1, 2, 3, 4, or N sublevels.
- Use the [Java applet menu](#) in a separate frame.
  - Edit the structure template
  - Choose the "Generic" tab
  - Activate the left frame option
  - Adjust its width, foreground and background colors
  - Choose the "Table of contents" tab
  - Adjust Java applet menu parameters : depth, numbering, font...
  - Build the web site (Ctrl + E)
- Manually insert selected links in the page header, left or right pane.
  - Edit the structure template
  - Choose the "Page" tab
  - Enable and edit the Page Header (for instance)
  - Manually type text and insert links to ideas.

### 7.2.3. FAQ : Modify the main title

---

*I'd like to modify the main title which appears on each page.*

The display of main title is set by the current structure template.

To display or hide this title, edit the structure template (expert mode), choose the "Page" tab, and (un)check the "Generate a text header on each page" box.

To change the title, edit the text field near the check box.

To change the title appearance, edit the style template (expert mode), select the "PageHeader" class, and specify its CSS rules.

## 7.3. Text editing and images

### 7.3.1. FAQ : Type HTML text

*Can I directly type some HTML tags ?*

Yes, all HTML tags are taken in account, and the result depends on from which view you entered the text : "Normal" or "Html" view.

In the "Normal" view, text is displayed literally : if you type "<i> Once </i> upon a <b> time </b>", you'll get "<i> Once </i> upon a <b> time </b>" in the final display (check in the "Preview" view).

In the "Html" view, text is analysed and HTML tags are interpreted : if you type "<i> Once </i> upon a <b> time </b>", you'll get "Once upon a **time**" in the final display (check in the "Preview" view).

**Advice : don't use relative links in your text (they won't be analysed at generation time and may appear broken).**

### 7.3.2. FAQ : Paste HTML text

*Can I paste HTML text from another HTML Editor ?*

Trick (undocumented WIT feature) : you can insert or paste any HTML code in the text of a WIT idea, but you have to follow some rules (again), if you want that eventual inserted links (images or URLs) to be correct once the Web site is generated.

If there are images, anywhere in the text, follow the following steps :

1. Create an "extra" directory under the current WIT document directory (all its contents will be copied into the Web site root (say "/webroot") at generation time, in "/webroot/extra").
2. Copy the image files you want to insert in this "extra" directory.
3. In the idea text, type direct HTML tags, for example :  

4. You may also add <CENTER>and </CENTER> tags to center the images ...

Note that the reference begins by "../extra/" because the image will be copied into "/webroot/extra" whereas the HTML page file will be generated in "/webroot/text".

You may also use absolute reference : in this case, you don't have to use the "extra" subfolder.

The HTML tags will be replaced by the image (you may move and resize it with the mouse, or delete it, but you won't be able later to modify the HTML tags : just delete them and insert new ones).

In the same way, you can insert URLs, but keep in mind that they should be absolute or keep being consistent once the Web site is generated (they are considered as "foreign" references by WIT and are not analysed nor handled by it).

To be more general, if you have a special or sophisticated HTML code to insert (tables, forms ...), you may produce it with a dedicated HTML editor, and paste the result into your WIT idea page (choose the "HTML" tab for the insertion).

You can even paste some text from Word or other word processors. Try it !

### 7.3.3. FAQ : Paste HTML text from Word

*Can I paste HTML text from another Word processor ?*

YES. Current format and styles will be kept, as far as possible (page layout is more powerful with Word than with HTML).

Use the Ctrl+C / Ctrl+V sequence between Word and WIT.

See also :

► [FAQ : Paste HTML text](#)

### 7.3.4. FAQ : URLs

*I'm a bit confused on URL's: when I type a url into the edit window it appears to become an active link (which is great) and I assume it will export as a link. What's the advantage of the links entered under the item property? Where does this appear? I entered a test one and it doesn't seem to appear in the item body.. is the link used for the title?*

You may type absolute URLs in the text area of an idea (of a page). You may also add (absolute) URLs in the Links tab of idea properties : these URLs will be grouped together and displayed apart from the main text, such as a footer note, provided the corresponding options are enabled in the current structure template. In this case, edit the structure template (expert mode), choose the Idea tab in the dialog, and add the 'external links' item in the rightmost list (choose also the display order of your items). Save and you'll see your URLs in the preview tab at the place you specified. This applies to all the document.

(Why such a choice ? It is a personal point of view but I feel annoying all these links everywhere in hypertexts : the problem is to stay focused on the reading and I find it better to have linear reading, such as in a book, with footnotes.

I have written WIT in the purpose of publishing electronic books, and web sites with presentation and style rules.)

### 7.3.5. FAQ : Relatives URLs

*I assume images etc inserted into WIT in the normal WIT way will stay relative etc.*

Yes you are right.



### 7.3.6. FAQ : Side by side images

*I'd like to put images side by side instead of in top / bottom manner.*

It's is quite simple :

- Edit the style template (expert mode),
- Choose the `.ImageSubgroup` item (it is a sub block made of an image and its text legend).
- Add the property :  
`float:left;`

Provided there is enough room, the images will be displayed side by side, in all the Web site.

### 7.3.7. FAQ : Alternate text and images

*Is it possible to arrange a page with text, then a photograph below, then more text, then another image?*

Not easily, the text section can't be mixed with the image section. Web sites generated by WIT follow rules which apply to all pages (sometimes hundreds of them).

However you can :

- split your pages in smaller ones,
- display the image section before the text section (struture template in expert mode), and make the image section float left, so the text will flood at the right part of the screen and under the text group (style template, ImageGroup CSS properties). See the hitchcock-en.wit sample (first page, and templates values).
- you may apply the trick described in the next page.

See also :

► [FAQ : Insert image in the middle of text](#)

### 7.3.8. FAQ : Insert image in the middle of text

*How to insert an image, anywhere in the text ?*

Follow these steps :

1. Set the cursor in the WYSIWYG editor where you would like to insert the image.
2. Use the "Internal link" menu in the editor to insert an image
3. Select the image

This will insert a HTML tag like `` that you may change in the HTML view.

Note that the reference begins by `"../../wit-icons/"` because the image will be copied into `"../webroot/wit-icons"` whereas the HTML page file will be generated in `"../webroot/text"`.

You may also use absolute references.

The HTML tags will be replaced by the image (you may move and resize it with the mouse, or delete it, but if you want to modify them, you have to activate the pure HTML editor (HTML tab), and update the HTML code).

### 7.3.9. FAQ : Center images updated!

*I cannot figure out how to centre photographs.*

Images can be inserted, either in the WYSIWYG editor, either in the idea properties dialog.

**Images Inserted in the WYSIWYG editor :**

- click on the image, then click on the "center" icon in the WYSIWYG editor toolbar
- open the Style template, and edit the `.TextImage` item (located in the `.TextGroup` elements) :
  - remove any "float" property that could be mentioned there.

**Images inserted in the Idea properties dialog :**

- Edit the style template (expert mode), and choose the `ImageSubgroup` item.
- suppress `float` properties (you can comment them out with `/*` and `*/`).
- add the following properties :

```
text-align:center;
```

This means that all the images inserted through the idea properties dialog will be centered in the whole document.

### 7.3.10. FAQ : Insert forms and tables

*I'd like to insert forms or tables.*

You may insert forms or tables in your WIT pages, either in the pure HTML code edit view (HTML tab in expert mode), either in the normal edit view (Normal tab).

Although direct typing is easy, building tables and forms with dedicated tools prior to paste them in your WIT pages is a good practice.

Another good idea is to prepare your tables with Word or Excel, and paste them in your WIT page.

### 7.3.11. FAQ : Create a photo album

---

Photo albums creation is very easy :

Automatically [import folders](#) content with the "Ideas / Import folder" menu :

- Keep only image file suffixes
- Eventually check the "Recurse through subfolders" option
- Check the "Make album" option : WIT will also create thumbnails and treat any image as a separate idea.

You can also do it manually, and locally :

Automatically import images from a folder ("Ideas / Import images" menu ).

Create a thumbnails table with the idea subtree including the imported images ("Ideas / Build thumbnails" menu ).

See also :

- ▶ [Import folders](#)
- ▶ [Import multiple images from folder](#)
- ▶ [Build thumbnails automatically](#)

### 7.3.12. FAQ : Update a thumbnails table

---

*How to update a thumbnail table after some changes in the images ?*

Activate the HTML raw code view (expert mode).

Select and remove the HTML code which describes the thumbnails (the CSS class ".TocImagesGroup") DIV group), and remove the corresponding /DIV closing tags.

Ativate the "Page preview" view.

Create a new thumbnail table ("Ideas/Build thumbnails" menu).

### 7.3.13. FAQ : Undo edit changes

---

An Undo/Redo feature in text editing is available : just use Ctrl + Z (undo), and Ctrl + Y (redo).

In most dialogs, it is a single level undo/redo .

In the "Normal" and "Html" views it is a multi level undo/redo.

### 7.3.14. FAQ : Create links between ideas

---

*How do I create a hyperlink in one Idea to another Idea ?*

There are two ways to create links between ideas :

1. The first method is named "associations" between ideas. Select the idea in which you want to create a link, open its properties, and select the "Associations" tab, click on the "add" button in the "other associated ideas" section. This opens a selection dialog : select the ideas to which you want to link and validate. Associated ideas (links) will be shown in your page, according to the structure template option (expert mode).
2. The second method is a link in the idea text. You have to activate the "Normal" view (the WYSIWYG HTML editor), and click on the rightmost icon (a double chain). Select the ideas you want to link, and WIT will insert all the links for you. You may edit/change the inserted text (by default, it is the idea titles).

## 7.4. Structures and styles

---

### 7.4.1. FAQ : CSS

---

**Feature Not in freeware edition**

*"Experimented users (expert mode) may control Web site and pages structure, and define a uniform and elegant presentation style thanks to cascading style sheets (CSS standard). "*

*I'd like to do some changes or create my own css. Unfortunately, I do not see how this could be done. All css seems to be predefined and not possible to change to, only to a very small extent. I do not see how to create an own css either.*

1. The structure of the Web site you generate with WIT is predefined. You may choose the items to display and their order with the structure template parameters (expert mode).

2. To each structure item corresponds a CSS style item (a CSS class). You can't change CSS class names, but you can fully specify, with no CSS1 or CSS2 limitations, the style of each class. Please look at the examples, for instance the user's manual : activate Expert mode, and edit the style template. You'll see how predefined classes inherit from each other (indentation), and how each element can be defined. At that point, all the CSS language is available.

See also :

- ▶ [CSS 1](#)

### 7.4.2. FAQ : CSS user stylesheets

---

*I'd like to build my own CSS stylesheet*

**Feature Not in freeware edition**

You may add your own CSS classes and properties in the style template (expert mode).

You just have to refer to your own classes in the HTML tags you will use in the raw HTML editor (HTML tab, expert mode).

Note that most of predefined WIT CSS classes names are read-only. You still may change their properties.

### 7.4.3. FAQ : Custom templates, custom CSS classes

---

*One question on html export.*

Feature **Not in freeware edition**

*Is it possible to edit a personalised template. Or is it possible to edit existing templates. Also can I insert html into items? And can I edit the headers and footer html for each page (this might be covered if I can edit a template) ?*

Yes, it is possible to create and edit personalised templates : just switch to Expert Mode in the File/Options menu. This enable access to structure and style parameters of current template.

You may also insert HTML into parameter fields, but keep in mind that eventual relative URLs may not be resolved at web site generation (for example images) since the tool won't analyse the HTML code you insert.

Header and footer can be edited in the template, and will apply to all pages.

### 7.4.4. FAQ : Different backgrounds for different pages

---

*Can we have different backgrounds for different ideas / pages. It appears that presently one background applies to remaining pages ?*

You can specify a background for the whole site in the structure template.

You also can specify individual background for each page : edit the idea properties, choose the "Display options" tab, uncheck the "Use document defined general display options", and set the background. This takes precedence over the template rule.

## 7.5. Sound

---

### 7.5.1. FAQ : Getting rid of the preview browser noise


---

This sounds originates from Internet Explorer. To deactivate it, open the Windows configuration panel (sounds), and uncheck the sound associated to the "Windows explorer / Start navigation" item.


### 7.5.2. FAQ : Modifying WIT sounds

---

The files are located in "<DATADIR>\Sound".

 snd-start.wav is used when the program starts and ends,

 snd-err.wav is used to signal an error,

 snd-ask.wav is used when a question is asked.

These files can be replaced by other \*.wav files (by keeping original names).

## 7.6. Default WIT resources

---

### 7.6.1. FAQ : Adding new arrows, bullets... images

---

New images can be added to WIT default images.

They must be GIF or JPEG files, and copy them in the "Arrows", "Backgrounds", "Bullets", and "Rulers" subdirectories of <DATADIR>.

### 7.6.2. FAQ : Adding new structure templates in the toolbar

---

To add new structure templates in the "Structures" tab of leftside toolbar, place them in the "<DATADIR>\Structures\" directory.

The next time WIT is launched, the new templates appear in the toolbar.

### 7.6.3. FAQ : Adding new style templates in the toolbar

---

To add new style templates in the "Style Templates" tab of leftside toolbar, place them in the "<DATADIR>\Styles" directory.

The next time WIT is launched, the new templates appear in the toolbar.

### 7.6.4. FAQ : Modifying default HTML pages

---

WIT has default HTML pages for home page and frame top pages. To modify these pages, just modify their files with an HTML editor, in the "<DATADIR>\Templates" directory.

There are two file sets : one for English (ending by -en), one for French (ending by -fr).

- HomePage-en.html is the default home page. It must contain at least a link to the text/index.html page to address the rest of the site.
- titlePage-en.html is the title page, displayed in the top frame when defined.
- accessPage-en.html is the left frame page (if defined) : it is typically used to give shortcuts to main site items (table of contents, or any other choice). (By default, WIT automatically generates a page using a Java applet).

See also :

► [FAQ : Modifying site default home page and frames](#)

## 7.7. Site building

---

### 7.7.1. FAQ : Meta tags

---

*Is there a way to specify html meta tags for individual pages or branches?*

Yes, through keywords.

All keywords associated to an idea are copied in meta tags at generation time (if allowed by the structure template).

### 7.7.2. FAQ : Weight of generated HTML pages

---

*How can I get lighter HTML pages at build time ?*

If you encode your pages with iso-8859-1 charset (Western Europe), the size of generated HTML files will be the half of other charset encodings (Unicode, except UTF-8, where each character is 2 byte encoded).

If you use the UTF-8 encoding (better and suitable for almost all languages), the size of generated HTML files will be optimized (ASCII characters are 1 byte encoded, others are 2 byte, 3, 4 or more byte encoded...).

### 7.7.3. FAQ : Controlling the names of generated web pages

---

Feature **Professional ed. only**

The name of each generated HTML page may be chosen at idea creation time, by activating the function "*Choose name of generated pages*", in the "*File/Options*" menu, in expert mode.

### 7.7.4. FAQ : Modifying site default home page and frames

---

Feature **Not in freeware edition**

To define custom HTML pages for the home page and the pages located in the optional frames, use the structure template option "Prepared Html file" in the "General" tab of its dialog box (expert mode), and select the pathname to the page.

By default, WIT has minimum files, located in `<DATADIR>\Templates`.

See also :

► [FAQ : Modifying default HTML pages](#)

### 7.7.5. FAQ : Generated files

---

*When WIT generates a website does it recreate all pages and files even if they haven't changed since the last update?*

No, if you generate again the web site in an already populated directory (by a previous build), unmodified files will be kept, so you may optimize a subsequent FTP upload.

### 7.7.6. FAQ : What files to upload by FTP ?

---

*I uploaded my WIT project (the \*.wit file) on the server and there is no Web site.*

First, don't mix up the original WIT document (\*.wit file) with its resources (eg the images you refer to), and the generated web site.

Ideally, the WIT document is located in a subdirectory of the WIT "Project" directory, and the resources may be anywhere on the disk.

The generated Web site is located in a special directory that you choose at generation time. Let's say "C:\webgen", for example. (It's better not to generate the Web site right into the project subdir, so the source files and the generated ones remain separate).

The only files you have to upload are the generated ones (the "C:\webgen" contents). They **are** the Web site and their layout must be kept while uploading.

In "C:\webgen" you have the site main pages and styles,

in "C:\webgen\text" the generated pages,

and in "C:\webgen\wit-icons", copies of your original source images.

C:\webgen is expandable since all its contents are either generated or copied. You may delete it whenever you want, and regenerate the site with WIT.

Your WIT project directory and its data (\*.wit, source images...) are precious. Keep them safe, but there's no need to upload them.

While uploading "C:\webgen", you should keep the original file structure (don't move the files or web site links will be broken) :

- files in the C:\webgen dir will go in the site root (/),
- files in C:\webgen\text dir will go in /text dir,
- and those in C:\webgen\wit-icons will go in /wit-icons dir.

### 7.7.7. FAQ : Site FTP upload

*I'd like to see the files on the remote server when I upload my Web site.*

You may use an external FTP client such as the smartFTP freeware ([www.smartFTP.com](http://www.smartFTP.com), a really good one).

### 7.7.8. FAQ : Upload modified files only

*I'd like to upload only modified files (and not all files).*

You have to :

1. generate your Web site in the same directory as before : so only updated files will be modified,
2. upload only recently modified files, by checking the option "Transfer only modified files within last x minutes" in the FTP dialog box.

**NOTE** : be careful with images, which dates can be older than you think (creation or modification date). Be sure to transfer the whole site at least once (do not check the previous option the first time).

### 7.7.9. FAQ : FTP upload problems

*I get errors when uploading my files with FTP.*

FTP errors types are numerous, and they can have multiple origins. Most frequent causes (with variants) are :

- connection errors (bad user name, password, server...).
- insufficient permissions on the distant server (for instance read only files).
- transmission errors on the DSL or phone line (relatively frequent with the phone).
- heavy load on the network, and timeouts either on the sever on your computer.

In all cases WIT will give a basic diagnosis. To better understand the problem, I advise you to use an external FTP client (such as smartFTP, CuteFTP, etc...). You'll be able to graphically upload your files and get a better feedback.

Once your problem is solve, you may use WIT FTP again and benefit of its integration and its optimizations in only modified files upload.

### 7.7.10. FAQ : Images replaced by red "X" once on Internet

*When I browse my site on the Internet, no image is displayed. I get a red "X" instead, whereas everything is normal when I test my website on my local PC.*

This problem is well known : carefully check your image file names, specifically lower and upper case letters. For instance, "image.jpg" is different from "Image.JPG".

This is usually masked by Windows (which does not care enough about letter case), but is revealed by case sensitive operating systems like Unix. Your Internet Service Provider's machine is likely a Unix machine.


Another cause : images were not transferred by FTP (in this case, check for instance that the "transfer only recent modified files" optimisation is not active : the images you use in your site may have earlier dates).


## 7.8. Advanced features

### 7.8.1. FAQ : Building a contextual help system

For programmers, WIT helps to easily make contextual help system for software in development phase : here are the main steps (illustrated with Microsoft Visual C++, some minor changes should be made for other programming languages).


 In WIT, activate expert mode and choice of generated HTML pages names ("File/Options" menu).


 Create the structure of the user manual of your software. This structure should match the main functions of the software to document, and the main dialog boxes. Give an explicit name to the files associated with each idea.

 Write the user manual using WIT (define keywords, choose structure and style templates ; check that the structure template generates ideas in separated pages (tree : one idea - one page, see the "page" tab of the structure template dialog).

 Define where the web site will be generated, for example "c:\MyInstallDir\Help", and generate the site there.

 In Visual C++, overload the **OnHelpInfo()** method, in all your derived dialog classes (dialog boxes (derived from *CDialog*), views (derived from *CView*), tabs (derived from *CPropertyPage*) of property sheets (*CPropertySheet*)).

 In the registry (for example), create a key (for example *MyHelpValues*), where you will store values (strings) like : "valueName" = "helpPagePath". For example, for a *CMyDialog* dialog box to which the "My dialog" idea has been created with an associated file name "my-dialog", create the key : "**CMyDialog**" = "**text/my-dialog**". Help specific pages are generated in the "text" subdirectory of site root : "c:\MyInstallDir\Help" (no need to append a suffix to the page names).

 In the *CxxxxApp* class (where *xxxx* is the name of your being documented software), create a method such as :

```
void CxxxxApp::ShowHelp(CString helpId)
{
    // retrieve the registry key (example : MyHelpValues) where
    // the values such as "CMyDialog"="text/my-dialog" were
    // stored.
    CString regKey = ... ; // Retrieve the value associated to the value name "helpId".
    CString helpFile = ... ; // Retrieve the help web site root path
    CString helpPath = "C:\MyInstallDir\Help"; // for example, or use the registry
    if (helpFile == _T(""))
    {
        CString msg;
        msg.Format("Soon help on \"%s\", file %s ...", helpId, helpFile);
        AfxMessageBox(msg);
    } else {
        // Build the path to help file
    }
```

```

        CString helpFile = helpPath + "\\\" + helpFile + ".html";
        // Launch the browser on the help file (example MSIE)
        CString iePath = "C:\\Program Files\\Internet Explorer\\iexplore.exe"; // (or ask the registry for MSIE path)
        _spawnlp(_P_NOWAIT, iePath, "ie", helpFile, NULL);
    }
}

```

🔗 In the **CMyDialog::OnHelpInfo()** method, call the help display method, passing class name as parameter (context) :

```

BOOL DDispBookPage::OnHelpInfo(HELPIFINFO* pHelpInfo)
{
    theApp.ShowHelp("CMyDialog");
    return TRUE;
}

```

🔗 Compile and launch the *xxxx* program. Display the dialog handled by *CMyDialog* and press F1 key : the web navigator is launched and displays the correct page (here "C:\MyInstallDir\Help\my-dialog.html").

🔗 User manual can then be modified, and again automatically generated in the same place : there's no need to recompile the software.

## 7.8.2. FAQ : Generating different documents in a single web site

Several documents may be generated in the same web site.

A simple method consists in creating as many void subdirectories as documents, in the site root directory, and then generating each document in its target subdirectory.

You then have to modify the topmost home page to have a link to each sub site (index.html file in each subdirectory).

## 7.8.3. FAQ : Dynamic behavior

*How to add a dynamic behaviour to my web pages with WIT ?*

Feature **Not in freeware edition**

Inserting script code is easy. You may insert Javascript, PHP, ASP or any server-side code, in the pure HTML edit view (HTML tab, in expert mode).

You may either directly type the code, or paste it from external development tools (after validation). The second method is the best.

You also have to configure WIT (File/Options menu) to choose the suffix of generated pages files : for example, PHP pages should have a .php suffix.

## 7.8.4. FAQ : Generating PHP pages

*How do I activate my PHP scripts in my pages ? Do I need rename them ?*

Feature **Not in freeware edition**

You just have three things to do :

- turn the WIT file suffix to .php (or phtml)
- insert your scripts in the raw HTML code (in the WIT HTML view tab)
- check that your web server (or your ISP web server) accepts PHP pages and recognize your suffix.

All WIT generated pages will have the new suffix, even the "pure" HTML ones ; it doesn't matter : if a page contains only HTML tags and is named - let's say - mypage.php, it will be correctly served by the Web server, and you'll see it as if it had been named mypage.html. So no need to manually change page suffixes.

## 7.8.5. FAQ : Importing ideas from CSV files

*How to import ideas from CSV files ?*

*What's these files structure ?*

To import ideas, proceed in two steps :

1. Prepare your data in a \*.txt file. This file will contain a record per line. Each record is made of fields separated by a special character (for instance ; ). The order of fields does not matter since you'll specify in the next step how they will be interpreted. Just check that all lines have the same number of fields (even when void). A faster method to produce a CSV file is to use a spreadsheet like Excel and save a document as a CSV file (\*.txt).

2. Data import. Use the WIT menu : "Import/Export -> Import ideas from CSV file". It will display a dialog where you'll specify how the record fields must match those of WIT ideas (press F1 for more information). You can also study the tutorial which fully explains this.

## 7.8.6. FAQ : Importing bibliographies in standard formats

*How can I import bibliographies in predefined library formats ?*

WIT has today its own internal format for publications management, but you can import bibliographic data from any CSV format provided you map it to WIT format through the import dialog box (in the menu "Publications / Import a list (text)", press F1 for help once the dialog is displayed).

If you want to create more specific collections (for example, a database of URLs with comments, images, etc...) and want to create an index of it, you also can create a WIT document, customize its custom fields (Menu "File / Options"), and import data in the same way as books (with the "Import/Export -> Import ideas from CSV file" menu).

You can also associate books or other ideas to ideas. What's more, ideas are indexable (with keywords). On the other side, books can be filtered by theme, whereas ideas can be hierarchised.

## 7.8.7. FAQ : Exporting ideas in a single file, or in a Microsoft Word file

*How can I export all my data into a single file, for instance a Microsoft Word file ?*

You can build a WIT website where all ideas are built in a unique HTML page :

- open the structure template (expert mode),
- choose the "Page" tab,
- and check the option "same level Ideas grouped on a single page from level 1"
- build your website
- open the single HTML file with Word
- save it as a Word document.

Note : you may also remove additional navigation items such as child links or the hierarchical position from the structure template to get a more compact file. You may also remove all bullets and images.

### 7.8.8. FAQ : Print the user's manual

---

*I would like to print the user's manual (all at once). Moreover (with MS Internet Explorer 5.5), page breaks ignore the images (they are cut anywhere).*

To print a document (the user's manual) on a single page, change the structure template as explained in the link below (or use the predefined "Book" template). All ideas are built in a single web page, and some elements (child idea links, hierarchical position...) are ignored.

When printing with MSIE 6, page breaks are properly set and images are not cut any more.

See also :

► [FAQ : Exporting Ideas in a single file, or in a Microsoft Word file](#)

### 7.8.9. FAQ : Table of contents and index update

---

*How to update the table of contents and the index ? I've added new ideas and keywords, and can't see any change in the table of content and in the index.*

WIT does not systematically rebuild the TOC and the index whenever there's a change : it would take too much time for important sites.

However, the TOC and the index are rebuilt in three cases :

- when the document is loaded,
- when the website is built,
- when the update is forced.

To force the update, open the structure template, activate the "Table of contents" or "Index" tab and close the dialog : the TOC and the index are rebuilt.

Moreover, do not make a confusion between site building (Ctrl + E key) where the whole site is built in a folder that you specify, and the page preview where only the selected ideas are previewed from files that are generated in a temporary directory.

### 7.8.10. FAQ : Table of images content

---

*I don't see all my images in the table of images. Why ?*

The idea must be eligible to be displayed in the table of content.

See also :

► [Check an idea for table of contents updated!](#)

## 7.9. Known problems

---

### 7.9.1. Focus on the preview area

---

When the preview area has the focus (if you clicked in it), specific idea management functions are disabled (greyed menus).

Click in the main view to take focus back and activate view functions.

### 7.9.2. Colors are not well displayed

---

You should use a 16 bit display resolution or higher (Windows desktop options).

### 7.9.3. Background image is not properly displayed (CSS background-image property)

---

CSS background-image is correctly managed by WIT. However, the image file that is specified by the URL is not automatically copied by WIT (because it is not identified as a resource).

Three solutions (the first is recommended) :

1. Copy the image in the "extra" subdirectory of document save directory, and check that in the CSS rule, the URL looks like `"../extra/image.jpg"`.
2. Use this image in at least one idea (in the "links" tab of idea property dialog box) : it will be so copied with other images in the `icons` subdirectory of site root. Check that the URL looks like `"../wit-icons/image.jpg"`.
3. Or, manually copy the image in the `icons` subdirectory of web site root. Check that the URL looks like `"../wit-icons/image.jpg"`.

In all cases use relative URLs.

See also :

- ▶ [Uri](#)
- ▶ [WIT Documents](#)

#### ● 7.9.4. Office 2000 installer starts just after WIT launch

---

This problem has no consequence for your Microsoft Office 2000 installation, nor for WIT.

Just press the Cancel button in the Office 2000 component installation message. This message won't appear any more.

### ● 8. Glossary

---

#### ● 8.1. CSS

---

*Cascading Style Sheets*, standard of style sheets definition, emitted by the W3C.

See also :

- ▶ [Definition : Style sheet](#)

#### ● 8.2. ISP

---

Internet Service Provider. The organization which hosts your web site.

#### ● 8.3. Style sheet

---

Set of presentation and style rules.

See also :

- ▶ [Definition : CSS](#)

#### ● 8.4. FTP

---

*File Transfert Protocol*, standard internet file transfer protocol, based on TCP/IP.

#### ● 8.5. HTML

---

HyperText Markup Language, web page description language, using tags.

#### ● 8.6. TCP/IP

---

*Transfert Control Protocol / Internet Protocol*, general and standard data exchange protocols between computers.

#### ● 8.7. W3C

---

World Wide Web Consortium, in charge of Internet standardization.

---

[Table](#) | [Images](#) | [Bibliographie](#) | [Glossary](#) | [Index](#) | [Faceted index](#)

Web Idea Tree 5.20 User's Manual - Page 1 / 1

Last update 03/29/2008 at 23:07:57

Copyright © Christophe Guibert, 2001-2008. All rights reserved.

