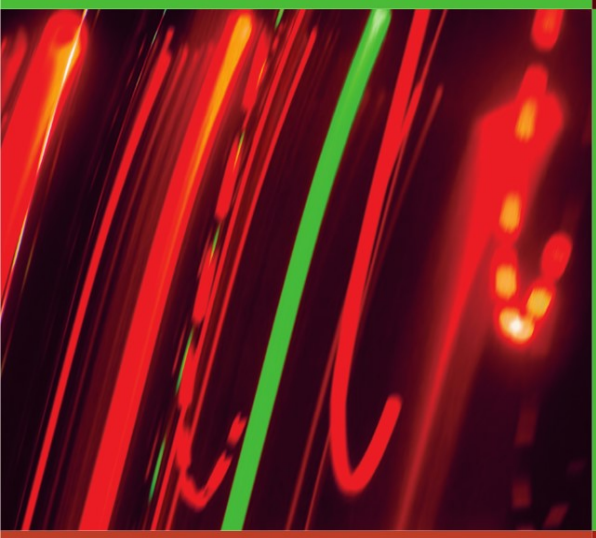


Anja Ebersbach
Markus Glaser
Richard Heigl

Wiki

Web Collaboration

CD-ROM

Springer

Wiki

Web Collaboration

Anja Ebersbach · Markus Glaser · Richard Heigl

Wiki

Web Collaboration

Foreword by Gunter Dueck

With 81 Figures, 47 Tables and CD-ROM

Springer

Anja Ebersbach
Ostengasse 10
93047 Regensburg
Germany
ebersbach@wiki-tools.de

Markus Glaser
Brunhuberstr. 14
93053 Regensburg
Germany
glaser@wiki-tools.de

Richard Heigl
Ostengasse 10
93047 Regensburg
Germany
heigl@wiki-tools.de

Translated from the German „WikiTools“ (© Springer-Verlag 2005, ISBN 3-540-22939-6) by Andrea Adelung.

Library of Congress Control Number: 2005930459

ISBN-10 3-540-25995-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25995-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springeronline.com

Printed in Germany

Cover design: KünkelLopka, Heidelberg
Typesetting: Camera ready by the authors
Production: LE-TeX Jelonek, Schmidt & Vöckler GbR, Leipzig
Printed on acid-free paper 33/3142/YL - 5 4 3 2 1 0

Foreword

A book about wikis!
That's what people need.
Because with wiki technology, lots of people can freely work together – they can even generate very large works in the intellectual realm. See for yourself:

Today, we still marvel at our massive church buildings, each constructed over a period of centuries, requiring an immense amount of labor and often bearing the cultural stamp of all of the epochs during which it was created. Someone just has to begin by placing stone upon stone and motivate the people nearby to help out a bit. In places where such enthusiastic fellow men and women lend a hand and donate materials, great things can emerge. And where they are absent? Either scant ruins remain, or the iron will of a pharaoh is required, an army of drivers, the sweat of a people and a mountain of gold. Great things can also be created in that way – take the Pyramids: a clear concept, no blending of styles, pure will.

Those are two very different paths. The one entails passionate people devotedly building something together for the common good; the other: a single will manages a variety of resources to achieve a set goal.

Wikis are tools with which lots of people with a minimum of organization, planning, money and time can create something together and communicate with each other from several scattered computers or over the Internet. Wikis are the technology for that first path of volunteers with a common idea.

This book introduces wikis and provides you with enough tools to create your own wiki; your own work platform. Yet the book will also invite you to join the animated discussion on what one can do with wikis and where it is better not to "abuse" them. It is the intriguing question of those two stimuli: enthusiasm and will.

Allow me to explain with an anecdote. Some time ago, I received a letter from a reader regarding my books. Someone wrote that he had inserted my name in the Internet lexicon 'Wikipedia'. He wrote that he initially only added a bit of preliminary data on me and would continue to work on it. I was mighty proud that I was now to be listed in an encyclopedia, and checked on the Internet right away under "Gunter Dueck", but I didn't find anything. I found strange messages indicating that there had been an article related to my name, but that it had been deleted. The reader I mentioned was angry, and he tried posting his article again, but it again was deleted. Days later, another "person" wrote something reasonable. That remained on the Internet for a few days, but disappeared again, ostensibly due to violation of copyright laws. Now wide awake, I then attempted to find the email address of the person deleting everything. I wrote: "Hey, why?" The answer: "The image most likely violates copyright, and the text presumably as well. I am one of the authorized persons assigned to quality and legal issues." I argued: "The picture of me was taken by my daughter Anne in our garden; I give it to everyone. And the text is taken from the cover of one of my books. I grant my permission to use that." Three hours later, "my" entry was back online again. Now I get nervous wondering if the article has been defaced or deleted. Do you understand what I mean? Anybody can do with me what he wants! If that is the case – is everything true that is listed in Wikipedia? Can I treat the information as being just as authentic as what I find in a book? Will anyone award me damages if, through a false entry in Wikipedia, I lose a bet or my reputation as a scientist? Questions abound regarding a variety that grows on its own power! Of course, you could also see it positively. "Wiki lives!" It changes, develops, grows – however, it needs to be weeded, and its garden protected from thieves (lexical vandalism).

Wikipedia is one of the truly colossal wiki projects. Ten thousand contributors are at work on a single intellectual monument. A number of masters run around and find out when someone has cheated. Controllers verify whether the building code is being followed. Anyone can participate whenever and however he or she likes. No time pressures, hardly any regulations, and no pay for the volunteer work – only a profound sense of accomplishment. "One stone of that pyramid is from me!" – That might be something a Wikipedia contributor might exclaim.

Wikis are exceptionally suitable for all such projects of several volunteers. Would you like to connect the parents of students of the Bammental grammar school? All of the members of a sports association? All of the astronomers of the world? All Linux freaks in

your company? Do you want to create something in a community with others? To maintain a community? Then you need a wiki! But which one?

The best one! As of today – I just checked – the German Wikipedia homepage reads: "We have just converted the software to MediaWiki 1.4. Please report any problems here…" This book also presents the wiki technology with the aid of the open source software MediaWiki, which you can download from the Internet. So, if you would like to use MediaWiki for your project, then you are in good hands – several volunteers are working on follow-up versions of "your" software – of that you can be sure! However, if you really want more, or if you perhaps even want it *all*, so to speak, then you'll enjoy the detailed description of the high-end software TWiki. This program can do much, much more – it can do it all, anything currently technically possible – it offers a cornucopia of supplementary functions, from presentation to drawing to calculating. And as such, up in the technical heights, where anything is possible – you'll have a bit more difficulty in the installation process, I believe, and will have to be more accomplished in its operation. What is the best wiki? "The standard!" call some, "Extreme wiki!" shout others. And, as usual, both sides are right.

So it's got to be a wiki? Well, that's no problem with this book! However, the book will also seriously discuss what a wiki cannot currently accomplish and what it should not even attempt. A wiki should not be "abused" for the "second path" of accomplishing things. The second path would be "finally" turning a community endeavor of spontaneous enthusiasts into "a real project." We need a plan! We have to organize who is going to do what! We keep lists on how much each person has accomplished! We check progress and define goals! We do everything efficiently and do not waste money! Does it always have to be the newest software? Can't we save more money?

Imagine if a company were building Wikipedia. Then there would be the equivalent of the pharaoh's will. A plan of necessary words would be generated. The words would be prioritized according to importance, experts present, and to the difficulty in writing new entries. Managers would fervently search for new sources where something could be copied or used more than once. They would set the pay for entries and monitor the rapidity of the work. The once volunteers would make sure they got everything done quickly – without paying attention to details – just quickly, according to plan, and the respective remuneration.

That would be the path of the "project" and of efficiency. A pyramid is built according to plan and schedule. It is made to be completely uniform and flawless – nothing about it is spontaneous! Nothing is voluntary. Everything bends to the central will: the project goal.

However, MediaWiki only invites volunteers to work on it! The wiki technology does not assign jobs or assess performance. It does not dole out punishment for insufficient output or errors. It does not organize workflow. So, if you create a wiki for yourself, you should know what a wiki will and will not be able to do. It can take a great deal of spontaneity and create something beautiful, common, or great. Yet it cannot truly be used as a tool to efficiently assert someone's will.

The authors of this book offer a fresh introduction to the topic. They are not afraid to take part in the discussion of the pros and cons of wikis. They discuss the current dialogue using several charming details. The book's style is inviting – it is very factually written, but somehow charming nonetheless. While reading it, I imagined the authors as the most passionately motivated of all wiki enthusiasts.
They write: a book about wikis!
And they are confident: That's what people need.

Gunter Dueck

The Authors

Anja Ebersbach studied Information Science, History and English. She has filled various teaching positions in the last four years at a variety of universities and technical colleges, and is active as a freelance IT trainer. She is currently working on her doctorate at the University of Konstanz.

Markus Glaser studied Information Science, English and Psychology and is employed as a freelance programmer and IT instructor. He is currently working as a scientific assistant at the library of the University of Regensburg and has just begun working on his doctorate at the University of Konstanz on Web-Based Collaborative Processes.

Richard Heigl studied History and German Philology. He completed his doctorate in Contemporary History at the University of Augsburg. He is a freelance instructor for communication training and software applications, and works as a consultant on organizational development with an emphasis on self-organization and cooperative learning.

Content

Preface

Why a book about wikis? About a year ago, Anja came back all excited from a conference on technology and social movements in Munich. There, she had taken part in a workshop on working and organizing with wikis. In the meantime, her excitement is also shared by lots of others. The popularity and renown of this small bit of software can primarily be attributed to the Internet encyclopedia Wikipedia. Yet a number of organizations have also discovered wikis as a simple and versatile tool for their work. For instance, the hackers of Berlin's Chaos Computer Club used it to organize their conference. We were drawn to wikis chiefly due to the opportunity to utilize wikis as organizational software for small groups. We wanted to find out more.

Yet even the attempt to install a wiki was a puzzle. Information and documentation on the Net were few and far between. There was no manual. The only book on the subject, by Wiki creator Ward Cunningham, was very informative, but not sufficient for our purposes. We sensed that wikis could be used for much more than developing texts in a cooperative manner. However, the wiki world is a jungle rampant with political discussions and racing technological development. What was missing was a survival kit, a practical introduction that recommended paths through the jungle. You are now holding an initial cartography of that jungle. Have a look around the wilderness, and become familiar with its inhabitants. Design your environment by setting up your own camp and blazing new trails.

Who Needs this Book?

While writing this book, we envisioned readers who already have some computer experience and are considering whether to work with a wiki or even install one of their own. Undoubtedly, veteran wiki users will also consult this book, and we are sure that especially the second part of this book will be of interest to them.

The Content Concept

Among the multitude of wikis, we have selected two to present to you: the widespread and relatively simple MediaWiki, and TWiki, as a piece of ambitioned wiki software.

- **MediaWiki** is fairly easy to install, and user-friendly in its operation. However, it also offers a series of features, such as a user administration, and it can be employed as a simple but good communication and organizational tool for groups of the most varied of sizes. In short: It is the ideal introduction to wikis.

- **TWiki** is very challenging to install and, especially for technical applications, assumes a good deal of skill and knowledge. Using TWiki also takes some getting used to. Yet for complex projects, it is worth the effort, because it offers a great deal of design options.

The caveman on the cover of the German edition inspired us in putting together our practical examples. Among other projects, he and his clan use a wiki to organize a conference in the Neolithic Age. Admittedly, the combination of 21st-century technology and stone-aged humans is not always consistent, but we hope it is demonstrative and a bit amusing.

In the forefront of the wiki discussion are the social methods of communication that crop up as a result. To this extent, the book also attempts to be something more than a pure software manual. Some will be irritated by the political fifth section. The wiki communities, as the free software culture before them, have provoked extensive socio-political discussions that are conducted at conventions such as the *Wizards of OS* or the *Chaos Communication Congress*. So pour yourself a glass of wine, get comfortable, and enter into a debate of issues discussed there. Don't be shy to spin it out further. We look forward to your reactions, contributions and feedback, and wish to offer a platform for these debates.

At our Internet site, www.wiki-tools.org, you have the opportunity to submit amendments, questions, criticism and suggestions – including with regard to the technical side of wikis.

How to Use this Book

The book is conceived to be read from front to back. However, it is more realistic to expect readers to jump to "their" chapters. That is why, next to the section headings, we have also included an indication of how deep into the wiki software a respective chapter delves. Authors need less previous technical knowledge than wiki and web administrators.

- Beginners should start with sections One and Two. **The Wiki Concept** provides a general introduction to wikis. It treats all questions that crop up with one's first confrontation with wikis. What are wikis? How do they function technically? How do you explain the fascinating phenomenon that open systems do not end in chaos?

- In **Our First Wiki**, we introduce MediaWiki, the most well-known wiki clone. Here, we will introduce you to practical work with wikis and invite you to experiment. We'll show you how to install MediaWiki so you can get an idea of how it functions. For instance, you will see how formats or tables are generated, and lots more.

- Don't be discouraged by **TWiki, the Jack-of-all-trades**. Installing TWiki requires a good deal of experience, and is more aimed at future and current web administrators. Yet this section also contains an introduction to using TWiki, which is also of general interest.

- In **TWiki as a Project Kit**, we introduce TWiki as a tool for the management of self-organized projects. Here, we'll show you further useful plugins and add-ons, such as the practical EditTable plugin or a diagram tool. It is difficult to estimate the degree of preliminary knowledge required on a general basis. The installation of the plugin takes some getting used to. However, the function of already installed plugins is easier. On the whole, we wish to demonstrate that you can do a great deal more with wikis than "just" writing encyclopedias. They basically support also any other type of project.

- **Tools with a Future** intends to expand horizons. On the one hand, it deals with current technical developments and important wiki projects. On the other, we provide a short summary of the social debates taking place within the wiki community.

- A **Glossary** and an **Index** will help you find your way through the book.
- A **CD-ROM** is included in the book where you will find the software discussed.

Our book is for male and female readers. However, for reasons of brevity, the pronouns used will not always expressly address both – but in our examples, both men and women work with wikis.

We Wish to Thank...

…first and foremost Christine Bühler, who, for years, has consistently been on the scene during critical phases of seminars, master's theses, articles and book projects of the most varied kinds. Radovan Kubani, our hopeful artist, who illustrated our book for us and has even more great ideas at www.rakuba.de. Gunter Dueck, not only for the foreword, but also for the corresponding stimulating discussion. Andrea Adelung from team interculturale for translating our book into excellent English. Our discussions in December 2004 with Jimmy Wales, the founder of Wikipedia, also provided us with valuable ideas and motivation.

For countless critical notes and concrete suggestions for improvement, we thank our volunteer editorial team: Andreas Schmal, who also supplied us with calories and Bytes. Andreas Legner helped out with corrections and constructive criticism, as did Thomas "Schnaks" Schnakenberg, who weathered out the final phase with us. Carsten Diederichs braved the installation instructions, among other tasks. Richard Hölzl supported us from Göttingen with long-distance diagnoses and the final chapter. Our thanks also to Benjamin Heitmann for his comments, and Dirk Brömmel, who stood by us with technical advice.

Regensburg, June 27, 2005

Anja Ebersbach
Markus Glaser
Richard Heigl

Typography

Various fonts are used in this book.

italics	Wiki pages and web titles and paths
bold	Buttons (and links functioning as such)
`Courier`	Entries and source code
[italics] `Courier`	This text is variable and must be customized by you.
Menu1 → Menu2	Steps through a menu tree
<key>	The corresponding key on the keyboard

Source code and entries are generally printed exactly as typed. However, please note the following symbols:

↵	The line is too long and for space reasons must be broken. The actual entry cannot have a break here.
·	A space. Is indicated where the number of spaces is of significance.

So that you won't get confused as to what type of source code is meant, you will find the corresponding symbol in the margin:

Wiki	Code of a wiki page
HTML	Source code in HTML
PHP	Source code in PHP
Perl	Source code in Perl
JS	Source code in JavaScript
CSS	Source code in CSS
CFG	Content of a configuration file
Shell	Entry at the console level

	URL
	SQL query
	Regular expression

Unless otherwise indicated, directories are written in Unix notation. Under Windows, slashes / must be replaced by backslashes \. If the path begins with a slash, it is an absolute path. Windows users must insert the corresponding drive letter in front.

I. The Wiki Concept

1 The Wiki Concept

Readers
Authors
WikiAdmins
WebAdmins

1.1 What is a Wiki?

Imagine you are surfing the Internet, and you stop at a site where you could and would like to add or modify something. For instance, you have a literary reference or link to add. Or you've noticed a typing error. Perhaps you even have a lengthy article that you'd like to display on a separate page. So, you just click on the "edit" button, change everything you wish, add a couple of ideas, confirm it, and the new page is online immediately! In a history, a listing of the saved, older versions of the page, you can view previous changes to the page as well as reverse your entries. If it all was a simple and transparent experience, you were dealing with a wiki. Wiki technology enables virtually anyone to completely edit pages without difficulty. Yet that's not all – anyone can contribute significantly to the structure of the site, simply by creating new links and adding new pages. This openness is the innovative and amazing aspect of wikis. The title of a book puts it in a nutshell: *The Wiki Way. Quick Collaboration on the Web*.

Tools

Wikis are downright fascinating tools. It has never been so easy to become a "correspondent" on the Internet, because the technical hurdles have been reduced to a minimum. People who hear about or use wikis for the first time often experience a bit of culture shock. "Anybody can come along and change my text!" is a popular reaction. The opportunities and consequences of free cooperation in the context of the typical work organization of our society inevitably lead to irritation, because we assume that a contribution from "others" will destroy our own work. We are simply not used to handing over control and responsibility – and to strangers at that. The Swedish data systems specialist Lars Aronsson writes:

"Most people, when they first learn about the wiki concept, assume that a website that can be edited by anybody would soon be rendered useless by destructive input. It sounds like offering free spray cans next to a grey concrete wall. The only likely outcome would be ugly graffiti and simple tagging, and any artistic efforts would not be long lived. Still, it seems to work very well."[1]

Excursion

You can't quite imagine it yet? Then let's take a short excursion, and try out whether or not it is really as easy and free as it sounds. Get on the Internet and go to the site http://www.wiki-tools.org. Click on "Php Wiki" in the left column. Now you are already on a wiki page. Click on *Sandbox*. In this sandbox, you are free to experiment at will. Click on **Edit**. Now, write over the existing text in the middle by typing your name or whatever else comes to mind. Then click on **Save** below the text field. Congratulations! You have just made your first entry in a wiki.

Definition

A wiki is web-based software that allows all viewers of a page to change the content by editing the page online in a browser. This makes the wiki a simple and easy-to-use platform for cooperative work on texts and hypertexts.

Note: Many wikis also correspond to the legal definition of open, free software. Most are subject to the GNU General Public License (GPL), which, among other things, prohibits a program to be converted into "proprietary" software. In this way, copyright laws prevent a program from being claimed as private property by a legal person for commercial purposes. Furthermore, the free use, distribution and editing of the program is ensured.

Origin

The first wiki, with the name WikiWikiWeb, was developed in 1995 by Ward Cunningham.[2] The software engineer from Portland, Oregon is considered to be a pioneer in the development of new methods, such as object-oriented programming, design patterns or extreme programming. Because he was dissatisfied with conventional word processing programs, Cunningham searched for a new documentation system that would better suit the needs of programmers. His goal was a relatively simple software that would enable collective work on software codes that could be published immediately. The new program would automatically document all editing steps to

[1] Aronsson 2002.
[2] Leuf/Cunningham 2004.

make changes easier to trace (document history). Ultimately, the first wiki server went online and has been in operation ever since.[3]

Connotation

"Wikiwiki" is a Hawaiian word that means "quick" or "hurry." The name stands for the programming characteristic of wiki software in which content can be made available in a quick and uncomplicated manner. As demonstrated below, there are a number of further developments of Cunningham's first Wiki, which we will discuss subsequently.

Note: When we refer to "wikis" below, we generally mean the concept, and less so any special implementation.

Functions

The use of wikis is dependent upon the goals of the community, the organization or the company that utilizes it. Although they were first intended for software development, they are now used in a variety of areas. Due to the further development of the wiki concept via various wiki clones,[4] wikis can integrate an increasing amount of functions.

Generally, we differentiate between two application options with wikis: They can be used as tools in a closed work group, or they can be directed at potentially everybody over the WWW. Wikis serve as knowledge management tools in planning and documentation. They can also be utilized as an open, web-based content management system (CMS) for the editing and management of a web presence or to supplement an existing CMS. You can use wikis as your internationally accessible notepad or as discussion forums for general and specialized discussions.

Areas of Application

In the meanwhile, the most varied of institutions have discovered the advantages of wikis. Groups within the so-called civil society, such as the Austria Social Forum or the Chaos Computer Club, use wikis as an organizational aid. Wikis are employed in the classrooms of Swiss schools. Yet even companies such as SAP, Web.de, Motorola or British Telecommunications employ a wiki clone as a decentralized intranet, since, in contrast to conventional groupware, it is considerably more user-friendly.

[3] http://c2.com/cgi/wiki

[4] For more on wiki clones, see Chapter 1.3.

Fig. 1.1

Growing significance

Wikis are becoming increasingly popular. Primarily due to the success of the free online encyclopedia Wikipedia (launched in 2001), wikis have become known to a wide audience.[5] At Wikipedia, the wiki concept is utilized to integrate and display encyclopedic knowledge "from the bottom up". In the German-language edition alone, over 160,000 articles were developed communally by the end of 2004. To date, however, the flagship of wiki technology remains the English language edition, for which, during the period from 2001 to the end of 2004, about 450,000 articles were written. Worldwide, 1.3 million articles have been published in over 200 languages. According to Jimmy Wales, the founder of Wikipedia, the site, with 400 million hits per month up to the end of 2004, was more popular than the websites of IBM or Geocities. Wiktionary[6] is an example of a wiki-based dictionary; Wikitravel[7] is an international travel guide. A

[5] Schwall 2004. Wikipedia is a successor project of Nupedia, also based on wikis.
[6] http://www.wiktionary.org
[7] http://www.wikitravel.org

few software instructional guides and aids (e.g. Wikitosh[8]) are based on wikis. Furthermore, wikis also serve as a professional information medium (e.g. Jurawiki[9]). Others have discovered wikis as an alternative form to forums and mailing lists.[10]

Potential

Basically, wikis are very young digital tools in which there is still great potential. Several further applications, such as learning systems or local news services, are conceivable. Wiki application opportunities for the self-organization of private or public organizations and businesses must be further discussed and tested. In addition, much more development of wiki software will also take place.

Let us briefly summarize:

Non-linear hypertext structure

Firstly, the WikiWikiWeb server technology enables the creation of associative hypertexts with non-linear navigation structures: Typically, each page contains a series of cross-links to other pages. The reader decides which page he or she will view next. In cases where larger wikis employ hierarchical navigation structures, these structures still play a secondary role.

Easy and extensive access

Secondly, using wikis, the technical hurdles and prior knowledge required for communication in and design of the mass medium of the WWW are reduced to a minimum. It is characteristic of wiki technology to allow externally generated texts to be edited "on the fly." The entry and formatting of a text is usually done using a few simple rules. For instance, sequential lines are formatted into lists by placing a star or dash in front of them. Also, pages within a wiki can be linked very easily by writing a sequence of words together without a space and with each word capitalized (called WikiWord or CamelCase, e.g. HomePage) or by placing a text in brackets.

No client software

Thirdly, regular users (clients) require no additional software, but rather can navigate, read, or alter content within wikis using conventional browsers.[11] Similarly, no applets or plugins must be loaded by users. Regular users also do not require extensive training courses. Cunningham was right when he described the wiki as "the simplest online database that could possibly work."

---

[8] http://wikitosh.com

[9] http://www.jurawiki.de

[10] see e. g. *WikiUserTypes*, www.twiki.org/cgi-bin/view/Codev/WikiUserTypes, 4/20/ 04

[11] This means that, in contrast to comparable systems, wiki technology does not differentiate between "back end" and "front end."

Social processes in the foreground

Fourthly, the simplicity of the software is the condition under which a number of communities and projects have been able to develop.[12] Not only is the technology of wikis interesting, but also the "wiki philosophy" and the debates on social perspectives linked to its use. For Internet projects based on the wiki concept, the discussion of purely "technical" problems can generally take a back seat to work processes, content-related issues and the social connections of the project. One could say that the wiki concept undoubtedly marks a new level in Internet technology and its usage.

Simple usage and technical hurdles

If wikis are tools that are so easy to use, why would one need a 400-page book? Let us point out a differentiation here. As a normal wiki user, you require hardly any previous knowledge. If you would like to install and maintain a wiki as an administrator, problems may arise for which more detailed explanations are needed. Between these two poles, plugins, for instance, offer a series of new possibilities that we would like to present. In addition: As easy as wikis generally are to use, the self-organizational processes that make wikis so fascinating can be very tricky.

The fundamental principle of the wiki technology, however, is still simple. Let us have a closer look at the technical side of wikis.

1.2 The Technology of Wikis

Readers

Wiki-Software is installed as a script on a server. The server produces small documents, so-called wiki pages or articles, that can be accessed via a browser. The content of the wiki page itself is written as simple text and then stored in a file or database. When a wiki-based Internet page is accessed, the browser first sends a query to the server that administers the data sets containing the wiki software. This data, which is in the form of simple text, must now be formatted for display in the browser.

To do this, the wiki script translates the file text (wiki code) or data set into HTML and embeds it in the web page (template) to be sent back to the browser. For example, the wiki script can be a PHP script that reads the raw page data from a MySQL database. This raw data is checked line for line, and the specific format commands

[12] The relationship between project and community varies. For example, at Wikipedia, a community has developed around a free encyclopedia project. On the other hand, the MeatballWiki is only a community without a central project.

contained in it are replaced by the corresponding HTML codes.[13] Subsequently, the page thus created is integrated in the layout template. Every wiki page has its own, distinct name indicating the subject of the page. In addition, there is usually a navigation menu and a few page-specific links of the displayed page. The most important of these links is the "Edit" link.

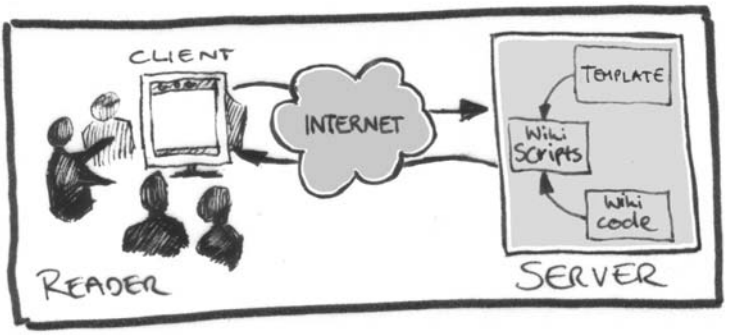


Fig. 1.2

Authors

If this page is then to be edited, the **Edit** button is used. This sends another query to the server. The same page is loaded again, only this time the contents are not converted to HTML format, but rather displayed in "raw form" in a large text field in an HTML form. The user can edit the text in this form and send a new version, which immediately replaces the old version in the database. When the page is accessed again, the new version is displayed.

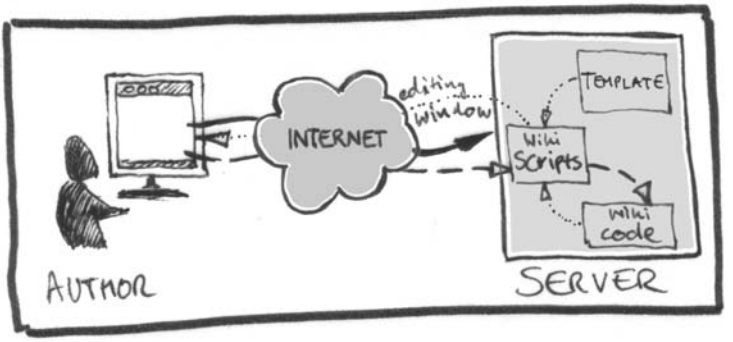


Fig. 1.3

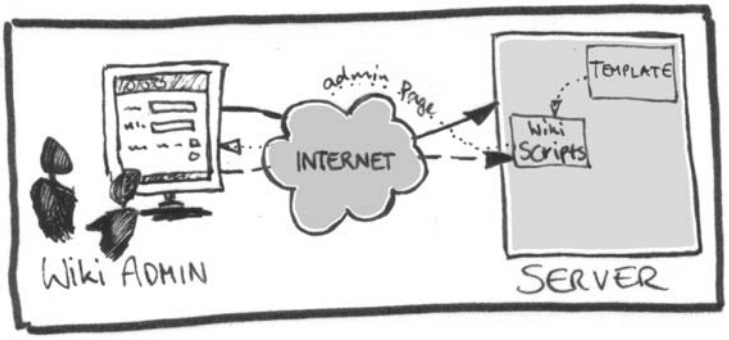
[13] In this step, all URLs are then clickable, and in place of all URLs that end in gif, .jpg or .png, in other words those displaying images, the corresponding image tags are set (the images themselves are subsequently loaded by the browser).

Visitors do not need to know any programming language or HTML in order to use wikis. Wiki pages are written in simple ASCII format, just like emails. There is a series of conventions that you should become familiar with sooner or later, but they are generally easier to learn and more "intuitive" than HTML. For instance, a blank line separates paragraphs. When this page is saved, the system translates the blank line to HTML, that is, it adds a <p> at the respective spot. The wiki link syntax, which we describe in more detail below, is also important.

Wiki Admins

The primary task of wiki administrators is to maintain wiki content and ensure the smooth operation of working in a wiki. They have more extensive rights than regular participants; for example, they can delete pages or block individual user access. To do this, wiki admins usually have their own interface or special pages in the wiki to which only admins have access.

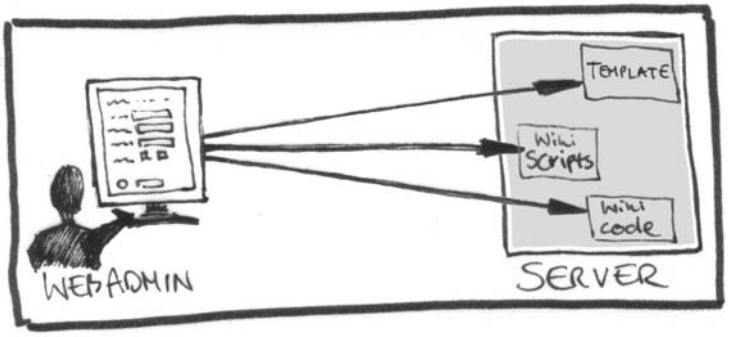
Fig. 1.4



Web admins

Yet even a wiki cannot exist without some technology. On this level, the web admin is in charge of software installations, maintenance and updates. In contrast to the other groups mentioned, web admins have direct access to files without needing to detour through the wiki interface.

Fig. 1.5



The categories mentioned up to now indicate a model which divides the access of participants into levels that differ technically in type and depth. We add to these the lowest level, which includes the infrastructure, the server and operating system, as well as the necessary software, web server and database. They are maintained by a system administrator.

Level model

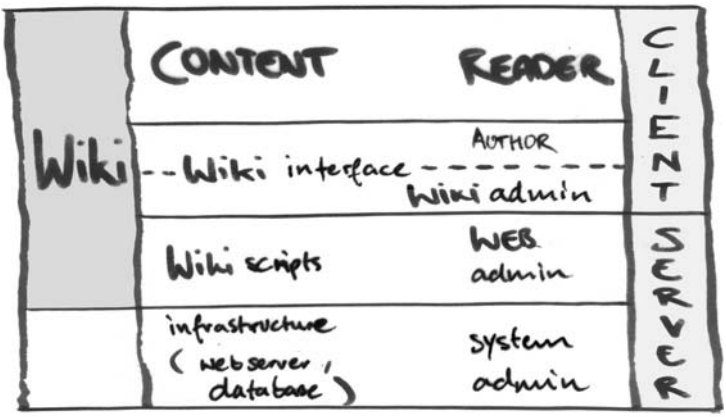


Fig. 1.6

Thus, while the system and web administrators have direct access to the server and must not necessarily be integrated into the wiki community, wiki administrators, authors and readers navigate the web-based interface of the wiki.

Using the idea of the level model, we can also clearly see that, in contrast to a normal HTML page, the interface with which content can be created is found in the client realm. This means that, from a technical standpoint, conditions have been established in which a great number of people can participate extensively in the design of the content without needing to clear any major hurdles (such as access to the server).

1.3 Wiki Clones

With the development and utilization of the WikiWikiWeb, new challenges have emerged, and accordingly, new programs in which the wiki concept has been further developed. Meatball (itself a wiki community) claims that there are currently about 200 different types of wikis.[14] These programs are called clones, since they imitate the

[14] Many wikis are listed under: http://c2.com/cgi/wiki?WikiEngines.

original wiki, but have added some extra functions. Most of them have the term "wiki" in their name. Here are just a few examples:

- **UseModWiki** is one of the oldest and most widely-used wiki clones. Written in Perl, they include a number of small programs that enables a variety of additional functions. UseMod has had a substantial influence on the development of other wikis, such MediaWiki, whose formatting language (or, more precisely, its syntax) strongly resembles that of UseMod.
- **MediaWiki** was conceived for the needs of the Wikipedia encyclopedia project. It consists of several scripts written in PHP, and contains a few further developments (name spaces, sidebar, and messages regarding processing conflicts).
- **MoinMoin** is a simple and very widely-used wiki clone written in Python. MoinMoin, whose name reflects a friendly northern German greeting, enables user registration and has a plugin system for enhancements, among other functions. MoinMoin is very user-friendly, and offers a very pleasant layout.
- **PhpWiki**, a very promising wiki clone, has several administrative functions and a plugin architecture which enables the integration of a number of functions. PHP is the syntactic foundation for the Php wiki.
- **WakkaWiki**, as a simple, small wiki, has become very popular. The project lives on in various clones such as ComaWiki, WikkaWiki and UniWakka.
- **TikiWiki**, also written in PHP, already offers a whole series of useful features that can stand up to a comparison with existing content management systems and groupware. In this case, the wiki is just one component in an array of additional groupware features, such as forums, blog functions, newsletters, file and image galleries, and survey, chat or calendar functions.
- The Perl-based **TWiki**, with its many plugins and features, is, in addition to TikiWiki, the most comprehensive wiki clone. TWiki implements wiki technology more consistently than TikiWiki, because additional functions can also be realized with this technology to a large extent. TWiki, which was conceived to be utilized as a company intranet, is being used increasingly for commercial purposes, due to its high level of development.

You can find a few tips on which criteria you should use to select the right wiki for you in Section 17.2. For instance, one question is whether your wiki should enable user administration or not.

1.4
Characteristic Wiki Functions

Regardless of the wiki script used, all wikis offer a few technical core functions, which we will only mention briefly here and later in more detail.

Editing. The **Edit** button is the ultimate typical feature of a wiki. Only in extreme cases are specific pages excluded from the editing option. At Wikipedia, the capability to change certain pages such as the title page, has been made more difficult. A page can also be completely prohibited from editing. Since such blocking goes against the grain of the wiki philosophy, it should be avoided when possible.

Links. Each article can be linked to other articles, and thus form a new network structure. As demonstrated above, links can be created in most wikis using a WikiWord: Words are capitalized and written together without a space (CamelCase, which in itself is an example of such a WikiWord). CamelCase makes linking easy, but can also cause problems in some applications. That is why, in other wikis, links are generated by simply placing the title in square brackets. Regardless of which procedure is chosen, a link is thus generated that appears in the normal view. If a respective linked page within a wiki does not yet exist, it can be created with a simple click of the mouse on that link. In this way, wikis support associative links between pages in that they design and display links in an almost intuitive manner, whether or not an intended link exists. The new pages are now linked to the existing ones and thus part of the hypertext structure.

History. This function basically saves all previous versions or modifications of any single page. Here, it is possible to exactly track the editing process of an article, since all changes have been documented. The “History” function allows a previous version to be opened and saved again, in order to restore the original content (rollback). This concept is based on Cunningham's editing history, and is also a useful tool against unfriendly users who wish to destroy the page. However, it is being used more and more for regular coordination problems. The history can roll all the way back to the first version, but can also be limited due to space reasons. More complex wiki clones offer a so-called “Diff” function, which displays detailed alterations between two versions, such that authors do not need to compare two texts line for line.

RecentChanges. This page either provides a current overview of a certain number of recent changes to wiki pages or all changes within a predefined time period. It is produced automatically and cannot be changed by users. Some wikis, such as MediaWiki, offer so-called watch lists. Such lists enable selected pages to be monitored over an extended period of time. If you are partial to a particular article, you do not have to continually look through the list of all changes to all pages.

SandBox. Wikis usually offer instructions and introductions on their homepage, which serve to facilitate working with the system. In addition, new users, as well as experienced ones, can use the so-called SandBox or PlayGround to learn how to use wikis and try out various solutions without having to use a regular page. You have already been introduced to the sandbox. This test environment is nothing more than a wiki page that is emptied on a regular basis.

Search functions. Most wikis also offer a classic full-text or title search for the wiki pages. Thus, articles in a wiki can be accessed quickly. It has been our experience that if titles are well-thought out, the search can function like an index card system.

1.5 The Wiki Phenomenon

Brecht's Radio Theory

Bertolt Brecht, in his so-called "Radio Theory," written in 1930, wrote that radio has one side when it should actually have two. "It is purely a distributive apparatus; it just rations out."[15] His now famous proposal was to convert broadcasting from a distributive apparatus to a communicative apparatus. "Radio could conceivably be the greatest communicative apparatus of public life, an incredible channel system; that is, it would be, if it were capable of not only broadcasting but also receiving, of thus being able to make the listener not only listen but also speak, and not to isolate him but connect him."

Just a few decades later, an innovative mass medium became available in the form of the Internet, which has markedly accommodated Brecht's technical demands of connecting broadcaster and receiver. This was just as much the case with the classic visual and audio media as it is with the "Internet revolution." Yet technology and new inventions alone do not change anything. A complex and difficult reciprocal relationship exists between social progress and tech-

[15] Brecht 1930/1967, 129.

nical innovation. We can analyze this relationship with the aid of the following questions: Who uses this technology? What goals and forms are involved? Upon what dependencies does he or she rely? Accordingly, wikis can be used in an emancipated manner, or not.[16]

1.5.1 Creativity through Group Processes

Communities

The fact that the potentials and limits of wikis are now also being discussed beyond the realm of professionals is primarily due to the fact that communities have formed around the large Internet lexica who are further developing the possibilities of self-organization. These groups can exhibit a few successful projects that rapidly make reliable services available to the public using a fairly uncomplicated procedure. It has been demonstrated that the communication of large groups is more effective, and using it, groups can react faster to change than in traditional hierarchical organizational models. Wikis do not represent a unique phenomenon: Similar experiences were previously made with the "subversive" development of open source software (e.g. Linux).

Cathedral or bazaar?

Eric Raymond triggered a sizeable debate when he attempted to represent the various management methods metaphorically using the principles of the cathedral and the bazaar.[17] While conventional software development assumed that important programs had to be built like cathedrals, "painstakingly chiseled by individual druids or small teams of high priests who worked in complete isolation and were not allowed to issue any unfinished beta releases, […] the Linux community seemed to be like a large bazaar of wildly intermingled voices having a variety of goals and approaches which could produce a coherent and stable system only through a series of miracles. The fact that the bazaar appeared to work, and work very well at that, was a downright shock."

The appeal of wiki technology lies in the act of rethinking the familiar. Once again, hierarchical control models are at our disposition, and with them, valid ideas of why and how, through the division of labor, complex problems can be solved and products produced and distributed. It is no less than a question of other socialization models whose possibility becomes apparent.

---

[16] See: Ebersbach/Glaser 2004.
[17] Raymond 1999.

Large-group dynamics

People repeatedly ask why wikis work. This question has to be posed more precisely: Why and under what conditions do people cooperate in wiki projects without central control and external pressure? Group processes are a much discussed and investigated topic in the fields of sociology and education; so much that we cannot present it here in all its theoretical complexity. However, experiences with large group events have revealed a few principles which contribute greatly to the success of large group processes.[18]

Playful creation. "Why Wiki works? It's cool," is the brazen comment at Ward's Wiki. A loose, playful atmosphere and fun at work are important conditions for self-organized processes, because one's creative, social and practical skills can best be unfolded in such an environment. It is motivating when one can make his or her own designs or contribute an article for a large-scale project. Less attractive "obligatory" tasks do not necessarily fall by the wayside if their necessity is recognized.

Flat hierarchies. Flat hierarchies are decisive for creative, self-organized group processes. The responsibility for the process, not only for subareas, is transferred completely to those individuals performing that process. These responsibilities are integrated into the planning and workflow control processes as completely as possible. Newer methodological approaches for large-group events place great value in handing workflow and goal definition responsibilities to participants and those concerned. This concept requires of its participants a willingness toward the open nature of the process, as well as toward equally distributing not only the risks, but also the advantages.

Projects based on the wiki philosophy require flat hierarchies. This can be seen, for instance, in the fact that participants contribute considerably to designing the organizational structure of the wiki, e.g. through linking. Yet social structures also develop within a wiki – and they often differ from those the initiators had initially imagined.

Modification pressure and the complex topic. The pressure to modify (as intrinsic motivation) and the will to want to solve a problem represent, according to Raymond, an indispensable motor for the "bazaar." Working on one's own topic creates dedication.[19] Self-

[18] Especially at *Open Space* conferences. See: Petri 2000, Maleh 2000.
[19] Of course it makes a difference whether a goal is self-set or stipulated, or whether external goals have been taken on as one's own.

organization processes build on a responsibility that stems from interest in the matter. Inevitably, in step with the wiki philosophy, incomplete or faulty wiki pages are bound to remain unedited for a time. Only after someone has deemed it necessary will the page be modified or existing errors corrected.

The important thing, as Raymond has explained for the bazaar, is to be part of a worthwhile cause and that improvements in which one is involved become apparent. In addition, a complex topic representing an intellectual challenge promotes the dynamics of large-group processes. It develops its full strength through a fascinating and challenging topic which can, by all means, have a high potential for conflict.

Simple system, simple rules. The decision to sit down and join in is the greatest obstacle for self-organization processes. Successful self-organized group processes are often founded on very simple basic systems, because favorable – if complex – decision-making and modification processes depend only on a rough overall concept, access to all relevant information and clear basic conditions. Thus, the conference model *Open Space* functions with just a handful of rules.

Wiki technology, with its low technical access hurdles, is ideal for web-based group processes. Several wiki communities have implemented simple codes of conduct.

Open access. Free will and open access are vital conditions for motivation in self-organization processes. The success of the bazaar principle as well as the wiki philosophy is based on the fact that discussions are removed from alleged expert and specialist circles, right from the start. This creates transparency and incentive.

A large pool of participants testing the system is also a way to identify errors at an early stage. The system becomes more stable and can be more quickly adapted to the changing needs of its users.

For wikis, this approach is supported by the principle of "open postings." With wikis, users are invited to edit an existing page within their normal browser or add new pages. In contrast to the classic editing principle, articles are not first proofread and only published when completely error-free, but rather as soon as possible, so that users of a page can be integrated in the cooperative process.

Diversity of the participants. For the dynamics of self-organization processes and collaborative work, a climate of openness and mutual trust is necessary – despite inevitable conflicts. A variety of experiences, backgrounds and knowledge are seen as the basis of creative

processes and as an enrichment, and thus, every user is initially recognized as an expert.

The wiki philosophy is based on the assumption that those individuals will become involved who also want to contribute to the situation. Their knowledge and motivation are sufficient to contribute to the issue. A certain degree of heterogeneity can also be observed in participants of wiki projects regarding their areas of interest.[20]

Extremely flexible scheduling. A relatively flexible scheduling of one's work time within an overall process is a further motivator. According to the bazaar principle, which knows no deadlines, it is possible to tailor one's work time to suit one's own rhythm and individual daily life. Work begins when the time is ripe, and ends when it is finished. It is less bound to fixed schedules. Time pressures exist only when problems remain untouched.

Self-determined work. People involved in group processes and members of communities have very different strategies and just as varied an understanding of their own function within the overall relationship. In addition, strategies and self-conception are subject to continuous change, such that it is difficult to determine certain roles or types – perhaps it is not even advisable. However, for better understanding, it would be helpful to consider that each individual – once freed from a socio-economic background – enters into relationships with other participants via a wiki in a very multifaceted manner. Cooperation in open wiki projects is attractive because strategies, focal points, and work intensity can be self-determined to a large extent.

- **Reception behavior.** We have to distinguish between how often participants visit a wiki system, whether they read the articles fleetingly or closely, and whether they are searching for articles on a particular topic or across several subjects.
- **Writing behavior.** With regard to writing behavior, there are a number of variations. Some visitors of a wiki never write there. Others proofread, edit the style and layout, and make small changes. Others still, the specialists, acquaint themselves in detail with a subject and contribute articles with a great degree of

[20] See Section V. As in the case of many other Internet projects, we must mention the limiting factor that we expect the circle of active wiki users to continue to be limited to certain social groups for the present (keyword: digital divide).

content. In comparison, "generalists" jump from article to article and bring in knowledge from other areas.

- **Structural behavior.** This category refers to the extent to which responsibility for an overall project is assumed. It includes the question as to how intensively one participates in fundamental debates or voices considerations regarding general procedures. Some participants take on functions as mediators or moderators. Others assume regulatory tasks, such as checking orphaned pages. Accordingly, the technical administrators and maintainers also belong to this category. People in self-organized processes ideally receive functions through their authority and the trust they have earned through their work.
- **Social behavior.** This category encompasses atmospheric aspects. It refers to the form in which criticism and encouragement are imparted. Does one enter the discussion with a provocative or cooperative stance? On another level, the organization of the social and cultural periphery is also part of this realm, such as a regulars' table or seminar weekends.

"Types"

The forms broadly discussed here are naturally not pure, but rather overlap each other and evolve. Due to the interplay between the individual practices and goals, the overall relationship is continually restructured as a process. The members of a community have just as much of an effect on the individual through their actions as the individual does when contributing to the daily design of form and content of the community. All of these behaviors, including those that are passive and, in a broader sense, "destructive," are necessary to a dynamic community. However, the community can also be destroyed by them at any time. Knowledge of specific and general group processes within a wiki community is still in its early stages. At the 21st Chaos Community Congress in 2004 in Berlin, Jimmy Wales presented some initial considerations using the example of Wikipedia. A few of the "types" he outlined serve to illustrate the degree of diversity.

- **Bees.** Wales describes as "bees" those participants who perform very important work and without whom Wikipedia could not achieve or maintain its quality. Nevertheless, they are the least recognized group. They include generalists and specialists. They provide important content articles, proofread texts or negotiate with difficult users.
- **Sock puppets.** This group is comprised of people who publish under more than one account. This is done for a variety of rea-

sons. A few wish to preserve their privacy (such as a professor who also writes as a fan of Britney Spears but may fear a loss of authority). For others, such as those using multiple identities to manipulate polls, it represents a despicable attack on the mutual trust upon which open editing is based.

- **Judges.** This is obvious. These are people who focus on conflict resolution and decision-making. They are active in juries and arbitration committees. They organize polls and further develop proposals for regulations.
- **Moths.** This rather strange label becomes clear when considering that "moths are drawn to flames," as Wales explains. Flames in this case refer to flame wars, that is, heated and often insulting verbal duals. While people who start aggressive flame wars generally do not enjoy a good reputation, Wales sees the fact that individual participants seek conflict and do not try to avoid it as not necessarily negative action. On the contrary, these discussions can lead to vital advancements.
- **Vandals** are a common problem in open editing systems. They willfully destroy content, and yet they pose a much smaller threat to the community than is generally assumed. (see below)
- **People "outside" of the wiki** are often overlooked in terms of their significance. They continue to develop wiki technology as programmers. Even those individuals who primarily develop wiki content in other communicative media (e.g. chats or mailing lists) also play a role.

1.5.2 Limits of the Wiki Philosophy

Wikis are not automatic "successes", much less a cure-all. Productive group processes are always faced with destructive practices that even the large wiki communities have to combat right from the start. In problem analysis, we need to differentiate between whether the group dynamics have stemmed from a constructive start to the process and then slipped toward the negative, or whether wikis are simply not being accepted as normal tools.

Lack of interest

If wikis are not being accepted as tools and are thus not integrated into the daily work routine, they share the same fate as several knowledge management systems. Usually one person alone does the writing, and the others only read. Or the wiki system is not consulted at all. There are several wiki systems whose possibilities cannot be fully unfurled due to a lack of interest or out of lethargy. The causes

are manifold. Generally, social elements are underestimated when dealing with new software. More than a few users already have high expectations when a wiki system is made available, and are quickly disappointed when it is not met with immediate positive response. However: Even if the technical hurdles are few, using wikis must still be "learned;" people still have to be interested in or introduced to the system. This includes the realization that a wiki is never "finished" and that not everything is going to function properly right away. The fact that one is not dealing with a WYSIWYG system can cause apprehensions that need to be taken seriously.

The acceptance of wikis depends on the degree to which I as a person can truly benefit personally from using them. That is why it is still important that the wiki not be empty at the start, but rather provides a certain quantity of content that can be further edited or to which additions can be made. This also means that a small core group that uses wikis for itself and thus feels responsible for it is of great significance.

Social environment and working world

Using wikis, the work environment can be influenced, and at the same time, be dependent upon it. Using open systems in today's working world is met with many types of resistance. The lack of willingness of managerial persons (project managers, area managers, etc.) to permit open systems in private companies is only one example.[21] However, the same thing can occur in authoritarian organizational structures.
High workload, family and social responsibilities or social security worries often allow little room for free forms of cooperation which are admittedly also complex.

Vandalism

The problem of vandalism has already been addressed, as well as the fact that its impact is overestimated. It has generally been observed that in wiki projects, destruction and/or damage remains relatively insignificant. It is assumed that cracking an open system poses no great appeal to "serious" crackers. The WikiWikiWeb server principle provides its own evidently effective antidote with its version control. It enables the previous version to be restored at any time. In systems with a high visitor frequency, disturbances can be detected quickly, as systems with several participants tend to be more "stable." Wikipedia, for instance, places pages that are damaged often on a separate list, to which the administrators pay special attention. According to an IBM study, attempts at vandalism at Wikipedia are often eliminated within five minutes: "We were sur-

[21] Even in companies in which hierarchies are being disassembled, this does not necessarily translate into a higher degree of transparency of company structures.

prised at how often we found vandalism, and then surprised again at how fast it was fixed,"[22] reports Martin Wattenberg, a researcher at IBM TJ Watson Research Center in Cambridge, Mass. The fact that many people can control the process and anyone can take instant action appears to be a significant element in quality control. It only requires a corresponding sense of problem awareness that anticipates attacks. Yet even willful alterations to small details can greatly inhibit the quality without being immediately noticed. Another – last – method has already been mentioned: blocking the page – which means the end of the wiki principle for that page.

Attention-seekers

Greater problems stem from people who use wikis as a platform for attention-seeking or those who do not wish to conduct discussions cooperatively. Provocation and posing general questions can be useful in breaking through a rut in thinking. Various opinions on a topic often develop into "editing wars." So-called trolls knowingly incite flame wars with lengthy, superfluous or provocative articles. Such conflicts, which contribute nothing to the issue, cost a great deal of energy. "Wiki pages represent consensus because it's much easier to delete flames and spam then indulge them."[23]. In many wikis, trolls are kept away from the articles through discussion pages, so that, where possible, they only need to vent on this meta-level without "adding noise" to the real content[24]. In addition, at Wikipedia, a few mediation procedures and open instances have been established, such as ad hoc mediation commissions or openly-discussed exclusion petitions.

In very stubborn cases, there is also the opportunity to block certain users for a limited period or forever, using an IP list. In order to promote transparency and exclude arbitrariness, users can access a list of blocked users at any time and find out about the initiator and grounds for exclusion. Such blocking of certain IP addresses, in turn, leads to the problem that non-excluded users may be mistakenly barred from having continued access.[25] A further problem is that the disruptive parties can re-register at any time under a new name.

In his newly published book "Die heimliche Medienrevolution" ("The Secret Media Revolution"), Möller provides a comprehensive look at the problems and possible solutions in dealing with difficult

---

[22] IBM 2003.

[23] Why Wiki Works, http://c2.com/cgi/wiki?WhyWikiWorks, Jan. 15 2005.

[24] Aronsson 2002.

[25] Since IP addresses are not always issued on a permanent basis.

controversies and vandalism in blog and wiki community environments.[26]

Project portal excursion

The best overview of the discussion culture and decision-making processes can be had with a visit to Wikipedia. Let us take a short excursion to the project's Community Portal page.[27] Here, we find guidelines and conventions, discussion pages for admin candidates, moderation information and pages collecting opinion statistics. Completed problem cases are documented on the arbitration committee page. And of course, a visit would not be complete without taking a look at some of the discussion pages of individual articles.

Quality assurance

We can see: For quality assurance and conflict resolution, a few mediating instances, rules and practices have formed at Wikipedia. We find name and formatting conventions; well-made articles are presented as examples, and quality offensives are being performed in certain topic areas. A Wikiquette offers recommendations for cooperative communication with other users: [28] Suggestions such as assuming the good intentions of other users, objectivity, mutual help and encouragement, and kindness are proposed, as is the advice to keep cool in conflicts that will inevitably crop up. After all, there are always the arbitration committees. And yet, the overall character remains true to its democratic fundamental principles. Anything else would cause the project to fail.

Neutral Point of View

One especially touchy subject is the credibility and objectivity problem. At Wikipedia, this is addressed, among other ways, under the heading "Neutral Point of View." Since many people from around the world and having the most varied of political and religious views take part in the project, Wikipedia is obligated to formulate articles as neutrally as possible. The point is not to write them as objectively as possible – this is a common misunderstanding – but to present all aspects of an issue. Most wiki users have thus learned to express themselves in a conflict-free way, insofar as possible. Instead of writing "Apples taste good," one would instead write "Some people like the taste of apples." We will address the issue of apparent neutrality at the end of this book.

Open editing

Giving up the author principle is one aspect of collaborative work. It leads to a few questions: Is someone who has contributed to collaborative texts legally accountable? Who owns the text? It is true that traditional newspapers and encyclopedias also represent a collection of articles by a variety of authors, but in open wikis, there is no traditional relationship between publisher and author. Wiki texts

[26] Möller 2005. To an extent, also in: Möller 2003.
[27] http://de.wikipedia.org/wiki/Wikipedia:Portal
[28] http://de.wikipedia.org/wiki/Wikipedia:Wikiquette

are thus not directly subject to the compulsion of marketability. In such cases, the individual author, on the one hand, receives a much stronger, more independent role, while, on the other hand, he disappears in the open system as an individual author at the same time.

Open text

A further area of interest is the issue of ownership and copyright. Since many individuals contribute to content, the question must be clarified as to whether anyone can claim copyright on individual articles or even the whole collection. Wikipedia, for instance, allows every user the right to protect his or her own contributions. However, when the page is being saved, the user is informed that he or she may only benefit from one type of copyright, namely the GNU Free Documentation License (FDL). In short, this means that anyone may copy and use the text for other purposes as long as he or she makes the original text available to other readers, which is most easily done by linking to the Wikipedia URL.[29]

Another question which arises deals with how materials protected by copyright are used in wikis, and who is responsible for any arising damage. To date, there has not yet been a precedent case. However, the law in most European countries differentiates between newspapers having an editing department for which a publisher is accountable and bulletin board systems or services of an Internet provider, where individual users bear the responsibility. Wikis are more likely to be categorized with the latter.

Careful optimism

Wikis will also have their share of problems and setbacks. The wiki philosophy may see some things too optimistically. Nevertheless, previous experience has given reason to adopt an open and optimistic stance toward these developments. The problems known to date and mentioned here do not negate any grounds for optimism; if one considers human relationships as being permanent collective learning processes, one cannot simply say, "people are the way they are," and stop there. Instead, one must question the causes for obstacles to "learning" (prejudices, insecurities, lack of complete understanding of group processes). Brecht's comment on "Radio Theory" is good advice for dealing with the "impossibilities" of wikis:

"Not feasible in this social order, feasible in another, the suggestions, which are only a natural consequence of technical development, serve the propagation and form of this other order. [...] If you should consider this utopian, I kindly ask you to consider why it is utopian."

---

[29] Aronsson 2002.

II. Our First Wiki

2 The Installation

Readers
Authors
WikiAdmins
WebAdmins

Local installation

In order to become familiar with and test a wiki, it is a good idea to install a system on your own computer, instead of delving straight into the depths of the Internet. Firstly, you have complete control over the software used and its settings. On the Net, this is only the case if you are an administrator of the server on which the wiki is running. Secondly, almost all entries made to an online version can potentially be read by the entire world. You need to ask yourself whether the world would really be interested in every little test; in addition, copyright issues could quickly crop up. In a local testing environment, on the other hand, you can control who has access to your computer, and you can experiment without needing to worry every time about relevance or legal questions.

Installing wikis

Basically, installing wikis is always done according to a similar pattern. First you need to create and adapt the environment in which the wiki will run, which primarily means setting up the web server and, if necessary, the database. Then you copy the wiki software files to the proper location and adjust one or more configuration files to the system environment, either manually or automatically. Such adjustments mainly include the indication of paths, language and connection to the database. In addition, some wikis offer the possibility of performing some settings directly over the browser. The individual steps for installing and setting up MediaWiki will be detailed below.

Note: The installation described refers to the software versions on the supplementary CDs. If you prefer to download the current packages from the Internet, there may be some variations.

2.1 A Test Environment with XAMPP

Test environment

Since it is the nature of wikis to be a component of the WWW, you need to create a test environment for the actual software for local operation on your computer. In the case of MediaWiki, such an environment consists of a web server, the script language PHP (Version 4.3 or higher) and the database MySQL. The installation and interaction of these components is complex, and descriptions of them fill bookshelves. However, that is not within the scope of our book. Luckily, there is a package that allows us to install all required components with a minimum of configuration effort: XAMPP. This is the abbreviation for "Apache MySQL PHP Perl" (the X is a placeholder for the operating system) and thus for components that together result in possible standard equipment for a web server. If you should already have this software on your computer, you can skip this chapter and go straight to installing the MediaWiki.

Tip: If you already have another web server installed on your computer, it is possible that XAMPP may not work. For instance, this is the case with IIS.

At the time of this printing, XAMPP is available in Version 1.4.14. It can be found on the supplementary CD in the directory */xampp/[operating system]*. The newest version can be downloaded at the URL http://www.apachefriends.org/en/xampp.html.

2.1.1 XAMPP for Windows

Installing XAMPP

XAMPP runs under all versions of Windows. You require 64 MB of RAM and about 110 MB of free hard drive space, which is usually not a problem for conventional computer systems. Simply execute the .exe file in the corresponding directory of the CD. You will now be asked to indicate your language. Then you will see the installation program. Start it by clicking **Next**. Now you will see license information, which you must accept by clicking **I Agree**. You can then indicate the path in which you want the system to be installed. If you wish to follow the examples in the book exactly, enter *C:\apachefriends* here[1]. Clicking **Install** will start the process. The program creates the folder *[drive]\apachefriends\xampp*, which con-

[1] This is especially important for setting up *TWiki* in Section III.

tains all included components. In addition, the parameters in the respective configuration files will be accordingly adapted to the conditions of your system. No entries will be made to the registry and no system variables will be set. Thus, to uninstall the program, you just need to delete the directory in which the software is stored.

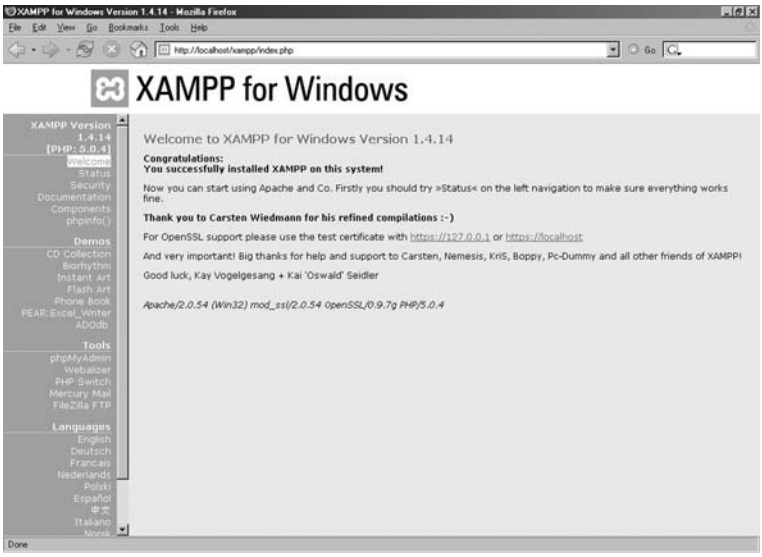


Fig. 2.1

Starting XAMPP

In order to start XAMPP, run the file *xampp_start.exe* in the directory xampp or select XAMPP BASIC START in the Start Menu under PROGRAMS → APACHEFRIENDS → XAMPP. Open a browser and enter the URL http://127.0.0.1 or http://localhost. This is always the local address of the test system. You should now see the start page of the XAMPP environment (see Fig. 2.2). As you have surely noticed, the DOS window that starts the XAMPP environment remains open. A few of the required programs run here. In order to close the test environment, however, we do not recommend simply closing the window, since some of the services running are independent of the window and will continue to run after you have closed it. For "clean" closure, it is thus advisable to run the file *xampp_stop.exe* in the same directory.

2.1.2 XAMPP for Linux

Installing XAMPP

Given the wealth of LINUX/UNIX variants, one should be careful with generalizations, but, in principle, XAMPP should run on all systems. It requires approximately 110 MB on your hard drive. Installation is easy. Transfer the installation package from the CD to your home directory. Open a shell. You should be logged on as an administrator. If this is not the case, switch using `su`. Then, enter the following command to unpack the archive:

Shell
```
tar xvfz xampp-linux-1.4.14.tar.gz -C /opt
```

That's it. XAMPP is now installed in the directory */opt/lampp*.

Starting XAMPP

You can start your test environment with the command

Shell
```
/opt/lampp/lampp start
```

Now open your browser and enter the URL http://127.0.0.1 or http://localhost. You should then see the XAMPP language selection, and after clicking your language, the start page. To end the test system, use the command

Shell
```
/opt/lampp/lampp stop
```

This will properly close all running components of the test environment.

2.1.3 What's Going On Here Anyway?

Interplay of the components

Once you have started XAMPP, a web server will be running on your computer. It waits for queries via the HTTP protocol, which are generally posed by a browser, and answers them with an HTML page or file. In local systems, the web server is always accessible at the address 127.0.0.1 or localhost. Simple homepages are transferred from the server to browser in this way.

Tip: If you are online at the same time, any user on the Internet can direct a query to your web server if he knows your IP address. This is of course only the case as long as XAMPP is active.

Tip: You can increase the level of security of your system under Linux by entering the command `/opt/lampp/lampp security`. You will then be asked to assign passwords for the operation of the individual components.

Under Windows, you can find a security check on the XAMPP start page. There you will find a link through which you can assign the necessary passwords.

More complex websites (e.g. wikis) generally require further processing of data. This is done with small programs called scripts. If you request a PHP or Perl script via a browser (usually identifiable by the .php or .pl extension), the web server first runs the program and then sends the generated output back to the browser. Scripts can, in turn, send queries to the MySQL database and thus read and further process data.

XAMPP start page

Once you are at the XAMPP start page, you will be asked to select your language (via a link). Then you will reach the actual XAMPP overview page. For the purposes of this book, only a few options are relevant. Under Status, you can see which components are active. Of course, it is important that MySQL and PHP have been activated. If this is not the case, an error has occurred during installation, and you should repeat the procedure. Under the heading Tools you will find the program phpMyAdmin, with which you can easily administer your database via the web interface (see Fig. 2.2).

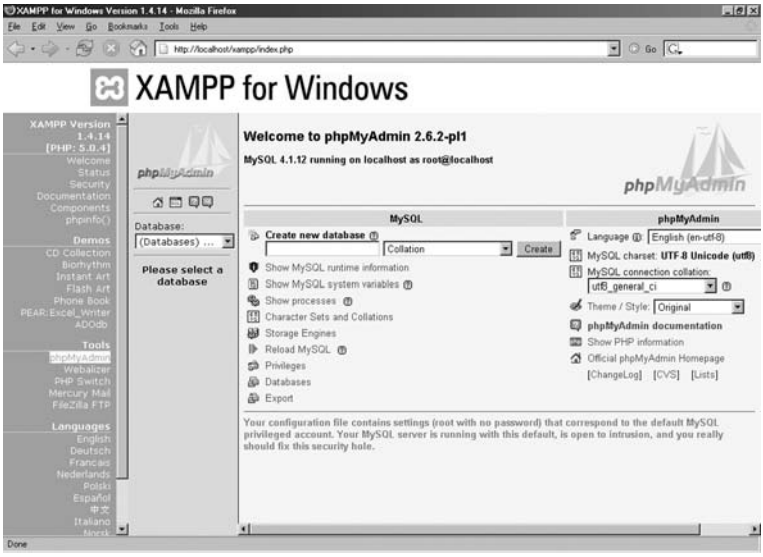


Fig. 2.2

Test the environment

The files of your own homepage can be stored in the directory *xampp/htdocs* under Windows and */opt/lampp/htdocs* under Linux. This is the standard directory that Apache accesses when reacting to a query. In order to test this, create the subdirectory *xampp/htdocs/test* or *lampp/htdocs/test*[2]. Now, open the text editor and save the following HTML code as *index.php* in the test folder.


```
<html>
  <head>
    <title>This is a test</title>
  </head>
  <body>
    <p>Your own pages are displayed here
    <p><?php echo "PHP running" ?>
  </body>
</html>
```

If you now access the address http://localhost/test, the file you have just created should be displayed. If the PHP output is also visible, then the installation has been successful.

Caution: Sometimes a file may be saved with an incorrect extension – e.g. with .htm instead of .php or vice versa. Under Windows, when attempting to rename the file, the error often occurs that the operating system automatically adds the old program extension. You can eliminate this in Windows XP, for example, by going to CONTROL PANEL → TOOLS → FOLDER OPTIONS → VIEW and deactivating (unchecking) the option "Hide extensions for known file types".

2.2 Installing MediaWiki

Copy files

After you have installed a test environment on your computer, you can now install MediaWiki. We have included Version 1.4 on the supplementary CD in the directory */wikis/mediawiki*. Alternatively, you can also find the current version at http://www.mediawiki.org.

Tip: When downloading, in some browsers (e.g. Mozilla), the file extension may be renamed from .tar.gz to .tar.gz.tar. This may lead to problems with decompression programs. Thus, make sure the file has the correct extension.

[2] To avoid complexity, the following examples will always indicate the "xampp" version.

2.2.1 Version 1: Local System

Copy the compressed file to the directory *xampp/htdocs* and unpack it with FilZip[3] under Windows. Under Linux, use the command

```
tar xvzf mediawiki-1.4.5.tar.gz
```

The directory *mediawiki-1.4.5* will be created.

Tip: If, instead of finding the directory, you see the file *mediawiki-1.4.tar*, just unpack it again.

To facitilate accessing the program, we recommend renaming the directory "mediawiki."

Linux: assign rights

If you are working with Linux, you'll need to make the directory mediawiki/config writable. The best way to do this is under mediawiki with the command

```
chmod a+w config
```

Setup via browser

To obtain updated information, first read the *README* file in the directory *mediawiki*. Then, to set up your wiki using the browser of your choice, enter the address of the installation directory; in our case, it is

```
http://localhost/mediawiki
```

This is the address under which we will subsequently access the wiki. You'll first only see the message "You'll have to set the wiki up first." You can find out how to complete the installation in the section after the next (2.2.3).

2.2.2 Version 2: Installation on a Web Host

To install MediaWiki in the Internet, you first need an address (URL) and storage space on the Net. So-called web hosts can provide one. We assume that you have FTP access to your web space. In addition, in order to operate MediaWiki, you will also need to be able to access a MySQL database and run PHP scripts.

[3] You can find this program on the CD under */tools/filzip*.

Unpack the compressed wiki archive to your hard drive, on a temporary directory called mediawiki. This creates a directory structure, and the data will be directly stored in the correct location. Now load the entire folder via FTP to the web folder of your web host. This is usually the folder you first access when you connect to your FTP server.

Now check the properties of the mediawiki/config folder on your web server. It must authorize writing capabilities for all users. If this is not set properly, you can change the properties directly with most FTP programs and make the folder writable.

For further installation, you will also need the access data to your database. This data includes information on the URL of the MySQL server, the name of the database, the database username and the accompanying password. If you have this information at hand, you can start the browser-supported installation by entering the URL of your homepage followed by "mediawiki," such as in this example:

URL `http://www.wiki-tools.de/mediawiki`

2.2.3 Completing the Installation in the Browser

You still have to complete a few settings before being able to work with your wiki. MediaWiki is very user-friendly in this regard, since all entries can be made via the browser. Just follow the link "Set the wiki up" with which you can access the installation page (see Fig. 2.3).

Checking the system environment

The script will first test your system environment to determine whether the existing software and its settings are suitable for the operation of MediaWiki. A warning is now issued to the test environment you have already set up, which relates to a PHP security setting. This is not a problem, however, as long as we are in a test system. You can change this setting by switching XAMPP to safe mode.

Naming your Wiki

In the next section of the installation page, you will be asked to provide some information on your new wiki. Some of these settings cannot be subsequently changed. This especially applies to the name of the wiki. It should thus be selected well. Good names are relatively short and to the point, e.g. TribeWiki. If you wish to use more than one word, you can use a WikiWord, by writing the words together with the initial letters capitalized.

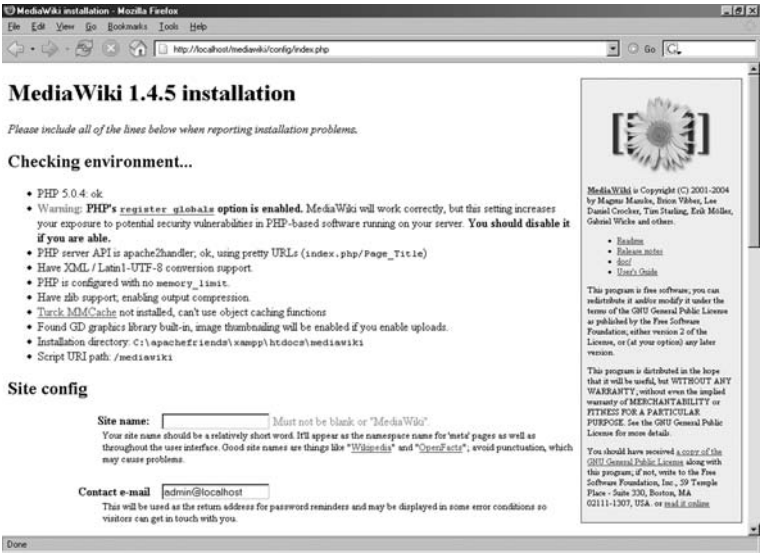


Fig. 2.3

Caution: Be careful with periods and special characters! Since some of them have special functions in wikis, their use in a title can lead to problems; they should thus be avoided.

Contact

Next, you should indicate an email address as a contact to which the MediaWiki can send a message in the event of a system error. In addition, forgotten passwords can also be sent to that address; i.e. it is possible that users will use the address to direct questions to you.

Language and coding

In the next field, you can set the wiki language and character code. Depending on what audience you wish to address and, above all, how international your group is, you may need to consider what language your wiki will be in if English is not your first choice. In the following field, your decision between Unicode or Latin 1 also depends on your target group. Unicode is more flexible and enables the display of many more characters, especially non-Western characters. However, it is not supported by older browsers. Yet because Unicode is the more future-oriented of the two codes, we recommend using it.

License

Now you can decide on the license that will govern your wiki pages. For an open wiki, it is not advisable to leave the question of licensing open, especially so as to avoid abuse of collectively produced wikis. You will find detailed information on ownership issues in Section V. Here, let us just say that the GNU Free Documentation License categorically excludes any subsequent commercial use of a text, whereby the Creative Commons License allows you to select

whether you want to allow others to be able to use parts of the text commercially.

Administrator account

In the next step, you will be asked to determine the user name and password of the administrator account. The administrator is initially the only person who can delete pages and block users. In an openly accessible wiki, we thus strongly recommend not using the suggested account name "WikiSysop," but rather selecting your own name and a safe password. This is also not a bad idea for a test system.

You do not need to change the "Shared memory caching" settings. They are only relevant for large wikis.

Database information

In the last section, you still need to provide information regarding the database. In our case, you can simply use the suggested entries. It is important to type in and repeat a password for the database to be created. Since the root password is not set in the MySQL standard installation, the MediaWiki setup can simply access the software and generate the necessary tables. This of course represents a considerable security gap. On a server on the Net, the root password should never be empty, because this account can delete your entire database. You only need your own "Database table prefix" if you wish to run more than one MediaWiki in the same database.

Tip: If you do not have access to your own server but are installing Wiki at a web host, you must first set up a database. In this case, you will receive the access data to be entered from your web host.

Complete installation

When you press the **Install!** button, the necessary settings will be made. If this has been successful, a further page will appear showing information on your current configuration. You will see a confirmation message at the end. Now you are almost done! The last task is to copy the file *LocalSettings.php*, which is located in the directory *xampp/htdocs/mediawiki/config*, to the directory *xampp/htdocs/mediawiki*, located one level higher. This step is necessary because, for security reasons regarding scripts under Linux, we have only set up write permissions in the config directory, and thus can only do the automatic settings here.

Open MediaWiki

After you have copied that file, follow "this link" or open the Wiki URL in the browser. You should now see an unadulterated MediaWiki before you (see Fig. 3.1).

3 First Steps

Readers
Authors
WikiAdmins
WebAdmins

Before we go into more detail about the opportunities offered by MediaWiki, let us provide you in the following chapter with a general impression of what it is like to work with the wiki.

3.1 The Wiki at First Glance

Description of the main page

At first glance, the start page hardly differs from a normal WWW page. The wiki is in view mode. The display can be divided into four areas (see Fig. 3.1):

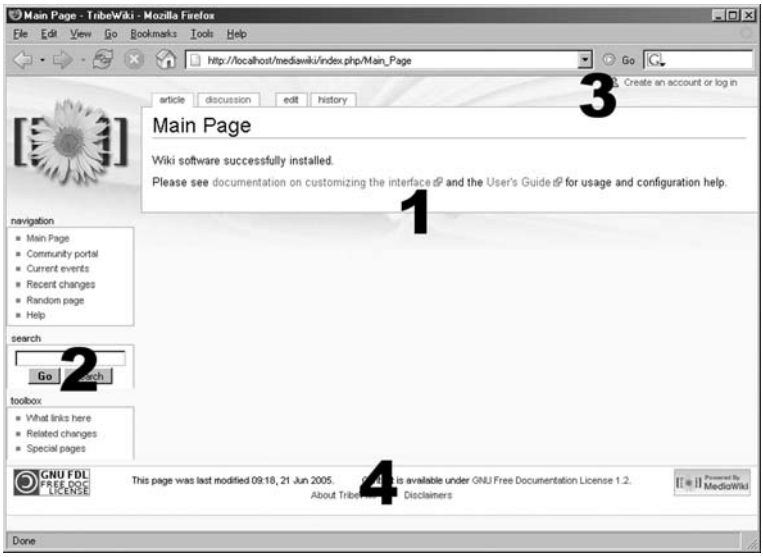


Fig. 3.1

- **Work area (1)**. The core of the wiki. This is where the actual content of an entry, which can be edited by anyone, is displayed. The form in edit mode and the difference and version numbers are also displayed in this area.
- **Navigation, search and tools (2)**. This area enables access to the articles according to various aspects. The image at the top left serves as a link to the main page. A few other functions can also be invoked from here.
- **User area (3)**. Login for registered users. They can enter personal settings or access pages they have edited.
- **Information (4)**. Shows a variety of information about the page, as well as licenses that apply to this wiki.

Work area

The work area contains all significant elements of a wiki. Here, all actions take place that can be performed on a page. It can be further divided into:

- **Tabs**. They provide the important functions for editing a page. It is the home of the most important wiki link: **edit**.
- **Page title**. The title describes the content of a page in one or two words. It also indicates which mode is currently active.
- **Content**. All users can design this area freely.

Up to now, there have only been a few pre-fabricated entries to TribeWiki. Now it is time to breathe life into your wiki and make a few changes.

3.2 Hello World

First entry

In tune with an old tradition in computer literature, our first entry will be "Hello World". To do this, first click on **edit**. After a brief loading period, the edit view of the wiki appears (see Fig. 3.2). Its central element is the text box containing the source text of the start page. Above this is a toolbar offering formatting aids. In the lower area, you will find the control elements.

You can alter the source text of the main page. To do so, position the cursor on the place in the text box in which the change is to be made (e.g. between the first line and the first paragraph) and type

Wiki

```
Hello World!
```

If you then click on **Save page**, your entries will be transferred to the wiki, and the normal view will once again appear, only now it will display your text.

Fig. 3.2

Create new page

One important aspect of the wiki principle is that every user can generate his or her own page. Thus, our next goal is to generate the page *Hello World*. It is not absolutely necessary, but still a good idea, to link a new page with the other pages of the wiki. So we will first create a link to *Hello World*. For this, we once again switch to edit mode. Now, enter the following anywhere:

```
[[Hello World]]
```

Wiki

and save the page. The double brackets cause the phrase “Hello World” to be interpreted by the wiki software as a link to the page *Hello World*, and a corresponding link is displayed on the *Main Page*. Since the page does not yet exist, it is colored red. If you now switch to *Hello World*, the edit view is automatically displayed. Here, you can once again enter any text. After you have saved it, you will see the newly created page. In order to return to the *Main Page*, click on the image on the upper left.

Tip: Try to avoid using the Back button as much as possible. Many browsers employ page caches to shorten loading times, so you may receive an obsolete version of the page. In the worst case, the

browser may send the form data from the edit mode again, and any new changes would be overwritten.

Now the link to *Hello World* on the *Main Page* appears in blue, since the page now exists. Of course, you can also create links to already existing pages using double brackets around the title.

Title selection

This brings us to the point of selecting a suitable title. Especially in wikis with several entries, it is not uncommon for authors to set links to pages they think will be relevant to their entry on a hit or miss basis. Thus, you should always make sure the title is short and describes the page as precisely as possible. Single words are suitable, but short sentences, such as "Why wikis work," can also be useful names. In addition, succinct page title names facilitate user orientation and article searches.

3.3 Initial Formatting

Both tasks described above represent the basis for working in a wiki. Principally, you can use them to take part in the generation and editing of a collaboratively administered page. However, the pages we have created up to now still look quite sparse, since we have completely ignored formatting and layout.

Formatting in a wiki

Although wikis are designed such that users need not busy themselves with layout questions, it is of course often desirable or helpful in understanding a text if options for highlighting and formatting text are available. They certainly do exist in MediaWiki. Since, however, entries are purely text-based, formatting cannot be immediately implemented, as is the case with a word processing system in a graphic interface. Rather, you must tell the wiki software that certain words are to receive special emphasis by using special character combinations.

To give you an idea of how the formatting method in a wiki works, first generate a test page as described in Chapter 2.2. We have chosen the name *Format Test* for our test page. Now enter the following text:

Wiki

```
Flying Sparks Tribal Initiative
The purpose of this page is firstly, to pro-
vide up-to-date information and secondly, to
coordinate our work.
We need YOU!
```

When you save the page as is, you will notice that even the sparse formatting in the text has been lost. This is because in MediaWiki, as in HTML, so-called whitespace, that is, characters that are not displayed (such as blank spaces and line feeds), are initially ignored.

Paragraphs

In order to insert a paragraph, you need to insert an extra blank line in the source code:

```
Flying Sparks Tribal Initiative

The purpose of this page is firstly, to
```

Italics

To lend the last sentence more verve, let's put it in italics. To mark text in italics, embed it in two apostrophes both before and after the area to be italicized:

```
''We need YOU!''
```

Bold

Strongly emphasized words in a text are generally set in bold type. This is done in MediaWiki using three apostrophes:

```
to '''coordinate''' our work.
```

Underlining is not one of the formatting options, because this type of emphasis is used to indicate links on the WWW, and confusion can be avoided by refraining from its use.

Headings

Headings are another simple way to add structure to your text. You can generate them by surrounding the respective text in double equal signs:

```
==Flying Sparks Tribal Initiative==
The purpose of this page is firstly, to
```

Since a heading is always on its own line, you do not need to mark the beginning of a new paragraph by inserting a blank line.

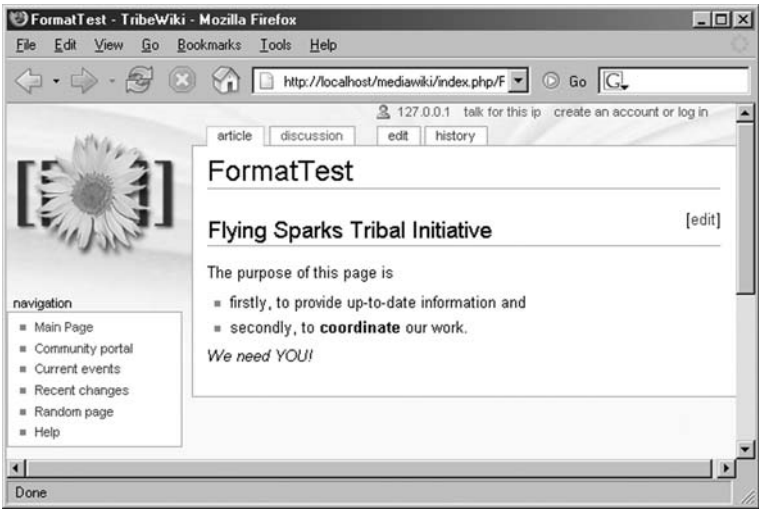
Lists

Lists are very useful layout elements, because they can be employed for quickly generated outlines or brainstorming texts to add structure. Lists are easy to generate. Place a star at the point in the text where a bullet should appear:

```
The purpose of this page is
* firstly, to provide up-to-date information ↵
and
* secondly, to '''coordinate''' our work.
```

The result of our newly designed page can be seen in Fig. 3.3. Of course, MediaWiki offers many more text layout options, which we will discuss in detail in Chapter 4.

Fig. 3.3



3.4 Vive la Difference: Versions

View history

You probably saved your changes more than once during the last section. If not, make another change to your text and save it. Generally, a wiki enables at least recent versions to be viewed. Click on the *history* tab. You will see a list of all former versions of the article and who edited them. If you click on a date, the respective version will be displayed. Use this function to undo changes. Simply open the older version, switch to edit mode and save it.

View differences

If you click on "current" ("cur") in any line, it will bring you to the difference ("diff") display. This shows you the differences between the current version and the version you select (see Fig. 3.4). Especially in the case of lengthy texts, you can quickly determine which areas have been modified.

Recent changes

Now, click on *Recent changes* in the navigation bar. Our page *Format Test* is also listed here. This function page always generates an overview of which articles in the wiki have been recently revised.

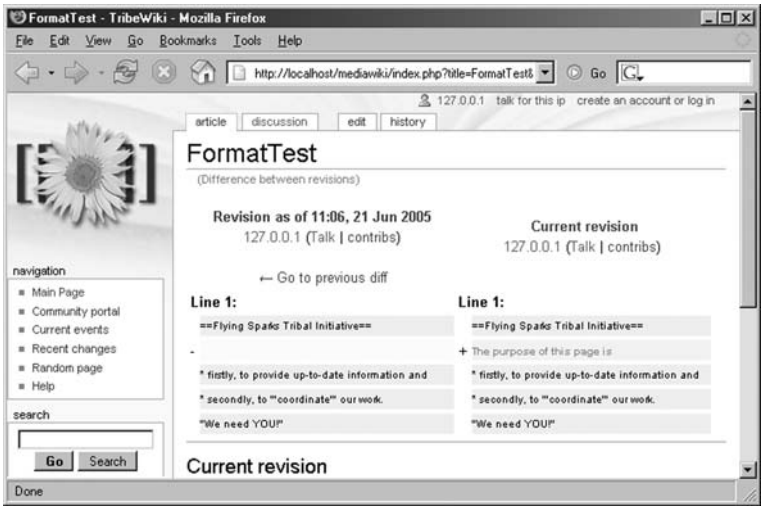


Fig. 3.4

3.5 Register as a User

On the last stop on our tour, we advise establishing a user account. A few of the actions described below can only be undertaken by registered users. In addition, it is considered proper etiquette in an open wiki to make yourself known if you are editing articles.

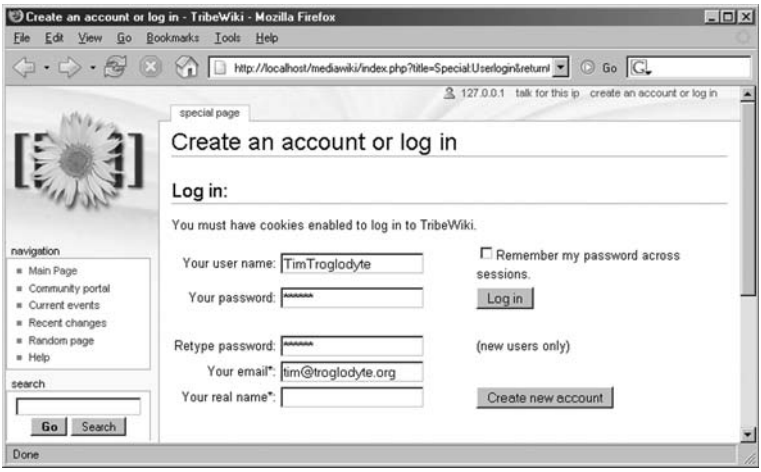
Create a user account

In the upper right corner of the screen, you will see a link to create an account or log in. Follow it to reach the registration form (see Fig. 3.5). Under "Your user name," select and enter a name, then select "Your password." You must retype the password on the next line to prevent typing errors. You should also enter your email address under "Your email," so that your password can be sent to you, should you forget it. When you have finished, click on **Create new account**. You are now registered as a new user, and logged in as well. You can verify this by looking up where the "Create an account" button had been; in its place is now a series of links, the first one being a link to a page with your user name.

Logging in and out

You can log out using the "Log out" link. On your next visit, you need only to enter your user name and password; the system already knows the rest.

Fig. 3.5


Create an account or log in - TribeWiki - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/mediawiki/index.php?title=Special:Userlogin&returnt
Go
127.0.0.1 talk for this ip create an account or log in
special page
Create an account or log in
Log in:
You must have cookies enabled to log in to TribeWiki.
navigation
Main Page
Community portal
Current events
Recent changes
Random page
Help
search
Go
Search
Your user name: TimTroglodyte
Your password:
Remember my password across sessions.
Log in
Retype password:
(new users only)
Your email*: tim@troglodyte.org
Your real name*:
Create new account
Done

4 The Core Functions of MediaWiki

Readers
Authors
WikiAdmins
WebAdmins

The following will illustrate the main functions of a wiki. This will be done step by step, using the tab pages, which enable various types of access to a respective page. Which tabs are visible and thus accessible depends on your status within the wiki. Thus, registered users and administrators have access to more functions than anonymous users.

4.1 Editing

Editing window and toolbox

When you click on the **edit** tab, the edit mode of an article is invoked; that is, the text on the page is loaded along with the wiki-specific format characters ("tags") in an editing window, and can now be manually modified or supplemented.

Note: For lengthy texts, it is also possible to edit it in sections. In this case, you will find the **edit** button to the left of the section heading.

Above the window, a toolbox also appears that can be of use when formatting text. It works in roughly the same way as in a graphic word processing program. You highlight the text to be formatted and activate the desired tool. The software initially only sets the corresponding tags around the highlighted text. As in manual editing, you won't see the effects until you select Preview or after saving it.

Toolbox functions

The functions of the toolbox are as follows (the second column shows the tag inserted by the corresponding tool):

Tab. 4.1

Image	Replacement	Function
	`'''bold'''`	Text appears bold.
	`''italics''`	Text appears in italics.
	`[[wiki page]]`	Links to an internal wiki page.
	`[www.google.de]`	Links to an external page on the WWW.
	`==Heading 2==`	Text appears as a Heading 2.
	`[[image: example.jpg]]`	Integrates an image previously loaded in the wiki (see Chapter 6.3)
	`[[Media: example.mp3]]`	Links to a previously loaded media file, e.g. an audio or video file.
	`<math> </math>`	Mathematical formulas in LaTeX can be placed within the set tags.
	`<nowiki> </nowiki>`	Any existing format instructions are not executed within these tags.
	`--~~~~`	Depending upon whether you are logged in or not, this function inserts your IP address or user name and a date stamp.
	`----`	Inserts a horizontal line.

Generally, in the editing window, you also have access to an "Editing help" page, which you can access by clicking on the link of the same name. Unfortunately, this page is not yet filled with content in MediaWiki, and is thus awaiting your constructive assistance (see Chapter 9).

Cancel edit mode

By the way, you can leave the edit mode at any time by clicking the **Cancel** button. This will display the page in its previous state without integrating any of your changes.

Preview

If you are not sure whether your formatting efforts have had the desired effect, you can view the results with the **Show preview** button without the changes being saved (see Fig. 4.1[4]).

[4] Perhaps you have noticed that we have changed the logo. You can find out how to do this in Section 7.1.

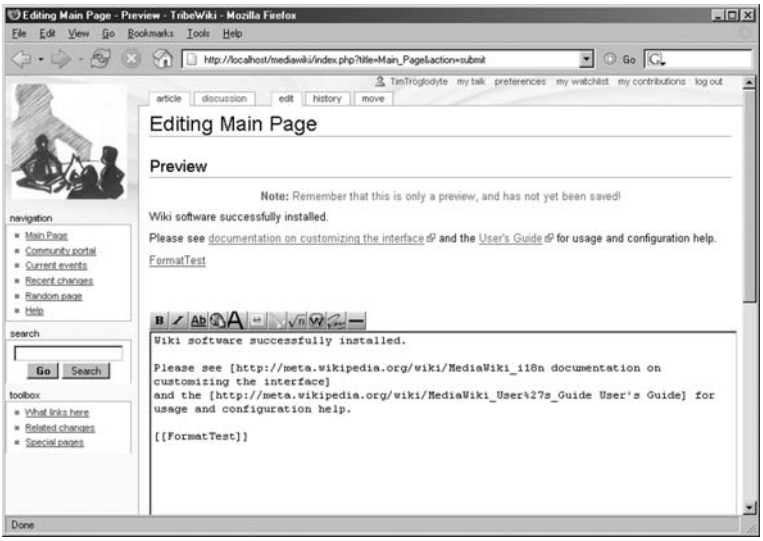


Fig. 4.1

Generally, the preview is displayed above the editing window. That means you might have to scroll up a bit to see it. You can make further changes in the editing area without having to leave the preview mode.

Caution: Do not click on **edit** again while in preview mode, otherwise you will lose your previous changes!

The preview mode has the advantage that the user receives the opportunity to proofread his text in formatted form, as well as that the history page is not loaded up with documents that are edited repeatedly solely due to faulty formatting or careless mistakes. However, note that the versions displayed in the preview mode are not saved versions. Thus, do not forget to save your work in the proper manner once you are happy with the way it looks.

Summary

If you wish to save your text, you can enter a comment on your text or changes in the Summary box under the editing window. This space has a capacity of 200 characters. Anything longer than that, such as when copying and pasting, will not be included. The summary is displayed in black italics on the *Recent changes* page, as well as the history and differences pages.

Caution: After you have saved a page, a summary cannot be edited without altering the main text area again. In the event of serious mistakes or significant omissions, you would thus have to feign editing in order to edit the summary, i.e. by making a small change, such as adding a blank space or blank line within the text.

Conventions

With regard to the summary, a few conventions have established themselves when using wikis in order to make working with them easier for everyone. One important guideline is that one should always provide a summary. Even a short description is better than none at all. Especially if you have deleted or modified another user's text, you should comment on it, so that the alterations are not interpreted as being malicious. In addition, modifications that are not explained are more likely to be undone. However, lengthy explanations should be reserved for the discussion page.

If you have only added a sentence or two to an article, it is advisable to copy them completely into the summary box.

Tip: Think of an abbreviation to use before the summary that will indicate to everyone that the summary includes all of the modified text, so that people will not have to go through the trouble of looking up the page on the *Recent changes* page. For example, one common abbreviation used at Wikipedia is `ft` for "full text."

Minor edit

Registered users additionally have the option of defining their modifications as "minor edits," i.e. to indicate relatively insignificant changes, such as in correcting a typing error or making any formatting changes. Before saving, simply mark the box next to "This is a minor edit" below the summary.

On the other hand, there are major edits that mainly affect the content of the article. These suggest to the user that it is worthwhile to look at the modifications, since a significant development of the text has taken place. This may even be the case if only one word is changed.

Differentiation between minor and major edits is relevant, since registered users have the option of hiding unimportant alterations on the *Recent changes* page. The decision as to whether modifications are major or minor is, of course, a subjective one, and may lead to readers missing out on important information. That is why you should think carefully about how other users would assess your modifications. It is for good reason that this function is unavailable to non-registered users, to prevent abuse.

Tip: If you have saved alterations as being minor by mistake, we advise just feigning an editing procedure and making a note in the summary that the previous edit was significant.

4.2 Discussions

Discussion page

The content of a respective article can be discussed on the discussion page. When working jointly on a text, the need for a content-based discussion will inevitably arise, especially if participants do not yet know each other: Any misunderstandings must be eliminated or ambiguities debated. Since this communication is very important but – as we know from dealing with forums – can often be very tedious and have little constructive value, it is taken out of the actual text and placed into a discussion page. Furthermore, the content on an article page reflects the current consensus of that topic. Users only wishing to read the article to gain information (such as in the encyclopedic entries of Wikipedia) are usually not interested in a discussion of the further development of the topic.

Taking part in discussions

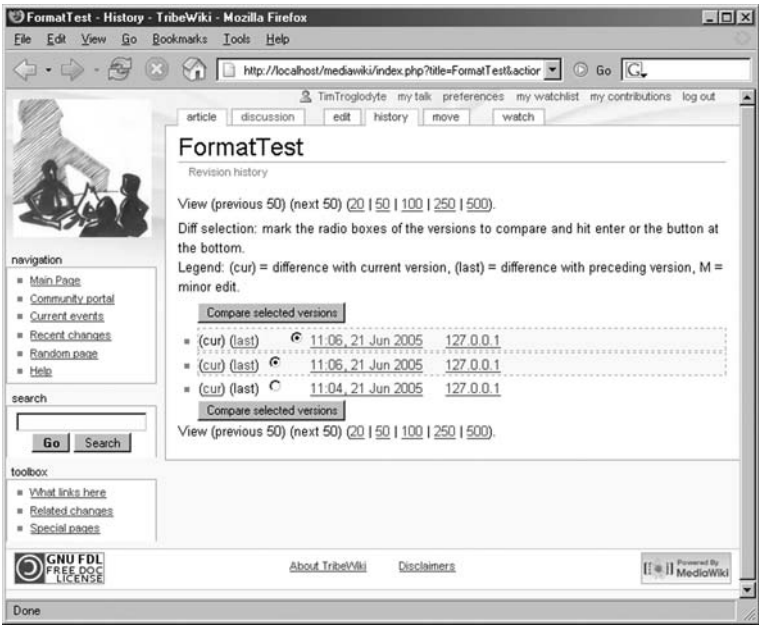
The discussion pages can be edited in (almost) the same way as the article pages: If you go to Edit, you can modify comments already written. However, since discussions should be easy to follow for subsequent users and thus must be documented, this function is of little purpose. By clicking on the plus sign in one of the tabs in discussion mode, you can immediately add to the discussion without needing to load all previous comments to the editing window. A subject heading appears under the toolbox. If you enter text here, this heading appears as a second-degree heading to your discussion entry, which is added after the other comments. This creates a structure very similar to that of a forum. However, remember that one of the greatest advantages of a wiki is that you can edit the entire text. Thus, if you have comments regarding a specific entry that has been discussed further up on the page, you should place your comment directly under this entry. Only new ideas or participation in a current discussion should be included with the "+" function.

4.3 Comparisons

History page

When you click on the **history** tab, you can view the development process of a text from the most recent modification to the first time the article was saved, i.e. its history. Deleted pages, which are no longer visible and may only be accessed by administrators, are the exception to this rule (see Chapter 8.2). Each line represents a former version, and includes various additional information, such as the point in time in which the text was saved. The versions can be compared in a variety of combinations.

Fig. 4.2



The lines are constructed as follows:

- If you click on the beginning of a line, that is, on "(cur)," you will get a comparison of the respective version and the current article.
- If you click on "(last)", you will see a comparison of the version in that line with the preceding version (in the line below it). Thus you will only see the changes made by the participant that generated that edit.
- Using the radio buttons, you can mark any two versions and view the differences between them by clicking on the **Compare selected versions** button at the bottom of the list.
- The radio buttons are followed by the time and date when the edit was saved. They lead directly to the previous version.
- If the originator of the edit is logged in, you can then see his or her name which leads to the respective user page. Otherwise, only the IP address is visible.
- If an author has provided a summary of his edit, it appears at the end of the line in italics and brackets. It is not linked.

Diff page

Above and below the version list, you will find the button **Compare selected versions**, with which you can compare the versions you have marked to the smallest detail. The result is displayed on its own page (see Fig. 3.4). On the left, the older version is shown, and on the right, the newer one. Only those lines are displayed in which modifications were made, whereby the text highlighted in yellow is deleted data, and the green lines represent newly added text. Under that display, you will find the newest of the two texts again in full.

Moved pages

If a page has been moved, that is, if its name has been changed (see below), the entire history of the article before and after the renaming is displayed on the history page, however the act of renaming the article itself is not listed. This is only visible in the history of the page with the old title.

Rollback

One action often performed is reverting to an older version of an article (e.g. due to vandalism), called a rollback. To do this, on the diff page, open the version to which you want to revert. If you wish to make that version the current article and thus undo all subsequent versions, go to **edit** and then **save page**.

Note: The entries to the history page are not deleted; thus a subsequent user could also roll back your rollback.

When you view an old page, you will see two arrows under the heading with which you can jump to the previous or subsequent version.

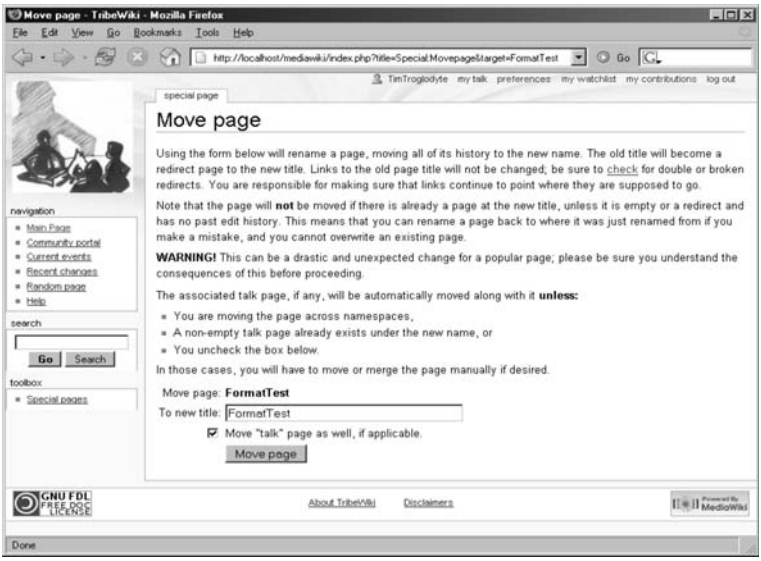
4.4 Moving and Renaming

Renaming pages

Sometimes it is necessary to rename a page, e.g. if the name has been misspelled or does not comply with the stipulated conventions for names, or if the scope of the article has changed. In this regard, the terms “move” and “rename” mean one and the same, because when you change the name of a page, links to that page, which are generated using the name of the page, become invalid. Thus, the page has been moved.

Open the page you wish to rename and activate the **move** tab. You can enter the new name in the space provided. By clicking on the Options field, you can also determine that the corresponding discussion page be moved, as well. Once you have verified the move by clicking the **move** button, the page is renamed and, at the same time, the old page becomes a redirect page, i.e., all links leading to it are automatically forwarded to the new page.

Fig. 4.3



Tip: The “move” function ensures that the entire history of the page before and after the renaming remains traceable. The alternative, namely cutting out a text and placing it into a new page, has the crucial disadvantage that the previous development of the page can no longer be traced. If this procedure is nevertheless necessary, e.g. if the content of a title is divided, it is advisable to note in the comments where you have obtained the text.

Rolling back a move

If you wish to roll back the move of article A to B, that is, if you wish to give the page its old title, this is only possible if the redirect page has not been edited in the meanwhile. In such a case, move B back to A and have an administrator delete B so that not too many redirects remain in the wiki.

Tip: In order to continue to be able to use the old name of the file for the new content, we recommend a switch. For this, you'll need an interim page which can be deleted once you are finished with it. First move A to Z, then B to A, and finally, Z to B.

4.5 Watching

As a logged-in user, you will see a tab at the top of the wiki pages called **watch**. When you click this tab, the current page and respective discussion page are added to a list of articles in whose development you are particularly interested. The tab then changes to read "**unwatch,**" and by clicking it, you can delete the page again from your watchlist.

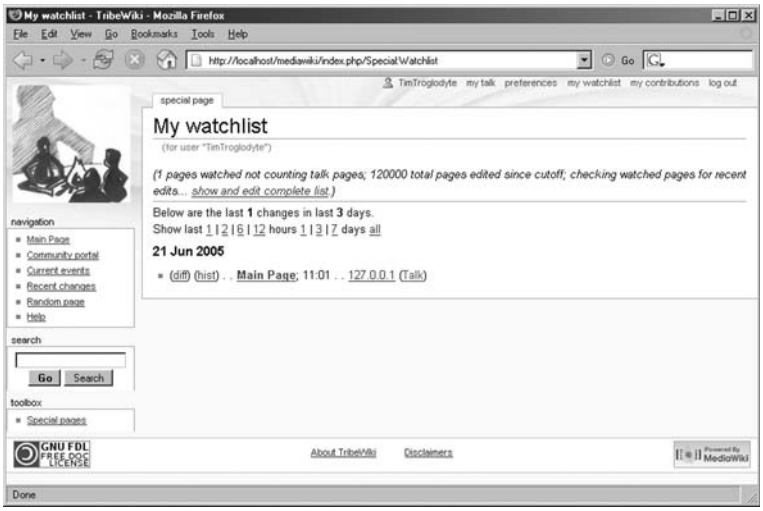


Fig. 4.4

You can open your personal watch page via the Special Pages list or the "my watchlist" link in the top row of links. The watchlist functions the same way as the "recent changes" list, except that only the pages you have selected to watch are included. Modifications to those pages are listed from the most recent edit back to a certain time limit.

Every line shows details about the most recent editing procedure: the date, whether it was a minor edit, a link to the page, the difference to the last version, the user name and summary. Currently, there is no option to hide minor edits here.

Tip: In the *Recent changes* list, watched pages are displayed in bold lettering. Thus, even if you never look at your watchlist, setting watch status can still be worthwhile.

4.6 Protect

If you have the status of a system administrator (see Chapter 6.3), you have access to the **protect** tab and can use it to protect a page from being edited by others. A protected page can then only be modified by administrators. Other users only have the option of viewing the source code. You can release the protected status with the **unprotect** tab.

Important pages

This function can be used to exclude a few important pages from modification. For instance, it is reasonable to want to protect your start page from anyone being able to edit it at will. However, in order to maintain the spirit of open wikis, you should at least generate a suggestion page in which new concepts for the title page can be discussed. Of course, the discussion page can also be used for this purpose.

Fig. 4.5

Protection from vandalism

Unfortunately, it is also necessary on occasion to prevent modifications to a page temporarily if differences of opinion on its content have heated to the point where constructive work on the article is no longer possible. Since such blocks are always at the discretion of an administrator, at least two participants should have administrator status, in order to avoid centralist tendencies (the "dual control principle").

4.7 Deleting

Administrators can remove pages and their complete history with the **delete** command. Such pages are placed into an archive and are no longer accessible over the WWW. However, they are only removed from the database when specifically deleted by a sysop (see Chapter 8.2). The deleting of pages should be done with the utmost of care, since one is generally dealing with a work authored by several participants, who, under certain circumstances, would not be happy to discover that their contributions have been removed from the wiki. There is often a page in wikis where potential "delete candidates" are suggested and discussed before an administrator may remove them from a wiki (with "may" we refer to a social, not a technical, limitation). Good delete candidates include the redirect pages mentioned above, which are generated when pages are moved.

Fig. 4.6

5 Formatting

Readers
Authors
WikiAdmins
WebAdmins

In addition to standard emphasis in the form of "bold" and "italics" cited above, you can utilize a series of further options in MediaWiki, even if the possibilities of formatting with wikis are clearly not comparable to those of conventional word processing programs such as Word. In addition, there is no WYSIWYG ("What You See Is What You Get"); that is, the formatting is set, as is the case in HTML or LaTeX, in the form of tags, i.e. ASCII characters in editing mode, and is only visible after saving it.

The formatting codes can be divided into two major types. The first are tags that are very similar to HTML codes and work in the same way. That is to say, you place the text to be formatted between an initial and a final symbol, such as `<b>Text</b>`, which displays the word "Text" in bold. On the other hand, one can also set wiki-specific markers that precede the text or surround it. It does not matter which type of symbols you use; the only difference is that the wiki-specific tags cannot be used for text spanning more than one line, whereas HTML tags can.

5.1 Formatting Characters

Bold + italics

As mentioned, two apostrophes before and after the text to be formatted italicizes the text, and three apostrophes around the text allow it to appear in bold. If you want your text to be both bold and italicized, you simply use five apostrophes.

```
'''''coordinate''''' our work.
```

Wiki

With regard to font and size, the selection is not all too great. In addition to the standard font Verdana, you can make your text Courier by setting it between `<tt>`, which imitates a typewriter font and looks fitting especially for technical explanations.

Wiki
```
firstly, to provide <tt>up-to-date informa-
tion</tt>
```

If you would like to generate a caption for a picture or a table or use a small font for any other reason, you can use `<small>` to do so.

Wiki
```
firstly, to provide <small>up-to-date infor-
mation</small>
```

Underlining/ striking

The tags are also very handy for striking and underlining passages using `<strike>` and. `<u>`, respectively:

Wiki
```
firstly, to provide <strike>old</strike>, no
<u>up-to-date</u> information
```

Superscript/ subscript

For mathematical expressions, the options `<sup>` for superscript and `<sub>` for subscript are available:

Wiki
```
<sub>P</sub>=1/2 D s<sup>2</sup>
```

Instead of `s<sup>2</sup>`, you could also write `s²`. However, this only works for superscript formatting, and here only with numbers as exponents.

5.2 Special Characters & Co.

Now we come to the numerous special characters, punctuation marks, symbols and language-specific diacritics, such as a tilde or accents. As in German umlauts in HTML, these symbols are replaced by a type of letter pattern that always begins with a `&` and ends with a `;` such as the Greek letter sigma: `Σ`.

A complete listing of all special characters would be too extensive; thus, we include only a selection of the most commonly used characters.

Tab. 5.1

Symbol	Code	Symbol	Code	Symbol	Code
à	agrave	ã	atilde	ç	ccedil
á	aacute	ä	auml	ñ	ntilde
â	acirc	å	aring	ø	oslash

Space

We wish to call special attention to the space symbol. It is not possible to generate more than one space with the space bar on the keyboard, because the wiki software only registers one space. Thus, if you wish to add several spaces in a row, you must use the special character ` ` as you would in HTML. This symbol also prevents a line break, which can be very useful when generating formulas.

You should also use special symbols to generate unusual punctuation:

Tab. 5.2

Symb.	Code	Symb.	Code	Symb.	Code
¿	iquest	»	raquo	¶	para
¡	iexcl	§	sect	•	bull
«	laquo	†	dagger	—	mdash
→	rarr	←	larr	°	deg

In our example, we can now add a new line directly under the heading:

Wiki

```
&dagger;&dagger;&dagger; Our fire is being
taken away &dagger;&dagger;&dagger;
```

And, at the end of the page, the following explanation:

Wiki

```
No fire &rarr; no development
```

Especially in the realm of the natural sciences, we often need old Greek letters. For all symbols that cannot be generated using the Latin alphabet, there are the following conventions:

Tab. 5.3

Symb.	Code	Symb.	Code	Symb.	Code
α	alpha	γ	gamma	Δ	Delta
β	beta	Γ	Gamma	Θ	Theta

As you may have noticed, it is not difficult to figure out what the rest of the characters in the table would look like.

Formulas

The other components of mathematical formulas can also be generated with special symbols. Here is a small selection of them:

Tab. 5.4

Symb.	Code	Symb.	Code	Symb.	Code
∫	Int	∑	sum	±	plusmn
∏	prod	√	radic	∞	infin

For more complex formulas, you also have the option of employing TeX-Tags that are integrated into the text with `<math>Formula</math>`. These are either converted to HTML or – especially if the characters are too exotic – a PNG image is generated.

<pre> and <nowiki>

Using `<pre>` and `<nowiki>`, you have the opportunity of suppressing the formatting of text (with the exception of generating special characters), i.e. allow the conventional formatting tags to remain visible after saving the file. These two functions differ only in the fact that `<pre>` sets the text in Courier and does not ignore line breaks and series of spaces.

Tab. 5.5

Code	Result
`<nowiki>no fire → ''no'' [[development]] </nowiki>`	no fire → "no" [[development]]
`<pre>no fire → ''no'' [[development]] </pre>`	`no fire → ''no'' [[development]]`

Comments

If you want some of your comments to only be visible in the source text by the subsequent authors, you can mask them by identifying it as comment, as in HTML.

Wiki

```
<!-- unnoticed -->
```

5.3 Headings and Paragraphs

You can establish various levels of headings by surrounding them with the respective number of equal signs:

Wiki

```
=Heading 1=
==Heading 2==
===Heading 3===
```

This results in the following display:

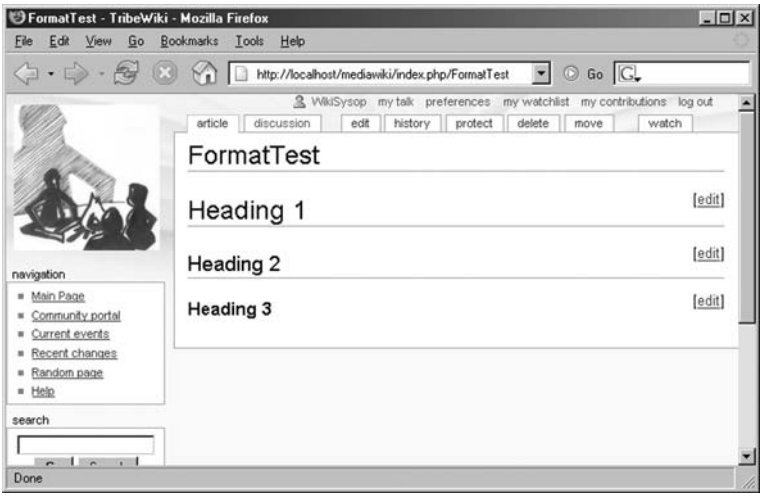


Fig. 5.1

At the same time, headings divide the article into sections which – using the proper setting – can be individually edited and accessed through links.

The headings can be classified in up to six levels, which corresponds to the <hx> tags in HTML. After the headings are formatted, an automatic line break takes place.

Now we can add a little structure to our tribal initiative page:

Wiki

```
= Flying Sparks Tribal Initiative =
== Purpose ==
The purpose of this page is...
== Description ==
no fire ...
== Demands ==
```

Line break/ paragraph indent

You can insert a manual line break within the standard text using <br>. This also applies to lists. In order to begin a new paragraph, place a blank line in the source code. You can insert an indentation with a colon at the beginning of the line.

5.4 Lists and Lines

Generating lists in a wiki is much easier than doing so in HTML code.

Unnumbered lists

In the case of unnumbered lists, simply insert a star in front of the corresponding list item. The farther down the level, the more stars are set in front of the paragraph:

Wiki

```
* First level
** Second level
*** Third level
** Second level
```

Numbered lists

You can employ the same principle with the # symbol for numbered lists:

Wiki

```
# Level 1
## Level 2
### Level 3
# Level 1
```

Mixed lists

A mixture of numbered and unnumbered lists is also possible:

Wiki

```
* Level 1 unnumbered
*# Level 2 numbered
*#* Level 3 unnumbered
*## Level 3 numbered
```

Definition list

If you wish to generate a glossary-like list, the definition list format is the perfect tool. In this case, the term preceding the semi-colon receives special emphasis.

Wiki

```
; Fire: provides energy and light
; Water: can control fire
```

Line

In addition to the possibility of structuring your text with lists, you can also add horizontal lines to your text with the following code:

Wiki

```
Text
----
Text
```

The last lists described are displayed in the following screenshot:

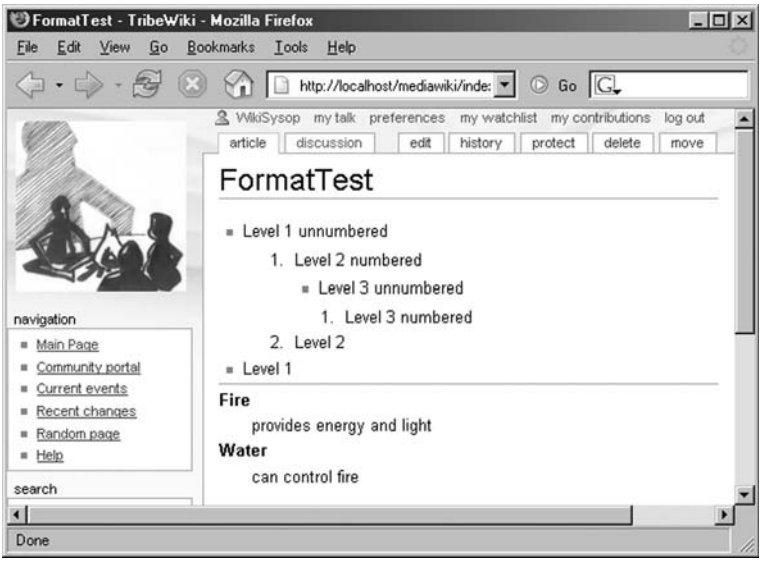


Fig. 5.2

Now we have the means to publish our demands for public access to fire:

Wiki

```
== Demands ==
# Fire must be accessible to all
# Its exclusive use by                    ↵
tribal executives is thus not acceptable
# Advanced measures must                  ↵
be undertaken:
** Sufficient supply of fuel
** Hazard control with water
```

5.5 Tables

MediaWiki supports three different kinds of table syntax:

- XHTML
- HTML and wiki-`<td>` syntax
- and so-called pipe syntax.

All three methods produce valid HTML code. The following will only focus on the third alternative, pipe syntax, because it is especially simple, space-saving, and it represents an element specific to

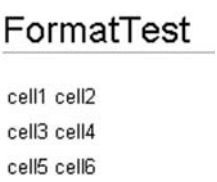
wiki. Here, the HTML tags are replaced by pipes (|), which results in a table, evident even in edit mode:

Wiki

```
{|
| cell1 || cell2
|-
| cell3 || cell4
|-
| cell5 || cell6
|}
```

You can see the result in Fig. 5.3:

Fig. 5.3



As is the case of HTML, the table is composed of three nesting levels:

Table level: You should always start a table on a new line. It is started using {| `param` and concluded with |}. The term `param` stands for optional parameters, such as background color or border thickness.

Row level: When you initialize a table, the first row is created along with it. Begin the subsequent rows with |- `param`. The row is automatically closed by starting a new row with |- or ending the table with |}.

Cell level: Cells can be generated in two ways. Either you set two pipes next to each other:

Wiki

```
| cell1 || cell2 || cell3
```

or you combine a new line with the pipe symbol:

Wiki

```
| cell1
| cell2
| cell3
```

The cell parameters, as the table and row levels, are situated behind the pipe symbol, but are separated from the cell content using a further pipe symbol.

```
| param || cell1 || param | cell2
```

Wiki

or

```
| param | cell1
| param | cell2
| param | cell3
```

Wiki

Caution: Every row must have exactly the same amount of cells, so that the number of rows remains constant. You should fill an empty cell with the space code ` `, so that the cell is displayed.

Let us return to the topic of parameters. They are based on HTML and can also be used in their abbreviated forms. Here, too, we differentiate between two levels:

On the table level:

Name	Function	Values
`align`	horizontal text alignment	left, right, center
`bgcolor`	background color	#000000-#FFFFFF
`border`	border thickness	number
`width`	width	number
`cellpadding`	internal cell spacing	number
`cellspacing`	external cell spacing	number

Tab. 5.6

On the row level:

Name	Abbr.	Function
`align`	al	horizontal text alignment
`background-color`	bc	background color
`valign`	va	vertical alignment

Tab. 5.7

On the cell level:

Name	Abbr.	Function
`align`	`al`	horizontal text alignment
`background-color`	`bc`	background color

Tab. 5.8

colspan	cs	several adjacent cells in a row are merged
rowspan	rs	several adjacent cells in a column are merged
valign	va	vertical alignment
width	wd	width

To make your heading or column headings appear in bold, when you open a row, instead of using the pipe symbol, use an exclamation point. Within the row, two pipe symbols before the respective cells are replaced by two exclamation points:

Wiki

```
{|
|-
! Control of !! Water !! Fire !! Both
```

Table heading

To give your table a heading, following the table initiation, that is, after the {| and any parameters, insert a |+ and then enter the column names.

Example

The following example contains all discussed table elements:

Wiki

```
{| border="1" cellpading="2"
|+Development of Humans
|-
! Control of !! Water !! Fire !! Both
|- align="right"
! primitive
| 70 || 20 || 10
|- align="right"
! average
| 20 || 50 || 30
|- align="right"
! sophisticated
| 10 || 30 || bgcolor="lightgrey" | 60
|}
```

Result:

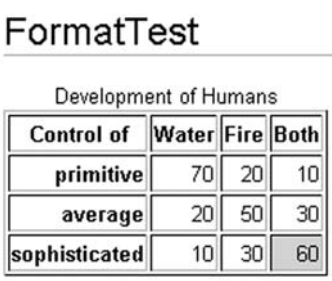


Fig. 5.4

MediaWiki also allows you to nest tables. To do this, simply start a new table:

```
{| border=1
| Elements ||
{| bgcolor=#ABCDEF border=2
| Fire
|-
| Water
|}
| are especially important.
|}
```

Wiki

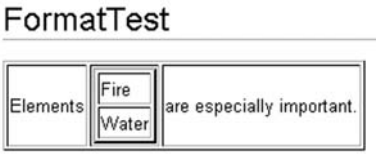


Fig. 5.5

Let us summarize the most important points regarding tables:

- {| ... | ... | ... |} encloses a table
- |- indicates the begin of a new row
- || separates cells
- ! ... ! displays a table heading
- Parameters following {| define the settings for the entire table
- Parameters following |- apply to the following row
- Parameters following | apply to the individual cell

5.6
Table of Contents

A table of contents is automatically generated for articles having more than three headings, unless

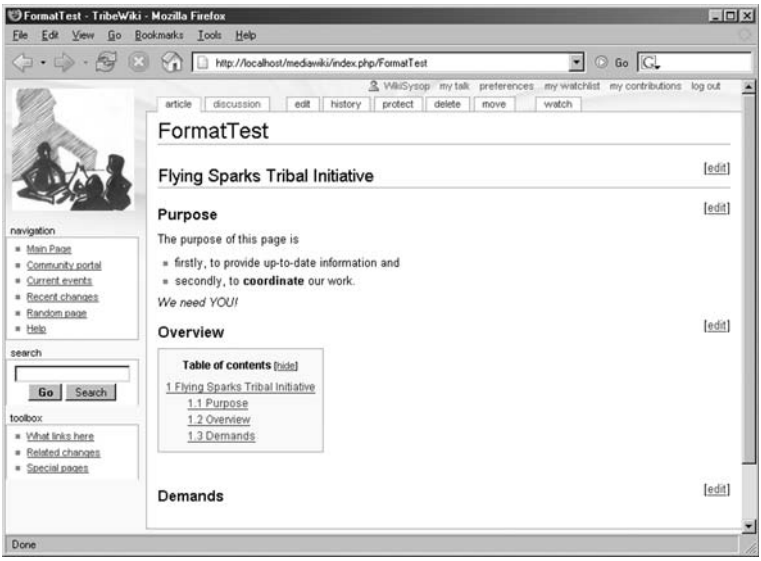
- a user's settings are configured differently (see Chapter 7.3)
- the code __NOTOC__ has been added in the edit box.

The numbering is generated automatically. The table of contents is located before the first heading, yet after any introduction that might be included. However, you can also insert the table of contents in another location with __TOC__.

Note: The headings of a template do not appear in the table of contents.

Entries to the table of contents are obtained from the headings of an article.

Fig. 5.6



6 Multi-Page Structures

Readers
Authors
WikiAdmins
WebAdmins

Up to now, we have dealt with formatting within one page. However, hypertext has the capability of establishing a relationship between various units through links. There is a variety of possible structures to use, and they have a pivotal influence on the manageability of a hypertext. It facilitates navigation through the pages, since related themes are linked. In addition, embedding an article into a network creates a context that can be helpful in classifying content.

6.1 Linking Methods

There are three levels on which links can be generated in a wiki: "interwiki" links within the wiki, internal links within a page, or external links. We will have a closer look at them below.

Interwiki links

We have already seen interwiki links in Chapter 2.2. They link to pages within a wiki. Thus, it is sufficient to cite the title of a page for such a link. Title pages are always distinct in a wiki; there can never be two pages with the same name. Remember: An internal link in its simplest form consists of a link target in double brackets:

```
[[page title]]
```

Wiki

Page titles consist of letters and numbers. It is also possible to use special characters. The following symbols, however, may not be used:

```
" # $ * + < > = @ [ ] \ ^ ` { } | ~
```

These characters fulfill special control and layout functions in the wiki source code or in PHP. Spaces may be used. They are stored within the system as an underline. Conversely, this means that underlines that you use in a title will be displayed as spaces.

Naming links

The wiki software takes care of formatting the links. It is assumed that the name of an article is distinct enough to indicate to the reader what the respective page is about. That is why the page title is simply used as the link name. Sometimes, however, it can be useful to have another description. This is the case, for instance, when you have a hierarchical order of pages for which a link "upwards" in the structure facilitates navigation. Also, synonymous descriptions, e.g. "tribe" and "clan," for two different pages should not be used, since they indicate the same thing. The necessary separation between link target and name can be achieved with a so-called "piped link":

Wiki
```
[[page title|description]]
```

As you can see, a horizontal line (pipe symbol) is placed after the page title, behind which you can insert a page description that can consist of one or more words.

Parts of words as links

Within the context of a sentence, a word can have various forms based on the rules of grammar. Since all of these forms link to the same page, the possibility was created to mark only the front part of a word as a link:

Wiki
```
[[Link]]s
```

In Preview mode, however, the whole word is identified as a link. The following construction is a rather exaggerated example.

Wiki
```
 [[caves|cave]][[dwelling]]
```

Here, two links are included in a single word. The first part of the word links to "cave", the second to "dwelling." One could surely debate about whether or not such a structure is very useful, especially since, when looking at it, one cannot detect that two different links are embedded in the word. However, the point is that it is possible.

Now it is time to give our wiki example some structure. We can divide the wiki into assorted pages that fulfill various tasks.

Wiki
```
Work areas:
* [[Info booth | Planned information booth]]
* [[Petition | Letter to the chief]]
```

```
* [[Summer Festival | Preparation team for ↵
the summer Festival]]
* [[Material | Link and material collection]]
```

Now, of course, the pages have to be filled with content. Don't forget that, for every page, there is also a discussion page. Thus, on the *Petition* page, you can directly read the current version of the text and make modification suggestions on the discussion page. However, we'll leave that to the members of the tribal initiative for now.

Internal page links

Especially in large documents, it is a good idea to establish links between elements of the document itself. This is especially very useful when you want to enable readers to be able to jump from an item in a table of contents to the respective portion of the document. Since the page name is the same for internal page links, you will need to define additional link targets within the article. This is a fairly intuitive process in MediaWiki: The sections serve as jump labels. All levels can be used with their names, specified as headings. Internal page links are identified by a hash symbol (#):

```
[[page title#section title]]
```

Wiki

If a link refers to the same page, the page title can and should be left out. Of course, internal page links have the same formatting options as in internal links. The only difference is that the wiki software will not verify whether the link target exists on the page. If the target is missing, the link is formatted as such but will have no effect.

MediaWiki article URLs

Wiki is a component of the world wide web. Thus it offers the possibility of external links leading to wiki, as well as links leading from a wiki to the WWW. As you can see in the address line of your browser, wiki pages are loaded via the URL in which, at the end of the address, in our case following http://localhost/mediawiki/index.php[5], a slash is placed and the page name is added. It is also possible to load the edit view of a page directly via URL. To do this, the page title as well as the desired action must be specified:

```
index.php?title=page_title&action=edit
```

URL

Please note that, in this case, any spaces are to be replaced with underlines.

External links

To link to an external page from MediaWiki, all you need to do is write the complete URL in the source code. Since it begins with http://, the software recognizes it as an external link and will format

[5] This path may vary depending on the installation; see Chapter 8.1.

it accordingly. In links to the WWW, a special character appears behind the text. It indicates to the user that he or she will leave the wiki using the respective link. External links of this type are not limited to the WWW with HTTP protocol. MediaWiki also understands https, ftp and mailto links, and assigns them their own corresponding symbols.

Formatting a URL

Placing a URL in a text makes the text harder to read and is thus not advisable (except when the exact address is the subject of observation). That is why MediaWiki offers two ways to format external links. The first is to simply surround them in brackets. Such links are consecutively numbered on the page from top to bottom. This is similar to the quotation procedure in scientific texts; however, it also does not facilitate smooth reading. If you wish to provide a description for your link, simply add the description following a space behind the URL in brackets:

URL

```
[http://www.wiki-tools.de·The page about the ↵
book]
```

More than one word is permitted. In this way, external links can be embedded in the text flow.

Now we can fill the material page of “Flying Sparks” with content:

Wiki

```
== Links ==
* The [http://www.humanitycouncil.de·        ↵
Humanity Council] publishes development re-
port
...
```

You can now link directly to these targets on the start page:

Wiki

```
== Statement ==
Tribes without Fire ...
* [[Material#Links|secondary links]]
```

Signing entries

It has become standard style in discussions to sign entries with the date and indication of the user's page, so that they can be allocated to participants. To do this, MediaWiki uses the abbreviation

Wiki

```
~~~~
```

When an article is saved for the first time, this code is translated into a link to the internal wiki page of the user and the current time. From

then on, this information is stored in the source text. It is replaced even if the user is not logged in; in such cases, the user name is replaced by his or her current IP address.

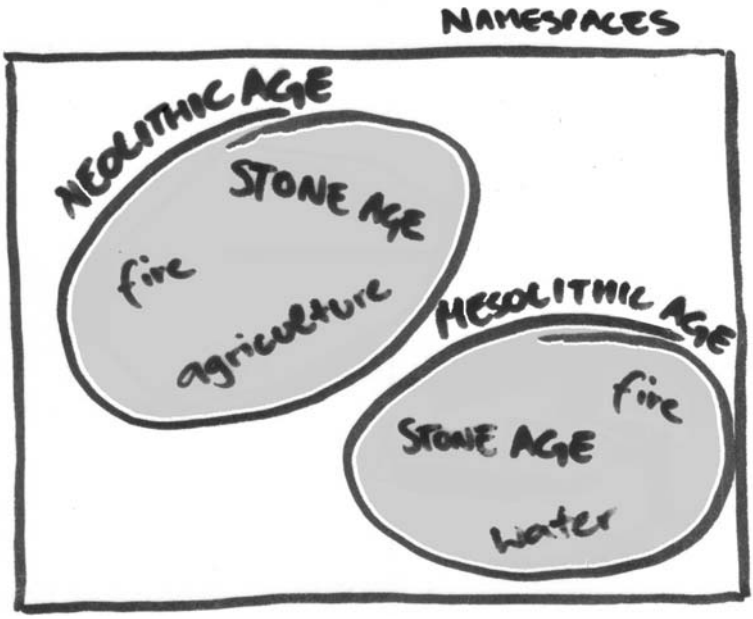
6.2 Organizing Content

The methods previously described create structures by linking to entries without providing any specific information about why there is a connection between them. MediaWiki also offers the possibility of grouping several pages according to function or content-related aspects. Namespaces and categories serve this purpose.

Function of namespaces

Namespaces allow a wiki to be divided into various areas. Pages having the same name can be established in different areas, so that, in principle, several groups of users can work on a wiki without worrying about whether a page name they have selected has already been taken by someone else. Thus, for example, the information booth and petition groups can each have a separate page in their own namespace with the same significant name *ToDo*.

Fig. 6.1



Namespaces in MediaWiki

However, in MediaWiki, the names of these areas are "hard coded." That means that they have been fixed in the source code of MediaWiki and may not be altered by users. Instead, they serve as a

separation between the various types of pages contained in the wiki. Entries with various namespaces may also possess additional functions. For instance, pages in the namespace *Talk* have an additional tab (+) that is automatically set in the articles at the end of the page. In the standard installation, there are 15 namespaces, the standard namespace (*Main*) and 2 pseudo namespaces, which are listed in Table 5.1. A page is allocated to the standard namespace as long as no information has been defined for it. The pseudo namespaces are used like normal namespaces, except that neither can pages be generated nor can existing pages be modified.

Tab. 6.1

Namespace	Function
Media	Pseudo namespace for uploaded images and files. Is replaced by the file path.
Special	Pseudo namespace for special pages (see Chapter 6).
(Main)	Normal pages. They are indicated without a prefix.
Talk	Discussion pages. They are created for every article (including in the following namespaces), for content discussion. Additional tab (+).
User	User page. Is generated for every registered user.
User_talk	See *Talk*.
[Project]	Information about the wiki or the current project.
[Project] _talk	See *Talk*.
Image	Description of an image or other files. Is automatically supplied with information regarding version and file sources.
Image_talk	See *Talk*.
MediaWiki	System messages. A complete list is available under *Special:AllMessages*. May only be edited by registered users.
MediaWiki_talk	See *Talk*.
Template	Templates may be integrated in other pages (see Chapter 5.4).
Template_talk	See *Talk*.
Help	Help pages describe how to perform various tasks in wiki. No clear separation of content with *[Project]* namespace.

Help_talk	See *Talk*.
Category	Pages can be allocated to categories which are then accessible in this namespace. See below.
Category_talk	See *Talk*.

Most pages in the namespaces are generated automatically. Special pages are generated by the system before installation, media and image pages are created when a file is uploaded, and user pages when a user is active in a wiki. 'Talk' or 'discussion' pages are also generated along with the respective pages. This leaves only the pages in *Main*, *[Project]*, *Template*, *Help* and *Category*, which can be designed freely.

Accessing pages in a namespace

To access a page in a particular namespace or, if it is not yet available, to create one, simply place its name before the page title and separate it with a colon:

```
[[Help:How do I set a link|]]
```

Wiki

In the example, the pipe symbol was also added. It prevents the namespace from being displayed along with the link, which facilitates reading. If you enter a namespace that the wiki does not recognize, the page is generated in the standard namespace *Main*. The portion of the link indicated as a namespace is supplemented by the title of the article and not identified as an area name. A page can only be allocated to one namespace. The same page name in another namespace generates a new page. In addition, namespaces only exist on one level; that is, a namespace cannot be subdivided into further namespaces. That is why a new namespace is created for each discussion page, instead of linking the namespace with the discussion page. In searches (see Chapter 6.2), the search area can be limited to individual namespaces.

Individual user namespaces

Unfortunately, there is currently no way for normal users to receive a listing of all namespaces. The generation of an individual user namespace is also only possible with access to the scripts (see Chapter 7.1.). However, one can use the syntax of the namespaces and simulate an area in the namespace *Main*, by adding an individual name as a prefix followed by a colon to the pages of a desired area. The general advantages of namespaces remain intact. For instance, it is possible to use pages having the same name in various “areas” and thus enable various groups to work on a single wiki. Since this individual “namespace” prefix is a component of the page name, one can now see a display of all pages within the “namespace” by performing a search of the prefix.

These individual namespaces are ideal for our fire project. For instance, we can place all of the meeting minutes in an area with the name *Minutes:* Each entry would be named for its respective date, such as *Minutes:12152004*. A further possible use is for work groups who would most likely want to generate several individual pages. The Summer Festival Group, for example, could generate their task list under the name *SummerFestival:Todo*, whereas the Information Booth Group would call their list *InfoBooth:Todo*, without any conflict between the groups. In addition, the pages are automatically allocated, so we know that every page whose name begins with "SummerFestival" belongs to that group.

Interwiki links

To create links between various wikis, a convention is used that is similar to that of namespaces: The interwiki links. Instead of using the name of a namespace, simply use the name of another wiki. However, the system must recognize these names, so that the URL can be replaced. You can find a few pre-defined wiki links in Tab. 6.2.

Tab. 6.2

Prefix	Wiki	URL
`TWiki`	TWiki	http://TWiki.org/cgi-bin/view/
`Wiki`	WikiWikiWeb	http://c2.com/cgi/wiki?
`WikiPedia`	Wikipedia	http://www.wikipedia.org/wiki/
`Google`	Google	http://www.google.com/search?q=

As you can see, you can link to other destinations besides wikis, such as search engines like Google. You can find a complete list of pre-defined interwiki prefixes in the file maintenance/interwiki.sql. You can learn how to make your own entries in Chapter 8.2.

Categories

A second type of organizing pages is through categories. This refers to a categorization of page content under a category name. These names are described in the *Category* namespace. Unlike namespaces, any number of categories can be generated and by any user. This is done by creating a page in the namespace Category. All you have to do to allocate a page to a category is insert a link to the category page in the source text.

Wiki

```
[[Category:MyCategory]]
```

Now a note appears at the end of the page indicating under what name the article has been categorized. If you wish to create a link to the category page itself, you must prefix the word "Category" with a colon. Just as in the case of creating new pages, you can create new categories with the categorization procedure.

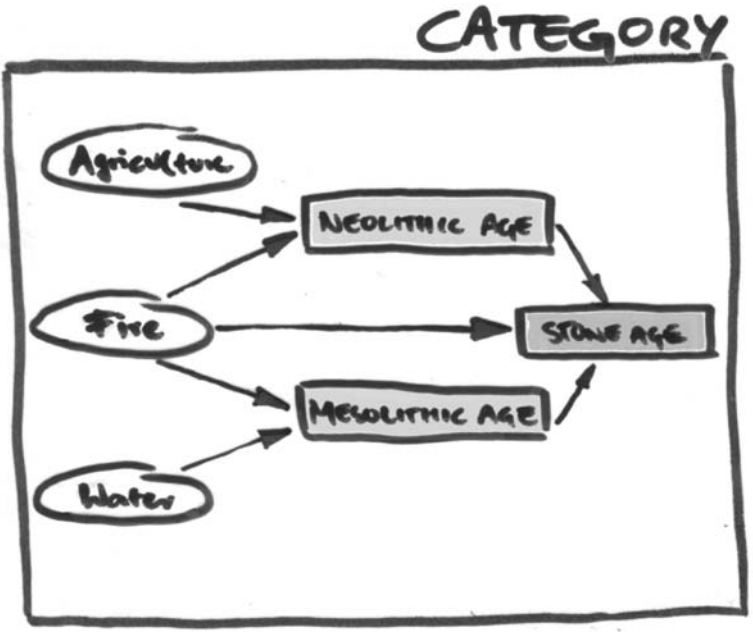


Fig. 6.2

Multiple categorization and position

A page can be allocated to more than one category. To do this, several of these category links are included. It does not matter where the categorization is inserted in the source text. It will always be shown at the end of a page. Thus, you can also insert them in places where they best fit the content. However, the order of the categorizations in the source text determines the order in which they are listed.

Category pages

Category pages function in the same way that normal wiki pages do. That is to say, you can edit them with normal wiki syntax. However, a category page has two further components that cannot be edited because they are generated automatically. The first is an alphabetical list of all pages under this category name. The second is a display of any existing sub-categories. Incidentally, it is also possible to organize categories into categories and thus create a hierarchical ordering system. To do this, insert a subcategory allocation on the respective category page.

Access to category pages

Access to category pages can be done via the special page *Special:Categories*. All existing category names are displayed on this page. They are ordered such that the categories having the most pages are at the top. Categories that have been created but do not yet have any pages allocated to them are not listed here. Since the order of the categories changes continually and the list is incomplete, this page is not suitable for first-time browsing. However, it is a simple way to insert a subcategory that contains all of the other groups as

elements. They are then displayed on that page completely and in alphabetical order.

Using categories

Categories may be used to organize the content of a wiki. Another use is the generation of a hierarchical catalogue which facilitates systematical navigation within a wiki. In addition, in smaller wikis, one could collect a list of highly frequented pages via these category pages (see also Chapter 4.5). The work of several user groups in a wiki can also be organized using categories: Here, each group is allocated a category. As you can see, the category names in MediaWiki offer a variety of application opportunities that stretch far beyond a normal categorization system. But please note that categories are not self-contained; an overlap of names can thus lead to problems.

Categories can help the members of our tribal initiative to maintain a record of all the material gathered during the course of their campaigns, so they can be found more easily. The campaign "Get Out of the Cave!", which had been conducted years ago, has left a few traces in our wiki: minutes of meetings (in *Minutes:*) concerning this campaign, an information booth (in *InfoBooth:*), and the campaigns of neighboring tribes (in *Campaigns:*). These pages are called:

Minutes:Get Out of the Cave
InfoBooth:Get Out of the Cave
Campaigns: Get Out of the Cave

Even though the corresponding pages are located in different namespaces, they were all allocated to the category "Get Out of the Cave" and can be found with a simple click of the mouse.

6.3 Multi-Page Design I - Images

After having discussed how to link pages, let us now address how to integrate external components into a page. An application that should be especially familiar to HTML designers is the displaying of images. Yet that procedure also enables the integration of original texts from other wiki articles.

Internal storage of images

Images can either be saved internally or linked to an external source. Both methods have their advantages and disadvantages. Internal storage ensures that the file is accessible and the address does not change without being noticed. Control over internal images remains in the hands of the wiki community. They can be written over,

but the modifications are recorded and can be traced in the same way that page histories can. In addition, internally stored images have their own description and discussion page. The disadvantage to internal storage is obvious: Especially in the case of larger wikis, storage space requirements are enormous, and hard to control, since the size of the uploaded files is unknown.

External storage of images

External images, on the other hand, are in the control of their originator. If an originator does not wish his image to be used further, he can simply remove it, and the external link will lead into emptiness. A great advantage to external images is that they drastically reduce and allow better control of the storage space required for the wiki. In addition, the link, in some cases, leads to the image's original context, or at least reveals its source location. This adds to the authenticity of the image. Yet even for image files that change often, it is advantageous for disk space and maintenance reasons to link them externally. On the other hand, especially in the case of highly-frequented wikis, it may prompt the accusation that one were only trying to save on bandwidth, a valuable commodity on the Internet, by avoiding the need to send large files from one's own server.

Copyright

Especially with regard to images, we must once again specifically stress that linking external sources to your page requires the permission of the originator. While a wiki subjects uploaded files to its free license, this is not the necessarily case for external files. Of course, only those images and files may be loaded that are permissible under the source's license. This especially means that a user who uploads material must possess the respective rights. In order to avoid a great deal of trouble, just remember, when you are not sure: It is better to ask once too often than not enough! It is conventional to allow usage in exchange for a link to the source page.

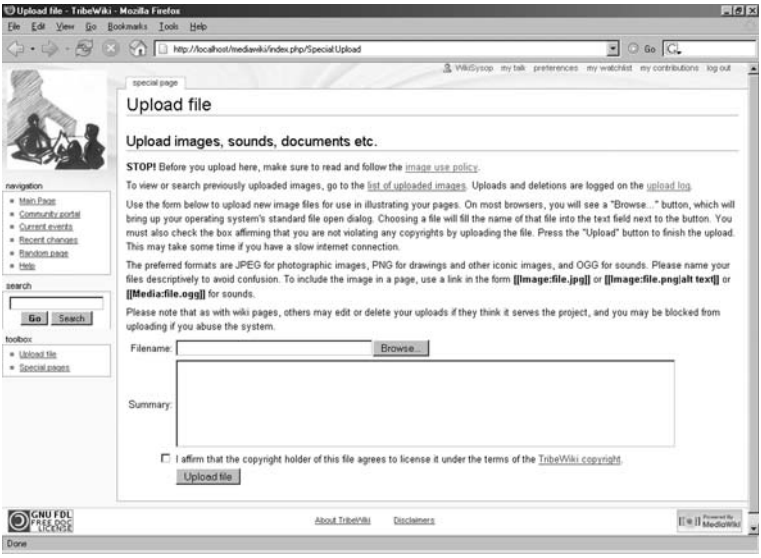
Uploading images and files

There is a special page, *Special:Upload file*, to upload images and files. You can find it either in the list of special pages or via the **Upload** link in the Tools area. Remember that you can only access the page if you are logged into the wiki.

Tip: In the standard installation of MediaWiki, uploading images as well as establishing external links are not allowed. To change this, set the variable `$wgDisableUploads` in *MediaWiki/LocalSettings.php* to `false`. For more information, see Chapter 8.1.

On the Upload file page, you will find a form:

Fig. 6.3



In the "Filename" space, enter the path and name of the file you wish to upload. Alternatively, you can click on **Browse**. A dialogue window then opens in which you can select a file. The required path information will be entered into the form. Now you should enter a short description of the image in the "Description" field. This gives potential viewers an idea of whether or not the file is of interest. Lastly, you must confirm that you are the rightful owner of this file and wish to publish it under the wiki's copyright. With the button **Upload files**, the file is then integrated into the wiki. As we have indicated, this procedure applies to all allowed file types. If a file of the same name already exists, you will see a warning page on which you can decide what to do. If you click on **Upload again**, the current action is cancelled and you will return to *Special:Upload file*. If you click on **Save file**, the file will be overwritten. However, as mentioned, the old version of the file remains in the history list. You will also be warned if the file size exceeds the limit of 100 KB. Furthermore, any spaces in file names are automatically replaced by underlines in order to conform to wiki page names.

Tip: The name of an uploaded file can no longer be altered. Thus, it is advisable to give the file a significant name.

List of all images and files

A list of all images and files in a wiki can be found under *Special:Imagelist*. In addition to a search function and a selection of displays (see Chapter 6), you will also see a list containing information about the respective files. For each entry, from right to left,

there is a link to the image page and to the file itself, information on the size, the name of the user that uploaded the file, the upload time, and finally, a description of the image. If you follow the link "(desc)", you will come to the corresponding wiki page with the image.

Fig. 6.4

All of these pages are located in the namespace *Image*. Here, you will first see the image itself with a description that you can also modify. Underneath, in the section *Image history*, there is an automatically generated list with the various versions of the image and the corresponding version functions. The section *Image links* is also automatically generated, and displays a list of the pages in which the file has been integrated.

Tip: Remember that the editable description of an image may no longer be applicable if an image has been overwritten; the description may need to be edited accordingly.

Integrating an image

Now we come to the real purpose of our exercise: integrating an image on a page. The (pseudo) namespaces come into play again here. If you only wish to supply a link to the image without actually displaying it on the page, set a link to `[[Media:imagename]]`. To create a link to the image page (with additional information), place a

colon before the namespace: `[[:Image:imagename]]`. Both versions that can be displayed on wiki pages can be formatted just like normal links (see Chapter 6.1). To display external images, just enter the URL of the file in the source text. The wiki software will recognize it and insert the original image at that spot. Note that a space must be inserted after the URL; otherwise the end of the address cannot be detected.

Wiki

```
The logo http://www.wiki-tools.de/ ↵
images/title_short.jpg is...
```

Formatting internal images

Integrated internal images, in contrast to external pictures, can be enhanced with a series of formatting commands that are entered with a vertical line separating them, as in:

Wiki

```
[[Image:gaotc1.jpg|thumb|50px|left|Cave1]]
```

Each of these entries is optional:

- **left/right/center/none.** Determines the horizontal alignment of an image. Any existing text is arranged around the image. For "none," no text is positioned near the image.
- ***size*px.** Determines the image width in pixels. The height is automatically adjusted.
- **frame.** If this parameter is indicated, the image will be displayed in its original size with a frame and caption (see below). Any size information will have no effect. Standard alignment for the image is along the right margin.
- **thumb.** The image is displayed as a thumbnail for preview purposes and includes a symbol which indicates the tooltip text "enlarge." Standard alignment for the image is also along the right margin.

The last parameter contains a description of the image. This description is displayed as tooltip text when the mouse is rolled over the image. Texts of this sort are important because they can also be seen by people with vision impairments who surf using voice output. In addition, they provide standard users with further information on the display. In both the "thumb" and the "frame" display modes, the description is added as a caption. Here, it is also possible to add basic formats (e.g. bold and italics), as well as links. However, please note that these formats are translated into HTML code in the normal image display, and thus will appear as tags in the tooltip text.

Tip: If no information is provided for a caption, the last entry in the image link will be displayed instead. Thus, we recommend always indicating the caption explicitly or inserting the pipe symbol at the end. This causes the file name to be used as the image caption.

A few examples will demonstrate the use of images:

Examples

Image gallery. First, let us generate an image documentation for our "Get Out of the Cave!" campaign that consists of a row of 50-pixel wide thumbnails positioned side by side:

Several images

Wiki

```
[[Image:gaotc1.jpg|thumb|50px|left|Cave1]]
[[Image:gaotc2.jpg|thumb|50px|left|Cave2]]
[[Image:gaotc3.jpg|thumb|50px|left|Group]]
```

The results are displayed in Fig. 6.5. It is important to indicate an alignment with the left margin. If you fail to do so, or if another alignment is selected, the thumbnails will appear one below another. If you enter more images than will fit on one line, they will automatically break to the next line. This results in a sort of table-like display, such as one would expect from a photo gallery.

Fig. 6.5

Illustration with a complex caption. Embedding an image in a text for illustrative purposes is probably the most frequent type of application. In our case, our formatted caption will also link to two other pages.

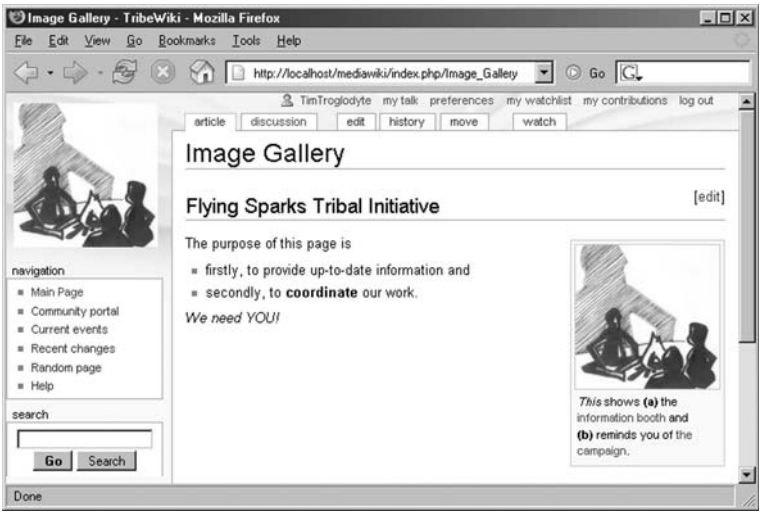
Captions

Wiki

```
==Flying Sparks Tribal Initiative==
[[Image:gaotc1.jpg|frame|right|                    ↵
''This'' shows '''(a)''' the                       ↵
[[Infobooth:Caves|information booth]] and          ↵
'''(b)''' reminds you of [[cave campaign|the       ↵
campaign]].
]]
The purpose of this page is
```

As you can see in Fig. 6.6, the image is displayed in a frame on the right edge of the page. The text flows to the left. The formatting of the caption corresponds to what we would also expect on a normal wiki page. If you roll over the image with your mouse, you will see that the formats from the caption have correctly not been applied to the rollover text.

Fig. 6.6



In continuous text

Graphics in continuous text. To display special symbols in text, e.g. lightning bolts that are not available in the set of characters, you must use images.

Wiki

```
The symbol [[Image:Internet.jpg|15px|Image     ↵
1]] indicates...
```

The corresponding text excerpt is depicted in Fig. 6.7. The key to smooth embedding in the text flow lies, on the one hand, in leaving out the alignment attribute. This signals that the image is to be displayed directly in the spot in which it is located in the source text.

Secondly, the correct size must be selected, so that the height of the image matches that of the text.

Fig. 6.7

Tip: Since the size relates to the width of the graphic, you may need to experiment a bit before finding the correct expansion factor. It is a good idea to work with the preview function instead of saving each attempt.

Aligning images

Aligning images on a page using the attributes `left` and `right` causes the text to flow around these images. There are a few opportunities to influence the positioning of the file. If you want the upper edge of the image to line up with the first line of text, place the image link before the first word in the paragraph. If you insert it after the first word, the first line will be written in full and the image will be attached to the second line. Generally, if an image is announced in a line, it is not displayed until the start of the following line. The end of the paragraph has no effect on the alignment. In fact, the entire subsequent text, including sections and tables, will be displayed next to the image. In order to make the paragraph begin underneath the image, place the following line directly before the respective paragraph:

```
<br style="clear:both">
```

Wiki

The line is then pushed far enough down until it has passed the edge of the image.

Two images at the same height

It is possible to position one graphic image on the left edge and another at the same height on the right edge. To do this, write the links for the images with the varying alignment attributes one directly after the other. The order does not matter. The text will then flow between the illustrations. Text located between the two images is joined with the normal continuous text and thus can, under certain conditions, appear before the first image.

Tipp: The use of areas surrounded by `<pre>` and images in the same paragraph can lead to both elements covering each other; this should be avoided.

Other types of files

Depending on how the wiki is set up (see Chap. 8.1), it is also possible to upload sound and other media files, as well as compressed files, in addition to images. The wiki software will always link them as files and not embed them like images. A link in the *Image* namespace thus links to the description page of the file. If the file itself is to be linked, the link to the file must be put in the pseudo namespace *Media* .

Links to these uploaded files can be collected on one page. You may remember that we set up a Material page for the tribal initiative. Here, in addition to "Links," we can also establish a "Downloads" section, through which users can access existing documents (e.g. audio recordings, letters, scanned stone tablets, etc.).

6.4 Multi-Page Design II – Templates

What are templates?

Templates refer to (source) texts that can be integrated in other pages of a wiki. This enables frequently used components, such as form letters or checklists, to be swapped out, so that they do not have to be created all over again every time. Furthermore, templates also serve to provide a certain uniformity. In addition, the use of templates offers a degree of flexibility, since modifications can be made centrally and affect all pages accessing that template. Accordingly, possible applications for templates include the administration of messages, the depiction of uniform navigation areas, or simply the homogeny of a group of pages.

Integrating templates

The namespace *Template* is reserved for templates. They are integrated into a page by setting the template name in two curly brackets:

Wiki

```
{{mytemplate}}
```

As usual, non-existent templates are displayed with an editing link. Otherwise, the source text in the template is put in place of the link. If you want to explicitly link to a template page (in order to edit it, for instance), you need to place the normal internal link in double square brackets.

Note: If a page contains an integrated template, the latter is displayed in edit mode below the edit box.

Tip: You can use any page of a wiki as a template. Just enter its complete name (with the namespace). For pages in *Main*, it will suffice to set a colon in front of the name.

However, for reasons of clarity and simplicity, you should use this function sparingly, and use the corresponding namespaces for templates conceived as such.

Possible content

Templates can contain anything that can be on a normal page, including dynamic elements such as links and variables. One possible application for such templates would be the link to the personal page of someone responsible for a particular area. Since this person could change, one could establish a template citing the respective current contact person.

Place of execution

All commands are executed as if they were directly embedded in the page. This is especially true for allocation into categories. Thus, if a template is allocated to a category, all pages using that template will also be allocated to that category. It is possible to nest templates, that is, use a template within another template. This results in an easily maintainable structure of recurring elements of a page.

Variables

Up to now, the use of templates has been relatively fixed and static. When generating a template, either it does not matter in which pages they are integrated or the author must have a relatively precise idea of what kind of pages the template will be used for. For example, for messages of a general nature, it does not matter what content is on a page. In order to offer more flexibility in relation to the article using a template, there are built-in variables. They are placeholders for certain pieces of information on the wiki, the page or the current time. Variables are integrated just like templates, and when a page is accessed, the current values are shown. A list of built-in variables can be found in Tab. 6.3. Note that the names are case-sensitive, and thus writing letters in upper and lower-case makes a difference.

Tab. 6.3

Name	Function
{{ns:*Name*}}	All namespaces have original identifiers in English that are the same in all MediaWikis. Is replaced by the name of the namespace in the language of the wiki.
{{SITENAME}}	Name of the wiki
{{SERVER}}	Sets a link to the server, that is, that area in the URL from http:// to the first slash.

{{localurl:*Page*}}	Displays everything after the slash in the URL of the corresponding page.
{{localurl:*Page*|*query*}}	As above, only with the additional indication of parameters.
{{CURRENTMONTH}}	Number of the current month
{{CURRENTMONTHNAME}}	Name of the current month in the respective wiki language.
{{CURRENTDAY}}	Number of the current day.
{{CURRENTDAYNAME}}	Name of the current day in the respective wiki language.
{{CURRENTYEAR}}	The current year
{{CURRENTTIME}}	The time in the format HH:MM
{{NUMBEROFARTICLES}}	Number of articles in the *Main* namespace that contain at least one link.
{{NAMESPACE}}	Name of the namespace of the page.
{{PAGENAME}}	Name of the page.
{{PAGENAMEE}}	Name of the page as it is displayed in the URL. Generally with underlines instead of spaces.

You can use these variables on any wiki page. For example, using

Wiki
```
{{SERVER}}{{localurl:Hello World|action=edit}}
```

it is possible to set a link to the editing view of *Hello World*, or using

Wiki
```
{{CURRENTDAY}}.{{CURRENTMONTHNAME}}
{{CURRENTYEAR}}
```

to display the current date. Note that in the first case, the entire URL of the page is generated. This is the only way to access the functions of a page. Using the link

Wiki
```
[[Hello World&action=edit]]
```

you can create a page with exactly the name indicated and no link to *Hello World*.

Substituting

If you place the key word `subst:` in front of the variable, when you save the content, the variables will be written to the source text at that spot. For instance: If you place the following line in the page *Hello World* and then save it,

```
{{subst:PAGENAME}}/{{CURRENTTIME}}
```
Wiki

you will see when editing the page again that the source text now looks like this:

```
Hello World/{{CURRENTTIME}}
```
Wiki

This mechanism is useful if you want to leave a time stamp on a page. However, it is relatively complicated, since it is a one-time action, and entering the variables is lengthier than just looking at the clock and copying down the time. On the other hand, the same formalism applies to the integration of templates. In this way, the content of a page can be copied to one's own page and then customized accordingly.

The fire initiative uses this mechanism to avoid having to fill out standard text every time the minutes are recorded:

```
== Minutes of MM.DD.YYYY ==
===Agenda===
__TOC__
===Attendant===
===TOP1===
===TOP2===
```
Wiki

On the other hand, there is a checklist for the information booth which shows what preliminary and subsequent work needs to be done. This is re-entered and customized for every information booth, and checked off in the wiki.

Integrate as source text

If you wish to integrate a template without having wiki interpret the text contained in it as source text by the wiki, just position `msgnw:` in front of the name of the template. This causes the entire template text to be enclosed in a `<nowiki>` tag and not processed.

Variables in templates

Variables often appear in templates as well. In such cases, the value of the variables of the accessed page and not those of the template are determined. Thus `{{PAGENAME}}` does not display the name of the template, but rather the name of the page in which it is integrated. This enables a series of page-specific links to be created which primarily lead to functions or special pages regarding the article. For example, the template

```
[{{SERVER}}{{localurl:Special:Whatlinkshere| ↵
target={{PAGENAME}}}} Backlinks]
```
Wiki

creates a link with the name “Backlinks” on the page that displays all incoming links (see Chapter 7.1). Note that the actual URL is generated again here.

Parameters

Especially in the case of templates that dictate layout but not content, it is advantageous to transfer values to the template. You can use your own parameters for this purpose, which can be set when the template is accessed. Transfer of values takes place within the curly brackets that surround the template, and can be done in two different ways:

Wiki

```
{{Templatename|value1|value2|...}}
{{Templatename|name1=value1|                    ↵
               name2=value2|...}}
```

In the first case, the parameters are simply numbered consecutively, and in the second, they can be addressed using a name. The parameters are integrated in the template by setting either its number or name in triple curly brackets. For example:

Wiki

```
{{Salu|Ms|Stone}}
{{Salu1|name=Stone|address=Ms}}
```

Invokes the two templates *Salu*

Wiki

```
Hello {{{1}}} {{{2}}}
```

and *Salu1*:

Wiki

```
Hello {{{address}}} {{{name}}}
```

The result is the same in both cases; the sentence “Hello Ms Stone” appears. The first version is less complicated when writing it, since you do not need to name the parameters. However, if you have a template with several values, it is simpler to give them names. In such a case, the source text of the template is easier to follow. Furthermore, the order of the entries does not matter; thus, you can be sure that the correct information will end up in the correct spot. This advantage should not be underestimated, especially in large-scale projects!

An example

In the following example, we will further clarify the possibilities offered by templates by working on the design of a uniform participants page for members of a project group of our tribal initiative. We want the page to have a uniform header with a display of current

messages, an area where participants can introduce themselves, as well as a section that can be designed freely. The participants are encouraged to make this page their personal start page, and to have a look at it at least every couple of days. The template to be used, *Template:ProjectParPage*, looks like this:

Wiki

```
{{ProjectHeader}}
{{ParInfos|
name=FIRSTNAME LASTNAME|
contact=EMAILADDRESS|
area=AREA|
image=[[Image:OWNIMAGE|150px|center]]}}
== My responsibilities are ==
* TASK1
* TASK2
== Personal area ==
You can design this area any way you wish
```

As you can see, the information that participants will later customize is written in capital letters to set a visual signal.

The template *ProjectHeader* provides the title, links that are to be on every project page, and an area for important messages that the participants should read. In order to make the layout a bit more attractive, we employed a table:

Wiki

```
{| width=100% style="border:1px solid        ↵
   #000000;background:#CCCCFF"
| style="font-size:20pt;font-weight:bold;     ↵
  text-align:center" | Fire Project
|-
| <hr>
|-
| Important Links: [[FireProject|Start]]     ↵
  {{ProjectContactPerson}}
|-
| Messages: <br>                              ↵
  '''''{{ProjectMessages}}'''''
|}
```

The cells were formatted using CSS, and they determine the table background as well as the text format and alignment. The name of the project, *FireProject*, has a permanent code, because it will not change during the course of the project. The contact person, how-

ever, is integrated in a further template, *ProjectContactPerson*, in which only the link to his page is indicated.

Wiki
```
[[User:ContactPerson|Contact Person]]
```

The messages are also obtained from a separate template. This has two advantages. Firstly, the messages are easy for users to write, because they do not have to search through lengthy code to find the right spot. On the other hand, the *ProjectHeader* template will probably only be able to be edited by administrators. Due to the swapping out of messages, any user can make entries. They are in the form of simple text displays:

Wiki
```
Don't forget: The deadline for suggestions is
Dec.31!
```

Now we have designed the project header. Next, we will turn to the uniform presentation of the participants. As you saw in the template *ProjectParPage*, all relevant information is transferred to the template *ParInfos* in the form of parameters. Ultimately, all we have to do is position the information in an attractive way. To do this, we'll once again use a table:

Wiki
```
{| align="right" style="border: 1px dotted  ↵
   #000000"
| colspan="2" | {{{image}}}
|-
| '''Name''':
| {{{name}}}
|-
| '''Contact''':
| {{{contact}}}
|-
| '''Area''':
| {{{area}}}
|}
```

Links, as in the case of `image,` as well as normal text, can be used for the parameter values. It is even possible to use blank spaces, as is done in the case of `name`. The MediaWiki correctly identifies them as a value. The same is true for links, which, although they contain the separating pipe symbol `|` , can still be correctly identified.

Generating a page with a template

Up to now, we have only generated one set of templates. If a user wants to create his own page, he must first make a copy of the prototype *ProjectParPage*. To do this, the first version of the participants page is to be saved with `{{subst:ProjectParPage}}`. Now the participant has his own copy of the page, in which the corresponding placeholder data, such as `first name last name`, just needs to be replaced. Then the uniform page is finished:

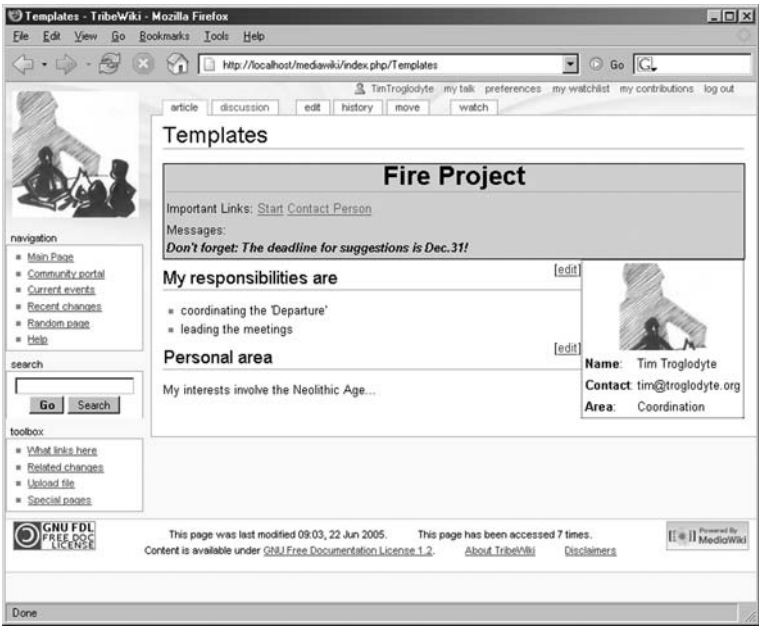


Fig. 6.8

6.5 Forwarding with Redirects

A final means of multi-page design in MediaWiki is found in redirects. As the name indicates, they serve to redirect a user from one page to another. This can come in handy when a page has "moved;" that is, when it has been copied to another page with a new title. In addition, several values indicating the same thing can be consolidated in this way. Thus, for instance, you could avoid having double or deviating descriptions of *WikiWikiWeb* and *Wiki*, by forwarding the user from one article to the other. Redirects can be installed on a page as follows:

`#REDIRECT [[`*`pagename`*`]]`

Lines below this command are automatically deleted when the page is saved. At the top of the target page, a message appears indicating that the page has been redirected. There is also a link to the original page. Thus, if you wish to cancel the redirection, you can access the old page via the link and edit it. To manually prevent a redirection, you can also add the following parameter to the URL in the address line of the browsers: `&redirect=no`.

7 Components: the Function Pages

Readers
Authors
WikiAdmins
WebAdmins

Special pages

Although many necessary tasks in a wiki can be completed with the basic functions Edit, History and Search, there are a few procedures that require more complex programming. For example, these involve statistical functions, administrative settings or the categorization of pages according to certain criteria. These functions are carried out by the special pages, which we have already mentioned. These pages are a group of entries in the namespace *Special* whose content is at least partially automatically generated and generally cannot be altered. A few of the special pages can be accessed via the navigation or tool area. A complete list of all special pages is available under *Special:Specialpages*.

Lists

These pages often list wiki articles (see Fig. 7.1):

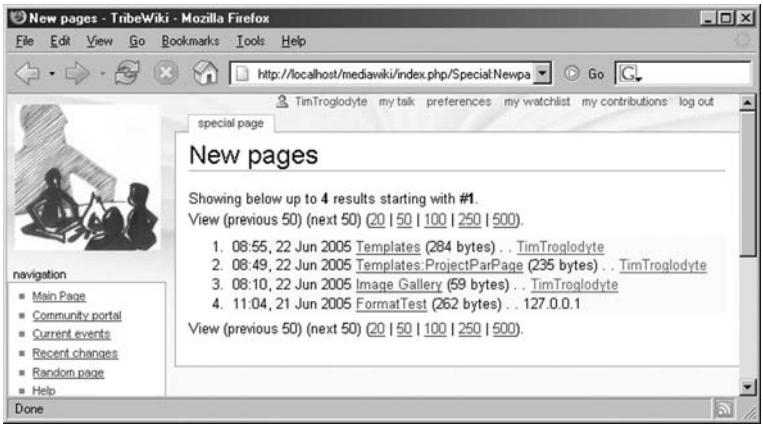


Fig. 7.1

They begin with an indication of how many results are displayed, followed by the number of the first result displayed. In the default setting, up to 50 finds are shown on a page. If the number of results exceeds 50, the navigation links “Next 50” and “Previous 50” are

activated. You can change the number of articles displayed by clicking on the corresponding number after the navigation link.

Parameter transfer

Information that a function page requires to "know" exactly what it should do, for instance, to what page it refers, is transferred by parameters in the URL. For the page *What links here*, it looks like this:

Wiki

```
http://www.wikitools.org/MediaWiki/index.php ↵
?title=Special:Whatlinkshere                 ↵
&target=Main_Page
```

The main page, the target of the backlink page, is transferred with the command `&target=Main_Page` at the end of the URL. If there is more than one parameter, they are attached and separated from each other by an ampersand (`&`). One result of parameterization via the URL is that one can no longer simply use an internal link to refer to that page, since, using that formalism, no transfer of parameters is possible. The solution lies in the use of real URLs. They can either be completely manually entered or "constructed" with the aid of variables. In the following example, two values are transferred:

Wiki

```
[{{SERVER}}{{localurl:Special:Shortpages|    ↵
limit=5&offset=3}} Short Pages]
```

As you can see, the parameters are indicated after the pipe symbol. If more than one value is to be transferred, the individual entries must be separated by a `&`. You can find frequently-used parameters in Tab. 7.1.

Tab. 7.1

Name	Function
`target`	Page to which the function page relates
`limit`	Number of results displayed per page
`offset`	Number of the first element shown on the result list

Now let's have a closer look at the individual special pages.

7.1 Navigation

Most of the function pages search the wiki for certain criteria and provide a list of articles as a result. They are designed to improve the navigation, availability and balance of the wiki. Some are considered to be so useful that they have been directly integrated into the navigation area.

Recent changes

The *Recent changes* page lists the subpages of the wiki that have been recently modified (see Fig. 7.2). The most recent changes are at the top. In addition to the display function for lists we have mentioned, there are also two further ordering criteria. You can obtain a list of the changes made in recent days, and a list of changes since the last access to that page. The latter link can be found in the line "Show new changes starting from." The following date corresponds to the server time at the point when the page was last accessed, and is updated every time. Note that the server time does not necessarily correspond to the time on your computer; for instance, the server might be located in another time zone. The advantage of this function is that you only see those changes that you have not yet viewed. Also, you can see at a glance if no edits have been performed, since the hit list would then be empty.

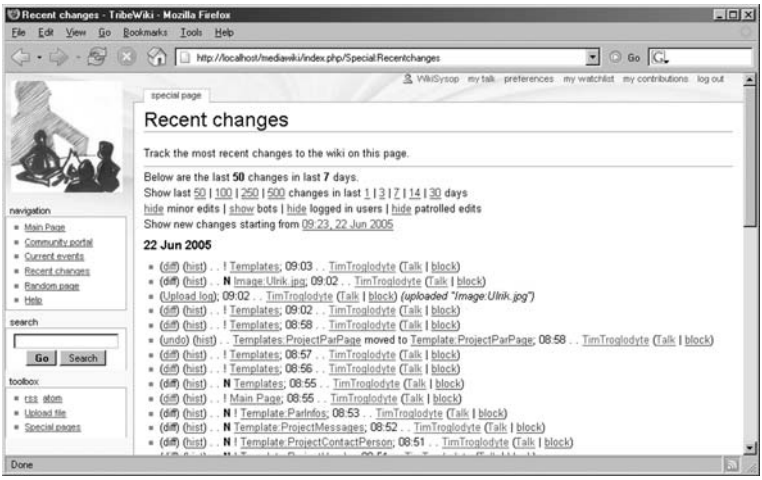


Fig. 7.2

Result lines

The individual result lines consist of several links. *Diff* indicates a view of the differences between the latest version of a page and its predecessor. This corresponds to the last change. *Hist* provides a list of versions of a page. Next, we see information on how the page was edited. If only two dots are shown, normal editing took place. If there is an N, the page was newly created. You will see a red exclamation point in some lines. This is part of a control system, and indicates whether the edit has already been viewed by anyone (see below). The next piece of information is the title of the page that was modified, followed by the point in time in which it was edited. This is followed by the name or IP address, including the talk page, of the user who made the modifications. Last but not least, any comments are shown that were made regarding the edit.

Settings

You can determine whether minor edits are displayed via the user settings (see Chapter 7.3). You also have the opportunity to activate the enhanced display of *Recent changes*. This primarily summarizes several changes by a user on the same day. In such a case, an arrow appears at the beginning of the line. If you click it, the individual edits will unfold below.

Use of the page

The special page *Recent changes* allow you to be at the pulse of the wiki, so to speak. On one hand, you find out what the current hot topics are that are being fervently worked on, since they will have several edits. On the other hand, using that page, you can check new entries to its content. This makes *Recent changes* an indispensable tool for those who have taken on the role of a maintainer in a wiki.

Control system

If several participants take on quality assurance tasks, they have to coordinate their work with each other so that they do not all end up checking the same page. There is a control system for this purpose. Pages that have not yet been checked are marked in the Recent Changes area with a red exclamation point. If a user follows the "Diff" link, he or she will find a further message in the difference display, "Mark this article as patrolled." When this button is clicked, the exclamation point disappears in the *Recent changes* display, and everyone can see that the modifications have already been checked.

RSS feed

The versions page offers the possibility of having the content read out via newsfeed. This is done with special programs that regularly retrieve and process the content of the feed. In this way, one can keep abreast of changes to the content. In the current version, the formats RSS and Atom are available. The corresponding links, if available, are in the toolbox.

Random page

The special page *Random page* does exactly what it says. It displays a random article from the namespace *Main*. This is especially useful in a wiki having an encyclopedic character, because it offers an introduction platform from which to "browse the wiki." This page once again stresses the deliberately playful nature of a wiki.

What links here?

The option *What links here* harbors a very simple, yet just as powerful function. It lists all pages containing links to the current page (backlinks). In principle, this switches the direction of the link; one can now follow it back the other way. If you want to view links on another page than the one you are on, you have to change the value indicated in the URL after `&target` to the respective page (see the beginning of this chapter).

Use of the page

There is an abundance of applications that make this function particularly interesting. For instance, it is possible to obtain an idea of the context in which a page is integrated. When you know what topics link to a specific text, you can draw conclusions as to the direction and perhaps even the quality of the article. Furthermore, you

have the opportunity to see to what degree and in what areas an article is integrated in the wiki. If deficits become apparent, remedies may be able to be found.

Backlinks for page organization

When employed appropriately, backlinks are a good source of information. Backlinks to a user page list all pages in which the corresponding user has made signed contributions. Since this is particularly common practice for discussions, one can find discussion contributions made by a participant. Templates issuing certain messages function in a similar manner. For example, if you have a template that identifies a page as "urgently requiring editing," you can use backlinks to find all articles carrying this predicate. Similarly, you can establish your own category system using this mechanism, with which you can set a link on the category page for every article belonging to a certain main category. The pages of this group can then be discovered via the *What links here* function. Of course, this is superfluous in MediaWiki, because categorization is possible. However, there are several wikis that have no categorizing function. Yet backlinks are a component of just about every piece of wiki software.

Recent changes linked

The page *Special:Recentchangeslinked* displays changes made in the last seven days to pages that have been linked by a particular page. The use of this special page corresponds to that of *Recent changes*.

Recent changes linked displays pages that have been linked from a single page. If you want to be absolutely certain that the list is complete, you should make sure the watch period you select is long enough. One advantage is that the integrated templates are displayed and linked. They can be found and edited more easily than if you first switch to edit mode and then painstakingly search for places that integrate the templates.

Use as a watch page

Establishing a watch page is very similar. You create a page that links to all articles whose development you wish to follow. Via Recent changes linked, you can now obtain a list of all edits to those particular pages.

Upload file

If you are logged in, you will find a further special page in the navigation area: *Upload file*. This was discussed in detail in Chapter 6.3.

Navigation pages for readers

Under the link *SpecialPages*, you will find further navigation functions in the namespace *Main*. They can be divided into two groups. The first serves to make interested readers aware of certain articles or facilitate their access to them. This group includes:

- ***All pages***. This contains a complete list in table form of all pages in the main namespace of the wiki. Here, you can enter a

combination of letters from which point the all articles starting with those letters will be displayed. This is very useful for large wikis. In addition, you can determine from what namespace the articles are to be shown so you can find templates easier, for example.

- ***Favorite pages***. The articles are ordered according to frequency of viewing. The number of hits is put in brackets.
- ***New pages***. The pages are listed in the order of their creation. Caution: This refers to the first edit. Subsequent edits are ignored here.
- ***Oldest pages***. The longer a page has not been edited, the farther up it will be on this list. Caution: This refers to the last edit, not the creation of the page.
- ***Categories***. A list of all category pages. The ones with the most elements are shown on top.

Navigation pages for authors

The other group of navigation pages represents aids for authors so they can see which articles still need work. These pages include:

- ***Wanted pages***. Displays all pages to which links have been set but which have not yet been created. The link in brackets (backlink function!) shows the pages on which these links appear.
- ***Orphaned pages***. All pages that are no longer linked by any other page are shown here. They are no longer accessible through normal browsing with mouse clicks.
- ***Unused images***. Shows all images and files that were uploaded but not integrated into any page. Since files take up a great deal of disk space, these should be monitored and, if not used for an extended period of time, perhaps deleted.
- ***Uncategorized Pages***. Here are all the pages that have not been allocated to a category.
- ***Dead-end pages.*** These are pages from which no links lead to any other page.
- ***Short pages***. The pages are organized according to size in Bytes. The shortest pages are listed at the top.
- ***Long pages***. This list shows the longest pages first.
- ***Image list***. Shows all images and files according to name, size or date as desired, and thus offers an overview of the content of the wiki that does not originate from articles.

A special feature of MediaWiki is the page *Book sources*. We mention it for purposes of completeness. Here, you can enter the ISBN number of a book to automatically generate links to a few online bookstores where you can order the book.

7.2 Search Function

In the profusion of various pages and their potentially relatively unstructured order, a search function is essential. The MediaWiki search function is designed such that, if you know the name of a page, you can jump directly to it by entering it in the search field.

Go and Search

You will find the search function in its own box under Navigation. As you can see, there are two buttons, **Go** and **Search**. The only difference between them is that **Go** will take you right to the corresponding page without a detour, and **Search** brings you to a search results page with a list of all results. **Go** thus offers the aforesaid direct access. Confirming your entry with <ENTER> has the same result as **Go**.

Search display

If, however, the page does not exist, or if you have pressed **Search**, the special page *Special:Search* opens. Similar results are not part of the search. You will primarily find two types of result lists. The top, "matching titles," shows you what page titles match your search query. If the term appears within the limits of how many lines are displayed on a page, according to preference settings (see Chapter 7.3), it will be displayed along with the line number and its context under the page name. The second list, "matching text," searches the content of the pages for the desired word. Here, too, their sections are also displayed.

Settings

If your search has not brought the desired results, it may be due to the fact that not all namespaces are taken into account. On the lower part of the results page, you can set which areas are to be searched. There is also a search field here in which the word you have entered is displayed. You can thus conveniently repeat and, if necessary, modify your search.

Possible search items

You can use regular words as search items. The search is not case-sensitive. Only entire words are sought, whereby a dash also counts as a word separator. You cannot search for every word. A query for "the" is virtually useless, since this word appears on practically every page. There is a list of such common words, so-called stop words, that are excluded from the search. Wildcards, as you know them from the Internet search engines, are not permitted. If you enter more than one word, the order is irrelevant; the individual

elements are automatically joined by "and." If the order is relevant, put the phrase in quotation marks.

Boolean searches

Boolean searches with "and," "or," and "not" are taken into account in the search. This means that, for instance, it can indicate a list of alternative words if you join the words with "or." In that case, all pages are sought in which either of the two words appear. If you wish to search for pages containing both words, you can join them with "and;" however, that is the default setting. If you want to exclude a certain word from your search, precede it with "not." To determine the order of the joined words, use brackets. For example, in a search for


```
Wiki and (Tool or Tools)
```

we will get "wiki tool" and "wiki tools," but not "wiki wiki." Using these criteria, you can formulate very differentiated searches.

7.3 User Administration

Why register users?

Although one can actually take care of all important work in a wiki on an anonymous basis, most wiki systems offer user administration. There are several advantages to such a system. *Firstly*, it enables a user to make personal settings that will always apply to that person. *Secondly*, the name of the user is added to the version history for every edit, so that other participants can have an idea of who represents what opinion. This, in turn, offers an individual recognition and responsibility, and promotes a certain cohesion in the community. *Thirdly*, there are wikis which, precisely for that reason, make a registration with a user name a requirement for participating in the wiki. The exclusion of anonymous contributions is not possible without a user administration. Note that the real identity of a participant need not be made public, since user names can also be pseudonyms. *Fourthly*, a few wikis offer the possibility of adjusting access rights depending on a respective page and various tasks. However, MediaWiki has only very limited functions in this regard.

Registration

The registration process was described in Chapter 3.5 and can be found under the link "Create an account or log in" in the upper area of the wiki. This registration or login is then active until the browser is closed or the user explicitly logs out (also in the upper area). You will only need to register once if you work at a specific computer that is set to accept cookies if you have checked the option "Remember my password". Do this only if you are the only person us-

ing that computer; otherwise, someone else could log in under your name and submit articles to the wiki. To cancel the permanent login option, simply log out explicitly using the "Log out" link.

Permissions System

The MediaWiki permissions system is relatively straightforward. If you give yourself a user name, you only have "user" permissions. You can now upload files and move pages. Yet there are two higher levels with more rights. These are "sysop" and "bureaucrat." Here, you can protect and delete pages before editing. Furthermore, you have access to a few special pages that deal with the maintenance of the wiki (see Chapter 7.5). The only difference between the two is that only bureaucrats can make other users sysops. Of course, the "WikiSysop" account that we have generated during the installation of the wiki retains bureaucrat rights.

Make a user an administrator

The page *Make a uner into a sysop* is used to grant other users administrator privileges. Here, just enter the respective user name, select whether he or she is to receive "bureaucrat" privileges like your own, and click on the confirm button. That's it. If anonymous users are not allowed to create a new account, you'll have to also arrange this as an administrator (see Chapter 8.1).

User toolbar

Once you have logged in, you will see a series of further links in the top bar that enable you to access your personal pages. There is of course your user page. On it, you can add information about yourself, for example. As a complement to this page, you'll also find a Talk page that serves as a means of communication with other users. In step with the spirit of simplicity, both pages are nothing more than normal wiki pages that are used for a specified purpose. However, this does not prevent you from placing articles on your Talk page or conducting discussions on your user page.

User page

Every registered user has his or her own user page. On it, you'll find two new entries in the toolbox on the lower left. "User contributions" lists the edits the user has performed in the wiki. Should it no longer be up to date, you will see a rollback link with which you can restore the last version of the user. For your own contributions, you will find the link "my contributions" in your user toolbar.

Watchlist

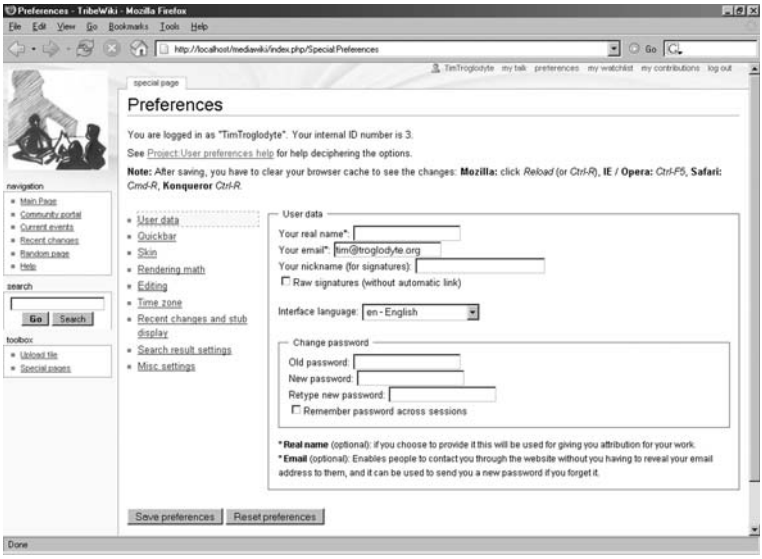
If you have placed one or more pages in your *watchlist* via the *watch* tab, you can monitor recent changes to that page. You will receive lists on a daily basis. Per line, you will find the name of the watched page, a link to the "differences" from the last version, as well as a list of the complete *history*, the participant that made the modifications and, if you have sysop rights, the opportunity to block that participant (see Chapter 7.5). Thus, you have all important administrative functions for a particular page at your fingertips. You could place your own page on your watchlist, in order to observe the reaction of the rest of the wiki community to it. If the list becomes

too long, you can delete several entries from the list simultaneously by clicking on "show and edit complete list." Now you can remove pages by checking them and clicking on the button **Remove checked items from watchlist**.

Preferences

In the user toolbar, you will find the tab "preferences." It will take you to a display in which you can select a number of settings regarding how the wiki should look and behave when you are logged in (see Fig. 7.3). We will discuss the various options in order of appearance below. Once you have indicated all of your preferences, click on **Save preferences** to apply them. Note that old page data may still be in the browser cache and may need to be emptied before you can see the new settings. If you wish to apply the default settings, click on **Reset preferences**.

Fig. 7.3



User data

Under "User data," you can indicate preferences regarding your real name and email address. You can also give yourself a nickname that will be used when you employ the signature abbreviation ~~~~. You can also change your password here as well.

Quickbar settings

The item *Quickbar settings* is only relevant in the skins *Standard* and *CologneBlue*, and determines the location of the navigation area. You can set the desired display design under the item *Skins*.

Rendering math

"Rendering math" is interesting for those who frequently deal with mathematical formulas. Here, you have the option of displaying the formulas as images, HTML forms or text. The ideal setting depends on the browser used and the bandwidth of your Internet connection.

A few interesting options can be found under "Editing." You can set the height and width of the edit box. This is useful if you are working with a very low or very high screen resolution. Further settings include:

Edit box size

- ***Edit box has full width***. This sets the edit box to the maximum size.
- ***Show edit toolbar***. Only works if your browser can display JavaScript.
- ***Show preview before edit box and not after it***. If this option is not selected, the preview will be displayed beneath the edit box.
- ***Add pages you edit to your watchlist.*** Pages edited by you are automatically placed on your watchlist.
- ***Mark all edits minor by default.*** Useful if, for instance, you check your wiki for spelling errors.

"Time zone" enables you to "inform" your wiki of any difference in the time on your server and your local time. All times indicated would then be converted to your time zone.

Recent changes and stubs

You can determine the amount of pages listed under "Recent changes and stub display" under the item of the same name. In addition, you can suppress the display of minor edits. If your browser is JavaScript enabled, you can switch on the enhanced display for *recent changes* (see above). Finally, this section allows you to set a threshold for the number of words under which an article is marked as a short article (stub). Such articles are then displayed in another color and are considered as requiring editing.

Search results

The item "Search result settings" enables you to determine how many results are displayed per page. "Lines to show per hit" indicates how many lines of context of an article are to be displayed. If you set this number very high, the environment will always be shown with the search item. "Characters of context per line" refers to the length of this context display. Furthermore, you can also indicate in which namespaces a search should take place by default.

Miscellaneous settings

"Miscellaneous settings" offers a range of options:

- ***Show hoverbox over wiki links***. Determines whether the URL or internal destination of a link should be displayed as a tooltip in the hoverbox.
- ***Underline links***. This makes it easier to recognize links.
- ***Format broken links***. If you deactivate this option, links to pages that do not yet exist are not highlighted but rather appear

with a question mark that serves as a creation link. Short pages receive an exclamation point after the respective link.

- ***Justify paragraphs***. Justified paragraphs are generally more difficult to read and are thus not set by default.
- ***Auto-number headings***.
- ***Edit pages on double click (JavaScript)***. Using this option, you can switch to edit mode by double clicking on the text. This can greatly facilitate your work if you have to make several changes.
- ***Enable section editing via [edit] links.***. If you deactivate this option, an article can only be edited as a whole.
- ***Enable section editing by right clicking on section titles (JavaScript)***. This option is independent of the above link display. If activated, you can edit the section by right clicking on the respective section title.
- ***Show table of contents (for pages with more than 3 headings).***
- ***Disable page caching***. Do this if you often receive older versions of a page on your display. The disadvantage is that you will have to put up with longer loading times.

Those are all of the settings with which you can personalize your wiki.

7.4 Information about the Wiki

Statistics

A further group of special pages provides information on the wiki software and the state of the wiki itself. Let us first look at the *Statistics* page. In the first section, it shows information regarding the number of pages in the database, the number of pages that are probably legitimate content pages (not including, for instance, talk and special pages), and the number of hits and edits. In the second section, you will find the number of users and administrators. This page actually has a purely descriptive function, serving to measure the size of the wiki and how often it is frequented.

Version

On the *Version* special page, you can view general data on the software used, information regarding copyright and originators of the wiki software, and links to websites of the programs used. Please note that MediaWiki software is always subject to GPL laws.

User list, Admins list

Two pages provide information on the users of the wiki. The page User list displays a list of all registered users and their permissions. Users who have only registered with an IP address and may have established a user page are not included in this list. On the page *Admins list*, all users with administrator rights are listed (for more on user rights, see Chapter 7.3).

List of blocked IP addresses

Time after time, there will be users who do not observe the stipulated rules. To stop such users, administrators can block their IP addresses and thus prevent them from editing pages for a period of time. During that period, the individual's IP address is listed in the *List of blocked IP addresses and usernames*, along with the name of the administrator who blocked that address and the reason. If you are logged in as an administrator, you can also lift a block early. This page, of course, is also interesting if you yourself are among the blocked users. Firstly, you will discover why you have been blocked and secondly, you can contact the corresponding administrator to clear up any misunderstandings.

7.5 Maintenance

Maintenance page

A number of pages aids in administering and maintaining the wiki and its community. First and foremost is the *Maintenance page*. Using it, errors in the structure and content of a wiki can be detected. When you access the page, you will find an overview of its possible functions. Unfortunately, the first function is the "Disambiguations pages," which is not fully functional in our version of MediaWiki. However, it serves to list all pages employed to separate terms that are the written the same way but mean different things.

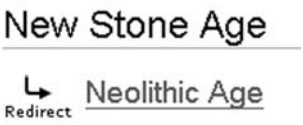
Double redirects

The second function, "Double redirects," searches for all redirects that in turn lead to another redirect. This is unnecessary, since the original redirect can lead right to the target page. One example: Page A links to Page B, which leads to Page C. In practice, automatic redirects only work once. That means that a reader accessing A ends up at B and cannot be redirected automatically. Thus, it is a good idea to eliminate the intermediate step from A to B and redirect from A to C. In the listing of double redirects, you will see the original Page A, Page B to which it links, and in brackets, Page C, so you will know the target. Behind A is a link to the corresponding edit mode, so you can make the correction immediately.

Broken redirects

The link "Broken redirects" brings you to redirects that link to non-existent pages. Since these redirects do not work (see Fig. 7.4), they should be removed or a target generated.

Fig. 7.4



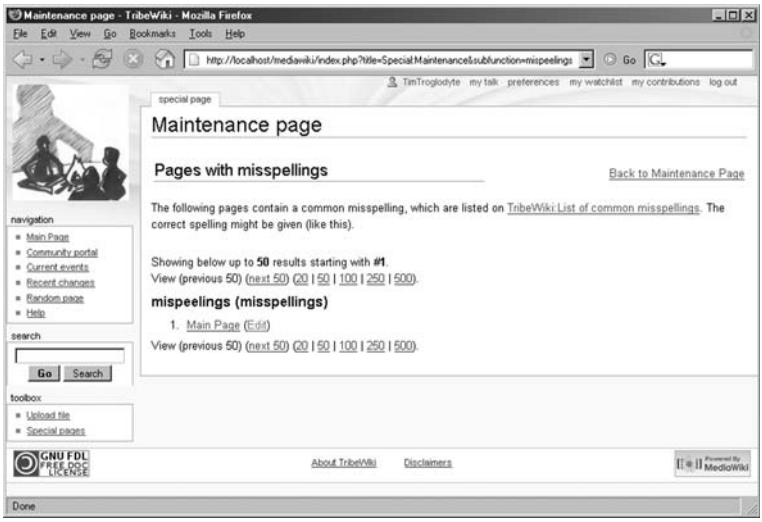
Pages with misspellings

An interesting method of standardizing spelling can be found in the search for "Pages with misspellings." Of course, MediaWiki has no built-in spell checker. Instead, the search is aimed at entries on a page *[Wiki-Name]:List of common misspellings*. The list of errors on this page simply consists of a nonnumbered list of spelling mistakes. Generally, the correct spelling is indicated in parentheses:

Wiki

```
* mispeelings (misspellings)
```

Fig. 7.5



Other text that is not part of the list is ignored. Thus, you can put the list in alphabetical order, for instance, and subdivide it in sections according to the respective letters, or make comments on the misspelled words. The first word of the listing is the misspelled word. However, the error message on the function page shows the complete line of the listing and thus also information on the correct spelling (see Fig. 7.5). Directly underneath, the pages are listed that contain the misspelled word.

Find missing language links

For international wikis, it may be desirable to see to what other languages a page has been translated. If your wiki is to be bilingual or multilingual, such parallel pages should be even indicated systematically. In Wikipedia, this is done via language links characterized by a language prefix in front of the actual page name, as in:

```
[[en:Main Page]]
```

for the English version of the main page. The function "Find missing language links" lists all pages that do not have a link to one of the indicated languages. The language links are generally interwiki links (see Chapter 6.2). However, the corresponding abbreviations do not exist in the standard installation and, if desired, must be done manually (see Chapter 8.2).

Export pages

That rounds off the functions of the Maintenance page. However, there several further handy special pages. For instance, the page "Export pages" can generate an XML file containing the information on the indicated pages. To do this, you only need to enter the title of the page that you wish to export in the text box on that special page. Each title must be in its own line; otherwise it will be treated as one long title consisting of several words. If you deactivate the field "Include only the current version, not the full history," all changes will be issued. Caution! This will create very large files. After pressing the button **Submit Query**, the wiki returns an XML file. You can either view it in the browser (not very useful, since you can see the same information in formatted form on the actual page), or save it for further processing as an .xml file. Note: Page decoration and formatting is not transferred. The information is primarily intended for the exchange of pages between wikis or MediaWikis.

Wikis and XML

On the whole, the connection of wikis and XML has been treated as a relatively minor topic. With the rising numbers of users and especially wiki systems, and the associated need for swapping between wikis, this is certain to change in the future. After all, there are already efforts underway to create an XML description of the entire MediaWiki formatting and page information under the heading Wikipedia DTD.

All system messages

A list showing "All system messages" might at first seem to be of little use. Here, you will find a three-column table with all system messages and inscriptions on the right. On the left, there is a link to a page in the MediaWiki namespace. That is an indication of why this area exists. Each message of the wiki is recorded on one of these pages and can be edited by administrators. Modified messages are highlighted in red. In the middle column, the standard values are shown, such that, in the event of an unjustified modification, they

can be easily accessed again. In a few messages, you will notice that variables have been used. These are identified by $ and a number, and are filled directly by the software. The content is usually a page name or number. This should be evident from the context.

Logs

The page *Logs* provides you with the opportunity to follow all changes in the MediaWiki. This refers to uploaded files, deletions and page and user blocks. They can either be shown as a complete list or filtered according to user or page title.

Block IP addresses

Last but not least, there are two more pages that can only be accessed by users with sysop status. As mentioned earlier, there is the possibility of blocking disagreeable users from editing texts in the wiki for a period of time. This is done via the page *Block user*. Here, you can enter an IP number or a user name.. In addition, it is necessary to enter a block period. This consists of a number and indication of the time period or the word "indefinite." Possible periods include "seconds," "hour," "days," "months," and "years." If no deadline is entered, the address will not be blocked. Blocking users should only be a last resort to fight vandalism, and should be reasonably justified.

Undelete pages

Administrators can delete pages. Such pages, however, are only marked as deleted in the database, but they otherwise remain preserved. Thus, in case a page has been deleted by mistake or wrongfully, it can be restored. The special page *Restore deleted pages* serves this purpose. It provides a list of all articles that have been marked as deleted. Simply select the corresponding file by clicking it. A page is then displayed with the newest version of the page to be restored. All older versions are linked at the bottom of the page. However, when you click on **Restore!**, all versions are always restored.

8 Administration

Readers
Authors
WikiAdmins
WebAdmins

Administration of a MediaWiki, as we have seen, can in part be carried out via the browser. In addition, however, there are a few tasks that can only be performed with access to the server and the files located on it. The following will delve somewhat deeper into the processes and settings significant to the operation of the wiki.

8.1 File structure and *LocalSettings.php*

To modify the MediaWiki on a file level, it is a good idea to first obtain a rough overview of the functions and corresponding directories, as well as the most important files that determine the performance of the wiki.

Directory structure

The subdirectories of MediaWiki directories and their functions are listed in Tab. 8.1:

Tab. 8.1

Name	Function
config	This is the location of the installation script that was explained in Chapter 2.2. In addition, the file *LocalSettings.php* is generated in this directory during installation.
docs	Documentation offering administrators an introduction.
extensions	Contains extensions of the MediaWiki. Primarily intended for future plugins, which are listed here.
images	Images and files that are uploaded to the wiki are stored here.
includes	The actual core of the MediaWiki. Here, you will find all files concerning the editing and structure of wiki pages.

irc	Connection to Wikimedia chat. Not relevant to the wiki.
languages	Translation of wiki messages to the corresponding language. Is read during installation and used for the generation of the message pages.
maintenance	Tools for administrators primarily for maintenance, updating and exporting of the database.
math	Files required for the integration of mathematical formulas.
skins	CSS files and images needed for the respective skins. Also the templates for the structure of MediaWiki pages.

Parameters in LocalSettings

Many of the basic settings you can perform in MediaWiki are made via entries in the file *LocalSettings.php*, which is located in the main directory. This is actually a PHP script that determines a few fundamental variables. If you are not a PHP programmer, this should be no cause for worry, since the settings are done using a simple pattern:


```
$parameter = "value";
```

Lines beginning with a # are marked as commentary and thus have no effect on the software. The value of the parameters are determined during the installation of the wiki and adapted to the conditions of your system. In the following, we will discuss the interesting settings that you can perform more or less without risk. You should only alter the other values if you have experience in dealing with PHP and dynamic websites. In any case, to be safe, you should make a copy of *LocalSettings.php* that you can restore in an emergency. The file can be edited with any editor; however, syntax highlighting is advantageous, because the variables and values can be rendered obvious.

Tip: If during installation you have copied *LocalSettings.php* into the main directory and not moved it, the directory *mediawiki/config* should still contain the original version of the file.

Contact information

During installation of the MediaWiki, you determined the name and contact email address. This information can be found in *LocalSettings.php* as parameters. The name of the wiki is set in `$wgSitename`. We do not advise subsequently changing the name, because several wiki pages use the set name, especially those in the respective namespace. On the other hand, it is possible that the

administrator's email address my change. It is stored in `$wgEmergencyContact` as well as `$wgPasswordSender`. As you can see, both addresses can be different. The first contains that person who receives an email when the wiki software detects a problem. The second indicates from which address passwords are sent. This should be an address to which users can send a reply in the event of questions.

Adapt interface

The look of the wiki is partially determined by parameters in *LocalSettings.php*. The path containing the image that you see in the upper left corner of the MediaWiki is set in `$wgLogo`. Note that the path must conform to the operating system of the server. Under Linux, slashes are used to separate directories. Entries here are case sensitive. Under Windows systems, case does not matter, and directories are separated by a backslash. The file indicated should be an image file that does not exceed 135 x 135 pixels. Larger graphics are cut off. The general look of the wiki is determined by skins. The default skin for users not logged in is *MonoBook*. This is set in the parameter `$wgDefaultSkin`. Here, you can enter one of the skins listed. If you misspell it, the default skin will be set. To use your own skins, see below.

Designing the URL

Depending on the configuration of your system, it is possible to design the URL of a wiki page in a different manner. Normally, an article is accessed via a parameter in the file *index.php*:

```
http://WikiPath/index.php?title=PageTitle
```

URL

The second variation is shorter and nicer. Here, only path and title can be seen.

```
http://WikiPath/index.php/PageTitle
```

URL

During installation, the script checks your configuration and sets the variable `$wgArticlePath` accordingly, i.e. if possible, the shorter variation is used:

```
$wgArticlePath = "$wgScript/$1";
# $wgArticlePath = "$wgScript?title=$1";
```

CFG

`$wgScript` is the path, and `$1` the placeholder for the page title. If you prefer the long version of the URL, uncomment the second line by deleting the # and placing a # at the beginning of the first line.

Uploading images and files

Uploading images and files is not possible in the standard installation; thus it must be explicitly allowed. To do this, set the parameter `$wgDisableUploads` to `false`, by removing the hash at the

beginning of the line. Note that the value `false` is not contained in quotation marks. Under Linux, you must make sure that the directory *images* is writable. You can also determine what types of files will be accepted for uploading. They are set in the variable `$wgFileExtensions`, which is normally not set in *LocalSettings.php*, but rather in *includes/DefaultSettings.php*. There, it contains the following standard entries:

CFG

```
$wgFileExtensions = array( 'png', 'gif',     ↵
'jpg', 'jpeg', 'ogg');
```

DefaultSettings.php is not modified! If you want to add further file types, you must insert the above line in *LocalSettings.php*. Then you can expand the list however you like according to that pattern; that is, separated by commas and enclosed in single quotation marks. Note that a few file types harbor risks, especially if they are or contain executable programs. Thus, the following types are generally excluded from uploading:

```
html, htm, php, phtml, php3, php4, phps,
shtml, jhtml, pl, py, cgi, exe, scr, dll,
msi, vbs, bat, com, pif, cmd, vxd, cpl
```

If it should be necessary to upload these types of files, you can permit the uploading of .zip archives, pack the files in question into such an archive, and then upload them to the wiki.

Image sizes

Using `$wgUseImageResize=true;` you can enable the modification of image sizes. The option is switched off during installation, but it is a very good idea to activate it, especially to be able to generate thumbnails. If you have installed ImageMagick, you can use it to modify image sizes. The parameter is `$wgUseImageMagick=true`. The option of integrating LaTeX formulas in a page can be activated using `$wgUseTeX=true`.

Database access

Settings for access to the database can be found in the parameters `$wgDBserver`, `$wgDBname`, `$wgDBuser` and `$wgDBpassword`. You have already entered these during installation.

Your own namespaces

In addition to the namespaces already configured in the installation, you can add more. To do so, you first need a further variable in *LocalSettings.php* with the name `$wgExtraNamespaces`. All namespaces are numbered, and your own begin at 100. Even numbers indicate normal areas, uneven numbers denote discussion namespaces:

```
$wgExtraNamespaces =
  array(100 => "Flying Sparks",
        101 => "Flying Sparks_Discussion" );
```

CFG

You must now determine the behavior of the areas by setting two more variables. In the file *includes/DefaultSettings.php* you will find the entries to the variables `$wgNamespacesWithSubpages` and `$wgNamespacesToBeSearchedDefault`**.** Copy both to *LocalSettings.php* and expand the list with the numbers of the new namespaces. The first variable determines that new pages may be generated in the namespace, while the second determines whether the area is to be automatically included in searches:

```
$wgNamespacesWithSubpages = array(
  -1 => 0,
  0 => 0,
  ...
  100 => 1,
  101 => 1 );
$wgNamespacesToBeSearchedDefault = array(
  -1 => 0,
  0 => 1,
  ...
  100 => 1,
  101 => 1 );
```

CFG

Using this procedure, you can add up to 156 of your own namespaces in your wiki.

Limiting page access

The file *LocalSettings.php* also contains information regarding who may view and modify the pages of the wiki. Granting permissions in MediaWiki is done according to the whitelist principle. This means that all pages that cannot be freely accessed are put on a list (the whitelist) and cannot be viewed or edited. If there is no whitelist, there are no limits. There is also a list for user accounts. Individuals on that list have the right to establish new user accounts. In the default setting, which is not specifically listed, every user, whether logged in or not, may perform any task. The necessary parameters are set as follows:

```
$wgWhitelistEdit = false;
$wgWhitelistRead = false;
$wgWhitelistAccount = array ( "user" => 1,
"sysop" => 1, "developer" => 1 );
```

CFG

Thus, there is no whitelist for editing and none for reading pages. That means that non-registered users are not excluded from having access. In addition, visitors can register as users. If you want to limit editing privileges only to registered users, you should activate the edit whitelist:

CFG
```
$wgWhitelistEdit = true;
$wgWhitelistRead = false;
```

Here, it is not possible to release only certain pages for general editing. On the other hand, for a read whitelist, the individual pages must be explicitly cleared for reading:

CFG
```
$wgWhitelistRead = array("Page1", "Page1");
```

This means that you will have one or more openly accessible introductory pages for your wiki, but that all others can only be accessed upon registration.

Permitting user registration

The previously discussed settings assume that users will first make themselves known to the system before reading or editing anything. This is done by logging in, or if a user does not yet have an account, registering. In principle, this means that reading and editing is still open to anyone. If you would like to prevent this (e.g. because you have a fixed team of authors), you must limit the possibility of opening new accounts to sysops.

CFG
```
$wgWhitelistAccount = array ( "user" => 0,
"sysop" => 1, "developer" => 1 );
```

If you, as an administrator, wish to register a new user, you must first log in yourself and then access the page *Special:Userlogin*. Since it is not listed among the special pages, it must be activated via a search or URL. Enter a user name and the email address of the new user and click **via email**. A random password will then be generated and sent to the user.

8.2 Database Structure

Database tables

Almost all data created in working with MediaWiki is stored in the database. It is the final resort for edits to the data if no corresponding special page is available for the desired tasks. Thus, we provide an overview of the tables in the database below (see Tab. 8.2):

Name	Function
archive	Deleted pages are located here.
brokenlinks	List of links to non-existent pages.
cur	Contains the current versions of articles.
image	Information on uploaded files.
imagelinks	List of links to images and files.
interwiki	List of known interwiki prefixes.
ipblocks	Information on currently blocked users.
links	List of all functioning links.
math	Information on mathematical formulas.
old	Contains the old versions of articles.
oldimage	Information on old versions of uploaded files and images.
recentchanges	List of changes in the wiki.
site_stats	Statistical information.
user	List of users.
watchlist	Entries on the personal watchlists of users.

Tab. 8.2

Modifications to the database aim at the heart of the wiki, and could potentially cause a great deal of damage. When you make changes here, you should have a certain degree of experience in dealing with MySQL to prevent the loss of existing data.

A user's last edit

If you have identified a user as a vandal, it may be of interest to find out what pages he or she last edited; that is, for what pages no one has yet rolled back the destruction:

```
SELECT cur_title FROM cur
WHERE cur_user_text='name'
```

SQL

Your own interwiki links

A list of the known interwiki abbreviations is entered in the table "interwiki." The elements each have three components: First the abbreviation, that is the prefix, is shown. Then the URL appears before an entry, followed by a $1, which stands for a variable for the article name. Finally, a flag must be set to indicate whether the link is local or not. In the former case, the URL of your own wiki is entered in front, in the latter case, the link is considered an independent external link. You can add new entries as follows:

```
INSERT INTO interwiki
VALUES ("prefix", "URL$1", local)
```

SQL

"Local" either has the value 0 for local or 1 for external. In an interwiki link, the prefix is then replaced by the corresponding entry and, in place of the `$1`, the link name is entered.

Empty archive

To reduce the size of the database, you can empty the archive of deleted articles. Only do this if you are certain that none of these articles is needed anymore.

SQL
```
DELETE FROM archive;
```

For some articles, their history only plays a minor role (e.g. if you generate a personal to-do list). To delete the old versions of a page, remove all data sets in the table "old" whose titles match that of the page being deleted:

SQL
```
DELETE FROM old WHERE old_title = 'title';
```

Saving data

To increase data security, it is advisable to copy and archive the content of the database from time to time. In the event of data loss, at least a portion of the articles can be restored. You can make a copy of the database by entering the following command at the command line level on the server:

Shell
```
mysqldump -u user -p db > file
```

You have already indicated *User* and *Db* (database) during installation. For *file*, you can select any name. It is a good idea to add the extension .sql, because this file is one in a series of SQL commands. You will then be queried for the database password, and after you enter it, the copy will be generated. As an alternative to the command line, you can also use a client such as phpMyAdmin.

Restoring data

To restore data, load the database copy in MySQL.

Shell
```
mysql -u user -p db < file
```

If you wish to import the page from another wiki, all you have to do is retrieve the tables *cur* and *old*. The dump command looks like this:

Shell
```
mysqldump –u user –p db cur old > file
```

The import command remains the same. After you have retrieved the new pages to the wiki, you should execute the maintenance script that refreshes the link tables. This can only be done via the

command line level. To do this, switch to the directory *mediawiki/maintenance* and enter the following:

```
php rebuildall.php
```
Shell

Execution of the script can take several hours, depending on the database size. Since a few special pages depend on the data, it is worth the effort for reasons of consistency, because otherwise only incomplete data will be generated.

8.3 Design Questions

Skins

If you wish to utilize MediaWiki as a component of your homepage, it is a good idea to adapt the design of the wiki to your own site. Even as an independent site, you may want to lend your wiki a personal touch. There are two approaches you can use to change its appearance: You can design the skin by altering the CSS files in the directory *mediawiki/skins/common*. Changes to the page structure, on the other hand, are done via the templates in the directory *mediawiki/skins*. The template system has only been in use since MediaWiki version 1.3. Template-based layout, however, is the model for future versions and considerably easier to adapt. In addition, the skin, which is intended to enable one's own settings, is based on a template model. Thus we will concentrate on template-assisted design below. It is of note, however, that the skins *Standard*, *Nostalgia* and *CologneBlue* use a different system.

Template structure

Let us first have a look at the template *MonoBook.php* in *mediawiki/skins*. The crucial area begins as of the line

```
class MonoBookTemplate extends QuickTemplate
```
PHP

The schematic construction is depicted in Tab. 8.3.

Tab. 8.3

Area	Function
`<head>`	Determines the page title and the stylesheet to be employed.
`<body>`	The entire content to be displayed is located here.
`<column-content>`	Areas in which the article is displayed.
`<bodyContent>`	Article display.
`<column-one>`	All navigation bars and toolboxes are located here.

<p-cactions>	The tabs with the various views of a page.
<p-personal>	User-specific links.
<p-logo>	Logo.
<p-nav>	Navigation menu.
<p-search>	Search function.
<p-tb>	Tools.
<p-lang>	Links to other languages, if activated.
<footer>	Content of the footer.

As you can see, we are dealing with an HTML document in which the corresponding content is inserted. The area names, except for head and body, are each classes (class=) or IDs of tags that surround an area. In the template, you can now remove elements or add your own HTML components. Of course, you should make a backup copy first.

Entry in the navigation menu

Let us assume that you do not need the entry "Wikipedia Portal" in your navigation menu; instead, you wish to have a link to the participants list "ParList" of the Fire Project. You will find the portal entry in the area <p-nav>; it begins with <ul>. As you see, the entries are generated automatically. They are taken from a list set in *DefaultSettings.php* in the variable $wgNavigationLinks. There you will find the name of wiki pages that contain a title and link target of the menu item. All of these pages can be found in the MediaWiki namespace. In our case, these include portal and portal-url. Thus, you can change the menu items via the entries on these pages. If, however, you wish to add new items, you first must generate two new pages in the MediaWiki namespace and then expand the variable $wgNavigationLinks. In our case, the pages are called parlist und parlist-url. Thus, you insert the information needed in the position where the new entry is to be situated in the menu:

CFG

```
wgNavigationLinks = array (
  ...
  array('text'=>'parlist',
        'href'=>'parlist-url' ),
  ...
);
```

Editing skins

In order to change the skin design, you can format and position each of the elements via its name with a CSS command. We will now edit the skin *MonoBook*. We do not wish to re-invent the wheel here, but rather simply change the design slightly. You can find the

stylesheet data in *mediawiki/skins/monobook* under the name *main.css*.

Background

If you now wish to change the background, for instance, look for the entry `body`. This is where the URLs of the background image and color are set:

CSS

```
body {...
      background: #f9f9f9 url("headbg.jpg") ↵
      0px 0px no-repeat;
...}
```

You can change the color by altering the hexadecimal number behind the #. The corresponding values can best be established with an HTML editor or graphic program. To remove the background, delete that part of the entry from `url` to `no-repeat`.

Text color

The text color can also be set in `body` in the attribute `color`. Link colors can be modified a few lines down in the entries `a` and `a.new` for links to non-existent pages. The headings are set in `h1` to `h6`.

Personalized skins

Creating personalized skins is relatively easy. We want to install the skin *FlyingSparks*. Of course, you can use any name you like.

First, make a copy of the file *MonoBook.php* and change its name to *FlyingSparks.php*. You should also copy the directory *mediawiki/skins/monobook* to *flyingsparks*. Now you have to make a few changes to the new file *FlyingSparks.php*:

PHP

```
 1 class SkinFlyingSparks                    ↵
 2   extends SkinTemplate {
 3 /** Using monobook. */
 4   function initPage( &$out ) {
 5     SkinTemplate::initPage( $out );
 6     $this->skinname  = 'flyingsparks';
 7     $this->stylename = 'flyingsparks';
 8     $this->template  =                      ↵
 9       'FlyingSparksTemplate';
10   }
11 }
12
13 class FlyingSparksTemplate                  ↵
14   extends QuickTemplate {
```

Your skin is now installed. You just have to adapt the files of the template, as shown above for *MonoBook*, and your new layout is finished.

List of special pages

Last but not least, let us make one small modification: You can tailor the list of special pages to suit your needs and, for instance, change their order or eliminate unwanted pages. Information on these pages is contained in the file *SpecialPage.php* in the directory *mediawiki/includes*. To remove entries, simply delete the line that begins with the name of the corresponding page. To change their order, move the line to the desired position. However, make sure that, except for the very last entry, all entries end with a comma.

8.4 Wiki Spam

Spam, that is, automatically generated messages with the intent of advertising a product, has, to date, primarily appeared in the form of emails, and represents a fairly new problem in the wiki world. The route via email seems to have been far more attractive to the senders of such messages than an open site up to now. Besides, in wikis, spam has usually been able to be warded off with the aid of IP bans and simple rollbacks. However, an increase in the frequency of wiki spam (as well as spam in blogs, forums and guest books) has become evident. The reason lies in the assessment of pages by search engines. The more external links lead to a hit, the higher that page is ranked. The assumption is that a well-linked site is most probably relevant to many people. To receive as many links as possible, spammers place a mass of external links to their homepage in the wikis, blogs, etc. It does not matter that these entries will be deleted soon after; the important thing is that a certain percentage of such links are still online when the search engine robots visit a site. Of course, this type of spam is aggravating. It can decrease the motivation of Wikians considerably, and thus poses a serious threat to the community.

Nofollow attribute

Luckily, even the big search engine operators have recognized this problem and are in the process of developing a solution. This solution has taken on the form of a new attribute for the href tag which indicates that the search engine, when searching for contents to be indexed, need not follow a particular link:

HTML

```
<a href="url" rel="nofollow">No Spam</a>
```

This attribute is attached to all external links. Thus, these links lose their influence for the search engine's page ranking. In MediaWiki,

this attribute is used by default in Version 1.4 and higher. If you wish to switch it off,[6] add the line

```
$wgNoFollowLinks = false;
```

CFG

to the file *LocalSettings.php*.

Blacklist

Ever since MediaWiki 1.4, there is also the possibility of installing an extension which prevents links having certain URLs from being saved. These URLs can even be retrieved from a list located in the Net, to always be up to date. To install the extension, first obtain the SpamBlacklistExtension files from http://cvs.sourceforge.net/viewcvs.py/wikipedia/extensions/SpamBlacklist/ and copy them to the subdirectory *SpamBlacklist* of the *mediawiki/extension* directory. Then, you need to insert the following line at the end of *LocalSettings.php*:

```
require_once( "$IP/extensions/SpamBlacklist/ ↵
SpamBlacklist.php" );
```

CFG

Now you still have to indicate from what source the plugin should get its filter information. This is also done in *LocalSettings.php*, directly under the line cited above. You have two options: Either you can link to a file or to a page in MediaWiki. In the following example, a file was stored as a text file in the extension directory, and a separate blacklist with the name "My Blacklist" is set up in MediaWiki:

```
$wgSpamBlacklistFiles = array(
   "$IP/extensions/SpamBlacklist/          ↵
         wikimedia_blacklist",
   "DB: wikidb My_Blacklist",
);
```

CFG

You can find a continually updated list at http://meta.wikimedia.org/wiki/Spam_blacklist. The script "load lists" can download this list for you and save it in the file *wikimedia_blacklist*. It would be a good idea to have the script automatically run at regular intervals.

---

[6] The use of the attribute is disputed. For one thing, it is not certain whether a spammer first checks to see if the attribute is set. Thus, possible success will thus only be visible and effective beyond wiki on a long-term basis. Also, a link, for instance from Wikipedia, is by all means an important statement on the relevance of a site that should not simply be ignored.

The blacklist itself consists of a series of regular expressions that label the blocked URLs. In practice, this simply means that a period must be masked by a backslash. Such an entry might look like this:

Wiki
```
spammer1\.com
forbidden\.da\.ru    #do not include
```

As you can see, it is also possible to insert comments. They must begin with a #.

8.5 Security

Generally, operating MediaWiki is as safe as operating your web server. Thus, you should make especially sure that the PHP security settings have been properly set. If you use XAMPP to operate a public wiki, you should definitely switch to safe operation mode (see Chapter 2.1).

Uploaded files

The greatest security risk is posed by uploaded files. They may contain viruses or malevolent code, and should thus be treated with special care and under no circumstances executed unchecked. This especially means that, as described above, a few types of data may not be copied to the server. An additional degree of security is offered by setting the web server to not execute PHP scripts in the upload directory. To do this, add the following commands to the Apache *httpd.conf*:

CFG
```
<Directory "path/uploaddirectory">
  php_admin_flag engine off
  AddType text/plain .html .htm .shtml
</Directory>
```

Data in LocalSettings

A further potential security problem is that access data to the database are stored in the file *LocalSettings.php* in plain language. This is risky for two reasons. On the one hand, you must make sure that this data is not passed on unintentionally if you make a backup of the wiki software and make it available to others. Thus if – for whatever reason – PHP no longer runs, the source text of *LocalSettings.php* can be read via a browser. You can remedy this by storing security-relevant data separately and not releasing it for access from outside.

Thus, create the file *access.php* in a directory which is located on a level higher than the web directory of your Apache server. In the case of XAMPP, this would be outside of *htdocs*. This file looks something like this:

PHP

```
1 <?php
2   $wgDBserver       = "DBServer";
3   $wgDBname         = "DBName";
4   $wgDBuser         = "DBUser";
5   $wgDBpassword     = "DNPassword";
6   $wgDBsqluser      = "SqlUser";
7   $wgDBsqlpassword  = "SqlPassword";
8 ?>
```

Under Linux, you should make this file readable only for yourself and PHP, that is, the system. In *LocalSettings.php*, you must now also add the access data file after the line in which the default settings are integrated.

CFG

```
1 include_once( "DefaultSettings.php" );
2 include_once( "path/access.php" );
```

In this way, the web server can be prevented from simply displaying the file. It can only be addressed within the system.

8.6 Update and Uninstall

As an administrator of a wiki, you will most likely be interested in making sure your software is always up to date. This means that you will have to load a new version of MediaWiki to your system from time to time. The risk involved has less to do with getting the program to run than it does with making sure all existing data can be taken over completely.

Update

The developers of MediaWiki have recognized this problem and integrated an automatic update mechanism in their installation program. Before performing an update however, you should save a copy of the database, to be on the safe side. If you now perform an update, just install the new MediaWiki. Instead of the new database, however, enter the access data for the existing one. The wiki will identify it and make the corresponding adjustments.

Uninstall

If you want to uninstall your wiki, you need only do two things:

- Delete the MediaWiki files and the directory in which they are located.
- Delete the database in which the MediaWiki data is stored.

This removes all traces of the wiki.

9 Life in MediaWiki

Readers
Authors
WikiAdmins
WebAdmins

In the previous chapters, you have become familiar with MediaWiki and are now capable of using it creatively. Yet you are not alone in a wiki. It can only unfold its true potential if several persons can work with it together. In this chapter, we thus want to consider what life in MediaWiki can look like. You will find a detailed description of co-operative processes in Section IV.

9.1 More than Mere Text

Wikis generate hypertexts. This form of text, however, is still relatively new. Many people learn it passively by surfing the Net, or reading Help pages, for instance, but only very few have experience in generating hypertext. Yet that is exactly what the wiki needs.

Peculiarities of hypertext

You should first be aware of the fact that reading hypertext on a screen is coupled with phenomena that are not present with normal text:[7]

- **Content-related unity on hypertext pages.** A page on the Net is generally considered to be a unit.
- **Browsing**. The reader moves through the hypertext by following existing links. This may lead to a text not being read completely.
- **Skimming**. Texts on screen are not read as thoroughly as the printed word. The reader skims the text in the search of relevant information or new and promising links. This is additionally promoted by the fact that links interrupt the text flow due to their format.

[7] See: Kuhlen 1991, Chap. 2. and Nielsen 1996, 309ff.

- **Lost in Hyperspace**. The user happily follows the links as they appear. After a while, he knows neither where he is nor how to get back to the starting point.
- **Serendipity**. On a journey through hypertext, one often discovers things one hadn't been looking for but which still appear very remarkable and helpful.

These peculiarities are translated into concrete reader needs that should be observed. Write everything that belongs together on the same page. However, do not be shy about placing new thoughts in separate articles. This is promoted in a wiki through the necessity of indicating page titles, which lends each article a kind of motto.

Navigation elements

With a history and the **Back** button, browsers already facilitate navigation through the Net. Nevertheless, it is important to set up your links so that they support browsing. For instance, it can be a great help to include a link at the start of the page leading to a superordinate or parent page and can be found under a name such as "Page up" or the like.

Wiki
```
[[Parentpage | Page Up]]
```

If you employ these conventions consistently, the reader will have it much easier. You should also provide information regarding the spot within the wiki in which the reader is presently located. Several Internet pages thus indicate a path to the current article to illustrate the trail through the hierarchy of the website.

Wiki
```
You are here: Main > Stone Age > Development
```

This decreases the risk of your readers getting lost in hypertext.

Link names

In addition, it is important that they can gain an impression of what is behind a link. This means, among other things, that links should not consist of meaningless words, even if they are highlighted and are thus very noticeable. Instead of writing

Wiki
```
You can find the development report [[devel-
opment report|here].
```

choose instead to highlight the key word:

Wiki
```
You can find the [[development report]] here.
```

You should make sure that the links you set have a connection to their information. For instance, in the following text, it makes little sense to set the first link:

```
Humankind in the [[Year]] [[10000]] B.C.
```

While it may be of interest to find out what special events occurred in the year 10000, defining "year" as a link is irrelevant. Experience has shown that little thought is given to this point. However, do not hesitate to document unusual associations with links; that promotes the serendipity effect and makes the text more interesting.

9.2 Make Participation Easy

An empty wiki is not a pretty sight. Furthermore, every wiki should contain information that specifies what is to be treated in that web. That is why a few important pages must be established right from the start. Our tribal initiative has decided to set up the pages together in a sort of workshop one afternoon, and at the same time, learn how to work with MediaWiki.

Designing the main page

First and foremost, the main page must of course be designed. Describe the orientation of the wiki's content and clarify the status that the published work will have (e.g. discussion, idea collection or finished documents). You should be certain about the target group of your wiki. If you are aiming to allow many strangers to communicate with each other, the description must be a great deal more detailed. In this case, it may be a good idea to allow a separate page for this description. For the tribal initiative, the platform will more likely serve the internal discussion, where the group is more manageable and know each other. Thus, a description of the site's purpose would not need to be quite so exhaustive.

Editing help

Because we can assume that the members of a tribal initiative do not have much experience in using wikis, it is advisable to place the most important information on its use directly on the platform. One important site in this regard is *Help:Editing*. This information can be accessed in edit mode by clicking "Editing help," and should contain the most pertinent formatting rules. Since syntax in all wikis is the same, it will suffice to use the corresponding text from Wikipedia as an example. In addition, you can establish a separate page on which questions that pop up can be posted and answered, and which can be expanded in the course of the work.

Sandbox

In step with this concept, several wikis offer a sandbox that invites users to experiment. It is not a standard component of MediaWiki; you must generate it yourself. This page has no special functions, which means that it should be cleaned, that is, its contents deleted, manually on a regular basis. A few explanatory lines detailing the purpose of the sandbox can be placed on a separate page and embedded in the sandbox as a template. The advantage to this procedure is that the explanatory text cannot be destroyed by overzealous experimenting.

Establishing conventions

A significant component to every wiki should be the provision of a series of common conventions that promote the organization of the system. It is a good idea to record them on a wiki page, because, during the course of the project, they can be modified and tailored to the needs of the group. These rules, which our tribal initiative will define together, can include items such as work processes or name conventions. Even the "obligations" of the participants are a component of these stipulations, meaning for example, the setup of an information page which we assume will be read by everyone on a regular basis.

Note: Do not forget that every web presence should include an imprint that cites the contact address, including telephone number and email address, of the provider.

9.3 You Don't Write Alone

In conventional forms of publishing, you primarily write your text so that a reader can understand it. In a wiki, you also have to consider that your readers may also be authors. Thus, you are also writing for authors.

Edit conflicts

One central topic in this regard is avoiding edit conflicts. Often, problems materialize when two users edit the same file at the same time without knowing it. MediaWiki actually has a built-in mechanism that detects such conflicts within a certain period of time and makes them known. However, many authors prefer to edit a text on their own computer over an extended period of time (e.g. due to running online costs). In such a case, the automatic protection does not work; in addition, it is very probable that someone else has taken on the text. That is why it is advisable to agree that, when editing a text, one should add a short note to the respective page indicating that the page is currently being edited and will be available again within a certain period of time.

In Wikipedia, there is a prefabricated text module that is placed above an article before extensive edits:

Wiki

```
'''Attention''': This article is actively undergoing a major edit. As a courtesy, please do not make edits to this article while this message is displayed, in order to avoid '''edit conflicts'''.
```

This module can be set in the namespace *Template*. Our tribal initiative, for instance, has done so under the name "In Use," and can embed this message on any page indicating

Wiki

```
{{in use}}
```

Unfinished pages

Once in a while, someone notices that an article is unfinished but does not have the resources at that time to continue work on it himself. To coordinate work on the project, the Fire Group has added two further templates that show that a page urgently needs editing and that its content must be checked. One or two members who are especially attentive in maintaining the wiki have these two template pages on their watchlists, to facilitate quicker access. At regular intervals, they access the pages and use the backlink function. In this way, they receive a list of all pages that contain the corresponding template and thus need editing.

Formatting source text

A second important point to be observed while writing is that other authors must understand your source text. The more complicated the construction of your page is, the more difficult it will be for others to understand, for instance regarding how certain formats can be attained. You only succeed in establishing additional hurdles for participants to the creative process in the wiki.

The readability of a source text can be greatly improved using the following simple means:

- Use **new lines** to separate units. This is, admittedly, at times not easy in MediaWiki, since line breaks in the wrong place will spoil the process. On the other hand, the page breaks are not displayed when the page is formatted, such that, e.g. you can write commands to embed templates or variables in lines.
- **Comment** the source text. To do so, you can use normal HTML comment symbols (`<!--` and `-->`). Describe the structure of the page, the most important sections and special features that need attention.

- **Store complicated layout elements separately in templates** so as not to confuse inexperienced users, and comment on the way they function.
- Select **meaningful names** for pages, templates and variables that can be understood without a great deal of effort.

9.4 Usage Examples

This section is intended to serve more as motivation than instruction. Its aim is to demonstrate how the individual elements of the wiki interact and provide an incentive to find your own creative solutions for everyday problems on the web.

9.4.1 Main Page with News

Problem: After the visitor number of the tribal initiative's wiki increased dramatically, it was decided that the main page shall be secured so that only administrators can change it. Only a small area, "News," is to remain editable by everyone.

Solution: Locking the main page is done by an admin. He or she simply activates the "protect" tab on the main page and indicates as a reason: "Main page requires special protection." Now, only administrators can alter the page. A small area, however, will remain open. This is realized via a further page, *News*, which anyone is free to edit. This page is embedded into its respective location on the title page as a template:


```
{{:News}}
```

Note the colon, which indicates that the article is located in the main namespace. It thus prevents news from disappearing from the template namespace. The page can thus also be easily accessed separately.

9.4.2 Checklist

Problem: The planning of an information booth always occurs according to the same procedure. Authorizations must be obtained, the material ordered, and the public informed via the press. So as not to forget anything and to learn from experience, we need a checklist that can be expanded if needed for subsequent information booths. Since a few of the items on the list depend on conditions on-site, the respective current checklist should also be able to be adapted accordingly.

Solution: You can probably guess that we can best solve this problem using templates. There is a prototypical list that can be copied to a new wiki page for a corresponding project. First, the page *Templates:Checklist Infobooth* is generated. It contains a table with the items to be observed in one column, and check marks in a second column:

Wiki
```
== Checklist ==
{|
| Collect material ||
|-
| have authorized by __PERSON __||
...
|}
```

As you can see, there are a few holes here that are marked by underlines. This serves as an optical indication that the corresponding information must be added when the checklist is set up. In order to get your own list, there are two steps you have to take. You must first generate a page, preferably with a link:

Wiki
```
[[Checklist Information Booth in Neander-
thal]]
```

The content of the template is loaded to this page. To do this, the key word `subst:` is necessary:

Wiki
```
{{subst:Checklist Infobooth}}
```

Now, the blanks are filled in with the information, and the list is ready for use. When an item has been completed, the person responsible for it simply inserts his signature (`-~~~~`) instead of a

check mark on the list. If, during processing, it becomes apparent that an item to be checked off is missing, it must be added to the actual list as well as the template. This ensures that the information will remain intact for subsequent information booths.

9.4.3 Literature Database

Problem: Now that the tribal initiative has performed extensive research and gathered a good deal of material on the use of fire, a fair amount of disorder reigns. The wiki will be employed to counteract that phenomenon and organize the articles.

Solution: The data must be saved in the wiki in such a way that it can be found again easily. A certain degree of discipline is necessary (unfortunately not one of the tribal initiative's strong points); conventions must be established and, above all, observed. This especially applies to the selection of a title. Since it should be as explicit as possible, it should only contain the title of the work, in its entirety, if possible. The author, year, publisher and a few keywords are set up as categories. Here, you can make life easier for normal users by using a template that displays this information in a uniform way:

Wiki

```
{|
| Author
| {{{author}}} [[category:{{{author}}}]]
|-
| Year
| {{{year}}} [[category:{{{year}}}]]
|-
| Publisher
| {{{publisher}}}                                    ↵
            [[category:{{{publisher}}}]]
|-
| Keyword 1
| {{{keyw1}}} [[category:{{{keyw1}}}]]
|-
| Keyword 2
| {{{keyw2}}} [[category:{{{keyw2}}}]]
|-
```

```
| Keyword 3
| {{{keyw3}}} [[category:{{{keyw3}}}]]
|}
```

This template is embedded in the beginning of each entry, and forces authors to provide at least a few fundamental facts about the material:


```
{{subst:Literature |
author = Firestone |
publisher = Stone and Chisel |
year = 10000 B.C. |
keyw1 = fire |
keyw2 = water |
keyw3 = use }}
```

The rest of the page can be filled with a summary and any special comments.

There are several possible ways to find a particular article. If you know the title, you can obtain a summary by entering it in a search and pressing **Go**. If you know the author or publisher, you can use the category system for access. For a subject search, you can also use the categories determined by the keywords. If all of that is of no assistance, you still have the MediaWiki full-text search option.

9.4.4 Calendar

Problem: The members of the tribal initiative, all very busy people, must coordinate their appointments and, most of all, maintain an overview of the initiative's events. Because the cooperative platform of the wiki has proven to be successful, let us try using it for this task.

Solution: MediaWiki is stretched to its limits here. A calendar that automatically enters appointments in the correct spots and shows weekends would need more means of automatic editing than MediaWiki can offer. However, you can still generate a simple annual calendar by putting together two tables. The first contains all months. It basically serves to line them up next to each other. The second table displays the respective month. First, let us look at the annual table:

Wiki

```
{|
| width=16% valign="top" |  January
| width=16% valign="top" |  February
...
|-
| width=16% valign="top" |  July
| width=16% valign="top" |  August
...
|}
```

To make it easier to follow, we can enable the months to be displayed in two columns by selecting a column width of 16%. The respective month tables are then inserted:

Wiki

```
| width=16% valign="top" | January
{| width=100% border=1
| bgcolor="#CCCCFF" | 1
|-
| bgcolor="#FFFFFF" | 2
|
| bgcolor="#FFFFFF" | 3
|-
| bgcolor="#FFCCCC" | 4 appointment
|-
| bgcolor="#FFFFFF" | 5
...
|}
| width=16% | February
```

As you can see, the individual days are displayed in various colors. January 1 is marked as a weekend. This enables the week to be organized easily without having to manually enter the days of the week. Appointments are also highlighted in color. To facilitate editing, we recommend transforming the month names into section headers. Then, every monthly calendar can be edited individually and the desired spot will be much easier to find:

Wiki

```
| width=16% valign="top" |
{| width=100% border=1
|
=January=
|-
| bgcolor="#CCCCFF" | 1
```

Don't forget to suppress the table of contents using __NOTOC__. You can view the result in Fig. 9.1:

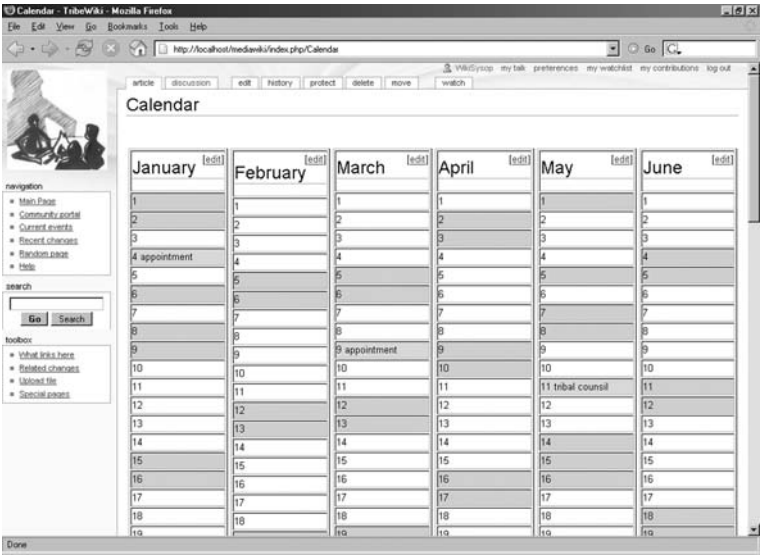


Fig. 9.1

As we have mentioned, a great deal of manual work is required for this process. Thus, MediaWiki is hardly suitable for more complex schedule management.

III. TWiki, the Jack of all Trades

10 Installing TWiki

Readers
Authors
WikiAdmins
WebAdmins

Currently, TWiki is without doubt the flagship of the wiki variants. It not only offers sophisticated permissions and user administration, various separate areas and its own forms, but also a plugin interface with which the system can be expanded by versatile additional functions, such as drawing, calculation or a database.

Required components

In contrast to MediaWiki, TWiki is written in script language. It does not use a database to store pages, but rather saves them as files in a directory structure directly on the hard drive. When doing so, it often refers to commands and programs that are present by default on Linux systems. While not impossible, this makes the setup in a test environment under Windows more difficult. For history administration, the tool RCS (Revision Control System) is employed, which records modifications and stores the files accordingly. Since this tool is not present or cannot be installed (in the case of web hosts) on all systems, TWiki has its own version of RCS which, however, is still in the test phase. All required components are listed in Tab. 10.1.

Tab. 10.1

Components	Function
Apache	Web server, administers access to files from the Net.
Perl	Script language in which the TWiki software is written.
Unix Tools	Diverse programs, especially Diff and Grep, that control access to and formatting of pages.
RCS	Revision software to monitor modifications to a file.

Before you install TWiki, you should think about what you would like to do with it:

- If you want to work with a group in TWiki, it is preferable to have your own server. The necessary installation steps are described under the **Linux System** information below.
- Of course, it is not a matter of course that everyone has his own Internet server. If you have stored your site with a **web host**, you will find the necessary information in Chapter 10.2.
- As a test system, it can also be run under **Windows** (see Chapter 10.3); in this case, however, installation is a bit more complicated, because TWiki is tailored to Linux.

In any case, prepare yourself for a somewhat complex installation process. A certain degree of experience in dealing with web administration will come in very handy. TWiki developers are also aware of the problem with the installation, and we can only hope that subsequent versions will be easier to set up.

We assume that the web server has already been established. If you have been working along through the book up to this point, you should have a version of XAMPP on your computer. Its installation is described in Section II.

10.1 Installation under Linux

Ideally, you should have root rights to install the program under Linux. Thus, you must log in as an administrator:

Shell
```
su
```

Normally, the necessary programs, Perl and RCS, will already have been installed under Linux. Verify this by requesting the respective version numbers:

Shell
```
perl -v
rcs -V
```

If you do not get an error message indicating that the command is unknown, the programs are installed. Otherwise, you must at least install Perl; there are alternatives to RCS that come with TWiki (see Chapter 10.4.2).

10.1.1 Copying TWiki

You can find the current version of TWiki under www.twiki.org or on the CD under */wikis/twiki*. Copy the file *TWiki20040902.tar.gz* to a directory to which your server has access, e.g. */opt/lampp/htdocs/twiki*. Unpack the archive with

```
tar xvzf TWiki20040902.tar.gz
```
Shell

This will generate a directory structure, and the files will be copied to their corresponding locations.

10.1.2 Configuring Apache

Determine executable directory

We first have to adapt the Apache configuration file *httpd.conf* so that the access rights to the individual directories are correctly set. You will find this file in */opt/lampp/etc*. The folder */opt/lampp/htdocs/twiki/bin* must be given permission for the execution of scripts. To do this, add the following to *httpd.conf*:

```
<Directory "/opt/lampp/htdocs/twiki/bin">
  Options +ExecCGI
  SetHandler cgi-script
  AllowOverride All
  Allow from all
</Directory>
```
CFG

Set up directory rights

The directory */opt/lampp/htdocs/twiki/pub* must be accessible and viewable to all.

```
<Directory "/opt/lampp/htdocs/twiki/pub">
  Options FollowSymLinks +Includes
  AllowOverride None
  Allow from all
</Directory>
```
CFG

On the other hand, the directories */opt/lampp/htdocs/twiki/data* and */templates* are not intended for the eyes of normal users.

CFG
```
<Directory "/opt/lampp/htdocs/twiki/data">
  deny from all
</Directory>
<Directory                                          ↵
  "/opt/lampp/htdocs/twiki/templates">
  deny from all
</Directory>
```

ScriptAlias

Repeat this entry for *templates*. Now you can determine the aliases for access via URL. In doing so, you may be able to make the web address shorter and you can mask the internal structure of your web directory:

CFG
```
ScriptAlias /twiki/bin/                            ↵
  "/opt/lampp/htdocs/twiki/bin/"
Alias /twiki/ "/opt/lampp/htdocs/twiki/"
```

Make sure that the transcription for scripts comes before the general alias; otherwise the settings will be incorrectly interpreted by the server. Restart Apache to adopt the changes. This is easiest by closing XAMPP and restarting it. If you now enter the URL http://localhost/twiki/bin, data or templates in your browser, you should receive the error message "Forbidden." If, instead, you see a list of files, check the entry again in *httpd.conf* for the respective directory.

10.1.3 Adapting Files

The scripts are designed such that the entire path of the Perl interpreter is called */usr/bin/perl*. You can track down the path on your system using `which perl`. If it should differ from the standard, you will have to modify the shebang lines accordingly. This procedure is described in Appendix A.

The files in the directory *twiki/bin* must be executable. You can achieve this by entering the following command in that directory:

Shell
```
chmod 755 *
```

The files in the directories *data* and *pub*, as well as the directories themselves, must be writable by Apache. The easiest way to accomplish this is by switching to the corresponding directory and setting the writing rights as follows:

```
chmod 666 */*
```
Shell

They also must be set for the directories themselves, with the following command in the *twiki* directory:

```
chmod 666 data pub
```
Shell

10.1.4 Adapting the Library Path

Edit the path entry in the file *twiki/bin/setlib.cfg* so that it links to your *twiki/lib* directory:

```
setLibPath = "/opt/lampp/htdocs/twiki/lib";
```
CFG

Now you can perform a preliminary trial run of your configuration. To do this, in your browser, open the URL http://localhost/twiki/bin/testenv. If you then see a text file beginning with a shebang line, that is, with `#!`, your Perl settings are incorrect. Otherwise, you will see an HTML page that starts with the heading "Test the environment for TWiki."

This concludes the Linux-specific instructions; you can now proceed with the installation in Chapter 1.4.

10.2 TWiki without an Admin-Account

Prerequisites

If you intend to run *TWiki* via a web host, you naturally will not have access to the server configurations. In addition, several web hosts do not offer SSH access, so that the files may not even be able to be edited directly on the server. Nevertheless, you do not have to give up working with wikis. However, one important prerequisite is that your web host offers *Perl* and allows you to use your own scripts. In principle, we proceed exactly as if we were installing a local copy, except that we use the parameters of the web host. We assume that you are familiar with the local installation of TWiki for your operating system.

Required information

You first need a variety of information about your web host.

- **Bin directory**. In which directory are you permitted to execute scripts? Usually, you will find a directory called *bin* or *cgi-bin*, if you log onto your account via FTP.

- **Web directory**. Which directory can be accessed via the Net? Standard names include *public_html* and *www*.
- **Perl**. What is the path to Perl? If you have SSH access, you can find the paths using `which`. Otherwise, a look at predefined Perl scripts may help.
- **Absolute path**. What is the name of the path to the directory of your web folder? Unfortunately, there is no way to guess this path. If you have SSH or Telnet access, you can have the path displayed using `pwd`. Otherwise, ask your host. If you cannot find the path, you can initially work with relative paths.

Unpack the TWiki installation file on your hard drive. If necessary, you will now have to adapt the shebang lines of the scripts in the local directory *bin* to the Perl path of your provider. This procedure is described in Appendix A.

Directory structure

Next, create a directory structure on the web server and copy the files from your local hard drive to the corresponding locations. The following folders are required:

Tab. 10.2

Local Folder	Function	Web Host Folder
twiki	Start files	Master directory of your wiki. It is the one you first access when logging in via FTP.
twiki/bin	Scripts	The bin directory defined above. You can also generate your own subfolders in this directory.
twiki/lib	Function library	This should be in the same directory as the *bin* folder.
twiki/pub	Publicly accessible files	In the web directory defined above or a subfolder.
twiki/data	Pages	Because these pages only need to be modified by the wiki software, it is advisable not to make this directory accessible from the web. If this is not possible, it is no great flaw unless you have secret data on your TWiki.
twiki/templates	Templates	See *twiki/data*.

You should wait to copy the *data* folder, because we first have a few settings to make for RCS. To do this, we need to perform a few preparatory tasks. If the folders *bin* and *lib* are not in the same directory,

you have to indicate the location of the function library in the Scripts directory. You can do this by editing the file *setlib.cfg* in *bin*. Enter the corresponding values in the parameter

```
TWikiLibPath = 'path';
```

CFG

Normally, you would enter the absolute path. However, if you have not been able to find it, just use the relative path for now.

You can now open the test script *testenv* in a browser. If you see two columns with lots of data, your installation has been successful up to this point.

10.3 Installation under Windows

Since TWiki has been tailored for use on Linux systems, we strongly advise against running public wikis under Windows. However, it may be useful to set up a version under Windows for testing or editing purposes.

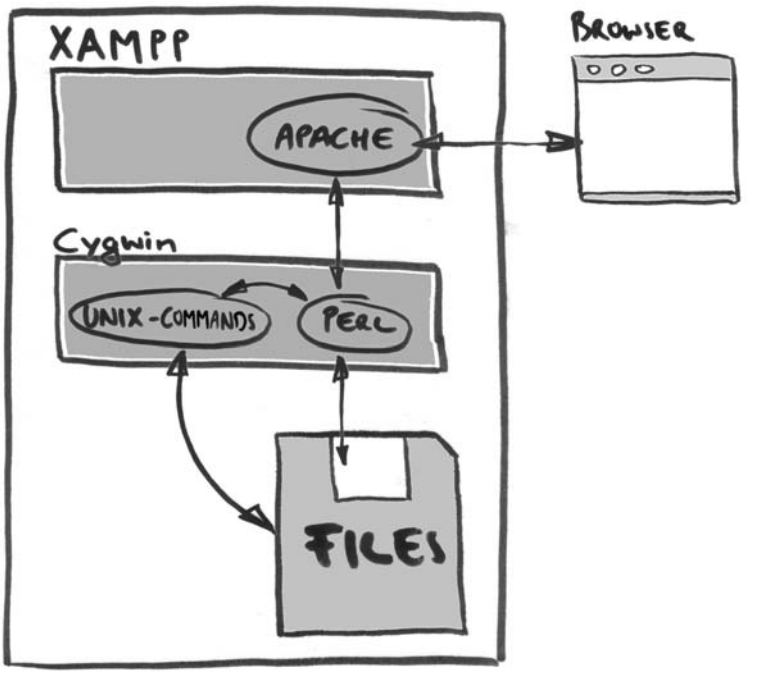


Fig. 10.1

To achieve this, we first must once again create a test environment. We have already explained the installation of XAMPP in the previ-

ous section (See Chapter 2). Now we need to teach Windows the most important UNIX commands. This chore is performed by a software package called Cygwin, which luckily also offers the option of setting up the Perl interpreter and RCS.

10.3.1 Setting up Cygwin

Starting the installation

You will find the latest version of Cygwin as of this printing, Version 1.5.12-1, on the CD included with this book. Since the package is composed of a large collection of individual programs, we have only included those programs relevant to TWiki on the CD.

Note: Of course, you can also directly download the latest Cygwin files from the site http://www.cygwin.com (see Appendix A). However, the following installation instructions refer to the CD version.

To set up the program, start the file *setup.exe*. After an introduction, you will be asked for the source of the installation. To install from the CD, select "Install from local directory." Next, you must indicate the installation directory. We have selected *c:\cygwin*, and the following path information relates to that directory. Furthermore, make sure that the "Default text file type" is set to "Unix." This prevents any subsequent complications in TWiki when uploading files. Cygwin should be set up for all users. In the next step, you will be asked to enter a directory in which the installation files are stored. Enter *[CD drive]\cygwin*.

Select components

Now you will see a selection window in which you can determine the components that are to be installed. In the first line that begins with "All," click on the word "Default." "Install" should now appear there, and all subsequent entries should also be set to "Install" (see Fig. 10.2).

Click on **Continue**. The required files will now be copied to the installation directory and set up. Lastly, you can indicate whether you would like to have icons placed on your desktop and/or start menu. This completes the automatic installation of Cygwin.

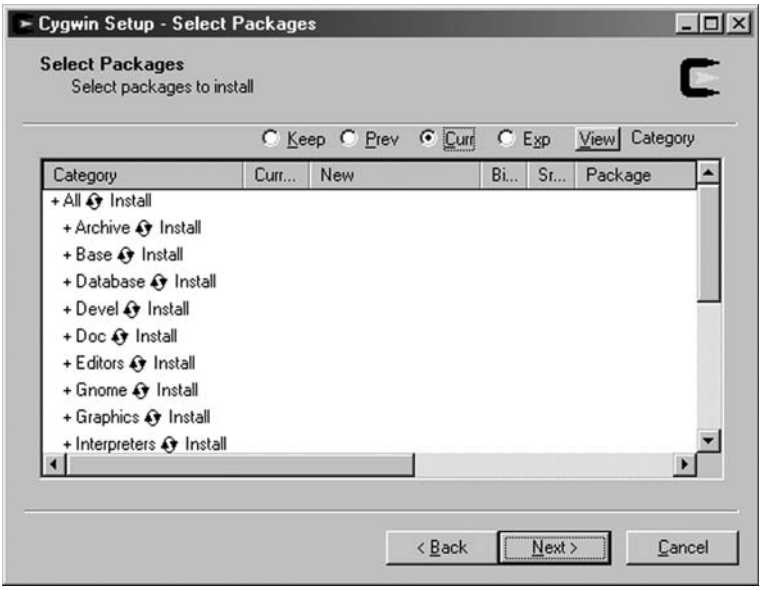


Fig. 10.2

Setting environment variables for Windows NT, 2000 and XP...

A few programs require the environment variable HOME in order to determine your user directory. Set this variable to *c:\cygwin\home\[username]*, whereby *username* stands for the name of the user registered in Windows. You can find this out easily by checking who is being logged off in the logoff dialogue (START → LOGOFF or SHUTDOWN). You can define HOME under Windows NT, 2000 and XP by clicking on the "System" symbol in the "Control Panel" (for XP, located under "Performance and Maintenance"), select the "Expand" tab and click on the **Environment variables** button. A list of existing environment variables will appear. With **New**, you can add an additional user variable to which you can assign the value explained above (see Fig. 10.3). Similarly, you should set up a TEMP entry linking to any temporary directory, e.g. *c:\temp*.

… and for Windows 98 and ME

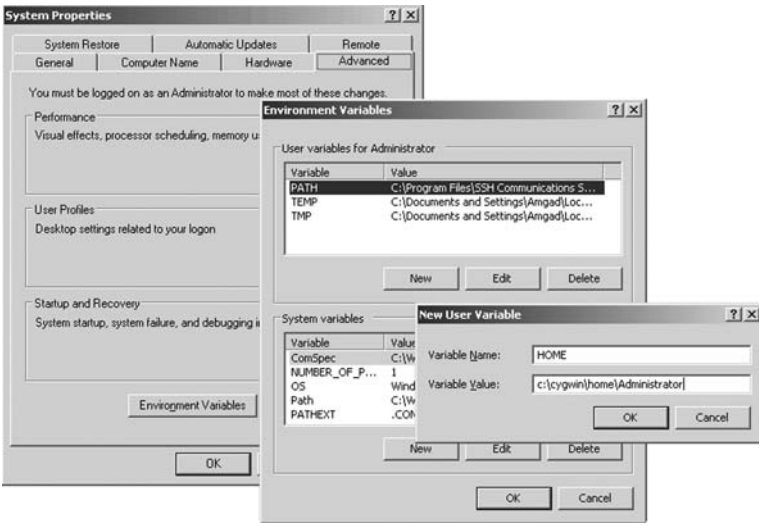
Under Windows 98 and ME, you must edit the file *autoexec.bat* in *C:*. At the end, add the lines

CFG

```
SET HOME=c:\cygwin\home\username
SET TEMP=c:\temp
```

Finally, for all versions of Windows, you should make sure that the temporary directory indicated actually exists, so that a variable does not link to nothing. Now you need to restart Windows 98 and ME.

Fig. 10.3



Trial run

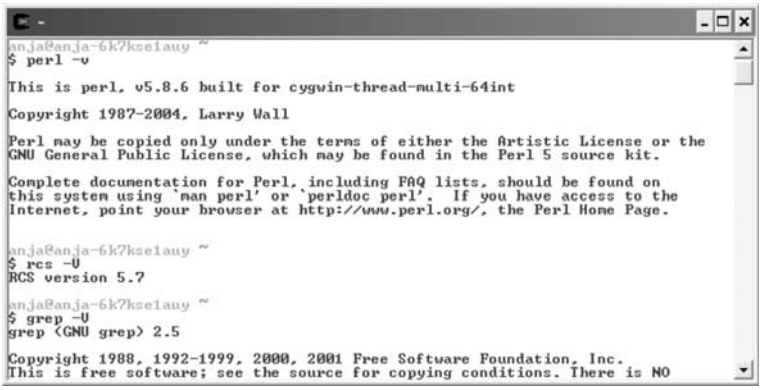
Start Cygwin for a trial run by clicking on the new icon on your desktop or in the Start menu, or by executing the file *cygwin.bat* in *c:\cygwin*. This will open a command line window that is very similar to the DOS box: The Bash shell. To verify whether all required programs are present, query the respective version numbers with the following commands:

Shell

```
perl -v
rcs -V
grep -V
```

After making the entries, the version number should appear (see Fig. 10.4). If this is not the case, the corresponding program cannot be found. You will have run the setup program again.

Fig. 10.4



Creating directories

Now you have to make sure that the TWiki directories are edited in binary mode. This is important, because otherwise additional characters (especially at the ends of lines) could find their way into the files during the version verification. First of all, create the required folders:

Shell
```
mkdir /twiki
mkdir /c
mkdir c:/apachefriends/xampp/htdocs/twiki
```

The last of the three directories indicated is the one in which the TWiki will be installed. It must subsequently be accessible by the web server; that is why we have created it right in the XAMPP *htdocs* folder.

Tip: If you should use paths whose names have spaces (e.g. "My Files,") you will have to enter a DOS path here, since spaces cannot be processed. To do this, enter the first six letters, a tilde and a consecutive number starting at 1 to indicate the differentiation. In this case, it would be `myfile~1`.

Now, connect these directories with their counterparts under Windows, and set them to be accessible in so-called binary mode. Only this mode enables smooth transfer of data between Cygwin and Windows:

Shell
```
mount -b -s c:/apachefriends/xampp/ ↵
      htdocs/twiki /twiki
mount -b -s c:/ /c
mount -b -c /cygdrive
```

The last line causes the newly connected drives to automatically be in binary mode. To test whether this mode is active, switch to the directory *twiki*, create a file and enter the following:

Shell
```
cd /twiki
echo hello world > test
cat -v test
```

The text `hello world` should now be displayed. If you find a `^M` or similar character at the end of the display, the switch to binary mode was not successful. In that case, repeat the above steps. You can delete the test file using

Shell
```
rm test
```

You now have a test environment that sufficiently reflects UNIX performance in Windows.

For a few additional functions of TWiki, you will need a Perl module that does not come with Cygwin Perl. You will find instructions on installing this module, if needed, in Appendix A.

10.3.2 Configuring Apache

Determine executable directory

Next, the web server must be informed that it is permitted to execute scripts in the directory *twiki\bin*. To do this, the Apache configuration file must be modified. This file is located in the directory of the program *c:\apachefriends\xampp\apache\conf* and is called *httpd.conf*. At the end of this file, add the following lines, making sure that the path is absolute, that is, stemming from the root directory:

CFG
```
<Directory                                          ↵
  "c:/apachefriends/xampp/htdocs/twiki/bin">
  Options +ExecCGI
  SetHandler cgi-script
  AllowOverride All
  Allow from all
</Directory>
```

Apache environment variables

When CGI scripts are executed, Apache creates a system environment in which certain variables are set. These variables, however, must be determined in advance. The following section of *httpd.conf* serves that purpose:

CFG
```
<IfModule mod_env.c>
  SetEnv RCSINIT -x,v/
  SetEnv TEMP c:/temp
  SetEnv TMP c:/temp
  SetEnv LOGNAME system
  SetEnv HOME                                       ↵
    c:/apachefriends/xampp/htdocs/twiki
</IfModule>
```

The paths must be indicated here with a / .

Script endings

The last entry for the server refers to the name of the executable scripts. In TWiki, they have no ending. Thus, Apache must be shown that files without an ending are to be executed as CGI programs. This is achieved with the following section of *httpd.conf*:

CFG

```
<IfModule mod_mime.c>
  AddHandler cgi-script .
</IfModule>
```

Note the path at the end! If the `<IfModule mod_mime.c>`-section is already present, just enter the middle line before `</IfModule>`. To make the modifications effective, Apache must be closed and restarted. That will render the web server ready for operation with TWiki.

10.3.3 Copying TWiki

Finding installation files

After having sharpened our test environment, we can now work on installing TWiki itself. The newest version can be found at http://www.twiki.org. We have included Version 20040902 on the CD, located in the directory *\wikis\twiki*. If you have used the paths indicated in this book up to now, you will find a version modified by us, Version TWiki20040902_win.zip, which will do some of the work for you. We assume you will use that version in the instructions to follow. If not, please see Appendix A. Copy the packed file to the TWiki installation directory on your hard drive and unpack it. This is most easily done by switching to the corresponding folder in the Cygwin shell with `cd /twiki` and then executing "unzip":

Shell

```
unzip TWiki20040902_win.zip
```

This unpacks the program files to the corresponding directories. They are all located in the TWiki directory and do not come into contact with Windows components.

At this point, you should test whether the configuration has been successful up to now. To do this, use a browser to access the URL http://localhost/twiki/bin/testenv.

Caution: Do not forget to start XAMPP first. If an error occurs when doing so, or if you cannot access http://localhost, this is most likely due to a typing error in *httpd.conf* . Pay special attention to the

fact that there are line breaks in this book where there should not be in the actual configuration file.

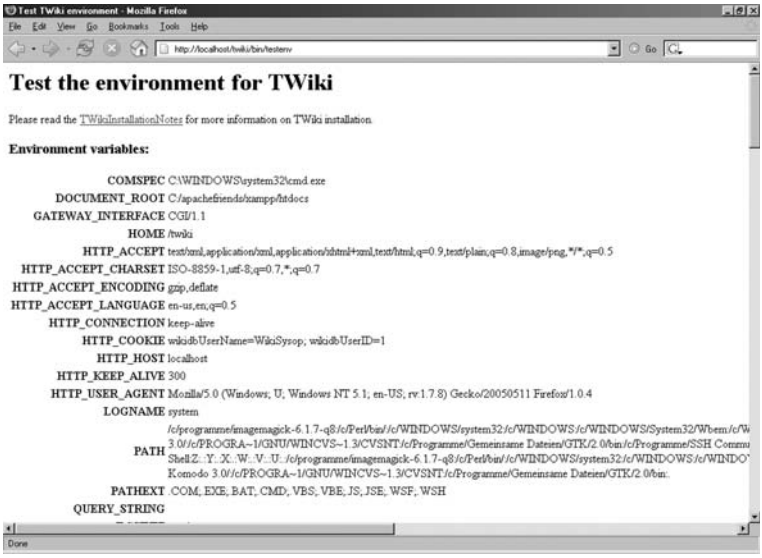
You should now see a long list with information under the heading "Test the environment for TWiki." If, instead, you see programming code beginning with `#!`, the directory *twiki\bin* has not been released by the server as a CGI directory. Check the path in the *Apache* configuration file. If you see the error message "Internal server error," the shebang line does not link to Perl. You can find out how to remedy this in Appendix A.

Thus concludes the Windows-specific portion of the installation process.

10.4 Completing TWiki Installation

The script *testenv* is intended to be a diagnostic tool with which to view the current configuration and find any errors present. You can access it in the browser via the URL http://[hostname]/twiki/[binpath]/testenv; thus, for example: http://localhost/twiki/bin/testenv.

Fig. 10.5



A list is first displayed with the values of the environment variables. In the area "CGI setup," you will find the installed and any required

Perl modules. A few optional modules are displayed that are not yet available. You can subsequently install them according to the methods described in Appendix A. The entry "user" shows the user name under which the scripts are executed. In addition, you will probably also see a warning regarding the RCS.

10.4.1 Changing RCS Owners

For the RCS, the system must have access to the key to the RCS files. If this is not the case, modifications cannot be made. Files are issued with "nobody" as the default owner. If your system does use any other name (and that is most likely the case), you will have to change the owner. For this procedure, you will find a link to an automatic script on the page; however, at the time of this printing, it seems to still be in a very experimental stage. Thus, we have included a small script with which you can make the changes. To do this, change to the directory *twiki/data* and enter the following command:

```
sh unlockrcs.sh username
```
Shell

You can find your user name on the Testenv page under "user."

Caution: Do not confuse the user entry with the log name, located further up.

The script creates a backup file called *old-data.tar.gz*. If something has gone wrong, you can revert to the former state by copying the backup file back over. This is done using

```
tar xzvf old-data.tar.gz
```
Shell

Otherwise, you can delete the backup file with

```
rm old-data.tar.gz
```
Shell

If you open *testenv* again, you will still see a warning, but this time, you can happily ignore it.

10.4.2 Adapting Twiki.cfg

The last step in the installation process consists of adapting the configuration file. It is located in the directory *twiki/lib* and has the name *Twiki.cfg*. Open this file in an editor (e.g. nano or vi). In principle, a series of paths must be adapted to fit your system's conditions. For Windows users: Use a normal slash / as a path separator. The file *testenv* provides valuable tips on how the variables in *Twiki.cfg* must be set. Thus, the value of

```
$defaultUrlHost = "http://localhost"
```

should match that of "HTTP_HOST." The value in the next variable corresponds to the relative path of the web folder *htdocs* to the directory *bin* or *pub*:

```
$scriptUrlPath = "/twiki/bin"
$pubUrlPath = "/twiki/pub"
```

The following parameters point to the TWiki directories and should also be written as absolute paths under Windows as well (but with /, not \), for example:

```
$pubDir =                                    ↵
  "c:/apachefriends/xampp/htdocs/twiki/pub"
$templateDir =                               ↵
  "c:/apachefriends/xampp                    ↵
     /htdocs/twiki/templates"
$dataDir =                                   ↵
  "c:/apachefriends/xampp/htdocs/twiki/data"
```

Next, select the proper version of the parameter `$safeEnvPath` that corresponds to your system. There are various sections for the variety of operating systems and software configurations. You can activate your variation, if necessary, by removing the hash at the start of the respective line. Under Linux, you need not make any changes here. For Cygwin, the lines are to look like this:

```
#$safeEnvPath = "/bin:/usr/bin"; # Unix/Linux
$safeEnvPath = "/bin";           # Cygwin
```

Do not forget to add it in front of the versions of the variables that are not longer valid.

TWiki can work with two different version control systems. Normally, it is advisable to use RCS. Under Linux, it is usually already installed. If, however, you have not got RCS on your system, you can use the RCSLite contained on the CD. We also recommend this option under Windows/Cygwin. However, RCSLite is still in a test phase, and is not recommended for use in production.

If you want to use RCS, check to see whether the program is located in the directory indicated under `$rcsDir` . Normally, you do not need to make any changes here. You can find the RCS path using

```
which rcs
```
Shell

whereby the program name is indicated at the end.

RCSLite

To use RCSLite, you must set the variable `$storeTopicImpl` accordingly:

```
#$storeTopicImpl = "RcsWrap";
$storeTopicImpl = "RcsLite";
```
CFG

As you can see, you only need to switch the comment symbol; the rest is done for you.

Paths to grep

Finally, it is important for Windows users to reset the paths to the grep commands. Set the entries as follows:

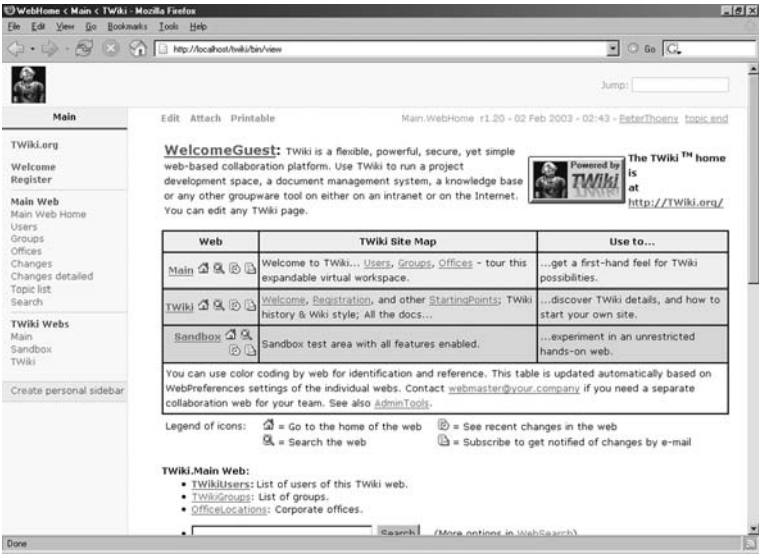
```
$egrepCmd = "/bin/grep -E"
$fgrepCmd = "/bin/grep -F"
```
CFG

That should enable TWiki to function. Further settings that can be made in *Twiki.cfg* will be detailed in subsequent chapters. Have another look at *testenv*. There should be no more error messages starting with the word "Error." Windows users may find a security message for the template directory, which is insignificant.

Opening TWiki

Now, the moment of truth has come: It is time to start TWiki for the first time, to verify whether all of the proper entries have been made and all components are available. Open the wiki with *view*. It has the same URL as *testenv*, with the exception that "view" is at the end of it. If you now see an interface as depicted in Fig. 10.6, your installation has been successful.

Fig. 10.6



10.5 Viewing TWiki Pages

Checking RCS

Now would be a good time to have a look around your TWiki to determine whether the installation has been executed correctly. At the end of a page, you will find the version of that page under "Revisions." If there is only one entry found here (on different pages), RCS is most probably not correctly installed. In this case, check the last entries in the error report of the Apache server to see if the program has been found. If it has not, the path `$rcsDir` in *Twiki.cfg* has most likely been incorrectly set.

Testing diff

Click on one of the greater-than signs to view the differences between two pages. If you only see empty pages, the Diff command could not be found. You would then need to check `$safeEnvPath` in *Twiki.cfg*. It must be set such that the Diff program can be found here.

RCS files unlocked?

Note the revision number of any page. Edit the page using the **Edit** option and save it with **Save**. The revision number should now have increased by one. If this is not the case, the RCS revision files have not been correctly unlocked. You would then need to restore the backup files in *twiki/data* and repeat the unlock procedure as described in Chapter 10.4.1.

Grep found?

Click on "Topic list." This will display a complete list of all pages in this web. If errors occur here, you should have another look at the

paths in `$lsCmd`, `$egrepCmd` and `$fgrepCmd` in the configuration file.

Write permission

Finally, load a file into TWiki. This is done via the link **Attach**. On the following page, enter a file and click **Upload file**. You should then see a table at the end of the entry. If it is not possible to upload files, this is probably due to the fact that the server in the directory *twiki/pub* has not received write permission.

Once all tests are successful, you then have a functional TWiki that you can work with.

11 Working with TWiki

Readers
Authors
WikiAdmins
WebAdmins

11.1 What are Webs?

Before getting to the components and functions of the start page of TWiki, that is, the first page displayed when the program is opened, we would like to provide a brief explanation of a special feature in TWiki: the webs.

A TWiki is divided into several webs. These are self-contained areas in which the pages are located. The nice part about them is that you can more or less operate several small wikis for various project groups, and yet only need a single TWiki installed on your web server.

The disadvantage is that, at least at the beginning, it may be a bit confusing – because, once your TWiki is installed, you already have a series of standard webs at your fingertips that exist equally and parallel to each other, but perform diverse functions. For instance, the sandbox enables uninhibited experimentation, while in the TWiki Web, one can determine important settings for the layout, among other realms.

Here is a brief overview of the standard webs:

Standard webs

Tab. 11.1

Web	Use
Main	Starting point in the TWiki. The user and group pages are also located here.
TWiki	Documentation of the TWiki as well as general settings and templates that apply to the entire wiki.
Sandbox	Playground in which wiki users can try out various functions without needing to worry about causing damage.

You can add any amount of your own webs to these pre-installed ones. Instructions for doing so are detailed in a subsequent chapter (15.1).

Topics

There is another noteworthy little TWiki quirk: The pages are not called "pages," but rather "topics," which, especially regarding menu items such as "Topic list" or jump labels like "Topic end," may lead to confusion. So remember: topic = a TWiki page.

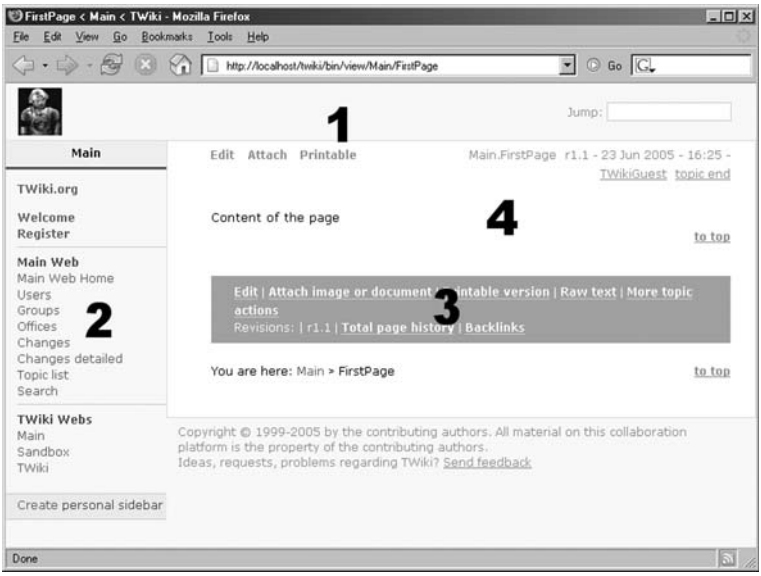
We should also mention one notation convention. When we refer to a topic in a certain web, we write *Web.Topic*.

11.2 A Website

Where is everything?

We are now in the *Main* web, which is invoked when you open the TWiki URL. All subsequent explanations, however, also refer to all other webs.

Fig. 11.1



We can divide this page into four general parts:

- At the top is the **Header** (1), normally highlighted in color and containing a logo. It serves as an orientation and permits fast access via "Jump."
- On the left, there is a **Menu sidebar** (2) containing the most important links and functions pertaining to the entire web. You

can also add your own links or diverse functions (see Chapter 16.1).

- The **Footer** (3) offers options you need to edit the content.
- The **Body** (4) is situated between header and footer, and can be filled with content via the edit box.

11.2.1 The Menu sidebar

We will now have a closer look at the most important components of the menu sidebar:

Register

If you wish to edit a page, you must first register in the login box that appears when you click the **Edit** button. You can register via a "borrowed" account, that is, as a "TWikiGuest" with the password "guest," or you first register as a regular user by filling out the registration form with a TWiki name and password. For the sake of consistency, your TWiki name should be a WikiWord, such as a first and last name without spaces, as in: *TimTroglodyte*.

When you register

Users

- an account will be opened for you. Using your TWiki name and password, you can now make yourself known at any time. Other users can see who is responsible for the edits you have performed.
- TWiki creates an additional user page with your TWiki name as its title and the further information in the registration form as its content.
- your name will be entered on the *TWikiUsers* user list.
- you will receive confirmation via email.

If you would like to verify whether your entry has been successful, click on "Users" in the menu sidebar. You should find your Wiki name in the appropriate spot in the alphabetical list of users. The names are linked to their respective pages. Behind that entry, the registration date is also indicated.

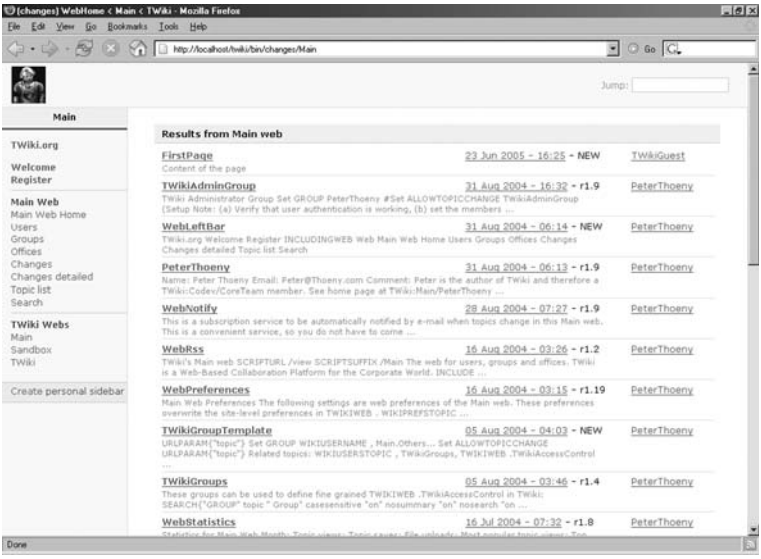
Note: Registration is not yet activated for the out-of-the-box installation; that means you can edit all pages without registering (see Chapter 14).

Changes

On the “Changes” page, you will see an overview of all pages that have been recently edited. The last revision is listed first. Access to older versions of a page, however, is not provided here. You would have to use the “Revision” option of the respective topic.

Caution: The focus of this function is on the names of the topics that have been edited. That is why only the most recent edit of a page is indicated.

Fig. 11.2



To the left, the name of the edited page is indicated, in the middle the date and revision number, and to the right, the originator of the edit. Underneath this line, you will see a pale gray shade indicating the start of the content of the page.

The “Changes detailed” page, on first glance, is very similar to the “Changes” page with regard to function. However, it is based on a different procedure, and will replace “Changes” in the next version of TWikis in order to enable more modifications to be found.

Topic list

As mentioned above, the term “topic” means the same as “page.” Accordingly, the “Topic list” is a list of all pages currently in the web in alphabetical order. These include:

- Pages created by you and the community.
- User pages automatically generated upon registration.
- Configuration pages
- Documentation pages of TWiki developers.

The structure of the "Topic list" is similar to that of the "Changes" page: From left to right, the page name, creation date and current revision number, and the user who edited the page are listed. Underneath, again, is the start of the page's content, highlighted in gray.

Search

The next important option in the menu sidebar is the search function. There, you can enter one or more search items in the text box. Using the radio button, you can determine whether the key words should be searched in topic names, in the content, or both, and thus, if desired, whether the search should be limited to a certain realm to speed it up. In addition, you can also set whether the current web or all webs in the TWiki are to be searched.

The customary rules apply to the search function:

- Words lined up next to each other are linked by a logical AND, meaning the function only finds those pages containing ALL of the search terms.
- If words are sought in a specific order, they must be placed in quotation marks, such as "Eric Raymond."
- To exclude certain terms from a search, place a minus sign in front of that particular key word.
- However, if your entire search begins with a minus sign, you should mask it in quotation marks, such as "–nowarns".
- Filler words such as adjectives and conjunctions are not taken into account during the search unless you place a plus sign in front of them. These so-called stop words are defined on the page *TWiki.TWikiPreferences* under the item `SEARCHSTOPWORDS`.

Here is another brief search example:

```
flame -romance
```

Here, all pages are searched having the word "flame," but that have nothing to do with romantic encounters.

Advanced Search

If you want to further specify your search, use the advanced search services. In addition to detailed selection criteria regarding the web and additional sorting functions, you also have the opportunity of entering your search as a regular expression. This is a notation convention especially popular with programmers for very flexible and thus powerful searches. You should observe the following "ground rules," which differ slightly from a simple search:

- The search patterns are either linked with a semicolon (AND) or a pipe symbol (OR).
- To exclude a word, do not use the minus sign, but rather an exclamation point before the word.
- Semicolon and exclamation point are to be masked with a backslash.

Here is the above example of a simple search as a regular expression:

RegEx
```
flame;!romance
```

Are you wondering why one would perform a search with regular expressions when a simple search would have the same result? Well, just then try to obtain the result of the following search pattern with a simple search:

RegEx
```
(torch(|es))?light(|s)?;!electric
```

Here, we are looking for "torch" in the sense of a flaming torch, either in the singular or plural, and "torchlight," in the singular or plural, but not the word "electric," because we are not looking for a flashlight, whose British counterpart is an "(electric) torch."

We should briefly mention here that you can embed your search into a page using the `%SEARCH%` variables. We will examine this option closer in Chapter 13.

Create Personal Sidebar

The last item in the menu sidebar is the command "Create personal sidebar." However, you will only be able to see it if you have not yet used it, since, when you activate it, you will enter the edit mode of the so-called *WebLeftBar* page. Here, you can design the lower section of the menu sidebar as you wish, e.g. by adding your favorite links. Even if you do not change anything in the text, you will now have your own little section of the menu sidebar in which at least the link to your user page is entered. Do not be confused by the many preset variables in the source text; rather, just try setting another link by entering the following text:

URL
```
http://www.wiki-tools.org
```

When you have saved the page, you will not only be able to see the modification in the menu sidebar, but also – in step with the wiki philosophy – on the page itself.

Note: Even a *TWikiGuest* can modify this section of the menu sidebar, although it makes little sense, since various people log in as a *TWikiGuest* and thus can change the entries.

This concludes our examination of the menu sidebar. We can now move on to the next component of the window.

11.2.2 The Information Bar

The information bar serves as a header: It tells you what page you are currently visiting, which version you are looking at, when it was last modified and by whom. In addition, you will find a couple of TWiki tasks, such as **Edit**, **Attach** and **Print**, which are also located in the toolbar and will be discussed in detail. The advantage of this doubling up becomes evident in long texts, because you do not need to scroll to the end of an article to access them. If you nevertheless need to quickly get to the footer, you can always use the jump label "topic end."

11.2.3 The Toolbar

Attach image or document

The function toolbar contains the most important tools for *TWiki* authors. In addition to the core function **Edit**, which opens the edit window (see below), you can add all types of files to the page using **Attach image or document**. This is a very useful instrument with which you can perform several functions.

For instance, it is possible to attach files of any format to a certain TWiki page for opening and saving. Along with a complete revision control, it facilitates cooperation between participants on a document, since any needed data located outside of the wiki can easily be made available to everyone. Access to the documents is also optimally organized with an access control on the page level.

These functions represent prime prerequisites for file sharing: Well documented and categorized, you can make available multimedia files, drivers, patches, etc.

In addition, you only need to upload the file once; you can use it on a particular page, in that web, or in the entire TWiki.

Attaching files

To attach a file, click on the **Attach** link. You will then access a dialogue page where you can make more specific settings before uploading the file.

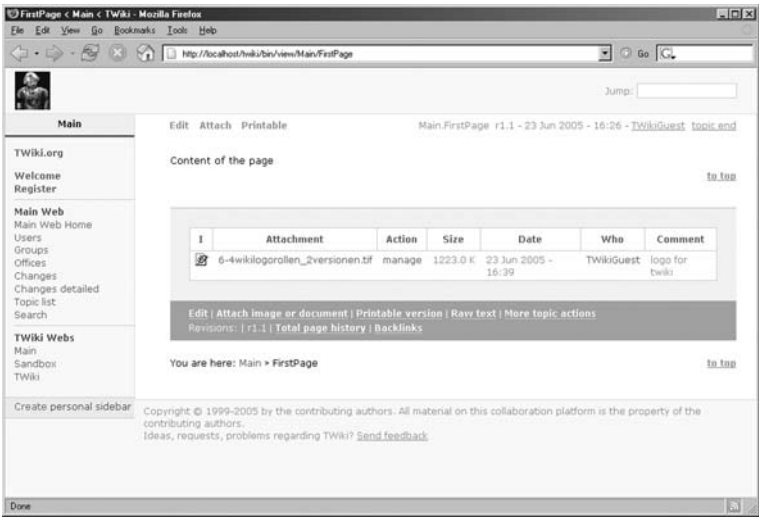
For instance, you can enter a comment or determine that a file will be attached in "hidden" mode, so that it is not indicated on the page as being attached. It can still be integrated into the page. However, the advantage is that at the end of the page, it does not appear in an attachment table. If you activate the "Link" option, it enables the attached image to be viewed in the editable text of the page. If it is not an image file, a simple link to the file is established.

Generally, any type of file can be uploaded. However, if the files are program files that might pose a risk, TWiki disarms them by adding an additional .txt, for instance to the name of a .php file.

Caution: You can limit the size of the attachment in the variable `%ATTACHFILESIZELIMIT%` on the page *TWikiPreferences*. The default value is set to 10000 KB. Larger files are not recommended anyway, since loading time would be considerably slowed or a timeout produced. For such cases, use an FTP site.

When you return to the page view, you will see the following table on the bottom, unless the file was hidden when uploaded:

Fig. 11.3



Here, anyone having authorization to edit the topic has the opportunity to download the file. In addition, you can use the **Manage** link to access a page in which all important settings regarding the at-

tachment can be made and/or modified, including moving and deleting a file:

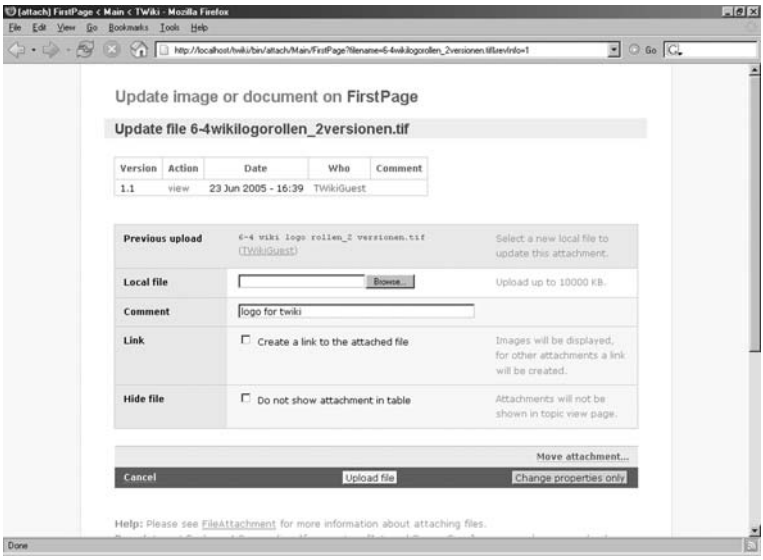


Fig. 11.4

The first table shows all attached files and their attributes. The `h`, for example, means that it is hidden.

The second table lists the various versions of a file. For every new upload, the revision number increases by one. To evaluate a certain version, you must click on the respective **View**.

If you only change the following attachment attributes, e.g. "hidden," it will suffice to confirm the change with "Change properties only."

There is one small detail to observe: If you have entered or edited a comment, the new comment will only be visible in the page view; in the revision table, the comment will continue to appear as it did when the file was uploaded.

Tip: In contrast to TWiki pages, uploaded files are not locked when you work on them. To avoid memory conflicts, you should thus note in the comment that you are currently working on the file.

Moving / deleting files

To move a file, activate the "Move attachments" link. In the dialogue box that appears, you can either move the attachment to another page (even one in another web), or delete it completely by selecting the web *Trash* and the page *TrashAttachments*.

Note: You can only attach files if you have editing rights for that page. However, this limit does not apply to moving files!

Linking to attached files

Once a file has been uploaded, there are several ways to link to them. For instance, you have the option to automatically set a link or a graphic in the source text by activating the "Link" option on the Attachment dialogue page.

This can also be done manually by using a variable (see below) with

Wiki
```
%ATTACHURL%/fire.txt
```

In the page view, the following link appears, with which one can access the file: http://www.wiki-tools.org/twiki/pub/TWiki/FileAttachment/Sample.txt.

It works the same way with a graphic image:

Wiki
```
%ATTACHURL%/cavepainting.gif
```

However, no link appears here, but rather the image itself.

To access the attachment of another page, you must employ one or two other variables.

If the page is located in the same web, you need:

Wiki
```
%PUBURL%/%WEB%/OtherPage/fire.txt
```

If the page is in another web of the same TWiki, use:

Wiki
```
%PUBURL%/OtherWeb/OtherPage/fire.txt
```

There is also the possibility of directly integrating an attached HTML file into the source code using:

Wiki
```
%INCLUDE{"%ATTACHURL%/Fire.txt"}
```

Embedding HTML files

The content of the file, that is, the HTML code, is then seamlessly embedded in the edit box and processed when saved.

Note: There is no access control for the individual attachments. If you wish to protect a certain file, attach it to a separate page and apply an access control code to it.

All attachments are saved in the web server directory *twiki/pub*. A separate folder is created for each web which, in turn, contains a folder for attached files.

Printable version

Let us now continue with the toolbar. There is not much to say about the function "Printable version:" When it is activated, all TWiki menu bars disappear, and you can print the page without all of the superfluous bordering elements.

Raw Text

The option "Raw text" lets you have a look at the source text of a page without having to switch to edit mode. This is especially useful if you are not authorized to edit a topic but want to see how certain areas have been generated, and e.g. if you want to adopt them with the copy & paste function.

More topic actions

Via "More topic actions," you can access a list of further very useful functions:

- **Move to Trash:** Here, you can delete the current page. The page is moved to the *Trash* web. If the topic name is already present there, you must first rename it. When deleting, you will receive the option to update links to the page being deleted, so that they will then lead to the page in the *Trash* web. You can perform this update in the current web or in all public webs of the TWiki.
- **Rename or move topic:** A similar procedure is used to move or rename topics, except of course that in these cases, you would not select the *Trash* web. Here, too, you can have the links be automatically updated.
- **Back-links (Ref-by):** This function generates an alphabetical list of all links that lead to the current page.
- **Child topics in *Main* Web:** Lists all pages in the current web (in this case *Main*), that were generated from this page.
- **Set new topic parent:** The parent page is actually the one from which another page is generated. Since it is indicated in the footer of the page as a path, it is sometimes necessary to change it for organizational purposes.
- **View previous topic revision:** If you know the number of a revision you are interested in, you can directly search for it here and have it displayed in the page view or source text. For older versions, however, there is neither an edit nor a save option. Thus, you can no longer modify the page.

- **Compare revisions**: If you are interested in comparing the differences between two revisions, you can have them displayed here. Just indicate the revisions you wish to compare.

Revisions

And now back to the toolbar. You can also access the revisions and their differences views from here: Click on the respective revision number or, to compare versions, on the greater-than/equals sign between them. If the older page is no longer on the list, however, you would have to resort to using the More topics page.

Total page history

The link "Total page history" provides a complete overview of the entire history of the page. All modifications between the revisions are also identified in a clear manner.

Backlinks

The last action available on the toolbar is "Backlinks." This may sound familiar; it was also on the "More actions" page: All links leading to the current page are shown.

11.3 The Edit Window

If you are logged in as a user and would like to edit a page, when you have clicked on the **Edit** button, you see the content of the page in its source text, and can then make any modifications or additions you like. All menu and information bars normally visible in normal view are hidden. You only see the options that apply to the current mode.

Fig. 11.5

11.3.1 Storage Options

Edit lock

Underneath the edit window, you can initially determine if you wish to lift the edit lock. This lock ensures that the edited page is locked for 60 minutes, meaning all users trying to access the page will receive a corresponding message. In addition, all modifications that you are currently making to the page will be included in the last version; that is, if you discover an error after saving the page and correct it within the 60 minutes, a whole new version of the page is not generated for that edit.

Note: The edit lock can, however, be circumvented. If you access a locked page, you have the option "Edit anyway," with which you can still make modifications to the document.

In order to release the page from the edit lock before the lock period has expired, activate the option "Release edit lock." Your own modifications will, however, be collected in a single version.

You can also set the edit lock period to fit your needs. To do so, access the file *lib/Twiki.cfg* and change the value of the variable `$editLockTime`.

Minor changes

You have already seen the "Minor changes" option in MediaWiki. The edit is marked as minor. That means that in TWiki, no message regarding the edit will be generated by the *WebNotify* service (see Chapter 15.5).

11.3.2 Saving Your Work

You have three easy ways in which to save your work:

- **Save**: The regular way to save pages. When you use this method, the edit lock is set.
- **QuietSave**: It has the same effect as saving with "Minor changes" and was introduced for convenience.
- **Checkpoint**: This is a kind of interim storage. The page is visible to all in this state. However, you do not leave edit mode, but, after such a save, can continue working in it.

Before final saving of the page, you can have another look at it using the **Preview** function. However, do not forget to save the file afterwards; otherwise your edits will be lost!
If, for some reason, you wish to cancel the editing procedure and return to the original page view, just press the **Cancel** button.

Access keys

Access keys are key combinations enabling the advanced user to work more quickly. Using them, it is no longer necessary to first grab your mouse and then click on buttons.

You can combine the following letters with other keys:

Tab. 11.2

Function	Key
Cancel	C
Checkpoint	K
Quiet Save	Q
Save	S
Preview	P

Which key combinations you need to press depends on your browser and operating system:

- *Netscape Navigator*, *Mozilla* or *Firefox*
  - Hold the <ALT> key down
  - Press the required letter.
- *Internet Explorer*
  - As above, except after letting the keys go, you must also press <ENTER> .
- *Mac*
  - Hold the <CTRL> button down
  - Press the required letter.

Formatting aids

At the end of the page, you will find a short overview of formatting commands in TWiki, e.g. how to italicize a text.

12 Formatting in TWiki

Readers
Authors
WikiAdmins
WebAdmins

In principle, formatting in TWiki is done in exactly the same way as in MediaWiki. Each wiki, however, has its own conventions. Thus, you must get used to a few new formatting signs.

12.1 Formatting Text

Generating a page

The best way to test formats is to use a separate page, which we will now generate in preparation for our formatting tasks. To do this, set a link to *FormatTest* on the page *Main.WebHome*:

```
FormatTest
```

Wiki

You will then see that there is a question mark after the link. This is TWiki's way of indicating that the page does not yet exist. When you click on the question mark, the page will be created. The page of origin will be entered as the parent page. You can recognize this by the fact that the parent page appears at the end of the page in the path "You are here."

Headings

TWiki headings always begin with three minus signs, and are followed by plus signs and then the heading text. A heading of level one, that is, the most important heading, has one plus sign, a heading of the second level has two plus signs, etc. No additional format indicators must be placed after the heading.

```
---+Heading 1
---++Heading 2
---+++Heading 3
```

Wiki

Headings can be used in up to six levels.

Formatting text

You can view the other formatting options in the following table:

Tab. 12.1

Description	Example	Result
Bold	`*Text*`	**Text**
Italics	`_Text_`	*Text*
Bold + italics	`__Text__`	***Text***
Fixed font	`=Text=`	`Text`
bold + fixed font	`==Text==`	**`Text`**

Fixed font text is written in a font with a fixed width (usually Courier) and displayed in another font color.

If you wish to prevent the source code from being interpreted (e.g. HTML), place it between `<verbatim>` tags. For instance, this is useful if you are describing an HTML formatting procedure. Such a description should be readable to the user and not interpreted in the display:

Wiki

```
<verbatim>
  <b>makes bold text</b>
  <i>makes italics</i>
  <a href="test.htm">linked</a>
</verbatim>
```

Caution: Do not put any spaces between the format indicators and the text; otherwise the formatting will not take effect!

Text color

With a bit more effort, you can change the font color of your text. In TWiki, there are variables (see below) for the most popular colors:

Wiki

```
%YELLOW%, %RED%, %PINK%, %PURPLE%, %TEAL%,
%NAVY%, %BLUE%, %AQUA%, %LIME%, %GREEN%,
%OLIVE%, %MAROON%, %BLACK%, %GRAY%, %SILVER%
```

To lend color to a paragraph, write the respective variable before the text, and follow it with `%ENDCOLOR%` to end the color mode.

Wiki

```
%RED% red color %ENDCOLOR% and %GREEN% green
color %ENDCOLOR%
```

Note: If you wish to switch from one color to another, you must still end the active color with `%ENDCOLOR%`, e.g. `%RED% red color %ENDCOLOR% %GREEN% green color %ENDCOLOR%`.

You can see a display of the previously discussed formatting procedures in the following screenshot:

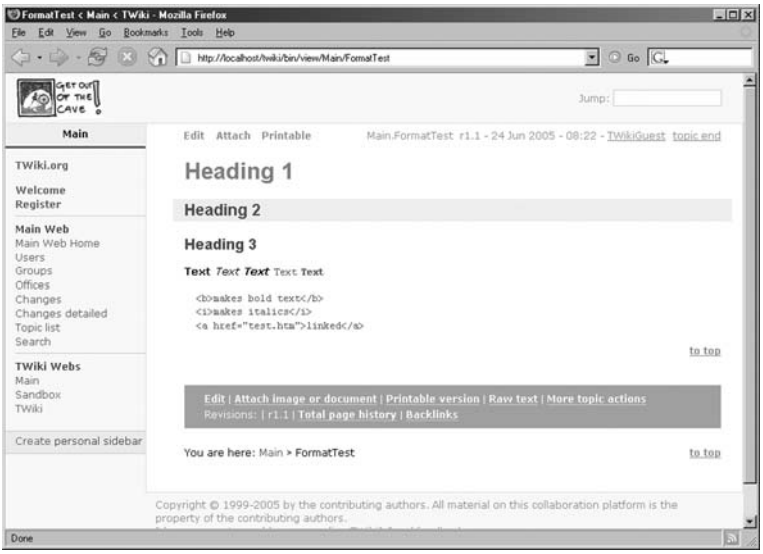


Fig. 12.1

Now that we have reviewed the text formatting options, let us turn to structuring options. You can start a new paragraph – as in MediaWiki – simply by inserting a blank link before it.

Paragraph

If you want to structure your content in the form of lists, there are several options, displayed in the following table. Please note that three blank spaces must be inserted before each corresponding format indicator.

Lists

List	Example	Result
Simple list	`···*·Fire`	• Fire
Nested list	`···*·Fire`	• Fire
	`·····*·Embers`	◦ Embers
Numbered list	`···1.·Fire`	1. Fire
	`···1.·Water`	2. Water
	`···A.·Fire`	A. Fire
	`···A.·Water`	B. Water
	`···i.·Fire`	i. Fire
	`···i.·Water`	ii. Water

Tab. 12.2

Definition list	`···$·Fire:·hot` `···$·Water:·wet`	Fire hot Water wet

12.2 Tables

The following three methods can be used to generate tables in TWiki: using a wiki convention, in normal HTML and with the aid of the `<verbatim>` tag. These three procedures will be explained below.

12.2.1 Wiki Convention

In the wiki convention, each line represents a row in a table; any spaces at the beginning of the line are ignored. The row can consist of several cells. Each cell begins and ends with a pipe (|). That is basically it!

In addition, however, there are also several formatting options available:

- | ***Content*** |: This cell is interpreted as a table header and thus – depending on the browser – is more strongly emphasized than the other cells.
- | **Content** |: (Two spaces on the left and the right) The cell content is centered in the cell.
- | **Content** |: (Two spaces on the left) The cell content is aligned with the right margin.
- | **Content** | |: The cell is combined horizontally with subsequent cells to make multi-span columns. If you only want one empty cell, you must place a space between both pipes. You can also join several cells by entering the corresponding number of |.
- |^|: The cell is joined vertically with the cell above it to form multi-span rows. It is also possible to join several cells in this way.

The following example table illustrates the TWiki table conventions:

```
| *Culture* | *Primitive* | *Developed* |
| Mesolithic Age |||
| Fire |  -  |  X  |
| Agriculture |^|  -  |
| Neolithic Age |||
| Fire |  X  |  X  |
| Agriculture|  -  |^|
```

Wiki

The final result looks like this:

Culture	Primitive	Developed
Mesolithic Age		
Fire	-	X
Agriculture		-
Neolithic Age		
Fire	X	X
Agriculture	-	

Fig. 12.2

12.2.2 Tables in HTML

You can also use plain HTML notation to make your tables. Here is a simple example:

```
<table border="1">
  <tr>
    <th> Mesolithic Age </th>
    <th> Neolithic Age </th>
  </tr><tr>
    <td> flint </td>
    <td> metal working </td>
  </tr><tr>
    <td> settlement </td>
    <td> agriculture </td>
  </tr>
</table>
```

Wiki

Here is the result:

Mesolithic Age	Neolithic Age
flint	metal working
settlement	agriculture

Fig. 12.3

As you can see, the automatic formatting found in tables employing wiki syntax is not used in this case. This means that you can determine the look of the table regardless of the skin of the TWiki, which, of course, is coupled with a certain degree of additional effort.

12.2.3 Tables with the <verbatim> Tag

The `<verbatim>` tag mentioned above is not only handy when formatting programming code; it can also be of assistance when generating simple tables.

Wiki

```
<verbatim>
  Culture    Primitive  Developed
  ------    --------  ----------
  Meso      wood      fire
</verbatim>
```

The text between the tags is displayed on a 1:1 basis. This notation can be useful if, for instance, the table is to be exported to another program as ASCII text.

12.3 Links

CamelCase

In contrast to MediaWiki, we can work with the famous, wiki-specific CamelCase in TWiki. Words written together with their initial letters capitalized, such as `SponsorContacts`, will generate a link to a new page called *SponsorContacts*. It is tagged with a question mark, indicating that the page has not yet been created. Of course, the link is also generated if the topic does already exist. If you wish to link to a topic in another web, you must first specify the web before the topic name: *OtherWeb.SponsorContacts*.

To prevent automatic linking of a WikiWord, place it between the `<noautolink> </noautolink>` tags, or add an exclamation point or `<nop>` without a space in front of the word.

Note: `<noautolink>` also works with entire tables. Make sure, however, that there is a space between the table end and the `</noautolink>` (this is a TWiki bug).

External links

Of course, you can also link to pages outside of the TWiki. The following Internet services are automatically recognized and linked accordingly: `http://, https://, ftp://, gopher://, news://, file://, telnet://, mailto://.`

Alternative with brackets

If you do not like CamelCase, but prefer notation using brackets, you can also use this method to produce links: The link `[[History of humankind]]` will still generate a page with a CamelCase name: *HistoryOfHumankind*. The advantage to notation with brackets is that you can separate the link text from the link reference; i.e. while, in the case of `[[Main.HistoryOf Humankind]]`, you could have this complicated phrase appear in the middle of your article without the brackets, if you use `[[MyWeb.HistoryOfHumankind] [history of humankind]]`, your link, appearing as "history of humankind," can fit better into the flow of the text. In the case of external links, you do not even have to separate the link reference and link text with brackets. A simple `[[mailto://tim@troglodyte.org Tim]]` will suffice to only have "Tim" displayed on the page.

Note: In order to be able to display square brackets, which normally constitute a link, you have to mask them with an exclamation point before the phrase.

Jump labels

For very long pages, it is a good idea to set jump labels at various points on the page containing internal page links. To do this, define an anchor by placing the anchor name after a hash mark, e.g. `#Summary` at the beginning of the line to which you want the browser to jump. As you can imagine, the anchor name should also be a WikiWord. To then set a link to a certain jump label on a page, just add the anchor name with the hash to the page name:

```
[[HistoryOfHumankind#Summary]]
```

If you link to a jump label on the current page, you do not need to add the page name.

12.4 Integrating Images

You have already indirectly seen how to integrate images in the text of your article in Chapter 11.2.3. However, let us review it once again briefly: To integrate an image, you first have to attach it to your page using the **Attach** function. In the options on the dialogue

page, you can then determine that a link to the attachment be included in the text. If you are integrating a graphic image, this means that it will be visible in the page view mode at the end of the article; the following line would be added to the source text:

Wiki
```
<img src="%ATTACHURLPATH%/image.gif">
```

To place the image in another spot on the page, put the line in the corresponding spot in the source text. By the way, you do not necessarily have to use the link function to integrate an image: You can also manually add the above line to the source text. As an alternative, the following expression also works:

Wiki
```
%ATTACHURL%/image.gif
```

Images attached to other pages or even other webs can be accessed via

Wiki
```
%PUBURL%/webname/pagename/image.gif
```

12.5 HTML in TWiki

All HTML tags can be used in TWiki. However, in a cooperative project, you should refrain from doing so when possible and use TWiki formats instead to keep the source text easy to follow and thus easier to edit. In addition, uniformity in formatting is no longer ensured if you employ HTML. Remember, too, that the text, which is surrounded by tags, is still interpreted by TWiki and converted by it to HTML anyway. This could lead to unwanted side effects.

If, however, you do use HTML, you must follow a few rules:

- Keep to the standards of HTML 4.0 and XHTML 1.0 as far as possible, since you would otherwise lose compatibility with various browsers.
- Refrain from stretching an HTML tag over several lines. The greater/less-than signs that surround the tag must be on the same line.
- Remove all blank lines, since TWiki would otherwise make new paragraphs out of them, which could pose problems in certain areas, such as in tables.

12.6 TWiki and JavaScript

Employing JavaScript with your TWiki application is practically effortless. To circumvent TWiki formatting, the JavaScript code should be masked with an HTML comment and `<pre>`-tags. Here is an example:


```
<script type="text/javascript">
  <!-- //hide JavaScript
    //<pre> suppress TWiki formatting
      window.alert("JavaScript is active");
    //</pre>
  // -->
</script>
```

Every time the page is accessed, a message box appears indicating that JavaScript is active.

12.7 TWiki Variables

TWiki variables fulfill two functions. *Firstly*, they stand for certain data that is only employed when the page is invoked and the source text is displayed in its normal form. For example, the placeholder `%DATE%` displays the current date. This can save the author a great deal of effort, and it makes the page more dynamic. *Secondly*, a few variables have a switch function. They enable several settings to be determined in TWiki. For instance, the `%NOAUTOLINK%` variable determines whether CamelCase is active or not.

You can identify a variable by the fact that its name is enclosed in percent signs; depending on the variable, comments can be inserted in curly brackets after the name, e.g. for an embedded search for `%SEARCH{"fire"}%`. There are pre-defined variables that are set when the TWiki system is installed or that access information from the server. Others can be set by the user. Here, we differentiate between the following options with regard to the area of validity of the variables:

- Variables applicable to the entire TWiki. They can be defined and edited on the page *TWikiPreferences* in the *TWiki* web.
- Variables set for a certain web. They can be found in the respective *WebPreferences* page.

- Variables pertaining to individual topics. They are inserted directly into the source text of that page.
- Variables that the individual user can determine. They are found on the respective user pages.

The variables of the lower levels overwrite the values set in the superordinate instances. If you wish to suppress this for certain placeholders, you must assign them to `%FINALPREFERENCES%`. This is a variable that prevents settings outside of the current level from being altered. For the entire TWiki, the code looks like this:

Wiki

```
···*·SET FINALPREFERENCES = PREVIEWBGIMAGE, ↵
WIKITOOLNAME, WIKIWEBMASTER, SMTPMAILHOST, ↵
SMTPSENDERHOST, ALLOWWEBMANAGE, ↵
READTOPICPREFS, TOPICOVERRIDESUSER
```

These settings can only be made at *TWikiPreferences*.

The following lists a few of the many variables available. You can find a detailed list on the page *TWikiVariables* of the *TWiki* web.

Pre-defined variables

- **`%WEB%`**: names the current web.
- **`%TOPIC%`**: names the current page.
- **`%INCLUDE{"ATopic"}%`**: includes the topic of another web.
- **`%INCLUDINGTOPIC%`**: cites the name of the page that integrates the current topic.
- **`%INCLUDINGWEB%`**: cites the name of the web into which the current topic is integrated.
- **`%SEARCH{"searchtext"}%`**: the search result is integrated into the page text.
- **`%TOC%`**: generates a table of contents for the page. The entries are composed of the topic headings.
- **`%WIKIUSERNAME%`**: cites the name of the user currently logged in.

Preferences variables

Now we will discuss a few variables whose content you can either edit on the TWiki, web, topic or user level, or across all levels. The most important settings are *TWikiPreferences* and the web-specific *WebPreferences*:

- **`%WIKILOGOIMG%`**: found in *TWikiPreferences*. Determines the image displayed in the upper left portion of the TWiki.

- **%WEBBGCOLOR%**: sets in *WebPreferenes* which background color certain elements of the page will have.
- **%WEBCOPYRIGHT%**: determines the copyright text at the foot of the page on the individual web level.
- **%EDITBOXHEIGHT%** and **-WIDTH%:** with this, users can set the size of the edit box on their page.
- **%RED%:** determines what color value in hexadecimal notation "red" is to have. This of course also applies to all other colors.

Changing values

A set variable generally has the following structure:

Wiki

```
···*·Set VARIABLENNAME = value
```

These definitions are often located on a lower level of a list. Depending on the variable, the corresponding value can be set by simply editing a page and changing the `value` to suit your needs. Usually there are explanatory notes on the page to assist you. To avoid redundancy, we will only discuss individual variables at the point where they are employed. This especially refers to Chapters 14 and 16.

In the lines below and above the settings, you can use normal Wiki code; only the sections having the form displayed above are recognized and interpreted as value settings.

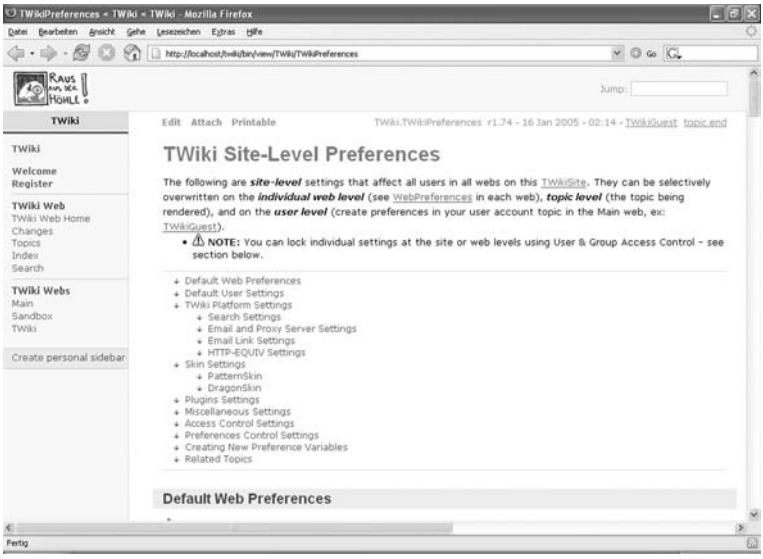


Fig. 12.4

Tip: Often the parameters on the page will disturb the overall look of the page and you may want to hide them. To do so, you can utilize HTML comments (`<!--` and `-->`) that you can insert in the lines above and below the parameters. They will still be visible in your browser's source text view mode.

Creating your own variables

It is also possible for users to create and assign their own placeholders by placing them in the respective spots using


```
···*·Set OWNVARIABLE = Test
```

Depending on what area the variable is to affect, set the value either in *TWikiPreferences* (for the entire Wiki), *WebPreferences* (for the respective web), the user page (for the corresponding user) or directly in the topic (for that topic only). If you now use the variable `%EIGENEVARIABLE%` in the source text, it will be replaced in the page view by the word "Test."

Note: To mask a variable, it is best to use the procedure described above in which an exclamation point is set before the variable.

13 Searching in TWiki

Readers
Authors
WikiAdmins
WebAdmins

A well-developed wiki is a sizeable database. However, it is largely filled with freely generated text, i.e. the data is not present in a structured form and thus not divided into various types and individually labeled. This results in the problem that stored information is hard to find. One possibility is to consistently classify articles right from the start and link them such that a type of catalogue is created in which one can click through subject headings and subheadings until one has arrived at the desired spot. However, this would mean additional effort that would not be able to be performed consistently when editing wars heat up. So, in order to get a handle on the (dis)order in the wiki, the search function is ultimately the best, if not only, opportunity.

13.1 The Search Function

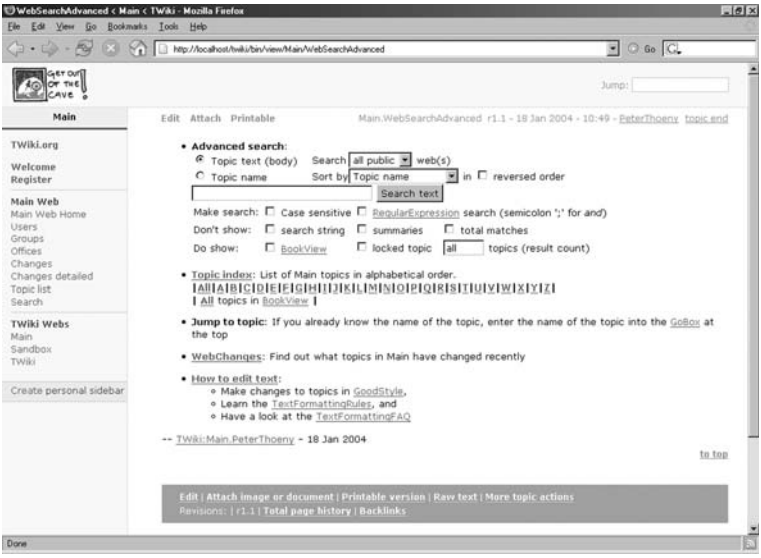
You will find the search function under "Search" in the navigation sidebar. When you click on it, the search form already discussed in Chapter 11.2.1 will appear.

If you search for, let's say, "leftbar," in all webs (the search is not case sensitive), you will get a list of results. The results are categorized according to webs. Per hit, you will see the title of the topic, the date of the last edit and the name of the user who last edited it. In addition, you will see the first lines of the source text of the article. At the bottom of the list, the number of topics found is indicated.

Advanced search

The advanced search offers more possibilities for displaying the result list and limiting the search. It can be found under *Advanced Search* or in the topic *Main.WebSearchAdvanced*.

Fig. 13.1



Here, you will find a wealth of additional settings:

- **Search … web(s)**. Here you can select which web should be searched.
- **Sort by …** The results can be sorted by topic name, last modified time, or last editor. If you select "in reversed order," the order is reversed.
- **Make Search …** If you would like the search to be case sensitive, put a check in front of the box for "case sensitive." With "regex," or "regular expression," you inform the system that the series of characters you type in is to be interpreted as a regular expression. You can find more information in Chapter 13.3.
- **Don't show …** Here, you can determine what information is to be displayed in the results list. A check in front of "search string" suppresses the display of the search item before the list, "total matches" the number of results. The most interesting option is probably that of hiding the "summaries" content lines and thus enabling a clearer overview.
- **Do show …** "locked topic" shows whether an article has just been locked. If, in "all topics," you replace "all" with a number, only that number of results per web will be displayed. With "book view," not only the titles of the articles found, but also their content, is shown. This name was chosen because the function can be used to print out collections of articles.

13.2 Effective Searching

Two factors are crucial to the quality of a search result. The first is the question as to how many of the relevant documents are found and how many remain in the dark. This value is called the recall. The second factor is how many documents are found that are not really desired and are left as "data trash." This value is called precision[1]. The selection of search items can considerably influence the quality of the search. The best types of words are those that most clearly describe the desired subject, yet still occur relatively frequently. If a word is used too seldomly, you will only find a small portion of possible results. If the word is too frequent, too many documents will be found, and the result will be virtually useless. To avoid the latter, there is a so-called stop word list (`SEARCHSTOPWORDS`) located in *TWikiPreferences*. All words on that list are simply ignored during a search.

Search method

The search method normally employed is based on simple comparison. That means all pages are displayed that in some way contain the word without sensitivity to case. This also applies to word fragments. If you enter more than one word, all words must be present in the document. The order is of no significance. Possible operations have already been covered in Chapter 11.2.1.

This search mode does not accommodate the display of various search items as alternatives (but see below). Note, too, that the search program does not perform any replacements. The search for "leftbar" results in several hits, but a search for the plural, or "leftbars," results in none. This is also an issue with regard to foreign diacritical marks; if, for instance, you search for "Duesseldorf," the German spelling "Düsseldorf" will not be found, and vice versa.

13.3 Searching with Regular Expressions

If the quick search methods mentioned above are not enough, you have another powerful tool in regular expressions (RegEx), to extract information from articles in a very precise manner. Regular expressions actually come from the world of programming, and refer to search patterns compared with a character string. RegEx is a

[1] See Ferber, 2003, Chap. 1.3.7.

complex topic that would fill bookshelves. However, we nevertheless want to try to give you an idea of what they are.

Boolean operations

In a RegEx search, there is also the option of using Boolean operations:

Tab. 13.1

Expression	Result
`Term1;Term2`	AND operation. Both terms must be present. This operator is specific to TWiki.
`Term1|Term2`	OR operation. At least one of the terms must be found.
`!Term1`	NOT operation. The term entered must not appear in the document.
`(expression)`	Collection of several expressions to make a single unit having processing priority.

There are a few more points to remember. First of all, make sure you do not insert any spaces. They are not ignored in RegEx, but rather processed along with the rest. That means if there is a space before a term, the character string "term" will be searched. Secondly, as the above table shows, words can be collected into a single unit using parentheses; these will be processed first. The order of the expression priority is PARENTHESES before NOT before AND before OR. Thirdly, employing brackets also enables unlimited nesting of expressions. This option facilitates very complex searches.

Examples

A few examples will illustrate the functions described. For instance, the word

RegEx `torch|torches`

is not only searched in the singular, but also in the plural. We can combine them to search for torchlights as well:

RegEx `torch(light|lights)`

Note that the entire term will be searched as one unit. Thus, pages containing just "torch" or just "light" will be ignored. A search for a torchlight from a flaming torch, excluding light from a flashlight, which in British English is called an (electric) torch, might look like this:

RegEx `torch(light|lights);!electric`

You can also make a search take into account differences in orthography is well, as in:

```
neighb(o|ou)r(s)
```

which effects a search for the American "neighbor" as well as the British "neighbour" in singular and plural.

Letter patterns

The real advantage to regular expressions is that you not only can look for words, but also letter patterns. For instance, you can use them to search for email addresses or telephone numbers in TWiki. The basic unit of such patterns is information on permissible letters:

Tab. 13.2

Expression	Significance
`[acf]`	One of the letters indicated.
`[a-g]`	A letter between (and including) `a` and `g`.
`[^a]`	All letters except the one indicated.
`.`	any character.
`[\-]`	The minus sign. The backslash functions as a mask, indicating that the following character is not a function (as in the normal function of a backslash as an area sign).

Instead of letters, you can also use numbers or other characters. If such characters have a special function in RegEx, you must "disarm" them with a preceding backslash. Then, for instance, you can search for links to films and audio files in TWiki:

```
\.mp[g3]
```

Determine frequency

It gets even more interesting when you can also indicate how often a certain character pattern may occur in a row:

Tab. 13.3

Character	Frequency
`*`	Never, once or more than once
`+`	At least once
`?`	At most once
{*n*}	Exactly *n* times
{*n*,}	At least *n* times
{*n*,*m*}	Between *n* and *m* times

The characters indicated are placed behind the expression whose size is to be determined. Now we can express a search for singular and plural words in the following way:

RegEx

```
Page(s)?
Box(es)?
```

URL search

You can find a normal URL with the following pattern:

URL

```
http://[a-zA-Z0-9._%-]+\.[a-zA-Z]{2,4}
```

A character string is searched beginning with http://, followed by an arbitrary number of alphanumerical characters and numbers and ending with a period or a series of two to four letters.

Email address search

The next regular expression search concerns email addresses, and works in a very similar way:

RegEx

```
[a-zA-Z0-9._%-]+@[a-zA-Z0-9._%-]+\.
[a-zA-Z]{2,4}
```

Remember that a search via RegEx always produces the entire article containing the respective search pattern. We will see regular expressions again in Chapter IV.5.2 when we discuss the processing of standardized data.

13.4 Embedded Searches

A sophisticated search, such as that for email addresses, may need to be performed more than once. Furthermore, the user may want the result displayed on a separate page. This can be achieved with an embedded search, which produces a freely formattable hit list in any location in a topic. Thus one can create free and dynamic pages, since they are generated from the version of the TWiki existing at any respective point in time.

Parameters

The embedded search is invoked with the variable `%SEARCH%`, which accepts a series of parameters:

Tab. 13.4

Parameter	Function
`"text"`	Search string in quotation marks.
`web="..."`	Name of the web to be searched. Several webs are separated by commas. If you want to search all webs, enter “all”.
`topic="..."`	Only the topics indicated are searched. Here, too, you can indicate more than one, separating them by commas. In addition, you can use

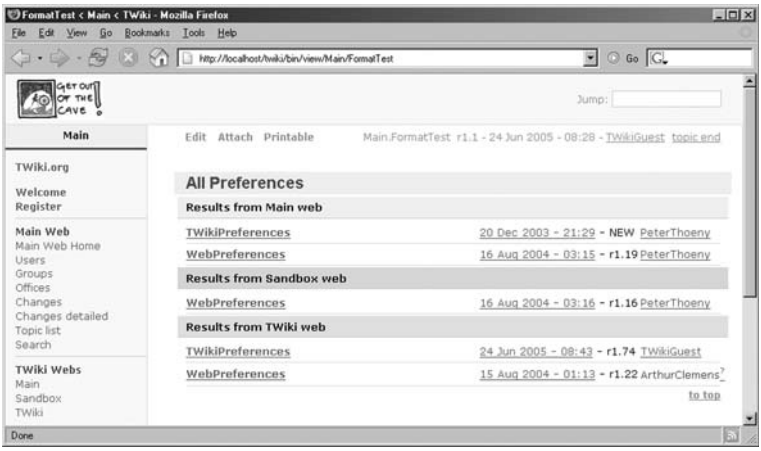
	the star as a placeholder for an arbitrary number of characters.
`excludetopic="..."`	The opposite of `topic`. The topics indicated are not searched.
`type="..."`	■ **keyword**. Normal search. ■ **regex**. Search with regular expressions.
`scope="..."`	Searches ■ **topic**. Only the title. ■ **text**. Only the content. ■ **all**. Everything is included.
`order="..."`	Results displayed in order of ■ **topic**. The title. ■ **created**. The date of creation. ■ **modified**. The date of the last edit. ■ **editby**. The last author.
`limit="..."`	Limits the number of results displayed.
`reverse="on"`	The results are shown in the reverse order.
`casesensitive="on"`	For a case sensitive search.
`nosummary="on"`	Only the titles of the resulting articles are shown.
`bookview="on"`	The entire text of a topic is shown.
`nosearch="on"`	The search string is not included in the result display.
`noheader="on"`	The search title is not displayed.
`nototal="on"`	The number of results is not displayed.

So, let us assume the gods want an overview of all settings in the wiki in addition to fast access to all *Preferences* pages. Since new projects and new webs are continually being added, a static page for this purpose would not make much sense. Thus, we generate a page called *PreferencesOverview* and put in a title and the following search:


```
---++ All Preferences
%SEARCH{"Preferences" scope="topic"          ↵
web="all" nosummary="on" nosearch="on"       ↵
nototal="on"}%
```

Thus, all pages in all webs are displayed that have “Preferences” in their title. Special information is excluded, with the exception of categorization into webs. If that, too, is annoying, it can also be suppressed using `noheader="on"`.

Fig. 13.2



Format result

And yet, the integration of the search into a page can go even farther. Using formatted searches, you can determine which information is shown on a page. This can even be taken to the point that you only have certain passages displayed from the pages found. This can be done with the parameter `format`, in which the desired display can be determined. If, for example, you want to display the above list of Preferences pages in a list, just include the `format` parameter:

Wiki

```
format="···*·$web: $web.$topic "
```

The web and the link to the topic are displayed. Note that the header is now automatically suppressed. This function can be individually set (see below). As you can see, the web and the topic title were replaced by variables that are appropriately filled in by the system. There is a whole list of such variables; we have included a few in Tab. 13.5.

Tab. 13.5

Variable	Function
`$web`	Name of the web in which the topic found is located.
`$topic`	Name of the topic found. Using the formalism `$topic(`*n*`, ...)`, the first *n* characters of the topic title are displayed followed by three

	periods. Instead of the three periods, you can have any series of characters displayed, e.g. . They will be inserted after *n* characters in the topic name.
$parent	Name of the parent topic. Formatting as in $topic.
$text	The formatted text of the topic.
$rev	Number of the last revision of the topic.
$date	Date of the last edit.
$wikiname	Name of the last author.
$createdate, $createwikiname	Date and author of the first version of the topic.
$summary	The first 162 characters of the topic as a summary.
$pattern (reg-ex)	In this spot, a regular expression may be employed to extract a certain text from the topic found.
$percnt, $dollar, $n	The percent sign, dollar sign, etc. and new line.

The following search string displays all skin files, their web and parent element in a table:

Wiki

```
| *Name* | *Parent* |
%SEARCH{ "Skin" scope="topic" web="all"      ↵
nosummary="on" nosearch="on" nototal="on"    ↵
header = "| *$web* ||"                       ↵
format="| $topic | $parent |"}%
```

The header of the table is first entered manually with two columns. The search result will also be entered in two columns. One blank line spanning both columns is issued per web. This is indicated in the header parameter.

Display individual results

Using a formatted search, data can be easily extracted from articles. This can be done with regular expressions that are processed with the variable $pattern. To be precise, the variable contains the text that corresponds to a certain expression. For example, to find the email addresses in TWiki, and only have these displayed, all you have to do is use the pattern cited above for email addresses. Since we are searching all topics containing these addresses, and at the same time wish to extract these addresses from the topics, the regular expression must occur in the search and as a pattern. So don't be shocked – it looks worse than it actually is. For simplicity's sake,

we have limited the possible addresses to small letters without special characters.

Wiki

```
%SEARCH{
 "[a-z.]+@[a-z]+\.[a-z]{2,4}"                    ↵
 scope="text" nosearch="on" nototal="on"         ↵
 web="all"                                       ↵
 regex="on" separator=", " multiple="on"         ↵
 format="                                        ↵
 $pattern(.*?([a-z.]+@[a-z]+\.[a-z]{2,4}).*)     ↵
 "                                               ↵
}%
```

As you can see, three parameters have been added. The parameter `regex` activates the search with regular expressions, and `separator` indicates the symbol with which the individual results are separated. The parameter `multiple` makes sure that a page is processed more than once if more than one result for the search pattern is detected. Otherwise, only one email address per page would be found.

Search patterns

The additional characters inserted before and after the expression in `$pattern` are required for internal processing. The expression `.*?` results in the first and not the last result found being used. The text in the inner parenthesis is utilized. It often happens that one wants a display of text located between two known characters. There is a fixed regular expression for this purpose:

RegEx

```
$pattern(.*?(from here.*?to here).*)
```

The starting and end point are included. If the end of the line is to be the end point, add the expression `[^\n\r]+`. This type of search is often used to display data stored in lists, e.g. on the user pages. The expression for this is as follows:

RegEx

```
$pattern(.*?\*.*?label\s*([^\n\r]+).*)
```

As you can see, the position of the interior parentheses has changed. This results in a display of only that which comes after the label. The expression `\s*` stands for any number of spaces.

Nested search

There is also the possibility of performing a further search within the search results. In principle, the internal search is indicated as a part of the parameter `format`. However, it must be reformatted so that the command is not executed too soon:

- Replace the leading `%` with `$percnt`.
- Mask the quotation marks: `\"`.
- Replace all `$` with `$dollar`.
- Deactivate the final sequence: `}$nop%`.

A search of all pages that link to pages containing the word “fire” would thus look like this:

```
%SEARCH{ "fire"                                    ↵
  nosearch="on" nototal="on"                       ↵
  format="···*·$topic:$n·····*·                    ↵
  $percntSEARCH{ \"$topic\"                        ↵
    format=\"$dollartopic\" nosearch=\"on\"        ↵
    nototal=\"on\" separator=\", \"                ↵
  }$nop%"                                          ↵
}%
```

Wiki

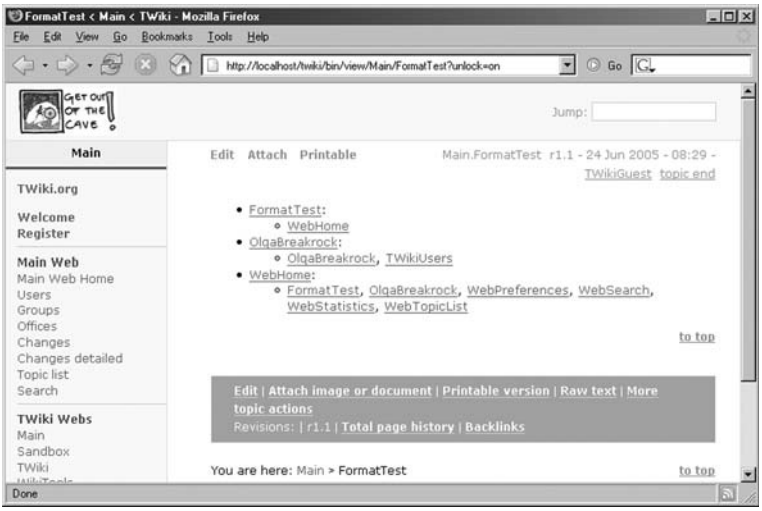
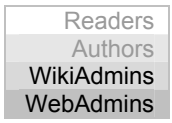


Fig. 13.3

Search requests can be nested in up to sixteen levels – but don't worry: We neither have an example of this nor any conceivable application.

14 User Permissions in TWiki



Up to now, anyone has had access to all pages. This is not always desirable, especially for project work where there is urgent need for a space for internal meetings and the exchange of information. TWiki distinguishes itself from other wikis in that it possesses a relatively reasonable user administration system with which access to certain pages can be specifically controlled. This is not yet activated in the standard installation, but can be activated in a few simple steps.

Caution: For user administration, TWiki requires the additional Perl module Digest::SHA1. It is not supplied with the standard issue of Cygwin-Perl. Its installation is described in Appendix A.

14.1 Authentication vs. Access Control

Concepts for user administration

Generally, there are two concepts when working with users: authentication and access control. The former means determining who is currently in the system. This requires a registration by name, and a login. Users are then identified, and actions they undertake in the wiki are associated with their name. However, authentication does not necessarily mean that the system is closed to the public. For one thing, you can offer and promote guest access that anyone can use to log in to test the wiki. In the standard installation of TWiki, this is done via *TWikiGuest* with the password *guest*. On the other hand, you can give all visitors of the site the option of registering as a user in the system. In such a case, there is a publicly accessible registration page with which users can obtain a user name and password.

Access control

If a user is known to the system, you can allow or prohibit him or her from doing certain things. This is what is meant by access control. For example, some areas are blocked to normal wiki users such that they cannot even see them (e.g. preferences pages). Or, you can limit the editing of pages affecting the public (e.g. the title page) to a certain circle of users.

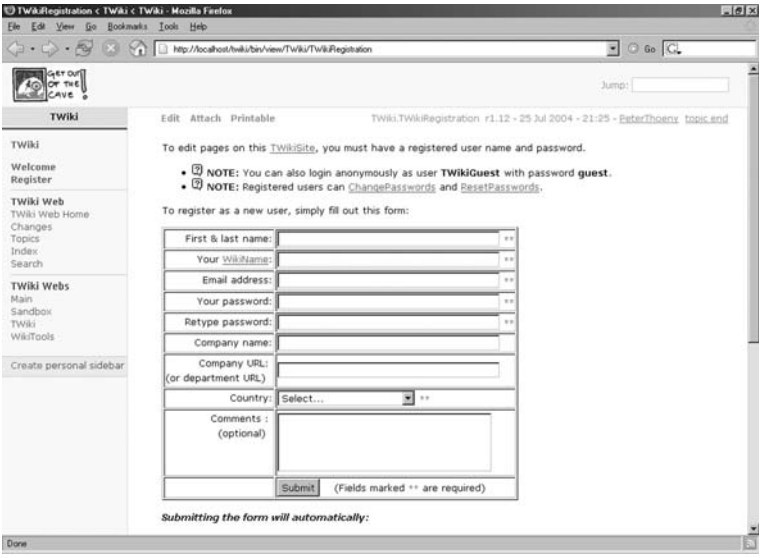
As you see, there are a variety of possible combinations of user and access controls. We will first have a look at how to activate and set authentication and access control. Subsequently, we will give some thought to access models and their sense or nonsense in the wiki.

14.2 Activating User Registration

Registration page

Before you activate the authentication, it is advisable to set up your own user account and register as an administrator. To do this, go to the page *Twiki.TWikiRegistration*.

Fig. 14.1



Caution: You should see a form here containing, among other things, a box to enter your password. If this is not the case, you have accessed the registration page for Intranet users. If so, move that page to *TWikiRegistrationIntranet*. Search for the page *TWikiRegistrationPub* and move it to *TWikiRegistration*.

It is a general convention in TWiki to use real names. If you do not wish to do so, you are of course free to select a pseudonym.

You can find out if the registration has been successful by looking at the page *Main.TWikiUsers*. Your name should now be on the list.

Note: If you have not set up the email function (see Chapter 15.5), you may receive an error message telling you that the confirmation email could not be sent. However, this message will not interfere with the registration process.

Defining administrators

Now, make yourself an administrator. To do so, access the topic *TWikiAdminGroup*. Alter the line in which the members of the group are indicated by entering your name:


```
···*·Set GROUP = YourUserName
```

Several users are categorized in a single group which then all have the same TWiki permissions. You can find out more about groups in Chapter 14.4.

You can also modify the template with which the user page is generated. It is saved in the topic *TWiki.NewUserTemplate* and can be edited like any normal page.

14.3 Setting Up Password Protection

Registration works via an authentication mechanism of the web server. It is set up such that access to a few of the TWiki scripts is denied without a login. The file responsible for this is *.htaccess* in the *twiki/bin* directory. In the standard installation, it does not yet have a *.txt* extension and is thus not yet activated.

Adapting .htaccess

You should first look at the file and make a few small changes. In the line beginning with `AuthUserFile`, replace `!FILE_path_to_TWiki!` with the path leading to the *data* directory of your TWiki, the easiest being `../data`. You can alter the message appearing during login in the following line, which begins with `AuthName`. If you still wish to keep your TWiki as open as possible, you should also enter login data for guest access here. Under `ErrorDocument`, replace `!URL_path_to_TWiki!` with the path in the URL of your TWiki (the section following the host name) that leads to the *Oops*-script. This script, located in the *twiki/bin* directory, is responsible for issuing TWiki error messages.

In the following lines, scripts are listed that from now on will require a valid user name. You should not change them, since a few interdependencies exist (e.g. *edit* and *save* must have the same authorizations). Change the name of *.htaccess.txt* to *.htaccess*. If you then attempt to edit a page in TWiki, you should receive a request to log in. Log in, perform a trial edit, and save it. If it works, you have successfully activated authentication.

Note: You should definitely copy the topic *ChangePassword* from the *TWiki* web to the *Main* web, so that registered users can change their passwords directly via their user page. Otherwise, this link will lead from the user page to nothing, which can be very confusing.

14.4 Access Control

Access to the files in TWiki can be controlled very precisely. The users are identified by their user names, which are known to the system via authentication. Since it would be very cumbersome to continually list all authorized users for every action, we can categorize users into groups. In keeping with TWiki logic, these, in turn, are also only topics in which the corresponding users are listed. A list of all existing groups can be found under *TWikiGroups*. Here, in a new system, you should only see the *NobodyGroup* and the *TWikiAdminGroup*.

NobodyGroup

The first group has no members, and for good reason: Using it, you can completely prohibit certain actions that are potentially hazardous (so that nobody has permission to execute them). Such actions include, for instance, renaming *TWikiPreferences* or *TWikiAdminGroup*. As you will see, this does not offer complete protection, since these permissions may be changed by anyone having editing rights to a page. However, it is an effective means of protection against accidental actions.

TWikiAdmin Group

The second group is the Administrator Group. By default, admins have editing permissions for all pages and may generate new webs. Of course, this too, can be changed. The members of a group are defined with the variable `GROUP`. In the standard installation, you will see that the developers of TWiki are still listed as admins. You will also see that several users are simply written one after the other, delimited by commas. Your user name should definitely be here as well, if you are responsible for the TWiki.

Creating new groups

The easiest way to add a new group is by filling out the form on the *TWikiGroups* page. Here, you just have to enter a fitting name. Make sure that the name ends with "Group." A new page is generated that already contains all of the elements the group needs. Importantly, you are already listed as a registered user in the `GROUP` variable.

You will also see a second variable here, `ALLOWTOPICCHANGE`. This determines who can modify entries on a page. Admittedly, it would be rather ridiculous to limit certain actions to the Admin Group while at the same time allowing everyone the opportunity to make themselves admins.

Allow vs. deny

This brings us to the access authorizations. There are two ways to define them. Either you explicitly define who may not execute an action, which makes it accessible to all the rest. This is an open security strategy with which you can exclude certain persons who, for example, have become conspicuous due to vandalism. The other method is to indicate who may execute certain actions. This is useful in actions critical to security, where you would like the circle of authorized persons is to be known by name. The two key words are "DENY" for the exclusive strategy and "ALLOW" for the inclusive approach.

Protecting individual pages

The two most frequent limitations relate to changing and renaming topics. The corresponding variables are:


```
···*·Set ALLOWTOPICCHANGE = User, Groups
···*·Set DENYTOPICCHANGE = User, Groups
···*·Set ALLOWTOPICRENAME = User, Groups
···*·Set DENYTOPICRENAME = User, Groups
```

Of course, you only have to make the entries that you need. If you only wish to enable inclusive access, you do not need to make any `DENY` entries. The first two variables relate to the editing of a topic and also include the option of attaching files. Thus, users prohibited from editing a topic may also not upload anything to that topic. The second pair of variables relates to renaming or moving a topic.

Hiding access variables

If you do not want to make it possible for everyone to immediately see who has access to a page and who does not, you can place the settings in HTML comments that begin in the line above the variable settings and end in the line below it. By doing this, you won't prevent users experienced in HTML source text from reading and nevertheless finding out this information. However, it is also a question of aesthetics to refrain from making such technical information immediately obvious to everyone in normal browser viewing.

Limiting viewing access

You can also limit viewing access to an article. The respective variables are `ALLOWTOPICVIEW` and `DENYTOPICVIEW`. The TWiki documentation describes this type of access control as "insecure;" however, we have been unable to reproduce the objections listed there.

Combining strategies

`DENY` and `ALLOW` can also be combined to exclude individual users of a group for a certain topic. Let us assume that the user *TimTroglodyte* is a member of the group *StoneAgeGroup*. However, recently, this user has conducted himself in a rude manner on a certain page that he is obviously attached to. The community decides to block him temporarily. This is done by setting the following permissions:


```
···*·Set DENYTOPICCHANGE = TimTroglodyte
···*·Set ALLOWTOPICCHANGE = StoneAgeGroup
```

This does not change anything for the other members of the group, and you do not necessarily have to change the group structure to define permissions for a particular topic. Principally, the cross section of both settings is permitted.

Caution: In the default settings, members of the *TWikiAdminGroup* have access everywhere (see below). If you wish to test the access controls, you should do so with users who are not in that group.

Web-level access control

Regulating access for each individual topic would not only take considerable effort during generation, but it would also easily lead to inconsistent settings if several people worked on them. Thus, there is the opportunity to set web-level controls that define which actions may be performed in a particular web. This is done via the respective *WebPreferences*. The corresponding variables for editing, renaming and viewing are:


```
···*·Set ALLOWWEBCHANGE = User, Groups
···*·Set DENYWEBCHANGE = User, Groups
···*·Set ALLOWWEBRENAME = User, Groups
···*·Set DENYWEBRENAME = User, Groups
···*·Set ALLOWWEBVIEW = User, Groups
···*·Set DENYWEBVIEW = User, Groups
```

If you wish to protect your web from unauthorized viewing, it is also a good idea to set the parameter

```
···*·Set NOSEARCHALL = on
```
Wiki

Otherwise, the web will be included in a full-text search. With regard to viewing restrictions, remember that a search requires registration as soon as blocked content is to be displayed. If a visitor does not have a user name, he or she cannot use the search in this case. The same applies to viewing topics and editing.

If you decide to protect a web from unauthorized access, you should also define the permissions of the preferences page. This is done, as usual, with the parameter `ALLOWTOPICCHANGE`, since it pertains to a single page.

Priorities in issuing permissions

This also means that a priority list must be generated for the event that topic and web settings are different. Normally, topic settings have priority. Thus, as you can see, decisions regarding access to individual topics are in the hands of the respective authors.

Wiki-wide permissions

The only setting that pertains to the entire TWiki is the permission to generate new webs. It is defined via *TWikiPreferences*. If you prefer not to have access limits in your TWiki, you should remove the following entries:

```
···*·Set ALLOWWEBMANAGE = User, Groups
···*·Set DENYWEBMANAGE = User, Groups
```
Wiki

since you otherwise will not be able to add new webs.

SuperAdmin Group

Now, there is still one serious security problem: An inexperienced user who tries to change settings changes the `ALLOWTOPICCHANGE` for a topic. In doing so, he makes a typing mistake or, if acting malevolently, knowingly enters a non-existent user. From then on, nobody can edit the page! One would have to lift the password protection temporarily in order to manually fix the problem. However, this would lead to considerable problems for users logged in at the time, because their permission verification only works through authenticated pages. To avoid such problems, a SuperAdmin Group can be set up that always has edit permissions for all topics. This group is activated by default and can be set up at *TWikiAdminGroup*. If you do not want such a group, or would like to grant another group these rights, you need to change the file *Twiki.cfg* in the directory *lib*. The two relevant parameters are

```
$superAdminGroup = "TWikiAdminGroup"
$doSuperAdminGroup = "1"
```
CFG

With the first parameter, you define the name of the Super group. The second is a switch. If it has a value of 1, the Super feature is activated; if it is 0, then it is switched off.

14.5 Strategies for User Rights

At this point, we should briefly contemplate possible access control strategies and their implications. Generally, the whole appeal of a wiki lies in the fact that there are relatively few hurdles regarding access and editing. Access control is diametrically opposed to this concept. On the other hand, in TWiki, we are confronted with the fact that many system settings are self-defined and can be modified. It is understandable that you may not want these pages to be edited by just anyone, because the rest of the community would have a lot at risk (e.g. with regard to pages for skins).

Closed system

A completely closed system offers the advantage that you have fairly precise control over who may view and edit content. In addition, there are no guest users, so you can always track which individuals have performed what edits. Closed areas in the Net, however, generally make an unfriendly and thus uninviting impression. Such a wiki would make sense if it only concerns the internal matters of a group. On the other hand, these wikis are often operated on an intranet, such that the public is excluded anyway. General wikis, accessible via a known URL, should at least have an accessible start page that explains the purpose of the wiki.

Viewing permitted, editing upon registration

The default setting of the TWiki user administration is to permit anyone to view pages but require a login to edit them. As we have seen, this creates the opportunity to target individual pages for limitations or grant privileges to certain users/groups. Although in this version, anyone is permitted to edit pages via the guest login, the request to register sends a clear message: “This is a self-contained area; you are intruding into our wiki.” Often, this is the decisive hurdle in participating in a joint project.

Editing partially permitted

Since we have determined above that not every user should be able to edit every page, it would be nice if the TWiki only had a login for those pages that have access limitations. Unfortunately, this mixed strategy is not within the scope of the software. However, there are ways to achieve this.

14.6 A Mixed Strategy for Authentication

In the following, we will consider the question as to how to accomplish having only certain pages require a login to be edited. It would seem obvious to use access controls alone to determine authorization variables. The problem, however, is that web servers only remember user names if an authentication is required. Without a user name, there can be no authentication. Thus, if you remove the *Edit* script from authentication, it cannot determine who is attempting to edit a topic. Therefore, it assumes that editing is taking place via a guest account and will never grant access permission to the protected pages.

Identifying users via IP

TWiki offers a preset mechanism that remembers who has logged in. To activate this, you must go to the file *Twiki.cfg* and set the parameter `$doRememberRemoteLogin` to the value 1. Now TWiki will save your IP address along with your login name in a file when an authenticated script is accessed. The next time any script is accessed via that IP address, the corresponding user name is retrieved from the file and is thus also known to non-authenticated scripts. So, we could conceivably remove the *.htaccess* protection for the *bin* directory. You can also define the storage location for the IP addresses in *Twiki.cfg*. The respective parameter is `$remoteUserFilename` and, by default, leads to the file *remoteusers.txt* in the *data* directory.

Problems

That leaves us with two more problems to solve. *Firstly,* we have to create a login link that triggers a one-time authentication, since the *Edit* script is no longer protected. *Secondly*, permanent storage and authentication via IP is very insecure, since IPs can easily be forged or, as is frequently the case, often change owners.

Login link

The first problem is relatively easy to eliminate. In the *bin* directory, there is a file called *viewauth*, which is an exact copy of the *view* scripts. You can protect the file via *.htaccess*:

CFG

```
<Files "viewauth">
    require valid-user
</Files>
```

You can comment on the entries for all other files with a # at the start of the line. The file *viewauth* now takes care of the login for us, since it is protected and thus prompts the browser for the registration dialogue. From then on, the user is known and is logged in via IP. However, you still have to insert a login link that will invoke the

protected script. The easiest way is to add it to the menu sidebar, that is the *WebLeftBar*:

Wiki

```
[[%SCRIPTURL%/viewauth%SCRIPTSUFFIX%/ ↵
%BASEWEB%/%BASETOPIC%][Login]]
```

This link causes the page currently being viewed to always be loaded with the authenticated script.

Logout

The second problem: Once you are registered as a user with an IP address, you will remain registered for life. This is not only insecure, but also not very practical, since – especially as an administrator – you have to switch users from time to time. A relatively short script can bring relief: To log out, the only thing that needs to be deleted from the corresponding file is the entry of the logged-in user. The script looks like this:

Perl

```
 1 #!/usr/bin/perl -w
 2
 3 # set path for libraries
 4 BEGIN {
 5     if( $ENV{"SCRIPT_FILENAME"} &&
 6         $ENV{"SCRIPT_FILENAME"} =~
 7              /^(.+)\/[^\/]+$/ )
 8     {
 9         chdir $1;
10     }
11     unshift @INC, '.';
12     require 'setlib.cfg';
13 }
14
15 # integrate necessary modules
16 use strict;
17 use CGI::Carp qw(fatalsToBrowser);
18 use CGI;
19 use TWiki;
20
21 my $query = new CGI;
22
23 # determine current user
24 my $theRemoteUser =
25       TWiki::initializeRemoteUser();
26
27 # load user file
28 my $text =
```

```
29       &TWiki::Store::readFile(
30         $TWiki::remoteUserFilename );
31 my @line = split( /\n/, $text );
32
33 # entry of new file
34 my $newtext;
35
36 # find current user in file
37 for (my $i = 0; $i < scalar(@line); $i++)
38 {
39    if (!grep(/\|$theRemoteUser\|/,
40          $line[$i]))
41    {
42         $newtext .= $line[$i]."\n";
43    }
44 }
45
46 # save new file
47 &TWiki::Store::saveFile(
48    $TWiki::remoteUserFilename, $newtext );
49
50 # display confirmation page
51 TWiki::redirect($query,
52    TWiki::getOopsUrl("Main", "WebHome",
53       "oopslogoff", $theRemoteUser));
```

First of all, a few preparations are made that refer to the work relationship (lines 4-13). This is necessary to be able to integrate the libraries in the next step (lines 16-19). The variable `$query` (line 21) will be required later when a confirmation message is displayed. In line 24f., the current user is determined if there is a corresponding entry. Otherwise, the value of `$theRemoteUser` is "guest." Now, the registration file must be loaded and split into individual lines for further processing (lines 28-31). Note that we do not use absolute file names and paths here, but rather the names specified in *Twiki.cfg* (via `$TWiki::remoteUserFilename`). Next, we see the heart of the script (lines 33-44): We put together the text (`$newtext`) of a new file in which all entries are adopted that do not contain the name of the current user (line 39). This text saves the script in the file (lines 47f). Then, we redirect the browser to the display of a logout confirmation. (lines 51-53). When doing so, we make use of the *Oops* scripts that is generally used in TWiki to display messages. The first two parameters of `TWiki::getOopsUrl` indicate the TWiki page to which the OK link leads. The next entry refers to the

template of the confirmation text (see below), yet to be generated. Finally, we transfer the name of the user that has logged out.

Tip: You can achieve direct redirecting to a TWiki page with the command `TWiki::getViewUrl("Web", "Topic")` instead of `TWiki::getOopsUrl`.

Confirmation of the logout

Why is this confirmation so important? We can only delete the server from the server's records. The HTTP authentication, however, works via the browser. It remembers the login data and sends it upon request (that is, in the presence of a protected script) back to the server, which once again enters the data in the remote user file. After our logout, it is possible that the browser could log the user back in unintentionally. That is why it is necessary, on one hand, to display a message explaining this situation and, on the other, to redirect the user to a non-locked page.

Tip: For the Firefox browser, there is the Web Developer, a very good plugin that enables the HTTP authentication to be reset (Miscellaneous -> Clear HTTP Authentication).

One possible display via the template system is printed below. For details on the templates, see Chapter 16.1.3. Save the following file under the name *oopslogoff.tmpl* in the *templates* directory:

CFG

```
%TMPL:INCLUDE{"TWiki"}%
%TMPL:DEF{"titleaction"}%(oops) %TMPL:END%
%TMPL:DEF{"webaction"}%
   *Attention*
%TMPL:END%
%TMPL:DEF{"heading"}%Logout%TMPL:END%
%TMPL:DEF{"message"}%
User %PARAM1% is now logged out.

Be careful! You have to shut down your
browser in order to complete the logout. If
you don't, the browser will log you on again
once you hit an authenticated page.
%TMPL:END%
%TMPL:DEF{"topicaction"}%
  [[%WEB%.%TOPIC%][OK]]
%TMPL:END%
%TMPL:P{"oops"}%
```

The result looks like this:

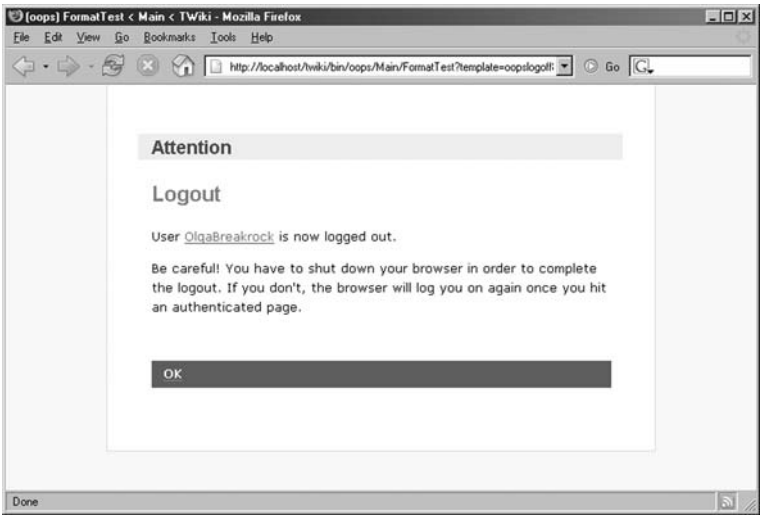


Fig. 14.2

In principle, the texts for ready-made areas of the *oops* messages are defined here.

Tip: In order to make sure that there are no "corpses" in the remote user file, you should delete them at regular intervals, e.g. via CronJob. Note, however, that users who are in the middle of editing procedures may lose their data.

15 Administering TWiki

Readers
Authors
WikiAdmins
WebAdmins

As an administrator, you are responsible for the operation of TWiki; thus, you will have a few duties that do not directly concern the "normal" user. For instance, you are especially responsible for the smooth operation of the system. This includes helping users with password problems, assuring security, and notifying users before any planned big changes.

To obtain admin privileges, you must be a member of the *TWikiAdminGroup* (see Chapter 14.4). You can find a list of the most important sites needed for administration work in the topic *TWiki.AdminTools*.

15.1 Administering Webs

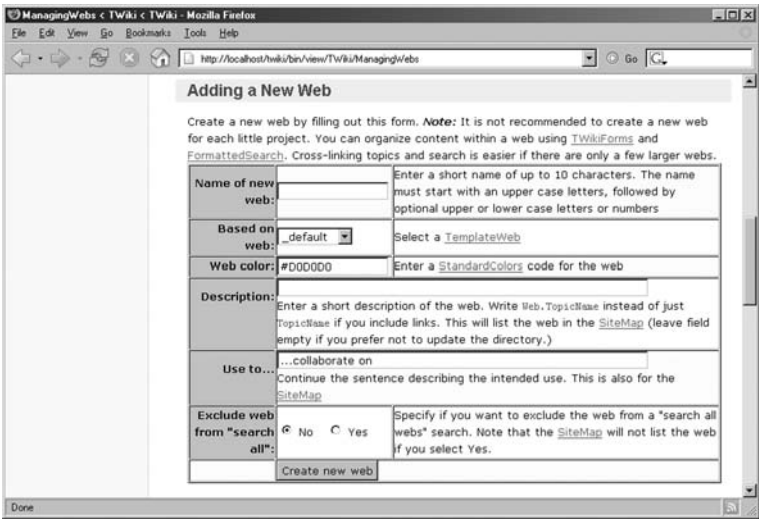
If you have various groups in your TWiki, you cannot avoid providing them with their own webs. However, separation into a variety of areas can also be useful for content purposes, since the individual webs are completely encapsulated. In the standard installation, only one member of the *TWikiAdminGroup* has permission to create new webs and edit existing ones. You can find this setting under `ALLOWWEBMANAGE` in *TWikiPreferences*.

15.1.1 Setup

To create a new web, you only need to fill out the form in *TWiki.ManagingWebs*. It is depicted in Fig. 15.1.

Name

Fig. 15.1



First, enter the name of the new web. In the newest version of the software, a WikiWord can also be used.

Note: This may lead to undesired side effects, since a WikiWord is also interpreted as such. However there are generally no pages in a web having the same name, so the web name is indicated with a question mark. In page templates, you can avoid this by placing a `<nop>` in front of the web name.

Template

Next, you have to select a template web. As the name indicates, this is a template for the web. Normally, you would use the template "_default" which, along with the web, also supplies a basic configuration of pages. These include all pages that start with "Web," thus *WebHome*, *WebChanges*, *WebIndex*, *WebNotify*, *WebPreferences*, *WebSearch*, *WebStatistics* and *WebTopicList*.

Tip: You can also create your own template web. To do this, generate a directory in *data* that starts with an underline, and copy all topics to it that you would like to have in the template.

The other possibility is to build the new web up on an existing one. When this is done, all pages from the old web that start with "Web" are taken over. However, make sure that any attachments are not copied along with the pages.

Color and description

Indication of the color of the web varies depending on the skin. In the *Pattern* skin, the tile of the navigation bar is displayed with the corresponding background color. In the *Classic* skin, you will see

the title bar with the toolbar in the respective color. The content of the two description boxes "Description" and "Used to…" are displayed in the topic *SiteMap*, which is integrated in the start page of the *Main* web, among others.

Search

Finally, you can determine whether the new web is to be searchable with the search function. Generally, you should permit this. However, if the web contains hidden pages, it is better to exclude it from searches, since such a search may access the hidden pages.

Create

If you now click on **Create new web**, the web will be generated, and you can now fill it with content. If you use a skin that accesses `WIKIWEBLIST` (e.g. *Classic*), you have to manually add the new web to this variable in *TWikiPreferences*.

15.1.2 Renaming, Deleting, Archiving

Finding links

There are no automated scripts for renaming or deleting webs. Therefore, the links to the old web must first be updated. Search the individual webs for pages containing the following entries:


```
Oldweb.
Oldweb/
```

Pay attention to the closing period or slash, which cause only the links to the web to be found. You must now change all links found into *Newweb* or, if you wish to delete the web, remove them. This is the most complex part of the job. For the sake of completeness, you should also update the entry `WIKIWEBLIST` in *TWikiPreferences*.

Change files

Then, go to the file level and change or delete those directories containing entries from the old webs. You can find them in *data*, *pub* and *templates* under the name of the respective web. You're done!

Archiving webs

To archive webs, you only need to pack the contents of the respective web directories from *data*, *pub* and *templates* into an archived file. Under Linux and *Cygwin*, you can use the following command:


```
tar czvf                                ↵
archivpfad/archivname.tar.gz            ↵
twikipath/data/web                      ↵
twikipath/templates/web                 ↵
twikipath/pub/web
```

At regular intervals, you should also first archive and then empty the contents of the *Trash* web, into which deleted pages are placed.

15.1.3 RSS Feeds

TWiki also offers the possibility of monitoring recent changes via RSS. This news service is provided by the page *WebRss*, which is available in every web. If you look at that page in TWiki, you will only find a collection of URLs and a few comments. To obtain the actual RSS document, you have to have it displayed in the *Rss* skin. Thus is the URL under which an RSS reader can pick up its information:

URL `*Web*/WebRss?skin=rss`

This feed contains all pages that are also found in *WebChanges*, including a short description. The newest pages and most recent edits are listed first.

Adapted feeds

You can also create an adapted feed that only monitors a selection of pages containing a particular key word. To do this, add to the URL the additional parameter `search`:

URL `*Web*/WebRss?skin=rss&search=*keyword*`

Now, only those edits to pages are shown that can also be found with a normal text and title search under that key word. Of course, it is also possible to generate a page of links to specialized feeds. Such a link can be integrated into a wiki page with the following command:

Wiki
```
[[%SCRIPTURL%/view%SCRIPTSUFFIX%/Web/ ↵
WebRss?skin=rss&search=keyword] ↵
[Feed to keyword]]
```

15.2 Topic Templates

When you generate a new topic, you will see that there is already text on the page, consisting of your user name and the current time. This text is defined in a topic template, and adopted when a new

page is generated. Templates of this kind are generated for various purposes; you can find an overview in Tab. 15.1:

Name	Application
`TWikiGroupTemplate`	Generating new groups.
`NewUserTemplate`	Setup of a new user.
`WebTopicEdit Template`	Generating a new page.
`WebLeftBarPersonal Template`	Setup of the user navigation sidebar.
`TWikiFaqTemplate`	Generating a new entry in *TWiki.TWikiFAQ*.
`ATasteOfTWiki Template`	Template for the slideshow *ATasteOfTWiki* (see Chapter 21.3)

Tab. 15.1

Special variable treatment

Except for the *TWikiGroupTemplate*, all templates are located in the *TWiki* web. Of course, you can alter these sample pages according to your wishes, just like any other wiki page. However, there is a slight difference. The variables you see listed in Tab. 15.2 are not adopted, but are replaced by the values when the page is generated:

Variable	Function
`%DATE%`	Current date.
`%USERNAME%`	Login name via the browser.
`%WIKINAME%`	Wiki user name, e.g. *TestUser*.
`%WIKIUSER NAME%`	User name with indication of web, e.g. *Main.TestUser*.
`%URLPARAM {"name"}%`	Value of the parameter `name` transferred in the URL.
`%NOP%`	Variable is simply removed. Can be used to mask other variables.
`%NOP{...}%`	As above, but spans more than one line.

Tab. 15.2

Personalized templates

You can also use templates for your own purposes, such as to generate uniform agenda pages for every project group. To do so, first create your own template page, e.g. *ProjectAgendaTemplate*. To now create new pages out of the above, you need a form with which the *Edit* is invoked. This might look as follows:

Wiki

```
<form name="ProjAg"                                    ↵
      action="%SCRIPTURLPATH%/                         ↵
              edit%SCRIPTSUFFIX%/                      ↵
              %INTURLENCODE{"%WEB%"}%/">
  <input type="text" name="topic"
```

```
        value="Project#CHANGE#Agenda" />
  <input type="hidden" name="templatetopic" ↵
        value="ProjectAgendaTemplate" />
  <input type="hidden" name="topicparent"   ↵
        value="%TOPIC%" />
  <input type="hidden" name="onlywikiname"  ↵
        value="on" />
  <input type="hidden" name="onlynewtopic"  ↵
        value="on" />
  <input type="submit" value="create" />
</form>
```

The form is depicted here:

Fig. 15.2

As you can see, a name is already suggested to the user, which he or she just needs to adapt. If the user presses **Create**, the page is generated. The possible parameters of the form are explained in Tab. 15.3.

Tab. 15.3

Parameter	Function
`topic`	Name of the topic
`templatetopic`	Name of the template used
`topicparent`	Name of the parent topic
`onlywikiname`	Only permits WikiWords as topic names
`onlynewtopic`	Only permits topic names that do not yet exist

Wiki variables can be allocated to the form parameters, since the parameters are filled with the corresponding values when the page is displayed, i.e. including when the form is displayed. If you wish to disable `onlywikiname` or `onlynewtopic`, you need to delete the corresponding lines. It is not enough to simply set the value to `off`.

15.3 Interwiki Links

Much like MediaWiki, TWiki also enables you to establish simplified links between various wikis and other websites. For this, place the abbreviation of the corresponding wiki or page in front of the actual link. The following link, for instance, invokes a Google search:

```
Google:Wiki%0ETools
```

You have surely noticed that, since the spelling was originally intended for wikis, you cannot use blank spaces. However, you can use `%0E` instead; similarly, to search for the inventor of the Wiki concept, type

```
Google:Ward%0ECunningham
```

Create your own interwiki link

A list of all compatible abbreviations can be found on the page *TWiki.InterWikis*. You may be wondering if you can also define your own interwiki abbreviations on this page, and you would be right. Simply expand the respective table. It has three columns: the abbreviation, the URL to be invoked, and any corresponding tooltip text. You can insert the variable part of the URL using `$page`. A link to *Wikipedia.org* will serve as an example:

```
| WP | http://de.wikipedia.org/wiki/ |       ↵
$page on Wikipedia |
```

As you can see, `$page` can also be left out if the variable part of the URL is at the end.

Tip: The tooltip function is switched off by default. It can be activated by setting the variable `LINKTOOLTIPINFO = on` in *TWikiPreferences*.

15.4 User Administration

Because administrators are responsible for access control (see Chapter 14), they are also confronted with the administration of users. A few actions, such as registering and changing data, can be performed by the users themselves. Others, such as administering groups, are taken care of by a community of users on its own. This leaves three questions that must be answered by an administrator: How can forgotten passwords be reset, users temporarily blocked and, last but not least, users deleted? Let us look at the answers to these questions.

New passwords

Resetting passwords works in two steps. On the page *TWiki.ResetPassword*, enter the user name and new password. The user himself can do this as well. As a result, you will receive a character string containing the user name and coded password. The best thing to do is copy this character string or otherwise note it. Then,

open the page *TWiki.InstallPassword*, copy the character string into the input box and confirm it. Now the password has been reset. However, this will only work if you are a member of the *TWikiAdminGroup*.

Blocking vandals

You can use the same mechanism for blocking users. In such cases, as an administrator, you can assign a vandal a new password. At the end of the blocking period, you inform that person what his or her new password is. Of course, this measure smacks strongly of paternalism and penalization, and should not be part of an administrator's standard repertoire.

Deleting users

If a user has permanently left the TWiki community, or simply wishes to obtain a new name, it can be useful to delete that user from the database. To do this, you first need to remove the account from the password list. The location of this list is indicated in *Twiki.cfg*; it is normally found in *data* and is called *.htpasswd*. There, delete the line beginning with the user name to be removed. The remaining traces can be directly removed from the TWiki. The page *Main.TwikiUsers* lists all registered users. Just remove the corresponding entry here. The topic of the users can also be deleted, if desired. If, however, the user was active in the wiki, several links marking his or her edits will then lead to nothing. You should also have a look at permissions administration. If the user was granted special rights, you should delete the corresponding entries on the pages. Otherwise, someone could register under the same name and automatically have the same rights. The best way to find these entries is through a search (see Chapter 13).

15.5 Email

Contact data

Emails are sent to inform users about page edits and to confirm the registration of new users. The corresponding parameters can be found on the page *TWikiPreferences* under the heading "Email and Proxy Server Settings." Set the `WIKIWEBMASTER` to the email address of the wiki administrator. In any case, you should make sure that the address indicated is accessible, since it will be distributed as a contact address during registration for any questions users may have. You can also indicate the name of the responsible person in `WIKIWEBMASTERNAME`.

Mail program

To send emails from TWiki, you naturally need to specify in what way and via which server the emails are to be distributed. There are generally two possibilities for this, either via the Perl module *Net::SMTP* or an external mail program such as *sendmail*. If you

want them to be sent via Perl, enter the name of the mail server in the parameter `SMTPMAILHOST`. If it is to be on the same computer, `localhost` will suffice. Accessing an external mail program is defined in *Twiki.cfg* in the variable `$mailProgram`. In this case, `SMTPMAILHOST` must remain empty. In `SMTPSENDERHOST`, you can also indicate from which URL the email was sent. If you use a proxy server for your Internet connection, enter the corresponding data in the variables `PROXYHOST` and `PROXYPORT`. Now your TWiki is configured to automatically send emails.

Email notification

You can verify whether your setup has been successful in two ways. You can either register yourself as a new participant in wiki. Unfortunately, this may lead to a series of unused dummy users if the email does not function right away. Therefore, it is better to test the notification of changes function. To register for that, just add your name to the list on the page *Main.WebNotify*:

Wiki

```
···*·Main.Username - your@email
```

This page exists separately for every web. When you add your name, you change a page in the *Main* web. Now, go to the command line of your server, switch to the *bin* directory, and add the following line:

Shell

```
mailnotify -q
```

You have just manually started the email notification service. If the mail settings are correct, you should now receive an email at the address indicated containing recent changes in the TWiki.

Automating notification

Of course, in order to run a decent notification service, you must automate the scripts. Under Linux, so-called Cron Jobs take care of this.

Shell

```
crontab -e 0 2 * * *                                    ↵
  (cd path/twiki/bin; mailnotify -q)
```

With this command, an email is sent every night at 2:00 a.m. (server time). The order of the time indicated is minute, hour, day, month and weekday. A star means “every time.” So, if you want to issue a notification every day, the “day” space must have a star, such as in the above example. Under Windows, there are a number of programs offered by third parties that simulate Cron. Microsoft lists a few under the URL http://www.microsoft.com/ntserver/partners/findoffering/serversolutions/Maintnce.asp.

Tip: One possibility with which to send all users (whether registered or not) general information are so-called broadcasts. These are messages that can be seen on every TWiki page via the toolbar. You can define its content in *TWikiPreferences* by making an entry to the variable `BROADCASTMESSAGE`. In the following lines, you will see two examples that you can simply copy and adapt accordingly. The second method is only possible when the *Pattern* skin is used. Broadcasts are especially useful if you wish to inform users about large-scale maintenance work or changes to the TWiki.

15.6 Spam in TWiki

Blacklist plugin

The general problem of spam in wikis has already been described in Chapter 8.4. In TWiki, as in MediaWiki, there are two means of counteracting the spam threat: the nofollow attribute in Links and a blacklist for IP addresses. Both can be achieved with the Blacklist plugin.[2] Once this extension has been installed, you can perform all relevant settings on the page *TWiki/BlackListPlugin*. There, you will find the entry `BLACKLIST`. It contains a list of IP addresses, separated by commas, which are prohibited from performing actions in TWiki. It is even possible to indicate entire address realms by simply leaving off the last number of the IP address. If an IP is blocked, inquiries coming from it will be delayed for one minute, and then an error message is issued. The message is defined in the variable `BLACKLISTMESSAGE`.

Banlist

In contrast to the blacklist, the `BANLIST` is dynamic and reacts automatically. Principally, it is based on a point system. Per action in TWiki, an IP address receives a certain number of points. If, within a stipulated amount of time, a certain number of points is collected, one can assume that the respective computer is manipulating data. In such a case, the IP address is automatically blocked. Incorrect classifications in machine-controlled blocks can occur. Thus, you can find a form via the `BANLIST` with which IP addresses can be removed from the list again.

Note: Due to a programming error, the `BANLIST` does not work until you have manually added an IP address and then deleted it again.

The parameters for automatic blocking can be found in the variable `BANLISTCONFIG`. The numbers indicate in order how many points are allocated for registration, the save/upload procedure, displaying a

[2] See Chapter 8.4 regarding the installation of plugins.

page in raw format, and other actions such as viewing a page. Then comes the number of points starting at which the address will be blocked, and the time in seconds during which the points will be added up.

Whitelist

In order to prevent administrators, for instance, who frequently work in the wiki, from being blocked by mistake, there is also a `WHITELIST` that contains all IP addresses and areas that are not allowed on the banlist.

Nofollow

External links can be supplied with the nofollow attribute by the BlackListPlugin. To do this, set the variable `NOFOLLOWAGE`. The number indicated determines the time in hours during which a link is to be supplied with this predicate after saving it. The number 0 switches the function off, and -1 always applies nofollow without a time limit.

15.7 Security Aspects

One could generally say that TWiki is relatively secure if you set access rights to the scripts and raw data very carefully. The scripts should only be executable by the system. In addition, except for the script directory, no other directory should be able to be accessed from outside via a web browser.

Holes in the security function

However, in previous versions of the TWiki version we are using from September 02, 2004, a serious security hole is open that you must close during your setup work. In addition, using the Search script, experienced users could execute almost any code on your server. These security holes even caused a stir recently when Spanish hackers managed to access data from a server of the Chaos Computer Club. As we have said, the hole in the current version has already been closed. If you have an older version, we recommend that you visit the site http://twiki.org/cgi-bin/view/Codev/SecurityAlertExecuteCommandsWithSearch, where you will find all information necessary to close the hole. You need to enter the following code in the file *search.pm*:


```
1 my $tempVal = "";
2 my $tmpl = "";
3 my $topicCount = 0; # JohnTalintyre
4
5 # fix for
6 # Codev.SecurityAlertExecute
7 # CommandsWithSearch
```

```
 8 # vulnerability, search:
 9 # "test_vulnerability '; ls -la'"
10 # Escape ' and `
11 $theSearchVal =~
12     s/(^|[^\\])([\'\`])/\\$2/g;
13 # Defuse @( ... ) and $( ... )
14 $theSearchVal =~ s/[\@\$]\(/$1\\\(/g;
15 # Limit string length
16 $theSearchVal =
17     substr($theSearchVal, 0, 1500);
18
19 my $originalSearch = $theSearchVal;
20 my $renameTopic;
21 my $renameWeb = "";
```

This code will disarm certain characters in the search string, thereby blocking any intrusion.

Free use of HTML and scripts

One problem which is more aggravating than dangerous is the lack of the possibility to limit the user of HTML and script languages. Free HTML can cause side effects with template tags and render a page undisplayable. This means one can no longer enter the edit mode from a browser to eliminate the error. Of course, you can always manually start the edit mode via the URL, but this requires a level of technical know-how that you cannot assume inexperienced users have. You can also bring about unpleasant effects with JavaScript as well. For instance, you could integrate a script that automatically closes the browser when a page is accessed. As in the previous problem, the normal user is at a loss. These problems are not hazardous; however, they can spoil the fun of the TWiki.

That is why you should perform a search for `<script>` tags in your TWiki on a regular basis. It will help you find all instances of these small programs and, if necessary, you can deactivate them.

Log files

TWiki keeps a record of requests made via the web. You can define which actions are included in this report in the configuration file *Twiki.cfg* . The corresponding parameters start with `$doLog` and are activated if their value is 1. You will find the log files in *twiki/data*; they are arranged by month and named according to the year and month, consistent with the following pattern:

```
logYYYYMM.txt
```

An entry in the log file contains the following tabular information (separated by | pipes): the date and time, user, invoked script, web and page, miscellaneous information such as whether the storage

process was "quiet," that a requested page does not exist or that a page has been revised, as well as the IP address. This information can give you an idea of what sensitivities and errors may exist during processing.

15.8 Backing Up Data

Any computer system is susceptible to technological failure. When this happens, total data loss is possible. That is why it is advisable to back up data in your TWiki at regular intervals and save the copies in a separate location.

Save all pages

Page data is stored in the directory *data*. To back it up, simply copy the entire directory or archive it. When this is done, the current versions as well as the history of the files are saved. Uploaded files are saved in the *pub* directory, which you should also back up. Under Unix, the command to back up data is as follows:


```
tar czvf backup_data.tar.gz */*
```

You must be in the respective directory. Form data does not need to be backed up separately, since it is saved as metadata in the respective topics and thus included in your backup.

Restoring data

To reload the data, unpack both archives to the directories *data* and *pub*. If you want to be extra careful, you should first empty those directories. This is especially true if you want to restore an older state of the wiki because, for instance, in the meanwhile, vandals have left their marks. This is the only way to make sure that no unwelcome remnants are still lurking in the webs.

15.9 Upgrading and Uninstalling

Since a new version of TWiki is released from time to time, it may happen that you may want to update your installation. Starting with the version accompanying this book, there is an installation script called *UpgradeTwiki*, which automates the updating process. To use it, generate a new directory and unpack the TWiki archive into it. Now you just have to decide in which directory the new TWiki is to be installed. Then enter the following command in the command line level:

Shell
```
./UpgradeTWiki absolutepath
```

If an error occurs, you will find instructions on manually updating your TWiki on the TWiki homepage at http://twiki.org/cgi-bin/view/TWiki/TWikiUpgradeGuide.
If you should actually need to uninstall TWiki for some reason, for instance because you are switching servers, you must do so manually; there is no uninstall routine. This is no big deal, however, because the TWiki stores all of its data in the *twiki* directory. Thus, if you delete that directory, all TWiki data will be removed. You should then undo the adjustments made to the Apache configuration file *httpd.conf*. To do so, remove all entries linking to the *twiki* directory:

CFG
```
<Directory "wwwpath/twiki/bin">
    ...
</Directory>
```

Also, delete the ScriptAlias and the alias for the TWiki directories, and remove the specific entries in the areas `<IfModule mod_env.c>` and `<IfModule mod_mime.c>`. These steps will remove all traces of TWiki.

16 Designing a TWiki

Readers
Authors
WikiAdmins
WebAdmins

Especially in your own online presence open to the public, it is a bit frustrating to only be able to use predefined, standard layout options. This is especially true if you intend to integrate the wiki as only one component of your homepage, or if you have already gone public with a specific corporate identity. Luckily, TWiki offers simple options to alter the layout. You can also achieve functional changes to the community using plugins.

16.1 Designing the Look

Setting skins

As in the case with MediaWiki, TWiki also offers the opportunity to control the look of the page using skins. The default skin that is active is the version of the *Pattern* skin that we use. Of course, you are free to change it; this is done, as in the case in so many other TWiki functions, via a topic. You will find the corresponding entry in *TWikiPreferences* , in the section "Skin Settings" under the name `SKIN`. You can see how the installed skins look in the *TwikiSkinBrowser*.

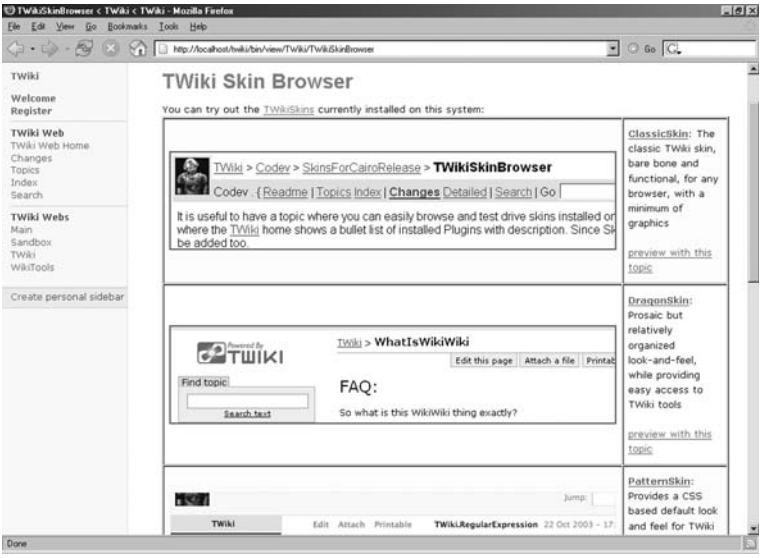
To activate one of the templates in the skin browser, set `SKIN` to the name of the desired skin, without using the word "skin;" thus, if you want *DragonSkin*:

```
···*·Set SKIN = dragon
```

Wiki

However, perhaps you initially only wish to give a few pages a new skin before you decide. To do this, attach the parameter `&skin=`*`skinname`* to the URL of the page you are currently viewing and thus access the page. Now the selected layout will be shown.

Fig. 16.1



Installing new skins

Admittedly, the selection of skins in the standard installation is rather sparse. However, you can install additional skins. A list of current skins can be found under http://twiki.org/cgi-bin/view/Plugins/SkinPackage. From there, you can go to the respective page of the skin and download an archived file which is present as an attachment. Unpack the file to the *twiki* directory.

Note: The procedure described here may vary somewhat for individual skins. Thus, you should read the installation instructions indicated for each skin.

Caution: Before you set up a newly installed skin, you should first test it on a single page to avoid any unpleasant surprises. For instance, in the case of *FreeSkin*, you must first define the edit buttons. If you forget to do this, you can only obtain the edit mode of the preferences page directly through the URL (`edit` instead of `view`).

Adapting the menu sidebar

In the *Pattern* skin, it is relatively easy to put together a menu bar. In every web, there is a topic with the name *WebLeftBar*, which is integrated into the page as a template on the left side of the article. If you access the page, you will see that a complete copy of the menu sidebar is located there. You can edit that sidebar just as you can any other page in the TWiki. For example, if you want to add a link to our homepage, just add the line

```
*[[http://www.wikitools.org][WikiTools]]*
```

This will insert a link in bold lettering. After you save it, you will notice that the changes have been immediately adopted.

Tip: In the template issued with the software, all entries are elements of a list. This is irrelevant to the display on the left side of the screen and can just as easily be left out.

16.1.1 Functional Elements

The most important question that crops up with regard to creating your own layouts is about functional links. How can one access the various views of a topic via links? You may have guessed that the scripts in the *bin* directory play an important role. These scripts are accessed with the corresponding page as a parameter. For a few functions, there are a few further entries to be made. You will find a list of scripts and their functions in Tab. 16.1:

Tab. 16.1

Script	Function	Parameter
`view`	display	■ **skin**: Display of the page using a particular skin. ■ **raw**: "on" displays the source code, "debug" additionally shows the meta information. ■ **topic**: Name of the article to be displayed.
`edit`	edit	■ **t**: Displays the time at the start of editing.
`search`	search	■ **scope**: Area in which the search is to take place. Possible options are "topic" for a search of the title, "text" for a full text search and "all" for both. ■ **search**: The actual search string.
`rdiff`	revisions	To show all old versions: ■ enter no parameters. To show old versions: ■ **rev**: number of revision to be displayed.

Comparing revisions:
- **rev1**: number of first revision.
- **rev2**: number of second revision.

Accessing the View script

Since we want to conceive our scripts for all pages, it is advisable to proceed as generally as possible. This primarily means replacing statistical information with the corresponding variables when possible. For instance, the *View* script is then invoked as follows:

Wiki
```
%SCRIPTURL%/view%SCRIPTSUFFIX%
```

The path to the script, as well as any defined file endings in *Twiki.cfg* are derived from the TWiki settings. The name of the topic to be accessed is clearly defined by the following variables:

Wiki
```
%INCLUDINGWEB%/%INCLUDINGTOPIC%
```

Note that the skin files are actually embedded in the article. That is why it is not enough to use the placeholders `%WEB%` and `%TOPIC%`, since they would always point to the skin itself. The list of parameters begins with a `?`, and several parameters are separated by a `&`. Thus, if you wish to have a link on every page which will display that page in the *dragon* skin, use the following lines:

Wiki
```
[[%SCRIPTURL%/view%SCRIPTSUFFIX%/                ↵
%INCLUDINGWEB%/%INCLUDINGTOPIC%                  ↵
?skin=dragon]                                    ↵
[in Dragon Skin]]
```

The print view can also be generated with this link, since it basically opens the same page with *Print* skin. So, you only have to replace *dragon* with *print* in the above commands and change the label.

Edit link

The edit script additionally requires the current time as a parameter. You can use the variable `GMTIME` for it:

Wiki
```
[[%SCRIPTURL%/edit%SCRIPTSUFFIX%/                ↵
%INCLUDINGWEB%/%INCLUDINGTOPIC%                  ↵
?t=%GMTIME                                       ↵
{"$year$mo$day$hours$minutes$seconds"}%]         ↵
[edit]]
```

As is the case for all TWiki pages, for a complex layout, it is indeed possible to write it directly in HTML. This also enables you to use

some of your own formats, such as those having CSS descriptions from the *FlexibleSkinStyleSheet* (see Chapter 16.1.2).

Revisions

Unfortunately, it is not possible to display the revisions view with skins that are controlled by TWiki pages in the way that you are familiar from the *Pattern* skin. You can only have all edits displayed at the same time. To do so, simply open the *Rdiff* script with the corresponding page.

Tip: You can integrate the revisions view via the templates. The templates for this display are located in the *template* folder, and are called *view.[skinname].tmpl*. The variable `%REVISIONS%` is available in these files, which displays the revisions view.

Backlinks

You can generate a list of backlinks with the aid of the built-in search function (for more detail, see Chapter 13). To do so, use the following command:

Wiki

```
[[%SCRIPTURL%/search%SCRIPTSUFFIX%/                    ↵
%INCLUDINGWEB%/SearchResult                            ↵
?scope=text                                            ↵
&amp;regex=on                                          ↵
&amp;search=%SPACEDTOPIC%%5B%5EA-Za-z%5D]              ↵
[Backlinks]]
```

Do not be distracted by the many cryptic characters. Those are circumscriptions for the special characters `&` (`&`), `[` (`%5B`), `^` (`%5E`) and `]` (`%5D`), that would otherwise disturb processing of the wiki page. Principally, this command searches for all pages on which the page name occurs in square brackets, thus as a link. Those are the backlinks.

Attachments

The function for attaching files is activated via a separate script. You only need to open the page using `attach` instead of `view`, and the upload dialogue will start.

Extended functions

To integrate extended functions (renaming, deleting, etc.), you need to access the *Oops* script. It actually produces the TWiki messages (including error messages). Since a wide variety of messages must be issued, it is invoked with its own templates that contain the corresponding message. This concept is also used for displaying extended functions:

Wiki

```
[[%SCRIPTURL%/oops%SCRIPTSUFFIX%/                      ↵
%INCLUDINGWEB%/%INCLUDINGTOPIC%                        ↵
?template=oopsmore]                                    ↵
[extended functions]]
```

The last functional link we will add is a field for direct access to our skin. The *View* script can be invoked using the parameter `topic`, which will take us directly to the page indicated or, if it does not yet exist, display a search dialogue. The parameter `topic` can also be issued via a form. To do so, the following HTML code will suffice to display direct access:

Wiki

```
<form action="%TOPIC%">
  <input type="text" name="topic" size="16"/>
</form>
```

Of course, this is only a minimal version. The form is sent by pressing the <ENTER> key. You can achieve greater user friendliness by inserting at least one button.

List of all webs

Now, let us have a look at two dynamic components of the navigation sidebar. As you can see, a list of those webs is displayed in the *Pattern* skin that are created in the wiki. This list can be displayed via the variable `%WEBLIST{""}%`. Without indicating any further formatting information, the names of the webs are simply written one after the other. That is why you have the opportunity to make detailed comments on the display in square brackets. You can enter any text you want here. In the spot where the name of the corresponding web is to be displayed, insert `$name`. In this way, you can display the webs as a linked list, for example:

Wiki

```
%WEBLIST{"···*·[[$name.%HOMETOPIC%]·                ↵
[$name]]"}%
```

When the script puts the page together, it displays the line indicated once for each web, whereby the corresponding name is inserted in each case. `%HOMETOPIC%` links to the start page of the web.

Highlight the current web

The `%WEBLIST%` variable also offers the possibility of highlighting a certain web. To do this, you first must define which web is to be marked using `$selected`. Then, use `$marker` to determine how the highlighting will look. Now, you just have to place `$marker` in the right spot in the display. For instance, to display the current web (stored in `%WEB%`) in bold, use the following code:

Wiki

```
%WEBLIST{                                           ↵
  "···*·$marker[[$name.%HOMETOPIC%]                 ↵
  [$name]]$marker"                                  ↵
  selection="%WEB%" marker="*"                      ↵
}%
```

The star indicating bold lettering is only shown if the web currently being processed (`$name`) matches the current web.

As a drop-down element

TWikis, in which several groups work simultaneously and thus have a number of webs, can quickly become complex if the entire list of all webs is always issued. A more elegant solution would be the integration of drop-down menus. For this, we need an HTML form that is filled with a `<SELECT>` element:

Wiki

```
<form action="">
  <select size=1 name="selection">
   %WEBLIST{                                    ↵
   "<option value="%SCRIPTURLPATH%/             ↵
   view%SCRIPTSUFFIX%/$name.%HOMETOPIC%">       ↵
   $name</option>"}%
  </select>
</form>
```

Unfortunately, we still have another hurdle to clear. Nothing happens when you select a web. This is due to the fact that the form is not sent out. A small JavaScript can come to the rescue here. The following attribute loads the URL indicated in `value` as soon as anything changes in the selection box:

JS

```
onChange=                                       ↵
"document.location.href=                        ↵
this.form.selection.options[                    ↵
this.form.selection.options.selectedIndex       ↵
].value"
```

Add the line in the `<SELECT>` element after `name="selection"`.

Now we still have the problem that the *Main* web is always displayed. Yet it would be nice if the current web in which we are located would also be preselected. For this, the HTML command `selected` must be inserted in the corresponding option. The `%WEBLIST%` then looks like this:

Wiki

```
%WEBLIST{                                       ↵
"<option $marker value="%SCRIPTURLPATH%/        ↵
view%SCRIPTSUFFIX%/$name.%HOMETOPIC%">          ↵
$name</option>"                                 ↵
selection="%WEB%"                               ↵
marker="selected"}%
```

List of topics

Similar to the list of webs, you can also have a list of topics generated that enables quick access to existing topics of a web as long as there aren't too many entries. The variable in which all topics are stored is called `%TOPICLIST%`. Here is a printout of the source code for a drop-down list:

Wiki

```
<form action="">
  <select size=1 name="aw"                          ↵
   style="width:100px;"                             ↵
   onChange="document.location.href=                ↵
    this.form.aw.options[                           ↵
      this.form.aw.options.selectedIndex            ↵
    ].value">
     %TOPICLIST{                                    ↵
       "<OPTION $marker                             ↵
         value="%SCRIPTURLPATH%/                    ↵
         view%SCRIPTSUFFIX%/%WEB%/$name">           ↵
         $name                                      ↵
        </OPTION>"                                  ↵
        selection="%TOPIC%"                         ↵
        marker="selected"}%
  </select>
</form>
```

The changes in comparison to the listing of the webs are highlighted in bold. Because the length of the entries in the topic list may vary greatly, it is a good idea to limit the width of the drop-down list. Thus, a `style` attribute has been inserted in the `<SELECT>` element.

16.1.2 FlexibleSkin

Installation

For large-scale use of the TWiki, such as integration into an existing homepage, it is unsatisfactory to only have predefined skins available. That is where *FlexibleSkin* comes in. You will find the corresponding archive on the CD included with this book in the directory */wikis/twiki/addons/*. Unpack these files to the *twiki* directory.

Back in the TWiki, there is a series of pages in the *TWiki* web that starts with "FlexibleSkin." It is best to simply open the topic *FlexibleSkin* itself. Here, you will find all necessary information and, more importantly, links to the relevant pages.

FlexibleSkin is based on an HTML table that divides each page into several areas, as well as a style sheet with which the layout of the areas is controlled. All relevant information is stored in TWiki itself. This means that there is a separate, special topic for each area surrounding the content. They are illustrated in Fig. 16.2:

Principle

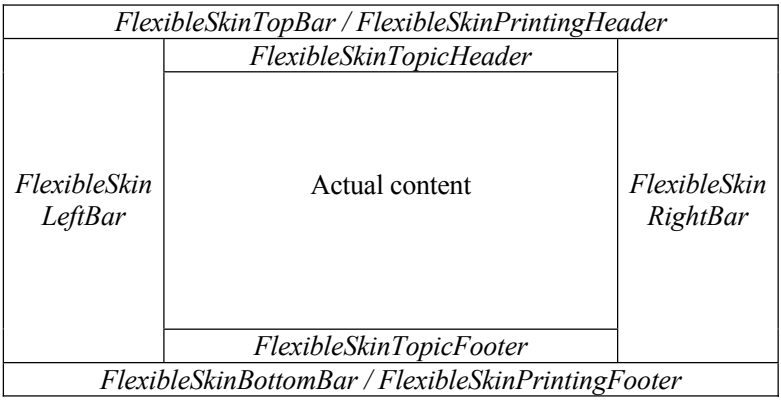


Fig. 16.2

The *Printing* pages are relevant for the print view. For example, an edit link would make little sense on a hard copy of a wiki article, and can thus be left out. The individual special topic areas are integrated when the pages are assembled. This means you have complete artistic freedom in their design. The standard version of *FlexibleSkin* is basically an imitation of the *Classic* skin, meaning that the *TopBar* and *BottomBar* pages and the print pages have already been assigned default values.

Define Include area

You should make it a habit of defining which sections of pages that are to be integrated into other topics can and should be included. In doing this, you will create the opportunity to add comments or descriptions in the template outside of the respective section. You can also prevent remarks (such as the user stamp when the page is generated) from being displayed by mistake. You can mark the source text to be embedded by inserting `%STARTINCLUDE%` at the beginning and `%STOPINCLUDE%` at the end.

Style sheet

In addition to the respective special topics, there is yet another topic that contains a style sheet: *FlexibleSkinStyleSheet*. Here, you can perform general settings that pertain to the overall appearance, or target specific areas. The classes corresponding to the special topics are listed in Tab. 16.2.

Tab. 16.2

Class	Topic
WebSkeletonTable	Pertains to the entire skeleton
TD.WebLeftBar	*FlexibleSkinLeftBar*
TD.WebRightBar	*FlexibleSkinRightBar*
TD.WebTopBar	*FlexibleSkinTopBar*
TD.WebBottomBar	*FlexibleSkinBottomBar*
TD.WebContent	*FlexibleSkinTopicHeader*
	FlexibleSkinTopicFooter
	FlexibleSkinPrintingHeader
	FlexibleSkinPrintingFooter

Now you can influence the interplay between HTML and CSS directly from the TWiki.

Web-specific skins

The use of *FlexibleSkin* is also appealing because the pages describing the layout can be defined individually for every web. That is why there are two different modes of loading the skin. With `skin=globalFlexible`, the template relates to the files stored globally in the *TWiki* web. However, if a skin is only set to `flexible`, the files from a particular web are used.

Tip: When developing your own layout, it is not advisable to set it as the standard, especially if several people are using the TWiki. Thus, include a link to the view of the current page in *FlexibleSkin* on the page *Main.WebLeftBar*. That way, you can still easily view any page with your skin.

Usually a uniform layout is selected for all webs with only minor differences. Unfortunately, you have to decide for every web if the global design is to be used, or whether you wish to generate a completely new one. However, we know a trick that will help you get around this. In every web that is to deviate from the global layout, generate all necessary topics. Now, integrate the version from the *TWiki* web into the special topics that are not to differ from the global layout by simply using `%INCLUDE%`. Thus, for instance, you would use.:

Wiki

```
%INCLUDE{"TWiki.FlexibleSkinTopBar"}%
```

Now you only have to generate those special topics that differ from the global layout.

Note: Remember that the embedding process turns all Includes in the global topic into second-class Includes. This means that special topics, and not the actual article, will become the

`%INCLUDINGTOPIC%`. You can fix this by replacing all `%INCLUDINGTOPIC%` and `%INCLUDINGWEB%` with `%BASETOPIC%` and `%BASEWEB%`.

Similarly, you can, of course, borrow the *WebLeftBar* from the *Pattern* skin for your own purposes. In addition to regular TWiki formatting, you also have the opportunity of integrating pure HTML. Note that in the finished HTML code of an article, the individual special topics are inserted in table cells. Thus, close all tags that you open in a particular special topic in that same special topic. It is particularly not possible to span a `<div>` section over an entire article.

TWiki also enables you to integrate JavaScripts into your page. In doing so, you can create a dynamic navigation bar, for instance, that opens and closes with a mouse click.

Define window titles

There is still one small blemish to correct. In the browser title bar, the content of the variable `%WEBTITLE%` is displayed. By default, it is not assigned a value, which is why the somewhat unattractive name "%WEBTITLE%" appears in this bar. You can set the variable on the page *TWikiPreferences* for the entire wiki, and on *WebPreferences* for the corresponding web. Just set the following line

```
···*·Set WEBTITLE = title
```

on the respective preferences page. You can also set the variable for a single topic, and thus display individual window titles by simply integrating the above line in the topic.

16.1.3 Advanced Options

If the possibilities offered by *FlexibleSkin* are not enough for you, you have two options. Either you use *FreeSkin* or you work directly with templates.

FreeSkin

FreeSkin continues along the path that *FlexibleSkin* has established. However, instead of dividing the page into several special topics and then integrating them into the rather fixed HTML skeleton, there are only two special areas, one before and one after the text. Their content is defined in the topics *FreeSkinPreTopic* and *FreeSkinPosTopic*. Nothing more needs to be added. Thus, you have complete freedom with regard to HTML design of the page. The great disadvantage to *FreeSkin* is that it can currently only be used for the display of articles. For the edit or revision mode, however, it

offers no templates. Thus, it is primarily suited to topics that are only available for viewing.

Templates

Therefore, an all-around solution can only be offered by writing your own templates. You can find templates on the server in the directory *twiki/templates*. They all follow the same name convention:

```
scriptname.skinname.tmpl
```

For scripts for which now templates have been allocated in the current skin, standard templates are employed, so that you do not necessarily have to generate all templates. You can write normal HTML code in the template files, but also use certain variables.

Standard template

The system is divided into two levels. On one hand, there is the standard template *twiki.tmpl*. It defines elements that other templates can access. A few of these predefined elements are already visible in Tab. 16.3:

Tab. 16.3

Name	Content
`sep`	element separator.
`htmldoctype`	Standardized `<head>` area.
`standardheader`	Standard layout header.
`simpleheader`	Simplified header of the standard layout, e.g. for the *Edit* and *Attach* script.
`standardfooter`	Standard layout footer.
`oops`	Error message skeleton.

You can integrate these elements using the command

CFG

```
%TMPL:P{"variablename"}%
```

If you wish to adopt entire swapped out files, you can do so with the following code:

CFG

```
%TMPL:INCLUDE{"filename"}%
```

The extension *.tmpl* can be left off. As in the case of the other templates, the standard master template is replaced when there is a file *twiki.[skinname].tmpl* that matches the skin. In this master template, you of course have the possibility of defining a few elements:

CFG

```
%TMPL:DEF{"elementname"}%
    elementtext
%TMPL:END%
```

Script templates

The second level is that of the concrete scripts. They are the ones that ultimately determine which HTML code is issued. To be able to use the elements from the master template, you have to integrate them with `%TMPL:INCLUDE{"TWiki"}%`. It suffices to indicate "TWiki" as the file name. The respective skin version is automatically searched.

Tip: If you define a few elements in the script templates, you can link to them in the embedded master template. This gives you the opportunity to parameterize the master templates. An example: In the standard template, the function toolbar is placed in the footer. This is defined in the master template. However, the commands that are supposed to be there are defined in the respective script template as `%TMPL:DEF{"topicaction"}%`. Thus, you can basically use the same footer for all scripts.

Placing text

Of course, one important question is how to place the actual topic text. This is stored in the `%TEXT%` variables. In addition, there are several other variables containing content components. A selection of them is listed in Tab. 16.4:

Tab. 16.4

Name	Content
`%TEXT%`	The actual content of the page. In the edit and preview modes it appears in raw form. Caution – the corresponding text box is not included in the variable.
`%EDITTOPIC%`	Edit link.
`%REVISIONS%`	Link to the last three versions of the topic.
`%TOPICPARENT%`	Name of the article from which the page has been generated.
`%PREVIEWBGIMAGE%`	Path to the background image of the preview mode.

Note that not all variables are always filled with content. If you create your own templates, you should always start from an existing template and adapt it to your wishes. In this way, you will also see what variables contain the functions you need.

Meta information

In addition to the source text of the topics, TWiki also saves so-called meta information. This is information about the page itself, its attachments, the parent page, and any related moves or forms. They are generally not shown when a page is edited.

Tip: You also have the option of having the meta data displayed along with the source text. To do so, open the page with the URL parameter `?raw=debug`.

The meta information is stored in meta variables. They are listed in Tab. 16.5:

Tab. 16.5

Name	Content
`%META{"parent"}%`	Name of the topic from which the current page has been created. In contrast to `%TOPICPARENT%`, you can also enter further options here: ■ **dontrecurse="on"**. Only the immediate predecessor is displayed. ■ **nowebhome="on"**. The web homepage is not included in the display. ■ **prefix=""** and **suffix=""**. Text that is to appear before and after the display. ■ **separator=""** Separator in the event that there is more than one predecessor.
`%META{"moved"}%`	If the page has been moved, a report of the move is found here.
`%META{"attachments"}%`	Display of the files that are attached to a page. ■ **all="on"** also shows the hidden attachments.

You cannot use these variables on normal pages. In your templates (e.g. *WebLeftBar* and the *FlexibleSkin* templates) and in the other templates, however, they are used according to the same principle as all other variables.

Your own parameters

In Templates, you even have the possibility of adopting four of your own parameters from the URL. They have to be transferred in the address line using `&param1` to `4`, and can be accessed in the script with `%PARAM1%` to `4`. You can find one application for this function in Chapter 14.6 under Logout Notification.

16.2 TWiki in Other Languages

There is, as of yet, no official support for the use of other languages in TWiki. Work is currently being done on a solution for the next version; as for now, we must be content with a few adjustments. The language problem in TWiki consists of two aspects: *firstly*, the display and error messages that must be issued in the proper language, and *secondly*, the fact that links containing special characters are not always displayed correctly. This means that we must perform modifications to the scripts.

16.2.1 Links with Special Characters

One known bug in TWiki is that automatic linking no longer works if the link name contains a special character. Processing stops at the first special character. This results in the following display when, for example, German link names are used:

Tab. 16.6

Code	Display
`[[ViveLaDiffèrence]]`	ViveLaDiffèrence?
`ViveLaDiffèrence`	ViveLaDiff?èrence
`[[HastaMañana]]`	HastaMañana?
`HastaMañana`	HastaMa?ñana
`[[DüsseldorfCity]]`	DüsseldorfCity?
`DüsseldorfCity`	DüsseldorfCity

Perl modules

To counteract this problem, a few small changes must be made to the configuration. First of all, make sure that you have installed the Perl modules `Jcode`, `Unicode::Map`, `Unicode::Map8` and `Unicode::MapUTF8`, if you are not running Perl 5.8 or higher. The *Testenv* page will let you know. If the modules are not available, you can install them after the fact with CPAN (see Appendix A).

Tip: If it is not possible to install these modules (e.g. because you have your web site at a web host), you can still try the following settings. In most cases, it will work.

Edit Render.pm

Now you have to edit the file *Render.pm* in the directory *lib/TWiki*. From line 40 on, add the following (bold) commands:

Perl

```
1 use TWiki qw(:renderflags                    ↵
 2  %regex $TranslationToken);
 3
 4 # ============================
 5 # Read the configuration file at compile
 6 # time in order to set locale
 7 BEGIN {
 8    do "Twiki.cfg";
 9 # Do dynamic 'use locale' for this module
10   if( $TWiki::useLocale ) {
11       require locale;
12       import locale ();
13   }
14 }
15
16 # Globals used in rendering
17 use vars qw(
```

Adapt Twiki.cfg

This enables local information from *Twiki.cfg* to be read. Of course, it has to be indicated there. To do so, you only have to change the content of two variables. For instance, for the German language version, set the following:

CFG

```
$useLocale = 1;
$siteLocale = "de_DE.ISO-8859-1";
```

The country code must, of course, be adjusted with regard to language. If you enter the links described above, it should be displayed correctly.

16.2.2 Translations

There is currently no official published translation of TWikis in other languages. However, you have the possibility of creating your own skins that also include text in other languages (see the example for German above). Admittedly, this is very laborious. On the page http://twiki.org/cgi-bin/view/Codev/TranslationSupport, you will find further tips on translations.

Norbert Windrich has already completed a portion of the work for the German language translation of the skins included on the CD. His approach is to replace the texts in the templates with variables that, in turn, can be filled with the corresponding information on the page *TWikiPreferences*. You will find all necessary files on the page http://twiki.org/cgi-bin/view/Codev/GermanTranslationForCairo.
Translation into other languages is also facilitated by the fact that all relevant output can be controlled through variables. Thus, it is enough to translate the entries for these variables, and the new language version is finished. Download the archive at the foot of the page and unpack it to the *twiki* directory. You should make a backup of the *templates* directory beforehand, though. Now you can find the page *TWikiPreferencesGermanPart* in the TWiki web. This page lists all translated texts. You will also see that in place of the function links, variables are displayed. Of course, they can still be clicked; just the correct labels are missing. To activate them, you need to copy the German portion of the variable declarations, which you will find on that page, and paste it in TWikiPreferences. Then you should be able to see the German version of the buttons. Unfortunately, the texts controlled from TWiki itself are thus not yet translated. The copyright lines can be found under the variables `%WEBCOPYRIGHT%`, and the navigation sidebar from the page *WebLeftBar* is integrated. You will have to adapt it yourself.

16.3 Plugins and Add-ons

TWiki achieves its special degree of power and flexibility with the aid of so-called plugins. These are small, add-on modules that you can install to supplement your existing software. A great advantage is that the core wiki remains relatively trim, which means that it does not drag around any unnecessary program code. Every TWiki operator can integrate those particular modules that are truly necessary. Furthermore, programmers of plugins can easily and cleanly expand TWiki and adapt it to their needs without endangering the stability of the core wiki. Plugins are often developed from a concrete necessity in using TWiki, so that there is a whole series of very useful supplementary functions. In addition, plugins enable programmers to react relatively quickly to new developments without having to issue completely new versions of TWikis. Thus, it is worthwhile to have a

look at the page http://twiki.org/cgi-bin/view/Plugins every once in a while[3].

As you can see, there are two types of add-ons found there. Generally, add-ons are independent from the existing functionality, and are executed as independent scripts or small programs. Plugins, on the other hand, expand the possibilities of existing scripts by enabling additional TWiki formatting elements, for example.

Plugins

In Tab. 16.7, we have included a list of especially useful and interesting plugins, to give you an overview of what they can accomplish. To find a plugin at *twiki.org*, you have to add "plugin" to every name:

Tab. 16.7

Name	Function
Calendar	Display of a calendar and appointments.
ActionTracker	Task planner for a project group.
Database	Database access from TWiki.
TWikiDraw	Cooperative generation and editing of drawings (requires Java).
SpreadSheet	Spreadsheet in TWiki.
SlideShow	Generate and display presentations.
Toc	Table of contents of *TWiki* webs.
Pdf	Generates a PDF document from a TWiki page.
Comment	Add comments to a page without having to edit it first.
Peer	Enables pages to be evaluated.
Poll	Conduct a poll in TWiki.
ProjectPlanner	Manage time and scheduling plans for projects.
SectionalEdit	Targeted editing of sections (see MediaWiki).
Session	Session-based authentication to circumvent the browser's mechanism.

So just how are these plugins accessed? In principle, they simply add new variables to the existing ones. They are inserted in the topic source text, just like normal TWiki variables. A few are already included in the standard installation, so we can test the concept right away using a concrete object. The list of installed plugins is found on the page *TWiki.InstalledPlugins*. You should also see the smilies plugin here. Edit any page, and add


```
:cool: and :)
```

[3] The plugins discussed in the book can also be found on the CD under */wikis/twiki/plugins*.

You will see that the characters from the smilies plugin have turned into smilies. You can find a more detailed list of possible graphics in the description page having the name *TWiki.SmiliesPlugin*. This kind of documentation is issued with every plugin, and we recommend looking at it before using the plugin.

Installing Plugins

Because there is such a wealth of add-on programs for TWiki (currently 135), we cannot discuss all of them here. A few are needed in Section IV, where we will have a closer look at them. However, our primary focus right now is the process of installing new plugins. It is a relatively easy task. On the overview page http://twiki.org/cgi-bin/view/Plugins/PluginPackage, you will find the entire list of installable plugins. Each one has its own topic. There, you can find directions for using the plugin, updated tips, and, most importantly, the installation instructions. Generally, you only need to download the archived file attached to the end of the topic and unpack it to the TWiki installation directory. This procedure will automatically copy the files to their correct directories. Cautious webmasters can also unpack the files to a temporary directory. A directory structure will be generated that is analogous to that of the TWiki directory, and you will know which files are to be copied to which directories.

Caution: A few plugins must additionally be configured on the file level. To do this, carefully read the corresponding installation instructions.

All plugins are activated automatically as soon as they are installed. However, you can prevent a plugin from being activated by adding it to the list of `DISABLEDPLUGINS` in *TWiki.TWikiPreferences*.

Unfortunately, we cannot provide a universal recipe for the installation of add-ons. Often, they are simple programs that you can either execute from your computer or server. A few of these programs require Java J2EE to be installed on your computer. You can download it at http://java.sun.com. Just follow their installation instructions.

FirefoxExtension

We would like to highlight one particular add-on, since it can make working with TWiki much easier. It is the FirefoxExtensionAddOn, which can be found at the URL http://twiki.org/cgi-bin/view/Plugins/FirefoxExtensionAddOn. If you use Firefox as your browser, this add-on gives you the option of inserting several TWiki formatting commands via a toolbar or context menu. You can install the add-on directly from the page indicated from Firefox. Then you just need to restart the browser, and you should see the

TWiki bar. Now you can either insert formatted text in the edit mode (tables, images, lists) or format highlighted text.

IV. TWiki as a Project Kit

17 Preliminary Thoughts

Readers
Authors
WikiAdmins
WebAdmins

How can wikis contribute to the improved organization of groups? Let's assume that we want to organize a project. What benefits would we garner from a wiki? How would we have to work with it? We want to demonstrate that wikis, in comparison to other systems (e.g. CMS), offer a number of advantages as an integrative organizational and communicative medium. They are just as appealing for companies as they are for political, social and cultural organizations.

The following chapter may seem sacrilegious to some Wikians. In the principles of project management and the wiki philosophy, two opposing concepts meet: a predefined goal, clear definition of work steps, resources and participants on the one hand, and on the other, self-determined goal definition, self responsibility for work methods and steps and an open number of participants. Yet, when we take a closer look, there are several similarities. Even Wikipedia is unthinkable without projects. It may be worthwhile to make use of experiences from the realm of classic project management for wikis and the wiki philosophy.

17.1 What is a Project?

Project management

The literature on project management attempts to define a project using three characteristics:[1]

1. It is a one-time event with a predefined result.
2. There is a predefined scheduled start and a predetermined end to the project.

[1] See e.g. Portny 2001, 26.

3. The amount of people, financial means, production means and equipment available are planned in advance.

However, these characteristics can be used to describe just about anything from construction projects worth millions to moving furniture around in an office. Its presumable clearness leads to the use of the term in daily language to refer to any random human activity. After all, it only describes the fact that a person predefines his activity in his head ("plans"), implements them and evaluates the altered reality for further action.

Taylor's Division of Labor

But contemporary use of the term is not arbitrary. "Regardless of what your project looks like, you always define it on the basis of the three same elements: output, start and end date, and resources," observes Portny.[2] The word "project" in project management is related to Taylor's principle of internal division of labor. According to this principle, all work steps are divided into their individual components (in project management, into "projects" and "work packages") and standardized. The individual work steps can thus be re-assigned and resource usage more precisely planned. As a result, efficiency and productivity of mass production can be increased considerably. However, prerequisite to this success is the exact observance of schedules and the extensive elimination of coincidence. Project and personnel alike are under central control.[3]

A project as a concept

The project concept, as it also forms the base of the wiki philosophy, counters that of the original meaning of "project." The older term "project," stemming from the 17th-century French word "projet," has a much broader target and is far less technocratic. It means something like "design," "urge forward," "push out," "throw forward," and "thrust out;"[4] all more reminiscent of the English verb "to project." The wiki philosophy, too, deals with reformulating systematic action, but with the intent of moving from passive reaction and optimization to rational planning. Here, too, the focus is on efficiency and cost-benefit issues, but the reaction to these requirements is different. The goal, which is far less clear, is not completely based on the primacy of an increase in corporate earnings. It is based on an idea of "people in free cooperation" (Spehr), aimed at promoting the

---

[2] Portny 2001, 27.

[3] The boom of "projects" in the working world fits in with a redesigning of production, right up to greater flexibility of employment relationships with the goal of a reduction in wage costs and a simultaneous integration of the immediate producers with more responsible autonomy (see Candeias 2004, 179).

[4] See Kluge 2002, 722.

creation of structures of self-administration and direct control of central decisions by the pool of participants. Participants themselves decide on goals, milestones and deadlines – thereby running the risk of areas not being processed and the production process running less directly.

Project: free planning?

Let us assume that you want to act in a relatively goal-oriented manner, achieve concrete results and maintain clarity regarding who is to do what when, yet at the same time, to decide on the process in as democratic and open manner as possible and not subject yourself to arbitrary efficiency criteria. If we examine the advantages and disadvantages of both project approaches, we find that they can be combined to a great extent.

Project phases

Wikis, then, can provide assistance in certain project phases (Tab. 17.1) or be employed as a supplement to existing systems. They can also serve as the central medium with which to manage all phases.

Tab. 17.1

	Phase	Goal	Concrete examples
1	Concept	The idea is developed.	■ Establish start group. ■ Collect ideas. ■ Write project drafts.
2	Delimitation	Composition of the project plan.	■ Define goals. ■ Plan work. ■ Plan costs. ■ Plan procedure.
3	Start	The team is put together.	■ Allocation of important tasks (admin, moderation, finances, PR work…). ■ Define rules.
4	Execution	The work is accomplished.	■ Query sponsors. ■ Do publicity work. ■ Occupy rooms.
5	Conclusion	The project is concluded.	■ Accounting. ■ Documentation. ■ Transfer to subsequent projects. ■ Evaluation.

Phase overlap in wiki philosophy

The phases in conventional project management are strictly separated from one another. If, in comparison, one follows the wiki philosophy, there is overlap between the first and third and the fourth and fifth phases. According to wiki philosophy, the construction of the team starts as early as the idea collection phase, and the circle of participants does not remain static, but rather can and should increase. This philosophy ranks volunteerism as its top priority. The "two foot" principle rules: Team members can also walk away from such projects. It nevertheless makes sense to establish responsibilities and clear rules after the second phase, at the latest. There is overlap toward the end of the project, as well. For instance, materials need not be collected for closing documentation, since, ideally, all necessary information is available on the wiki network.

Within the execution phase, tasks can and should be performed in parallel, even in classic project management; also: in software development, it is sometimes necessary to "jump back" when one has hit a dead end – a procedure, by the way, that is supported by the wiki history function.

17.2 TWiki as a Project Tool

Wiki: a fundamental decision

If and to what extent the planning and execution of a project is based on wikis is also always a fundamental decision about the character and work form of a project. Of course, one can also design traditional project management with wikis, but if the project is not open for real changes "from below," the question remains as to why one should employ wikis at all, which are tailored to dynamic and open processes. Congruent to its state of development, wikis are not always simple to use – although it is remarkable what we can already do today with a single wiki. Thus, each individual case should be examined before deciding whether or not using a wiki corresponds to the issues and to the group executing a project.

Advantages

To decide on the use of wiki technology as a central communicative and planning medium, one must weigh the pros and cons. Arguments in favor of using wikis include the following:

- Wikis are generally cost-free, open-source programs.
- A wiki should motivate the team or other participants to define structures and responsibilities in a cooperative manner. This results in the development of a strong integration effect, and participants will identify more deeply with the project.

- Through their openness, wikis promote transparency and provide an overview. Problems or faulty developments can be identified and discussed at an early stage.
- Wikis offer a simple, uniform system in which ideas can be collected. This can counteract a strict isolation of work groups and promote networked thinking.
- While several versions of a single document can circulate in a project, with wiki technology, it is possible to offer participants central access to the current version of a document.
- Wikis adapt themselves very rapidly to a project and mirror its project dynamics.
- The versatility of a single piece of software offers the opportunity to use it as a central publication and organizational system. The wiki system serves as a uniform administrative and reporting system, among other uses, in which decision processes or progress reports may be documented immediately.
- At the same time, the wiki enables a decentralized organizational structure which can cut down channels and work times. Editing, publishing and documentation concur in a single work step. For instance, content must not first be approved by an Internet editor before going online.

Problems

If you decide to work with a wiki, you will have to accept the following peculiarities:

- Based on the current state of development, wiki technology is not comparable to specially tailored, expensive project software. It offers no finished project templates, automation in process control, human resources planning or financial administration. It still lacks sufficient tools for scheduling and appointment planning.
- The decentralized hypertext structure can lead to initial coordination and acceptance problems. In the case of larger projects, complexities frequently arise that must be minimized. This means that navigation bars and overview pages must be conceived and maintained.
- Since a wiki offers so much freedom, it requires more attention and self-discipline from its participants. For example, they must adhere to stipulated conventions so that others can also quickly find the right information within the wiki. In addition, it is necessary to think beyond one's own work area with a view to co-

ordination ("What is everyone else up to?"). Otherwise, chaos, edit conflicts and redundant work are only a matter of time. Yet, as larger wiki projects have demonstrated, structured, successful work is possible under certain conditions, even in very large groups.

Which wiki?

Once you have decided on supporting your project with a wiki, you are only left with the difficult decision of which wiki software you should use. The following points are a list of criteria that may help you make that decision:

- **Programming language.** It is possible that you may have to make an adjustment or two to the scripts. It would thus serve you well if you were familiar with one of wiki's programming languages, such as Perl or PHP.
- **System requirements**. Especially if you do not operate your own server, you must make sure that wiki will run with the software available.
- **Technical capacity.** If your wiki is very heavily frequented, it is advisable for performance reasons to seek a database solution.
- **Installation effort**. What technical knowledge is required for installation, and to what extent is the set-up automated by routines?
- **User administration**. A wiki that is to contain sensitive data must offer the opportunity to block certain areas to general access.
- **Division in sections.** For user administration and the collective work of various groups in a wiki, separable sections can be advantageous.
- **Expansion through plugins**. Plugins usually offer a large variety of additional functions.
- **Interfaces**. An attractive look invites users to read and work. For one's own pages, being able to design them at will is an important aspect.
- **Convenient operation.** Can the wiki also be used by inexperienced users without difficulty? At the same time, does it fulfill the expectations of experienced wiki users?
- **Language**. Many wikis only have English interfaces. Can your community live with that?

- **Documentation**. Can one find sufficient information on the operation and maintenance of the wiki on the Net or in technical literature?
- **Further development**. Is it expected that errors in the software will be eliminated and that the software will continue to be adapted to demands in the future? Is the data stored in a format with which, if necessary, it can also be transferred to other systems?

17.3 Flat Hierarchies: Relationship between Moderation and Team

Conflicts

"Project," "cooperation," and "self-organization" all sound great. Yet what on paper looks like an easy and unproblematic strategy is suddenly plagued with a number of difficulties. As in any cooperative project, the same problems crop up again and again. Agreements are not met or adhered to. Personal differences and misunderstandings occur. You find out too late that individual participants are more preoccupied with other obligations than you had expected, to the extent they do not treat the current project with priority, or that they are overburdened. There are differences in opinion with regard to the procedure. Human resources fluctuations and changes to the project goal present additional problems. There are insufficient if any guidelines or procedures with which to solve problems. Conflicts are seen by many as disturbing. They cost time and nerves. However, conflicts can also be a source of dynamics and further development.[5]

Team = group of individual personalities

Whether you have an open wiki or are putting together a team for a larger project, you need to keep in mind that some participants have never worked together and will not be able to cooperate with each other ad hoc. In addition, there are lone wolves who wish to maintain as much autonomy as possible. A variety of work styles and communication methods will converge. The opportunities of co-determination and the margin for creation may make some people initially feel insecure and thus wish to receive clear-cut assignments. Respect and mutual trust must first be established through cooperation and continually updated.

[5] Dealing with conflicts is a question of experience, among other things. In any case, managing significant "disturbances" has priority over continuation of a project.

Thus, make sure that the participants have the opportunity to get to know each other personally, insofar as possible. The assignments, requirements and work pressure limitations must be clarified. The project requires simple but clear rules that are acceptable to all.

Broad involvement

We began by mentioning the significance of flat hierarchies for successful self-organized group processes. In the preliminary stages, the individuals in charge of implementing the project or whose participation is necessary should be included in the processes of determining goals and structure. Such individuals include project team members and their respective superiors. However, it is also advisable to include external individuals who are directly affected by the project or have marginal roles in it, a step which frequently supplies new perspectives.

Conviction

For the dynamics of self-organized group processes, it remains decisive that the individuals be truly convinced of the sense of the project's goal and see the result as a type of personal gain.[6] The participants must be familiar with the goal, ideally having worked it out together. This cannot be forced or demanded. Regarding the development of a project culture in open software projects, Stefan Meretz explains: "Whether or not a project truly results in a community cannot be manipulated. In contrast to prescribed projects in private enterprise, open projects are based on a strong degree of participant interest, specifically in that of good, usable software."[7]

Hierarchy?

The more people that are involved in a project, the more urgent is the question of coordination, supervision, but also the division of power, authority and hierarchy. This is not only related to the project itself; existential dependency relationships are often inhibitory, such as when superiors are integrated into projects as "watchdogs."

The extent to which hierarchies are functionally necessary was also contentious in the history of the emancipation movement. Ultimately, it depends on how the respective roles are filled and what real intervention options exist for the participants, in order to counteract any tendencies toward bureaucratization.

Moderation: accompany, don't direct!

If the group is responsible for the goal and the result of the project, someone will need to take on the role of moderator. The moderator serves as a sort of midwife, so to speak. That is to say, he or she provides assistance in maintaining the organizational environment and the communicative process. At the same time, he or she must make sure that stipulated conventions and discussion rules are observed. However, it is the group that stands in the foreground, not

---

[6] This is where non-democratic economic orders reach insurmountable limits. That, however, is a subject all its own.

[7] Meretz 2003, 106.

the moderator. Rather, the latter allows the course of the project, with regard to content, to be determined by the group. His or her motto should be: "Accompany, don't direct," refraining from overly zealous regimentation in order to preserve important group processes. The wiki philosophy, the conception of open software development and the notion of moderators in open-space conferences all rest on the same principle: that of leaving control of the process to the group.[8] It is the group that possesses the professional competence and responsibility.

Double duty

In web logs, Internet forums or wikis, moderators often also take on technical tasks, such as serving as wiki admins. They aid in user administration, text maintenance, etc.[9] Furthermore, within the debates and in the wiki system, there is an ongoing need for tidying up. Interim results must be collected and content supplemented or demanded; also, several tasks might crop up all at once. The immediacy of wikis, in which, contrary to moderated mailing lists, texts are published right away, bring stimulation into the discussion. Suitable "starting points" are required at central nodes in the system, especially on the start page, that aid in quickly bringing users to their desired content. In larger systems, a work plan is necessary here as well.

To moderate and be moderated

In wiki projects, participants frequently take on moderation duties or act as maintainers who take on additional tasks to maintain the functionality of the overall project, receiving wiki administrator privileges in the process. It is not unusual for moderators and admins to be elected and, following their 'term', subsequently pass the job on to others.

Thus, prepare yourself for a mutual learning process in which you may moderate and at another time be dependent on other moderators. Moderation alone is fast becoming its own work area, and should be taken on be people who are not obligated to provide content to the project, so that they can maintain a clear head for their job. Of course, your group can also agree that moderators be changed on a regular basis. Just as in the case of admins, it is advisable to have at least two moderators. Superiors are less suited to be moderators for the simple fact that they are heavily involved in the content, and conflicts of interest may quickly arise. It is better to either opt for external moderation, or for someone from the same or-

[8] See Meretz 2003, 103ff.
[9] See re: duties of moderation e.g. Dauscher 1998, 29ff.

ganization but a different department. The latter have a good idea of the subject matter, yet they are not mired in content issues. Of utmost importance is that the group trusts the moderators.

18 Conceptual Phase: Collecting Ideas and Outlining the Project

Readers
Authors
WikiAdmins
WebAdmins

18.1 Establishing a Base

Define the assignment

A project begins with your idea or an external assignment. If you receive an external project assignment, clarify as precisely as possible what the goal of the project is, what expectations the client has of you, and why the project was assigned. Discuss at an early stage the resources at your disposal, and don't be afraid to reject the project if you have the impression that financial or conceptual support will be lacking.

Example: Conference on the Neolithic Age

We wish to illustrate the existing possibilities using the example of a conference preparation process. Let us pick up the ball and continue rolling it. We will assume that our Stone Age people want to call a conference on the future, in order to discuss life in the Neolithic Age. Using this example, we will think through the procedure of such a project and introduce several features to you that you may be able to use for your own project. To organize a conference, we will conform to the general phases of project management (see Tab. 17.1). Thus, our example has a simple structure: The Stone Age people are their own clients.

To wiki or not to wiki?

Because the use of a wiki for the character of the conference is a fundamental decision, the group should decide as early as the conceptual phase if and to what extent they wish to work with a wiki. Excluding factors would be if the majority of participants or superiors – for instance, the old shamans – do not agree to the procedure and will not support the results because they fear having to surrender control, although they are crucial to the implementation of the project results. It is vital that ideas that are developed "from below" ul-

timately have the chance to be implemented the way they were intended.

Choosing TWiki

In answer to the question as to which of the clones we want to use, we have decided on TWiki. Our most important criterion was its variety of application options. TikiWiki and TWiki stand out in this regard. As we mentioned in the preface, TWiki represents a technically greater challenge for admins. We nevertheless chose it because this clone remains true to wiki technology, whereas TikiWiki contains the wiki only as one of several communication applications.

Introducing wiki

Therefore, immediately before or directly at the start of the conceptual phase, a separate wiki introduction meeting should be arranged. In our case, we are introduced to TWiki, and the participants are made familiar with the wiki philosophy.

This meeting concerns the integration of the wiki. Furthermore, it serves to eliminate any existing reservations and conduct a few practical experiments with TWiki. It is also a chance to discuss the classic wiki questions – regarding vandalism, etc. The occasion serves to point out that wiki is not a miracle technology, but that adopting it can be well worth it, and that its adoption is a decision that the participants themselves must make.

Tweaking TWiki

Especially TWiki, in the condition in which it is supplied to users, needs a bit of preliminary tweaking, for instance, by creating an HTML start page that offers an overview, and by generating a few webs and topics with initial content. Furthermore, the sandbox is to be activated and the most important syntax made available at a glance. Computer skills can never be assumed, so it is advisable to publicly discuss what skills and experiences participants have. It is a good idea to appoint a contact person that can provide support in the case of technical problems.

Constructive criticism

Another good reason to hold an introductory meeting is that some people wince at the thought of expressing their thoughts in writing, compressing the content, and exposing themselves to criticism. Many prefer to communicate orally, not least of all because they can be certain that the right people receive the information. This need should always be taken seriously, and can be addressed accordingly at such an event. Yet reciprocal constructive criticism and cooperative discussions take practice. By constructive, we do not mean avoiding conflict and stressing harmony, but rather criticism that is always paired with suggestions for alternatives. We are talking about assuming a position of respect for the work of others, since they identify with it.

Establishing rules

Let us get back to the hunters and gatherers. Basically, they need to lay down a few rules for interaction. Points to be considered include:

- Is there a check mechanism if, for instance, a part of the project gets behind schedule because certain content could not be generated? Who is responsible?
- They have to determine how and according to what rules the project is to be moderated. Which procedures exist to solve internal conflicts?
- How do you regulate the power differential that exists between admins, maintainers and moderators and normal users? A separate page should definitely be generated for such discussions. To prevent someone from blocking the project, a backup should be made at regular intervals and saved to a neutral server. To do this, at least one additional person is required who is familiar with the technology and has full access rights.
- Setting up central contact points has already been addressed. This includes establishing pages such as "News" or "Conventions."

Tip: Don't put too many expectations in your wiki. Not everything is going to function properly right off the bat. Never forget that, with all of the spontaneity, you are dealing with group processes and learning processes that take time. Associative thinking, creative seizing of ideas and cooperation are not prevalent in our form of society, nor are insight and understanding in organizational issues.

18.2 Collecting Cooperative Ideas and Generating Project Outlines

Project outlines

The hunters and gatherers must first collect ideas and ultimately formulate a draft paper (project outline). They already have several suggestions. A possible title would be "Get Out of the Cave. Prospects for the Future in the Neolithic Age." The advantages of sedentariness could be thematicized, or new techniques in cross-country plowing. Workshops on band ceramics are to provide a cultural supplement to the conference.

The draft paper describes the beginning and end, the title, goal and possible content of the conference. By making the ideas more concrete and compact, we can achieve more clarity for ourselves, and the paper can be presented to sponsors for their decision. Thus, we can already utilize two advantages of the wiki.

- More ideas can also be collected for the paper between meetings. They would be immediately worked into the paper and available online to all participants.
- The draft paper can be further developed in a cooperative way. Cooperative writing will also come into play for the subsequent organizational planning, applications, goal descriptions, task planning and invitational texts.

Collecting ideas

Now we can already start collecting ideas or query people's expectations on a wiki page for the next meeting, to summarize them in a text on a new page. The involvement of the people concerned in the formulation process increases the chance that the draft paper will be read and agreements will be better understood and accepted by everyone. Any modifications can vary greatly, from the correction of minor typing and grammar errors to the editing of long passages.

Brainstorming

Wikis can serve as a private notepad on which impressions, inklings, tips, questions, etc. can be noted down at random intervals. The hypertext structure can reflect and promote basic networked thinking. However, you can also use the wiki for cooperative forms of writing, such as collaborative brainstorming and other playful types of idea collection. One popular variation is to have the first person write one sentence, and the next person add a sentence to it spontaneously, and so on. For brainstorming, "fantastic" or "imaginary" ideas are also welcome, since they motivate participants to abandon habitual thought routines and think innovatively. "Food that won't run away," may have been one such invented idea on the verge of the Neolithic Age. Only after such a thought is expressed, has caused friction or prompted objection can it be seized by others.[10]

Seize ideas and new thought processes

Cooperative and associative thinking is a must, in which ideas are grasped and developed further in a cooperative manner toward a common goal. Initial fuzziness is not a hindrance; on the contrary – even if one does not agree with a point of view, an important impulse may be triggered that sheds light on aspects that had been previously ignored. This requires an attitude in which one asks what someone else had really been trying to say. A relaxed atmosphere means unusual ideas and half-baked thoughts can also be expressed, which can be the source of innovative thinking as well as "criticism."

[10] Which may possibly have developed from the desire for food that would not bite back.

Creativity is learnable

Cooperative writing occasionally requires spontaneity and creativity. It is an artistic process that triggers the interaction of rational thought and emotional perception in both hemispheres of the brain. Many people generally think they are not creative and that cooperative forms of writing are not "their thing." However, they overlook the fact that people basically do nothing more than think associatively. Thus, everyone is creative in some area: in landscaping, mixing music or even in destruction (for instance, some wiki "moths"). In creativity research as well as education science, the realization has meanwhile taken hold that creativity is learned and, for example, can be promoted with combinatorial analysis games and mind maps.

Blocks

Writing has always also had a psychosocial component. Here are just three tips in that regard:

1. Individuals seek to obtain attention from others. The fact that their own ideas are perceived and discussed plays an important role in interactivity, dynamics, creativity and motivation. That is why so many management guidebooks recommend good old backslapping to more or less manipulate employees. When it is about cooperation and creativity, not only open and goal-oriented criticism is required, but also truthful statements regarding what one liked or found helpful.
2. A hostile climate encumbers creativity by instilling a fear of disgrace. Also, open, associative thinking in a group is prevented by social blocks. Cave painter Hanna will think twice about what she says if her teacher or the cult community supervisor is within earshot – and vice versa.
3. Everyone has writer's block from time to time. The thought that a text must first be finished before being published is paralyzing. In such cases, it can be helpful to start writing as early as possible and begin with small articles. The wiki philosophy assumes that starting is more important than waiting for the perfect version.[11] "Ignore all rules," is the call Wikipedia makes to its new authors to illustrate the fact that they should not shy away from what has in the meantime become a mountain of conventions, but rather release texts as quickly as possible for further processing.

[11] Just think of Linus Torvald's method of releasing texts early and often.

Visuals

We must admit that computers are not the best tools for creative processes, since they indeed prescribe technical behavioral patterns and set limits. However, you should still take advantage of any existing room for design in creative processes. Especially when ideas are sought, the forms of design can differ greatly and consciously stretch beyond the scope of traditional structures, such as when images are also integrated because colleagues express their ideas and utopias via images or other data formats. Thus, include pictures and graphic images, and submit sketches and mindmaps from team meetings if they serve to substantiate your ideas. However, before you rush off and integrate a bunch of images into your page, you should frequently ask yourself if the visuals are truly fitting, and what functions they are to serve. Too many pictures can be overwhelming.

18.3 Keeping a Log with Wikis

Keeping a log with wiki

All possible sorts of texts can be generated in a cooperative effort: project outlines, invitational texts, blurbs or reports. Naturally, logs and reports also play a vital role in projects. After a meeting, the minutes of the meeting can be submitted to the wiki, either as a downloadable file, or, even better, as normal text.

In this regard, wikis offer the advantage that modifications can be made after the fact. Team members have another week's time to think things over. If, for instance, it was agreed that the "Finances" group will subsequently submit the cost estimate of the cave operator, it can be directly entered on the group's own wiki page. Similarly, unclear formulations can be corrected or existing ambiguities debated using all available texts. Such discussions can clear up misunderstandings at an early stage that otherwise would only crop up once execution of the project has already begun and that would endanger the success of the project. That is why it can also be of assistance to highlight new entries in a different color and specifically inform the project management of such entries.

During the meeting

Of course, meeting minutes can be entered in the wiki during the meeting itself, if the technical requirements are available. In addition, a beamer would provide visualization. However, the experiences of moderators and lecturers have shown that these measures are not optimal for a wiki project. *Firstly*, such technology assumes that the keeper of the minutes has sufficient experience in dealing with TWiki. Technical problems could quickly hamper the discussion flow and ruin the meeting's spontaneity, productive atmosphere and content context. *Secondly*, minute takers hide behind their lap-

tops, making it all the easier for them to withdraw from the discussion. This may be interpreted by the other participants as a lack of respect, since they are not sure whether the minute taker is taking part in the discussion or not. *Thirdly*, we must not forget the power of formulation. As soon as more than just the facts are recorded, it has a great influence on the course of the discussion and the direction of thoughts. In contrast to other moderation techniques, the other participants cannot directly influence the visuals. The minute taker may thus cause some individuals to feel insecure. *Fourthly*, because wikis are not WYSIWYG systems, they are not suitable for beamer presentations of this sort.[12]

Asynchronous or synchronous?

Möller sees great potential in the development of tools enabling simultaneous editing of texts in real time.[13] On the other hand, one of the great advantages of wiki technology seems to be the temporal equalization of discussion processes. Basically, both occur in group processes: acceleration and time pressure, as well as the desire for more time to expand certain ideas.[14] It has been observed that many Wikians do not simply start writing, but first do research and organize their thoughts. Kuhlen sees in "asynchronous communication forums" the advantage that a climate of balanced and informationally secure communication can emerge: "Not every question has to be answered immediately. The advantage of greater rationality is to be generally assessed higher than the presumable loss of spontaneity in the synchronous medium with its direct reactive communication styles. However, the success of asynchronous communication processes is largely dependent upon good moderation. This is especially true in learning environments where moderation performance is continually necessary, such as creating incentives, generating outlines and summaries, offering orientation aids and providing feedback for knowledge assessment."[15] Kuhlen's reference to asynchronous communication processes and learning environments, however, sees the moderator, who, among other tasks, must create incentives with sophisticated seminar designs, as the central figure. In deviation thereof, we have just presented our assessment of the role of moderator. Ideally, that person should not be a jack-of-all-trades, whose

[12] There is a very useful SlideShowPlugin for presentations; see Chapter 21.3.
[13] Möller 2005, 200f.
[14] Writings on large group processes repeatedly state that such processes can only be initiated and be capable of working out solutions under enormous time and problem constraints.
[15] Kuhlen 2004.

perfect seminar design serves an ultimately manipulative function, to which participants only "react."

18.4 Writing Texts with the Editor Plugin

KupuEditor

Let us return to our example. Within the context of planning the conference, several texts must be written, including by people who will not adopt TWiki syntax. This user group would be well served with the provision of WYSIWYG editors.

Development of WYSIWYG editors has recently begun for popular wiki clones. In the future, they will increasingly facilitate working with wikis. For TWiki, the Kupu Editor add-on pursues a promising approach. Kupu[16] is actually an HTML editor based on JavaScript that is compatible with both Mozilla and Internet Explorer. Developers are currently adopting, step by step, one version of this editor to the TWiki environment.

Using the Kupu Editor, it is possible to format the text graphically somewhat, similar to using a simple word processing program. However, note that an editor is only an attachment designed to make working with TWiki easier. You cannot achieve a full degree of functionality with it. Basic knowledge of TWiki syntax is still necessary, and the Kupu editor can only transform that which your TWiki version enables. For instance, the editor, as the entire TWiki, only has one font.

18.4.1 Installation of the Kupu Editor Add-On

A few steps are required to install Kupu. First of all, obtain the archive from the plugin page http://TWiki.org/cgi-bin/view/Plugins/KupuEditorAddOn and unpack it to the *TWiki* directory. This will copy the necessary files to their correct locations in TWiki. Then you need to run the installation script in the *TWiki* directory, which automatically sets the proper permissions:

[16] "Kupu" is a Maori word for "word," "statement," or "comment." "Kupu-kupu" is Indonesian and means butterfly, which has become the logo for *Kupu*.

```
perl KupuEditorAddOn_installer.pl install
```
Shell

If the Perl interpreter is not located in the */usr/bin*, you will need to adapt the shebang lines of the scripts accordingly. As indicated regarding the installation of TWiki, locate the path with `which perl`. If you should need to change the line, the easiest way is to use an editor (e.g. Nano). The files needing editing are:

bin/kupu_attachments
bin/kupu_html2TWiki
bin/kupu_TWiki2html
bin/kupuedit

The shebang line is always the first line. Just change the path */usr/bin/perl* to the corresponding path of your system. Now you should open the editor to test it. You can do so by opening any page in TWiki and replacing the URL "view" with "kupuedit." You should now be able to edit the file in a graphic interface.

If you have activated authentication, you must also include the four scripts cited above in the *.htaccess* file:

```
<Files "kupuedit">
       require valid-user
</Files>
<Files "kupu_TWiki2html">
       require valid-user
</Files>
<Files "kupu_html2TWiki">
       require valid-user
</Files>
<Files "kupu_attachments">
       require valid-user
</Files>
```
CFG

Finally, it is necessary to make Kupu available to all users by generating a link to the editor:

```
<a href="%SCRIPTURLPATH%                    ↵
/kupuedit%SCRIPTSUFFIX%/%WEB%/%TOPIC%">     ↵
Kupu Edit</a>
```
Wiki

You can either integrate this line in *WebLeftBar* or, more elegantly, include it directly in the *View* template. For the *Pattern* skin, you would make the entry at the spot indicated below:

CFG

```
%TMPL:DEF{"toolbar"}%
...
   * %EDITTOPIC%
   * <a href="%SCRIPTURLPATH%/kupuedit
     %SCRIPTSUFFIX%/%WEB%/%TOPIC%">
     KupuEdit</a>
...
```

Now you have a convenient access point to a WYSIWYG editor for TWiki.

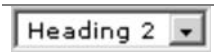
Opening Kupu

After you have successfully installed Kupu, the link **KupuEdit** appears next to the standard **Edit** button of a wiki page with which the editor is opened.

18.4.2 Features

The editor understands text and paragraph formats. Text formats are applied to the highlighted characters; accordingly, paragraph formats apply to the paragraph in which the cursor is located. Via the header, you can click on various format options. The following functions are available in the header:

Tab. 18.1

Icon	Function
	Allocate headings of up to six levels or highlight a quotation with "verbatim."
	Save.
	Set marked text in bold.
	Set marked text in italics.
	Expandable list with icons.
	Numbered list.
	Non-numbered list.
	Remove link. Only appears if a link is marked.
	Undo last command. To date, it does not work very reliably.

	Undo last undo. Also not very reliable. Switch between source code and editor view.

Tip: Using the button **Edit Source**, you can directly edit the source code to thus tap the full potential of TWiki syntax. For example, in the copy & paste function or when working with tables, non-functional or bothersome data in the source code found in the Edit Source view can be detected and corrected.

Setting links

Objects – texts, images or symbols – can be furnished with links. Just mark the corresponding object and enter the link in the top right column ("Internal & External Links"). Note that there are different kinds of links:

- Links that lead to a page outside of TWiki ("external links"). Mark the link text or image, select "External Links" in the right column, and enter the desired URL in the input box "Target:" (e.g. http://www.stoneage.com). Confirm it with **Make Link**. The link text is now displayed in blue and underlined.
- Setting a link to a target within TWiki ("internal links") is done in a similar way. After marking the link text, select a topic from the current web (*Main*, *Sandbox*…) or "All Topics" for the entire TWiki. Now you can select a TWiki page from the list of possible pages. After confirming your selection with **Make Link**, the internal link appears in a blue box.

WikiWords and Interwiki links can be automatically detected and also appear in a blue box.[17]

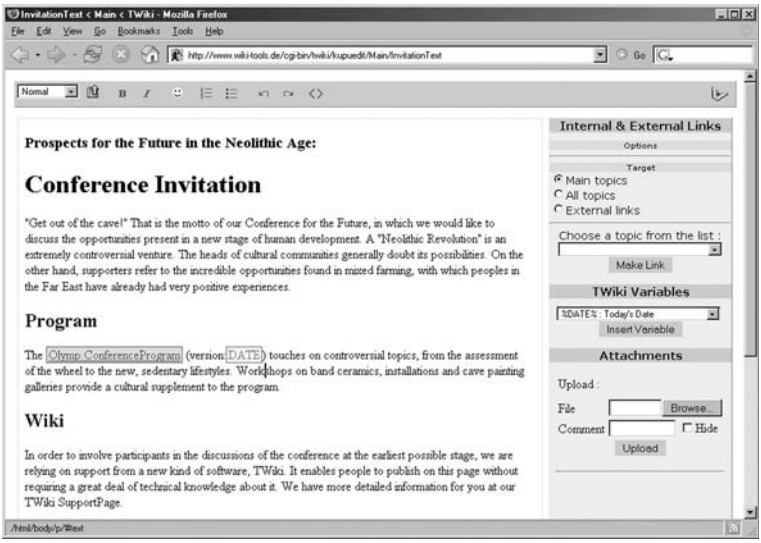
Inserting variables

In the right menu bar, you will find a drop-down list of TWiki variables. Go to that point in the text where you wish to insert a variable. Select the respective element by clicking it and confirm with **Insert Variable**. The inserted TWiki variable is then displayed in edit view in a green box.

The invitational text for the conference would look like this in Kupu:

[17] You can activate and deactivate this function in the right menu bar under "Internal&External Links – Options."

Fig. 18.1



Adding attachments

Furthermore, the Kupu Editor has a simple attachment administration function. It works in a self-explanatory manner: You can enter a file path in the input box or search for the desired file using **Browse**, then add it to the TWiki page with **Upload**.[18]

Uploaded files appear in the right menu bar of the Kupu Editor. Clicking on one of the files listed there will insert it at the cursor position on the Wiki page.

Adding tables

The table function is just as easy, allowing you to insert simple tables. Again, place the cursor on the spot where the table is to be placed. Select the desired features in the right menu bar under "Tables" and confirm with **Add Table**.[19] If you click on the table in edit view, you can edit features in the menu bar. At the same time, hold points appear that you can pull with the mouse while holding down the mouse button to change the dimensions of the table. When you right click the table, a dialogue box opens with which you can add or delete lines, among other actions.

Granting access permissions

With Kupu and the tool "Permissions," you can administer access rights for the current page in the right menu bar.[20] You can set reading rights ("View Permissions") and edit rights ("Change Permissions"). If, for instance, you click the **Change** button, a dialogue box

---

[18] To delete attachments or use the "hide" function, see Chapter 11.2.

[19] The significance behind the Kupu-specific features "table class," "plain," "listing," "grid," and "data," remain the developers' secret as of this printing.

[20] See Chapter 14 above.

opens in which you can enter the user name and groups that may edit that page, then confirm with <ENTER>. The name is now added to the list. You should check this procedure by switching to the code view using the "Edit Source" button or opening the page not with Kupu, but with the normal **Edit** button.[21] If there are no entries, editing is permissible for all users.

18.4.3 Importing and Exporting other Text Formats

Working with various text formats

Not only in the course of a project is it necessary to juggle several text formats; for instance, you may receive abstracts from lecturers in Word or RTF file format, or you may wish to transfer text from the wiki to your program of events.

Importing and exporting other text formats is a problem wiki technology shares with other systems. The Kupu Editor can be valuable in this regard.

Importing

If you would like to insert text from a Word file or a Writer file (Open Office) into a TWiki page, mark the corresponding text and copy it to the clipboard. Open the respective TWiki page with Kupu and add it to the desired spot with the "paste" function. Text and paragraph formats will remain intact, unless you are importing an RTF text, in which case the formatting is lost. In most cases, minor modification will be subsequently necessary.

Exporting

The same procedure applies if you want to export text from TWiki to a word processing program. Copy the desired text to the clipboard, and then insert it in the target document. Don't forget that, in the event that you have significant problems with this function, you can always attach files to the Wiki page and make them available to download.[22]

---

[21] As of this printing, the program does not always run reliably. You should see the variables SETALLOWTOPICVIEW and/or SETALLOWTOPICCHANGE with the corresponding users.

[22] See: Chapter 11. For importing and exporting Excel tables, see Chapter 18.4.3.

19 Composing the Project Plan

Readers
Authors
WikiAdmins
WebAdmins

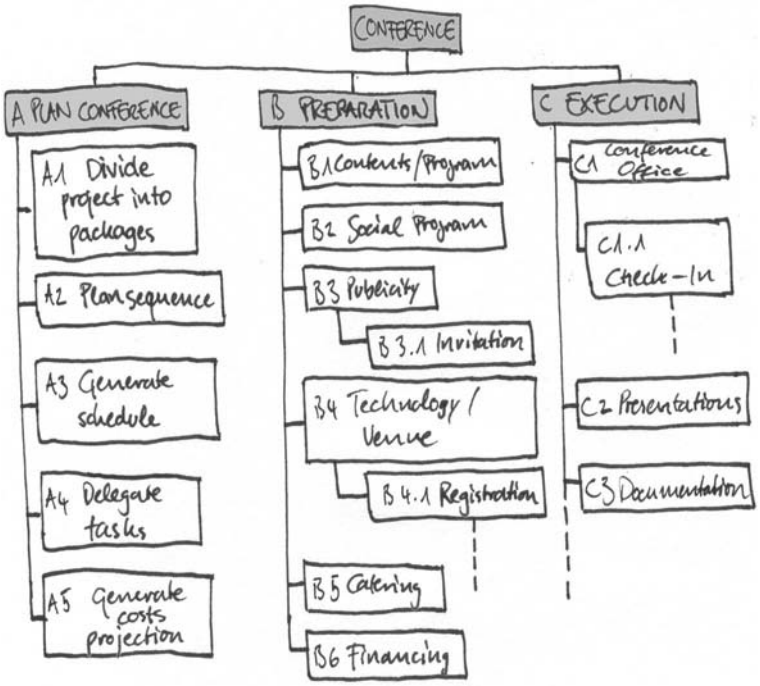
19.1 Preparing Work Packages

Project structure diagram

Once the initiators of the conference "Get Out of the Cave" have conducted basic discussions with friends, colleagues and stakeholders about the execution of the project, concrete individual planning can begin. In traditional project management, the next step involves first developing the project structure. In planning meetings, individual tasks are specified, divided into subtasks and individual steps (work packages) (see Fig. 19.1 and Fig. 19.2). This detailed and hierarchical division will help in the subsequent assessment of how much time is required for the individual project steps and when which tasks are to be completed. Thus, organizers are forced to develop a clear picture of the overall project.[23] Our hunters and gatherers first roughly define which subtasks they must complete in the course of the entire project. Their structure diagram for the conference looks like this:

[23] You can find assistance and ideas on planning work steps in respective technical literature, e.g. Portny 2001, 69-91.

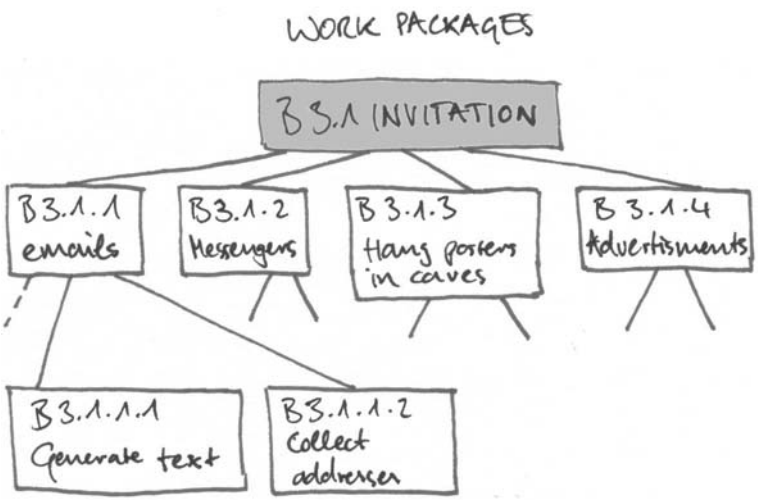
Fig. 19.1



Forms for work packages

With the aid of the project structure diagram, the project is then thought through, step by step, and all necessary work steps are noted in detail. The subtasks are further divided into individual tasks, known as work packages:

Fig. 19.2



Ultimately, with the work packages, they then have small individual steps to be executed for the entire conference.

Forms

To make their work easier, they first designed and copied simple forms, one for each work package. These forms contain

- an exact description of the task,
- what results are required from other tasks in order to perform the respective task (inputs),
- products and results that are produced with this subtask (outputs),
- roles and responsibilities for the package,
- the timeframe of the individual task and
- the required resources.

Often, the direct predecessors and successors of the respective task are noted. The paper forms are ideal for sorting on a wall and subsequently developing the procedure, deadlines and financial plans.

To-do list

A great deal of information can be filtered from these forms: overall cost, material requirements, time schedules, etc. Now TWiki comes into play again. The hunters and gatherers first transfer the work packages to a to-do list that can be viewed by everyone involved in the organization. The groups involved can note the status of their respective subtasks there. This results in an updated report on the status of the project.

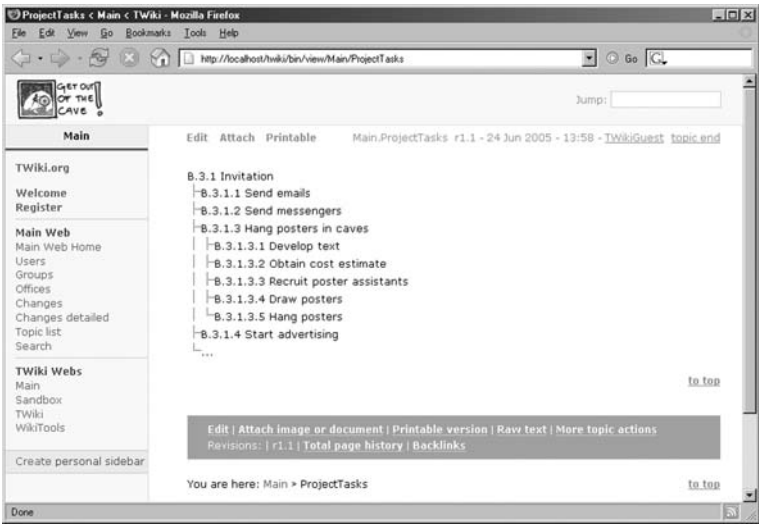
The groups must clarify how detailed the list is to be. A graduated list works well. Numbering can help in order to define precisely which work package is being addressed in discussions. It is possible, for example, that the publicity group and the content group both have a work package called "Send emails."

Note: These numbers must be assigned in a fixed manner and may thus not be generated with the numbered list function.

Project number list

The best thing to do is create a separate wiki page on which the project numbers can be clearly noted. It might look like this:

Fig. 19.3 RenderList plugin



RenderList plugin

The RenderList plugin comes into play for this list, which we will only explain briefly. The plugin, which should already be installed, enables lists to be formatted with either predefined icons or your own. We have selected the standard format `thread`, which we indicate in the variable `%RENDERLIST%`. Our list looks as follows in code view:

Wiki

```
%RENDERLIST{"thread"}%
B.3.1 Invitation
  * B.3.1.1 Send emails
  * B.3.1.2 Send messengers
  * B.3.1.3 Hang posters in caves
      * B.3.1.3.1 Develop text
      * B.3.1.3.2 Obtain cost estimate
      * B.3.1.3.3 Recruit poster assistants
      * B.3.1.3.4 Draw posters
      * B.3.1.3.5 Hang posters
  * B.3.1.4 Start advertising
  * ...
```

19.2 The Project Schedule

Project schedule

Now we turn to scheduling. The project schedule diagram only provides us with a view of the project that is not related to time. Scheduling is generally a complicated matter. It involves the assessment of the duration of tasks and the order of events.

The hunters and gatherers do not necessarily want to work on the individual steps in the stipulated order. Moreover, they are contemplating scheduling tasks such that overloads can be prevented and idle time can be better utilized. This is not a problem from the standpoint of the wiki philosophy, since it refers to the self-organization of mobilized groups.[24] The hunters and gatherers must once again decide for themselves how to proceed.

Note: It is a typical mistake to calculate project schedules "from the back" and thus incorrectly estimate the real time needed or resource requirements. Projects are then not modified in many cases, but rather the data is embellished. Do not be tempted to develop the schedule according to this "backing" method. It is better to do without some project content, since projects tend to need to fill too many expectations at once anyway.

Bar graphs

In our case, we will offer a visual in the form of a bar graph to depict scheduling. The bar graph organizes all elements in their order of performance along a time axis, such that the duration and time differences are also visible graphically. Such graphs are the same kind as those used to depict vacation scheduling in companies.

Unfortunately, TWiki as yet does not offer any solutions to achieve such a bar graph, generated in a group meeting, using the wiki principle. We can only generate the graph in another program and then load it into the wiki system as a finished graphic image. It would not be able to be edited using TWiki.

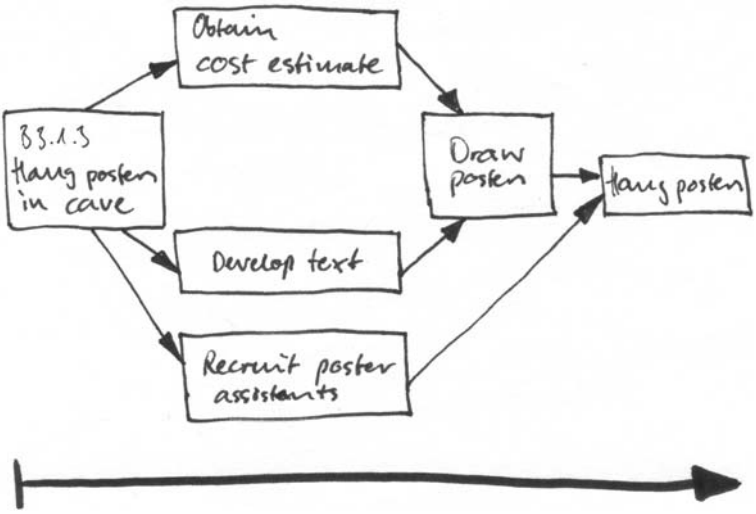
Integrating graphics

By and large, graphics and images are still employed conservatively in wikis. The integration of existing graphics are practical in that they also stimulate the right half of the brain, which is responsible for visual and spatial thought. Images are absorbed quickly, spark curiosity, and can improve clarity. In combination with the text, they can promote new ideas. Our group, for instance, has inte-

[24] This topic would go beyond the scope of this book. We feel it differs depending on the purposes for which it is used. Self-determination is not abandoned simply because planning has begun.

grated images of the planned venue. On their wiki, they also display statistics. It is also conceivable to transfer flowcharts, workflows or mind maps to the wiki. In classic project management, for example, procedural node network plans have been very helpful in depicting work packages and their dependencies (see Fig. 19.4).[25] The next section will explain how to draw these graphic images directly in the TWiki using a flowchart.

Fig. 19.4



Calendar function

Similarly, neither MediaWiki nor TWiki offers a sufficient calendar function that can be integrated into the schedule and used autonomously by the work groups. TWiki does have a calendar plugin, but the effort needed to install it is not worth the functionality it offers. That is why we would need to rely on other programs for this service.

[25] There, procedural node network plans are used to develop project strategy ("network plan technique"). These are graphic or tabular depictions of procedures and their dependencies. More detail on this topic would extend beyond the scope of this book; the illustration supplies a rough idea of the procedure.

19.3
Distributing Tasks and Forming Groups

Distributing tasks and forming groups

In the next step, participants accept a variety of tasks (work packages) and, if necessary, form work groups. Furthermore, it is advisable to appoint responsible persons for subtasks (finances, technology, catering, social program, content, and so on) who can help with the coordination when there are tie-ups, contribute their professional expertise, and keep an eye on the overall project. There are several different models for this job. Responsibility for subtasks can change hands in the course of the project, as is the case with moderation. The important thing is that responsibilities are frequently defined at certain points in time – just as in any democratic system.[26] In the end, all tasks and responsibilities must be clearly distributed so that the project can begin with a prospect for success.

Delegating tasks?

Some proponents of self-organizational processes may object at this point. They make the transfer of tasks dependent upon interests, the "passion" and self-motivation of participants. When this method is employed, tasks that are not assumed are also not delegated. It is true that there is Torvald's principle urging to "delegate everything that can be delegated," but here, the explicit accent is on voluntary assumption of tasks. Himanen goes the same route when he presents the thesis that the hacker ethic is a new work ethic that represents a counterdraft to the "protestant work ethic" of Max Weber.[27] Accordingly, in self-organizing relationships, only those tasks are fulfilled that people want to execute. Thus, it is possible that some subtasks will initially be left undone. For a long time, Linux was to a large extent an insular system for experts, because the needs of the "normal" user were completely ignored. The open-source community refers to the successes of this method, since ultimately all central subtasks were taken on.

We cannot unfold the advantages and disadvantages of self-organized processes here. Each individual group must decide which principles to follow in the distribution of jobs and subtasks. It is obvious that our conference cannot take place if central tasks are not completed on schedule. The ideal of self-determining schedules does not necessarily have to contradict this goal; it depends on the prob-

---

[26] In self-organized systems, "hierarchy" develops through authority that one has worked toward, or responsible persons are elected.

[27] See Himanen 2001, who explains in detail the motivation of hackers and free software projects.

lem.[28] It matters who defines the goals. Thus, in self-organized projects, too, there is a much discussed "inherent necessity" problem and routine work that is not linked to recognition. Also, there are "deadlines" – even if they come about for other reasons.[29]

Setting up group webs

Now work groups can be provided with their own webs, as described above, that they may develop independently from other groups. In our case, let us take a couple of organizers of the Neolithic Conference on the Future. For instance, if Walter has volunteered to get water, and Daniel has offered to catch the mammoths, and Lea has taken charge of the roasting skewers, they can come together in a work group called "Catering," and upon request receive their own web, in which, without worrying about the work style and requirements of other work groups, can develop their file structure as desired. However, we need to differentiate between two types of webs:

- Function webs, e.g. individual webs for registration
- Group webs, e.g. a web for the work group "Catering."

While function webs are available to all participants, insofar as this is possible with regard to data protection regulations, group webs tend to only be accessible for their respective members.

Granting permissions

Traditional content management systems offer very detailed permissions management. In such systems, permissions are often bound to existing functions and hierarchies. According to their original intention, the first wikis had no user administration. In the meantime, newer wiki clones offer possibilities to grant various degrees of user permissions. This can be practical because not every memo has to be open to the whole web, or because private data must also be protected. The hacker ethic puts it as follows: Information should be free, but personal rights and private data are to be protected. When establishing user permissions, you should consider their effect:

---

[28] Whether or not the article "Hacker Ethics" will appear on the Net today or in one month is less of a problem than a lack of staple foods (the "bread roll question").

[29] Beware of the danger of falling back into romantic, pre-capitalistic societal utopias. Further differentiated division of labor and systematic project coordination does not automatically lead to a limitation of human self-determination, as Marx thought when he criticized the fact that relationships created by humans rise above them. To be precise, division of labor is the actual prerequisite to self-determination.

- Some may interpret user rights as resembling a bouncer hindering access at the door. For others, the step of registering is also an instance of conscious decision for active participation in a relationship of work and discussion.
- Rooms that are limited are seldom opened again. Experience has shown that discussions shift to the limited rosters, and the open lists lose significance.

In any event, the dual control system should apply and not only be accessible to a single admin. Supporters, observers, helpers and those concerned with the project should be integrated into the wiki project to as great an extent as possible.

19.4 Outlining Structures and Procedures with the TWikiDraw Plugin

Organization charts

In very dynamic, self-organized projects it is also, and even especially, necessary to appoint contact persons and advisors for certain areas, so that the groups can make arrangements between themselves. We want to combine the sensible with the beneficial, and thus introduce a tool that is very easy to use: the TWikiDraw plugin. Using it, we can create visuals of structures and procedures, for example, such as in the form of workflows and organization charts.

Mode of operation

The plugin is based on a Java Applet that was developed by Peter Thoeny from the original JhotDraw. With it, you can place ready-made lines, shapes and text in a drawing window, similar to MS Powerpoint and MS Word, and supply them with links to pages within or outside of TWiki.

You should first install the plugin, since it is not a standard component of TWiki (see Chapter 16.3). Once you have moved the files to the correct directory, the plugin will be visible in three places on the *TWiki* web:

- on the web homepage under "TWiki User's Guide,"
- on the InstalledPlugins page,
- as the new topic *TWikiDrawPlugin*.

On the page of the plugin, you can test the drawing program and – if desired – make other settings (see below).

Syntax

Now, in order to draw on another page, just integrate the variable `%DRAWING%` to the respective spot in the source code. With the attribute `name`, you can give your drawing a name that, however, should not contain CamelCase. For example:

Wiki

```
%DRAWING{name="Drawing one"}%
```

If only one drawing is planned for that topic, it suffices to simply name the variable `%DRAWING%`.

Starting

In page view, you should now see the following switch:

Fig. 19.5

If you click on the switch, a drawing window opens with the tools you will need to edit your drawing:

Fig. 19.6

Closing

To leave the drawing program and return to the page, press either EXIT WITHOUT SAVING or SAVE AND EXIT in the DRAWING menu, depending on whether you wish to save your work or not. The drawing will now be attached in a frame on the page. If you wish to edit it and thus open the drawing program, click on the drawing. In the event that there are links in your drawing that you may activate by mistake, we recommend clicking on the drawing frame or on the **Edit** button.

Note: There is the usual X icon in the upper right corner of the window, but you cannot close the window with it. To close, you must use a menu option.

19.5
The Tools

On the left side of the interface, there is a toolbar offering the following functions:

Tab. 19.1

Icon	Function
	Selects, moves and edits the clicked object.
	Inserts text.
	Inserts text and attaches it to the object.
	Draws a rectangle.
	Draws a rectangle with rounded corners.
	Draws an ellipse.
	Draws a line.
	Creates a connection between two objects.
	Creates an elbow connection.
	Draws a polygon; the corners are created by mouse clicks.
	Enables freehand drawing.
	Decorates the clicked object with a frame.
	Adds a link to the clicked object.

Hot Spots

The last function listed in the table needs some explaining. In order to turn a figure, for instance a circle, into a so-called hot spot, that is, to supply it with a link, click on the URL icon in the toolbox and then mark the desired object. A small input window opens in which you can enter the URL or page name. When you leave the window by simply clicking on another spot in the drawing window, the path of the link will appear in red. However, this label is no longer visible in the page view.

Note: If you wish to edit or remove the link, you must once again activate the icon and then click on the red path. In the input box, you can edit the link. This applies to text boxes as well.

Attributes

You can change the attributes of an object, such as the fill text color or the shape of an arrow tip, primarily through the entries FILL, LINE and TEXT in the menu bar.

Move/delete/ align

The buttons underneath the drawing window provide a few more important functions, such as cutting, copying, duplicating, deleting and grouping, which are also in the menu. You should have highlighted the respective object first, though.

Hold points

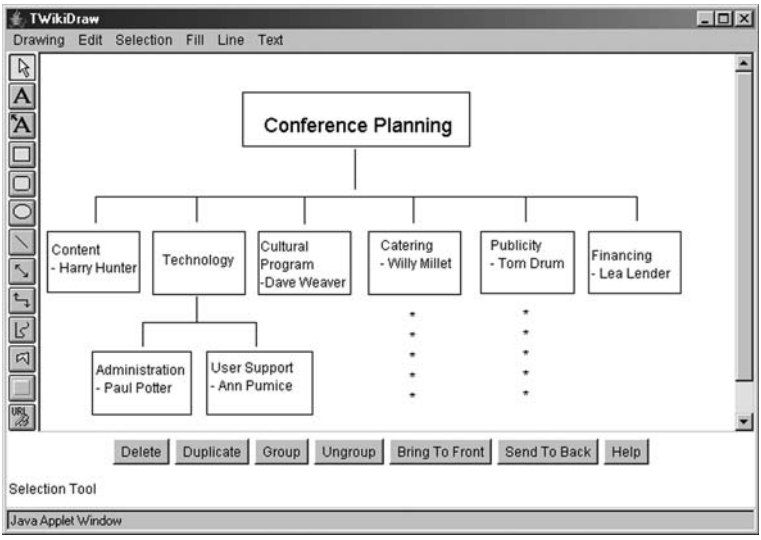
When marking the object, hold points become visible in various colors – depending on the figure. You can manipulate the figure with the mouse on these points:

- **White**. Changes the size of rectangles, lines and ellipses.
- **Yellow**. Changes the font size of text and the degree of rounding in rectangles with rounded corners. Also, the elbow connection lines can be shaped here.
- **Green**. Moves a connection arrow to another object.

Sample organization chart

To be able to present a drawing made with the plugin, we have designed an organization chart for our conference. It shows the various work areas of the conference planning and who is available as a contact person for questions.

Fig. 19.7



On the *TWikiDrawPlugin* page, you will find a few variables that you can set there and then apply globally to the entire TWiki:

Default settings

Tab. 19.2

Variable	Significance
`%EXTRA_COLORS%`	Includes additional colors that can be used in the drawing. Example: `Set EXTRA_COLORS = Aquamarine = #70DB93`
`%EDIT_BUTTON%`	Set to "1", this variable adds an Edit button over every integrated drawing. Example: `Set EDIT_BUTTON = 0`
`%EDIT_TEXT%`	Designs the initial link in a drawing not yet saved. Example: `Set EDIT_TEXT = Edit drawing`

19.6 To-do List with EditTable Plugin

Status of individual tasks

It is a great help when every group can report the status of their work to the other groups. Conversely, groups and individuals need information regarding how far along the parallel tasks have progressed. For instance, the group responsible for recruiting poster assistants has to find out whether or not poster production is running or has slowed, so they can inform the glue mixers of the delay in a timely manner.

In a further table, the current status of the tasks can thus be depicted. The EditTable plugin enables a table to be generated without a great deal of prior skills and without any confusing source code.

Mode of operation

With the aid of the plugin described above, you can enter content to tables in a simple manner. In the page view, nothing much changes at first: A button and a table appear that are connected with the plugin. By clicking on this **Edit** button, a special edit view is generated. However, you do not then get table code written in ASCII text in which you first must search for the correct cell, but rather a user friendly interface with text boxes, drop-down menus and/or date boxes. In addition, there are buttons with which you can conveniently add rows, undo changes or save your work. There can be more than one editable table on a page, but only one of them can be edited at a time.

19.6.1
Formatting the Entire Table

Formatting tables

To create an editable table, first set the variable `%EDITTABLE{...}%` in the corresponding place or insert the variable directly before an existing table. The parameters are now defined within the brackets for the edit view of the table.

Here is a very simple table with only two parameters as an example:

Wiki

```
%EDITTABLE{                                          ↵
  format="| row, -1 | text, 20, init |               ↵
  select, 1, one, two, three, four |                 ↵
  label, 0, %SERVERTIME{"$day $mon $year             ↵
  $hour:$min"}% |"                                   ↵
  changerows="on"                                    ↵
}%
|*No*|*Text Field*|*Drop-Down Menu*                  ↵
|*Time Stamp*|
| 1 | init | two | 07 Jan 2005 14:55 |
```

The page view then looks like this:

Fig. 19.8

No	Text Field	Drop-Down Menu	Time Stamp
1	init	two	07 Jan 2005 14:55

Edit

The same in the edit mode of the table looks like this:

Fig. 19.9

No	Text Field	Drop-Down Menu	Time Stamp
1	init	two	07 Jan 2005 14:55

Save table | Quiet save | Add row | Delete last row | Cancel

In the `%EDITTABLE%` variable, the input options of our table were first defined. Then we entered the content of two lines, namely the header and the first line, directly in the code, although it is not necessary to fill the table with content at the code level. However, the user should at least get an explanation if he or she sees an **Edit** button but no table.

19.6.2 Parameters

The most important parameter for the structure of the table is the `format` parameter. It defines the type and look of the input boxes for the entire table based on one line. All lines newly added follow that established pattern.

Using the remaining parameters, you can define the operating options of the edit view. Here is an overview of all parameters:

Tab. 19.3

Parameter	Description
`header`	Determines the text of the header, e.g. `|*Stone*|*Weight*|`; only matters if the table is still empty and is to be initiated with the **Edit** button.

`format`	The following input boxes are possible (words in italics are to be replaced with the corresponding values):
	■ text box (one line):
	`| text, *size*, *standard value* |`
	■ text box (several lines):
	`| textarea, *rows* x *columns*, <standard value> |`
	■ drop-down menu:
	`| select, *size*, *entry_1*, *entry_2*, etc |`
	■ fixed label:
	`| label, 0, *label text* |`
	■ row number:
	`| row, *offset* |`
	■ calendar box:
	`| date, *size*, *standard value* |`

changerows	If "on", lines can be added and removed. The opposite is "off."
quietsave	If "on", a **QuietSave** button is added.
include	Integrates the %EDITTABLE% variable of another topic. It accesses the first variable of the page cited.
helptopic	Inserts a link to a help page.
headerislabel	Headers are only readable, i.e. cannot be altered when the variable is set to "on."
editbutton	With this, you can define the text on the **Edit** button, e.g.: "Edit this table."

Here is another fact about the format parameter date: When you utilize this input option, the user can navigate within a calendar to select the desired date.

Escape symbol

If there are variables in the standard value parameters, they will be replaced with every save. For instance, you can use this for the time stamp (see above). If you do not want the values to be replaced, mask the critical symbols. Thus

Wiki
```
%SERVERTIME{"$day $mon $year $hour:$min"}%
```

becomes

Wiki
```
$percntSERVERTIME{$quot$dollarday $dollarmon ↵
 $dollaryear $dollarhour:                    ↵
 $dollarmin$quot}$percnt                     ↵
```

Let us briefly review:

Tab. 19.4

Code	Replaces:
$quot	Double quotation marks
$percnt	Percent sign
$dollar	Dollar sign
$nop	Prevents the dollar sign variable from being replaced.

In addition, it is possible to place the "literal" % variables in a drop-down menu, for instance, by supplementing them with a <nop> after the percent sign, such as in

Wiki
```
select,1,%<nop>X%,%<nop>Y%.
```

As you can see from the above example, in the table edit mode, several buttons are automatically added, some of which you recognize from the normal edit mode. You can influence a few of them via the `format` parameters:

Button	Description
Save table	Save table.
Quiet save	Save without notifying another WebNotify user.
Add row	Insert row.
Delete last row	Eliminate last row.
Cancel	Return to page view without saving.

Tab. 19.5

Example: To-do list

Armed with this knowledge, we can now design our to-do list. Here, we enter the task and responsible contact person under the respective project number. Furthermore, there are input boxes for the scheduled start, scheduled end, status, and estimated end of a task.

Wiki

```
%EDITTABLE{
  header="|*Project no.*|*Responsible*|      ↵
   *Task*|*Planned Begin*|*Planned          ↵
   End*|*Status*|*Finished by*|"             ↵
  format="|text,10|text,15|textarea,3x15|    ↵
   date|date|select,1,not begun,             ↵
   begun, stopped, finished | date|"         ↵
  changerows="on"                            ↵
}%
```

Project no.	Responsible	Task	Planned Begin	Planned End	Status	Finished by
B.3.1.3.1	Tom Drummer	Develop text	2005/04/11	2005/05/06	begun	2005/04/22
B.3.1.3.2	Tom Drummer	Obtain cost estimate	2005/04/01	2005/04/15	not begun	2005/04/08
B.3.1.3.3	Tom Drummer	Recruit poster assistants	2005/04/01	2005/04/29	begun	2005/04/22
B.3.1.3.4	Olga Breakrock	Paint posters	2005/05/02	2005/05/13	not begun	2005/05/19
B.3.1.3.5	Homer Sapient	Placard work	2005/05/16	2005/06/01	not begun	2005/05/27

Edit

Fig. 19.10

19.6.3
Determining the Format of Individual Cells

Up to now, formatting commands made in `%EDITTABLE%` applied to the entire table and thus for all cells. However, if you want to format an individual cell differently, e.g. if you want to define another input box, you can overwrite the table formatting with the variable `%EDITCELL{"..."}%` .

To do this, set the variable directly after the content of the cell, e.g. to define a text field:

Wiki
```
| cell content %EDITCELL{"text, 20"}% |
```

All of those input boxes are supported that you also recognize from the `format` parameter.

Caution: The `%EDITCELL%` variable only functions in connection with `%EDITTABLE%` .

Cell formatting is especially useful if your table is more vertical than horizontal and, for instance, consists of so-called key/value pairs. Here is a brief example:

Wiki
```
%EDITTABLE{ format="| label | text, 40 |" }%
|*Key*|*Value*|
| Name: | Jane Public |
| Gender: | F                                    ↵
  %EDITCELL{select,1, ,F,M}% |
| Occupation | Gatherer                        ↵
  %EDITCELL{select,1, ,Gatherer,Hunter}% |   ↵
| Town of residence: | Marble Valley |
```

Table edit view:

Fig. 19.11

Key	Value
Name:	Jane Public
Gender:	F
Occupation	Gatherer
Town of residence:	Marble Valley

Save table | Quiet save | Add row | Delete last row | Cancel

Problem zones

However, in a few ways, the plugin is stretched to its limits. We will name a couple of them here:

- It does not support a few formatting options, such as multi-span (connected) cells (| ... ||) and text alignment values within the cell (e.g. | centered | right |).
- From about the fifth row on, there could be processing problems.
- It is not possible to include two %EDITTABLE{}% expressions within a single code line.

Global settings

As is typical of all other plugins, you can also make global settings on the *EditTablePlugin* page regarding the following variables:

Variable	Significance
%CHANGEROWS%	Opportunity to delete or remove lines. Example: Set CHANGEROWS=on
%QUIETSAVE%	Defines whether a **QuietSave** button should be set as a default. Example: Set QUIETSAVE=on
%EDITBUTTON%	Labeling of the **Edit** button. Example: Set EDITBUTTON=Edit table
%JSCALENDAR DATEFORMAT%	Calendar format Example: Set JSCALENDARDATEFORMAT= %d.%m.%Y
%JSCALENDAR LANGUAGE%	Calendar language Example: Set JSCALENDARLANGUAGE=de

ProjectPlaner plugin

For internal use, TWiki offers its own ProjectPlaner plugin. In a separate web, report pages are generated for every project and subtask. Team members provide information on their project page regarding the status of the subtasks and work packages (project begun, estimated duration, time already invested…). Using this information, an overview page is generated that presents a brief summary of the status of the individual tasks using simple bar graphs. Due to the current stage of development, we will not go into this plugin any further, because we think that the effort required to install it does not merit its worth. You can largely generate its functions yourself using a few central overview tables.

19.7
Planning Costs and Financing with Tables (Spreadsheet Plugin)

Determination of requirements and cost projection

Using the estimated materials requirements of the individual work packages, the overall requirement for the project can be calculated. We can now generate a cost projection plan for our conference. Similarly, the requirements of the individual cost units can be determined; that is, we can estimate what expenses the "Catering" group or the "Content Planning" group will have. These groups can then manage their funds themselves.

Who, what, when

For smaller projects, such an overall cost projection, split into cost units, is completely satisfactory. In the case of larger projects, further temporal overviews are necessary. When are which expenses due? When will which personnel be required? What equipment and materials must be available where and when? Technical literature cites a variety of terminology (financial plan, operating resources plan, etc.) and methods. However, we wish to warn you that the limits of reification of human relationships can be quickly exceeded. Bureaucratic control concepts of "human resources," for instance, express themselves in the fact that said human resources are evaluated according to their work capacities, which need to be optimized and controlled.

Spreadsheet

Spreadsheets are an invaluable aid in maintaining an overview of finances. Although wikis cannot provide a highly developed spreadsheet program, its Spreadsheet plugin nevertheless enables us to work reasonably with tables in TWiki. Once again, the advantage of a wiki is that the data can be collected centrally in a document that is available to everyone and can be independently updated by the individual project groups. Thus, in the area of financial planning and control, new trails can be blazed with regard to transparency and self-determination.

Mode of operation

This plugin is included in the standard installation. It equips TWiki with conventional spreadsheet options. You can perform calculations based on entered formulas or existing functions, process strings – for instance, turning upper case letters into lower case – and formulate logical queries. To do this, in the edit view, add variables to your calculation that are replaced by solutions in the page view. This works within as well as outside of the table, which basically distinguishes this plugin from others. For example, you can display the individual items of a table column as a sum in a continuous text.

19.7.1 Syntax

The plugin is triggered by the variable `%CALC{"..."}%`. This variable can be situated within a table cell or in a normal continuous text, whereby several functions only make sense in connection with tables. The result of the function between the quotation marks is issued when the page is opened. For example, let us say you wish to calculate the average number of participants of three workshops visited by 14, 21 and 12 people respectively:

```
%CALC{"$AVERAGE(14,21,12)"}%
```
Wiki

In the page view, the result 9 is displayed.

Note: Calculations are always based on functions. Even an easy expression such as 1+1 must be expressed as a function: `%CALC{"$EVAL(1+1)"}%`.

Functions

Functions have the following schematic structure:

```
$FUNCTIONNAME(parameter)
```
Wiki

Its interpretation is done from left to right. It is possible to nest several functions, e.g.:

```
%CALC{"$AVERAGE(14,21,$SUM(4,8))"}%
```
Wiki

whereby the calculation in this case is done from the innermost to the outermost function.
The parameters of a function can be depicted as text, a mathematical formula, a cell address or cell area. This depends on the function itself. So, for instance, the sum function processes concrete values or information in cell areas.

Cell reference

If you wish to refer to the possibly changing content of fixed table cells, you can do so via the following coordinates:

R1:C1	R1:C2	R1:C3	R1:C4
R2:C1	R2:C2	R2:C3	R2:C4
R3:C1	R3:C2	R3:C3	R3:C4

Fig. 19.12

The R stands for row and the C for column.

To now be able to calculate with the content of one of these cells, you must enter the cell coordinates in the function `$T`:

```
%CALC{"$AVERAGE($T(R2:C3),$T(R3:C4))"}%
```

Caution: The convention of cell references only works in pipe tables, but not in HTML tables!

Table range

It is also possible to include an entire table range by indicating each of the top and bottom corner cells and separating them by two periods (“`..`”); e.g. `R1:C1..R3:C3` would completely include the first three columns of the above table.

In this case, it is not necessary to use the T function; you can transfer the table range directly as a parameter:

```
%CALC{"$SUM(R1:C1..R3:C2, R1:C4..R3:C4)"}%
```

Note: The cell data in the formulas can only relate to the cells in the current row and previous cells in the table. All cells below the current row cannot be addressed with that formula. If you insert a formula with cell data outside of the table, they will refer to the previous table.

Lists

If a parameter consists of several elements (of the same kind), e.g. in the calculation of an average, a list of values is transferred whose components are separated by commas, such as `%CALC{"$SUM(3, 5, 7)"}%`. Also, there are a few functions that were conceived specifically for editing and processing lists. Thus, for instance, the function `LIST` can be used to transform a cell range into a list and then further processed.

If it doesn't work…

When using the plugin, you may frequently make mistakes, especially at the beginning, since you first have to get used to the conventions of the program. However, the causes of the following undesired results can be found relatively quickly:

- **In the page view, the code of the formula is displayed but not calculated**: The plugin is probably either incorrectly installed or not installed at all.
- **Parts of the formula are displayed**: The syntax of the formula is incorrect. Perhaps a dollar sign was left out?
- **A 0 (“zero”) comes out as a result**: Your cell references may be referring to an HTML table.

Note: Remember that command lines in the plugin must have a period, not a comma. Thus, it would be 3.14 and not 3,14.

19.7.2 Important Functions

The plugin contains about 65 functions, the sum of which we unfortunately cannot present here. However, we will describe the most important functions as follows.

Caution: Please do not forget that when using a function, the dollar sign must be set before the name!

All examples with cell data refer to the following table:

	Content	Catering	Publicity
Human Resources Expenses	-5000	-1000	0
Equipment and Material Costs	-500	-3000	-2000
Income	0	5000	100

Fig. 19.13

ABOVE()
Addresses the cell range above the current cell.
Example: `%CALC{"$SUM($ABOVE())"}%`
Result: Displays the sum of the values above the current cell.

ABS(*number*)
Absolute value of a number.
Example: `%CALC{"$ABS(R3:C2)"}%`
Result: 1000

AND(*list*)
Logical And of a list.
Example: `%CALC{"$AND(1, 0, 1, 1)"}%`
Result: 0

COLUMN(*offset*)
Return of the column number under consideration of a possible offset.
Example: `%CALC{"$COLUMN(2)"}%`
Result: current column number +2

EVAL(*formula*)
Calculation of simple operations such as sum, difference, product, division and modulo. Nesting is also possible. The general rules of calculation apply.
Example: `%CALC{"$EVAL( (5/2) *3 - 1.2 )"}%`
Result: 6.3

EXACT(*text1, text2*)
Comparison of two character strings; returns 1 when both are the same.
Example: `%CALC{"$EXACT(cave, cove)"}%`
Result: 0

EXISTS(*page*)
Tests of *page* exists or not.
Example: `%CALC{"$EXISTS(DoesNotExist) "}%`
Result: 0

IF(*condition, value if true, value if false***)**
Branching using a condition.
Example:
```
%CALC{"$IF($T(R2:C2) > 0, profit,
loss)"}%
```
Result: loss

LIST(*range*)
Transforms a cell range into a list.
Example: `%CALC{"$LIST(R1:C2..R1:C4)"}%`
Result: Content, Catering, Publicity

LISTSORT(*list*)
Alphabetical sorting of a list.
Example: `%CALC{"$LISTSORT(R1:C2..R1:C4)"}%`
Result: Catering, Content, Publicity

MAX(*list*)
Returns the largest value of a list.
Example: `%CALC{"$MAX(R2:C2..R3:C4)"}%`
Result: 0

MOD(*number, divider*)
Modulo operation; returns a division remainder.
Example: `%CALC{"$MOD(15,4)"}%`
Result: 3

PROPERSPACE(*text*)
Inserts blank spaces in a WikiWord.
Example: `%CALC{"$PROPERSPACE(WebHome)"}%`
Result: Web Home

ROUND(*number, decimal places*)
Rounds off floating-point number to the number of decimal points indicated.
Example: `%CALC{"$ROUND(2.323256, 2)"}%`
Result: 2.32

SUM(*list*)
Sum of a list.
Example: `%CALC{"$SUM(R2:C2..R4:C2)"}%`
Result: -5500

T(*cell reference*)
Returns the cell content.
Example: `%CALC{"$T(R1:C2)"}%`
Result: Content

TODAY()
Returns the current date (server time).
Example: `%CALC{"$TODAY()"}%`
Result: current date

UPPER(*text*)
Transforms text to upper case.
Example: `%CALC{"$UPPER(Mammoth)"}%`
Result: MAMMOTH

Default settings

You can naturally also define default settings in the Spreadsheet plugin as well. Here are the variables you will need:

Tab. 19.6

Variable	Significance
`%SKIPINCLUDE%`	In integrated pages, the `%CALC%` variable is not processed. Example: `Set SKIPINCLUDE=1`
`%DONTSPACE%`	Indicated pages are excluded from the `PROPERSPACE` function. Example: `Set DONTSPACE=WikiWord`

19.7.3
Exporting/Importing Excel Files

When you work with tables in TWiki, sooner or later you will want to know whether you can transfer tables between office programs, such as MS Office or Open Office, and TWiki. There are several methods with which to do so. We would like to present a very simple program for that purpose that will enable you to

- import tables from office programs, HTML editors and texts structured via tabs to TWiki,
- conversely, export TWiki tables to office programs,
- perform table formatting and sorting tasks within the program.

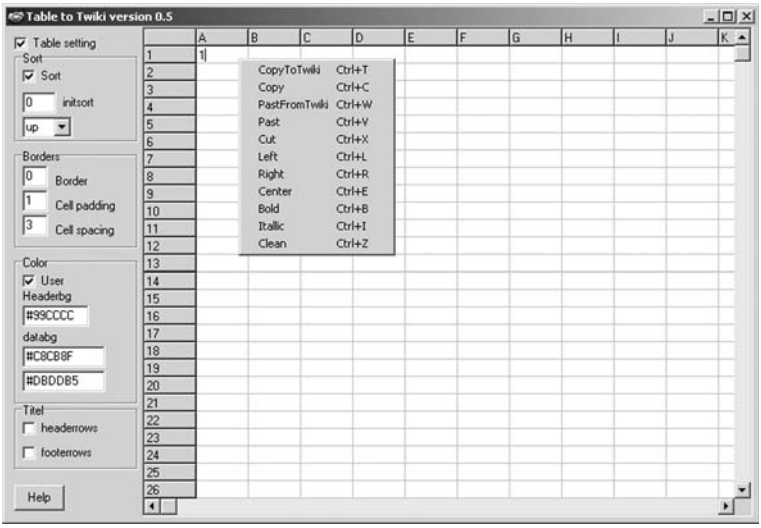
Installation

The program, written in Delphi, presently only runs under Windows. You will find it at http://www.twiki.org in the Plugins web under the Add-Ons as "Copy Table from/to Spreadsheet/table Programs Add-On." Download the file *CopyTWiki.zip* and unpack it to a local directory on your computer. When you then open the .exe file in the folder created, the program's interface will appear.

Mode of operation

The spreadsheet that you now see in front of you functions as a sort of clipboard, and is the interface between the various formats. For instance, you can copy your Excel table onto the sheet, edit and format it as desired and then copy it to your TWiki – or vice versa.

Fig. 19.14



The context menu plays an important role in the program; that is, the menu bar that drops down when you press the right mouse button. There, you will find the various transfer actions as well as a few formatting options. You have the following possibilities:

Tab. 19.7

Menu Entry	Function
COPYTOTWIKI	Copies the highlighted cells to paste to the TWiki.
COPY	Copies the highlighted cells to paste to an office program.
PASTEFROM TWIKI	Inserts cells that were copied from the TWiki source code.
PASTE	Inserts cells that were copied as an HTML table (not in the source code).
CUT	Cut out.
LEFT	Left cell alignment.
RIGHT	Right cell alignment.
CENTER	Centered cell alignment.
BOLD	Bold.
ITALIC	Italic.
CLEAN	No format.

On the left, you have further options for formatting and sorting by activating "Table Settings;" these include background color and cell spacing. However, they are largely self-explanatory.

20 Preparing for Your Event

Readers
Authors
WikiAdmins
WebAdmins

20.1 Planning Your Event

Public relations

The preparations for your conference are already in full swing. The lecturers have been invited, the program put together, and advertising has begun. TWiki will help keep all participants up to date. It serves as a bulletin board. The forte of wikis for project organization lies precisely in this double duty: It is an organizational tool as well as the basis of public relations and the integration of the outside world, e.g. by enabling the external viewing of spatial planning. In our specific case, it is helpful to involve participants as "collaborators" at an early stage.

Accordingly, you can present further web offerings:

- Separate discussion forums
- Feedback pages
- Carpooling and accommodation exchanges

Abstracts

This dual function can also be utilized to have lecturers integrate their abstracts into the pages themselves. Also, participants can write their questions and ideas right underneath, so that lecturers can better attune themselves to their queries.

20.2
Conference Registration

Registration procedure in the TWiki

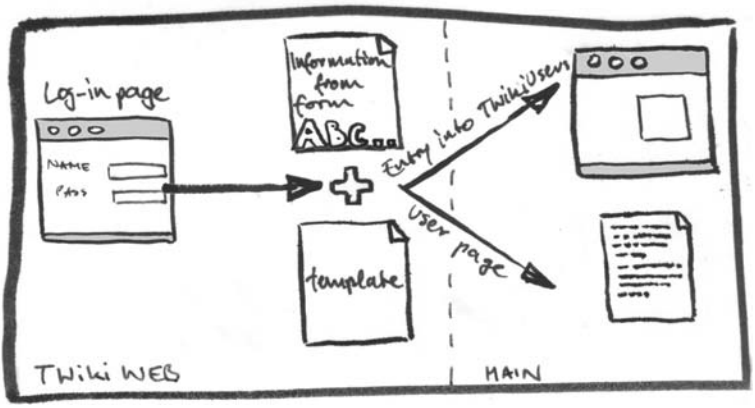
A few weeks before the conference, participants should be able to register for the conference over the Internet. Unfortunately, TWiki lacks a type of form plugin with database integration. To enable our visitors to register for the conference, we are limited to existing opportunities, and we can adjust them a bit. We will create a Registration web that serves only to present the registration form and safely store participants' data. Based on this data, various queries (or calculations) can be generated that a database management system would ordinarily perform. The suggested solution is only one of several options. We wish to demonstrate that you can solve a number of problems in TWiki by using the existing software in a creative manner.

20.2.1
What Happens in a Standard Registration?

Registration procedure

Before going into more detail on creating the conference registration, we should first clarify how a normal TWiki registration process works. This illustration will elucidate the procedure somewhat:

Fig. 20.1



Firstly, you register in the *Main* web. The data indicated is integrated into a template. This combination is stored as a user page in the *Main* web. In addition, an entry is made to the overview of all users (*Users* Topic).

20.2.2 The *RegistrationWeb*

Modifying the registration mechanism

Since it is not possible to generate a normal form with data integration using PHP/MySQL, we will now "misuse" the user registration form normally located in the *Main* web and the registration mechanism upon which it is based for our own purposes:

- First of all, we will generate a new web ("RegistrationWeb") that is to contain the registration form and the topics of the visitors and new users. The form should be excluded from searches using the "Exclude from search all" option, so no one can find out who is taking part in the conference.
- Then, the regular user registration page is copied to the new web and adapted to fit the requirements of the conference. The user registers for the conference as well as for TWiki itself, meaning a user page will also be generated for him or her.
- Next, it is important to make a few changes to secure the data in the user registration template and regulate navigation such that everyone can conveniently access his own data but not that of others.
- While a visitor is filling out the registration form for the conference, if he or she makes a mistake, the corresponding error messages should be displayed. Accordingly, if registration has been successful, a confirmation page should be presented. The templates for these pages should also be able to be modified as desired. For example, if your site is bilingual or being presented in another language, you would want a translation of that page to appear in the corresponding language.

20.2.3 Designing the Registration Form

Copy form

After you have generated the new web with the aid of the form on the *TWiki.ManagingWebs* page (see Chapter 15.1), transfer the user registration form *TWikiUserRegistration* (in the *TWiki* web) into our new web. To do so, copy the source text of the form to a new page.

That gives us two registration options within TWiki: one is the standard user registration in the *Main* web, which generates a page for the user and allows him or her to make modifications in the

TWiki; the other registration option is done in the new web, and registers the user for the conference as well as for the TWiki.

Note: The standard registration could be used by users who move about within the environment of the conference but do not belong to a group of official visitors and thus do not need to provide any information regarding prices, accommodation, etc., such as assistants. Thus it should be possible for these people to not need to register.

General design

To a great extent, you can now modify the copied form as you wish.

We can go into the edit view and alter the layout by adding a fitting heading, explanatory text, or images such as a logo. Any language changes should be made here. Formatting is done according to the standard TWiki rules.

Adapting input fields

Then it is necessary to alter the form input fields according to requirements, i.e. fields not needed should be removed and any additional ones inserted. As you can see in the edit view, the form syntax is pure HTML. The values entered are stored in the `name` variables that you indicate in the respective HTML tags. However, there are two things to remember. First of all, you cannot delete all input fields: "WikiName," "Name," "Password," and confirmation of the password should definitely be part of the form, which is sensible for the registration process. If you delete the fields or alter the defined names, the visitor will receive an error message when trying to register, and the registration will fail.

Secondly, we must differentiate between optional and obligatory input fields.

Setting up obligatory input fields

The optional fields may be, but do not have to be, filled out. The obligatory fields, on the other hand, are usually displayed in the page view with two red stars, and must be filled in by the user to facilitate registration. As the designer of the form, you can achieve this differentiation by beginning the names of the variables for obligatory fields with `Twk1` and with `Twk0` for optional fields. Thus, a registration page for our conference might look like this:


```
<form action="%SCRIPTURL%/register                    ↵
 %SCRIPTSUFFIX%/RegistrationWeb/%HOMETOPIC%" ↵
  method="post">
<table border="0">
<tr>
  <td align="right"> Your <nop>WikiName:
</td>
  <td><input type="text"                              ↵
             name="Twk1WikiName"                      ↵
```

```
            size="40" />
       =<font color="red">**</font>= </td>
</tr><tr>
  <td align="right"> Password: </td>
  <td><input type="password"                ↵
            name="Twk1Password"             ↵
            size="40" />
      =<font color="red">**</font>= </td>
</tr><tr>
  <td align="right"> Repeat password:
     </td>
  <td><input type="password"                ↵
            name="Twk1Confirm"              ↵
            size="40" />
       =<font color="red">**</font>= </td>
</tr><tr>
  <td colspan=2><hr></td>
</tr><tr>
  <td align="right"> First name: </td>
  <td><input type="text"                    ↵
            name="Twk1Firstname"            ↵
            size="40" />
      =<font color="red">**</font>= </td>
</tr><tr>
  <td align="right"> Last name: </td>
  <td><input type="text"                    ↵
            name="Twk1Name"                 ↵
            size="40" />
      =<font color="red">**</font>= </td>
</tr><tr>
  <td align="right"> Street: </td>
  <td><input type="text"                    ↵
            name="Twk1Street"               ↵
            size="40" />
      =<font color="red">**</font>= </td>
</tr><tr>
  <td align="right"> Town: </td>
  <td><input type="text"                    ↵
            name="Twk1Town"                 ↵
            size="40" />
      =<font color="red">**</font>= </td>
</tr><tr>
  <td align="right"> Postal code: </td>
  <td><input type="text"                    ↵
```

```
              name="Twk1Zip"                    ↵
              size="40" />
        =<font color="red">**</font>= </td>
</tr><tr>
  <td align="right"> Email address: </td>
  <td><input type="text"                        ↵
              name="Twk1Email"                  ↵
              size="40" />
        =<font color="red">**</font>= </td>
</tr><tr>
  <td align="right"> Fee: </td>
  <td> 30 Euro (discounted)
        <input type="radio"                     ↵
              name="Twk1Fee"                    ↵
              value="30 Euro (discounted)" />
        50 Euro
        <input type="radio"                     ↵
              name="Twk1Fee"                    ↵
              value="50 Euro" />
  <font color="red">**</font> </td>
</tr><tr>
  <td valign="top" align="right">  
        <input type="hidden"                    ↵
              name="TopicName"                  ↵
              value="%TOPIC%" /></td>
  <td><input type="submit"                      ↵
              value=" Send " />   
        (Fields,marked with
        =<Font color="red">**</font>=
        are obligatory)</td>
</tr>
</table>
</form>
```

Visual identification of obligatory fields is not done automatically, so you should add it. Otherwise, however, you are free to design the page as you like. For example, you do not have to limit yourself to simple text input fields, but can utilize the entire palette of form fields offered by HTML, such as radio buttons, drop-down lists, etc.

Here, you can see the finished registration form for our Conference on the Future, for which a Mr. Stonecake would like to register:

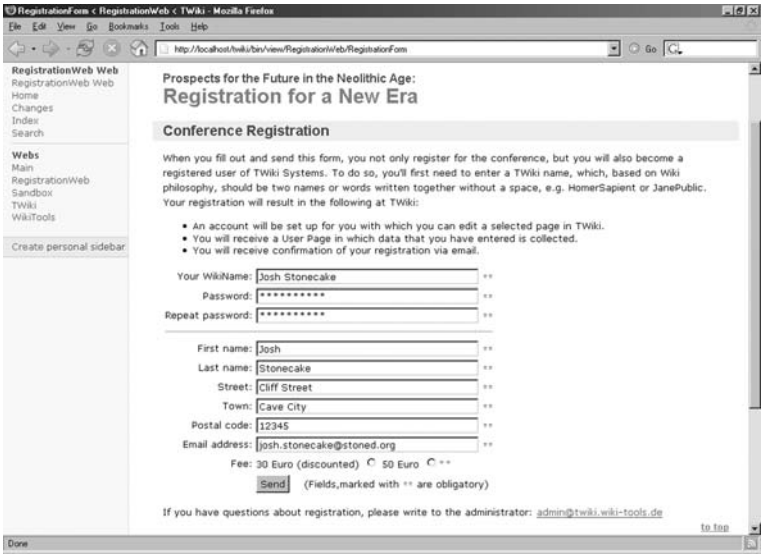


Fig. 20.2

20.2.4 Saving and Protecting Data

Preventing search function access

Now we need to make a few more small changes to ensure that the data entrusted to us will really be saved to the *RegistrationWeb* and be protected from access by third parties. You have already performed one of these measures when you set up the web and activated the "No search all" option. This step guarantees that the data in that web can no longer be searched. If you have forgotten to perform this step, you should do it subsequently in *WebPreferences*.

Note: The web will now no longer be automatically included in the left menu bar. If you need to access the web quickly, you should insert a link to it somewhere.

To make sure our user pages really land in the "hidden" *RegistrationWeb* and not in the *Main* web, we must go to edit mode and change the content of the `action` variables of the `<form>`-tags from `%MAINWEB%` to the name of the new web, e.g.:

Wiki

```
<form action="%SCRIPTURL%/register                    ↵
%SCRIPTSUFFIX%/RegistrationWeb/%HOMETOPIC%"           ↵
method="post">
```

Define user permissions in the template

The user pages will now be stored in the *RegistrationWeb*, but they can still be read by everyone.

To allow access only for the respective user and the administrators, user permissions must be redefined in the template itself that TWiki uses to generate the user pages. The template is located under the name *NewUserTemplate* in the *TWiki* web. On this page, we need to limit access permissions to the new user and administrations via the variables that regulate editing and viewing pages. The user name is stored in the variable `%WIKIUSERNAME%`. In order to continue to allow the template to be utilized by everyone and not only by users already registered in the TWiki, do not forget to add the `%NOP%` variable that will only then be removed once the user page has been generated. That variable prevents `ALLOWTOPICVIEW` from being identified as a switch and is temporarily disarmed. `%NOP%` is removed when the page is integrated and thus activates all access limitations:


```
···*·Set ALLOW%NOP%TOPICCHANGE =                ↵
         %WIKIUSERNAME%, Main.TWikiAdminGroup

···*·Set ALLOW%NOP%TOPICVIEW =                  ↵
         %WIKIUSERNAME%, Main.TWikiAdminGroup
```

Save user page in Main web

The individual user pages are thus sufficiently protected; however, the registered persons are still listed on the *Users* page in the *Main* web. To prevent general access to this list, you should use the `%ALLOWTOPICVIEW%` variables to make this page only accessible to administrators.

20.2.5 Navigation

With regard to data protection laws, we are now on the safe side. However, since we have locked so many pages in a variety of ways, we have to think about how users as well as administrators can access their own data.

Expand menu bar

The *RegistrationWeb* as well as the user pages contained therein are no longer accessible to the public and are not included in the list of webs. In order for users to view their own data, we have to set a link to the *RegistrationWeb*. The registration form must also be made available in the *Main* web so that people can find it quickly. Thus, we will set the registration page and the link to the RegistrationWeb in the left menu bar of the *Main* web. We will name the

link to the *RegistrationWeb* "Participants," since we wish to guide visitors to their pages; because once a user is registered and logged in, he lands on the *WebHome* page, where a link with the name "Your Page" is to be added with the variable `RegistrationWeb.%WIKIUSERNAME%`. The variable is then replaced by a link to the page of the logged-in user.

If the user is not yet logged in, the link of course will not work. The user would be transferred to the *TWikiGuest* page. That is why the user must be forced to log in. We can achieve this by blocking the *WebHome* of the *RegistrationWeb* for anyone not logged in. Then, the login window opens automatically. Here, the user can log into the system or be transferred to the registration form via a link. That link still points to the original TWiki registration page. To change it, you need to edit the template of the login message. You can find out how in the next section.

20.2.6 Confirmation Page/Email and Error Messages

Translating error messages

When a visitor sends a registration form but has not filled in the obligatory fields completely or correctly, TWiki generates the corresponding error messages. The templates for them are stored in the *twiki/templates* folder. They begin with "oops" and have the extension *.tmpl*.

Tab. 20.1

Files in *twiki/templates*	Error
`oopsregexist.tmpl`	User already exists.
`oopsregwiki.tmpl`	Name is not a Wiki name.
`oopsregpasswd.tmpl`	Password incorrectly confirmed.
`oopsregemail.tmpl`	Invalid email address.
`oopssendmailerr.tmpl`	Confirmation email was not sent.
`oopsregerr.tmpl`	Other error.

The confirmation messages are also sent via the "oops" templates. You can reformulate these templates in a regular editor, for instance, if you need to translate the messages into another language.

Caution: Depending on the skin selected, the templates may have been overwritten by specific templates. For instance, in the *Pattern* skin, the file *registernotify.pattern.tmpl* is used.

The template for the login message cited above is called *oopsaccessview.tmpl*. Here, you can set a link to the registration form for users not yet registered. Just add the link after the following line:

Wiki

```
You are not yet authorized to view this page:
<nop>%WEB%.<nop>%TOPIC%.

Please register at: Registration-
Web.RegistrationForm
```

Registration confirmation

If all entries have been made correctly, a confirmation page will be opened and an email sent to the user. The confirmation page is based on the template *oopsregthanks.tmpl*, which you could reformulate as follows:

Wiki

```
%TMPL:INCLUDE{"TWiki"}%

%TMPL:DEF{"titleaction"}%(note) %TMPL:END%

%TMPL:DEF{"webaction"}% *Confirmation*
  %TMPL:END%

%TMPL:DEF{"heading"}%Thank you for register-
ing!%TMPL:END%

%TMPL:DEF{"message"}%
···* You can view and edit your information
on your personal TWiki page %TOPIC%.
···* A confirmation email has been sent to
%PARAM1%.
%TMPL:END%

%TMPL:DEF{"topicaction"}%
[[%WEB%.%TOPIC%][OK]] %TMPL:END%

%TMPL:P{"oops"}%
```

Mr. Stonecake has registered for the conference. Upon successful registration, the following page will be displayed:

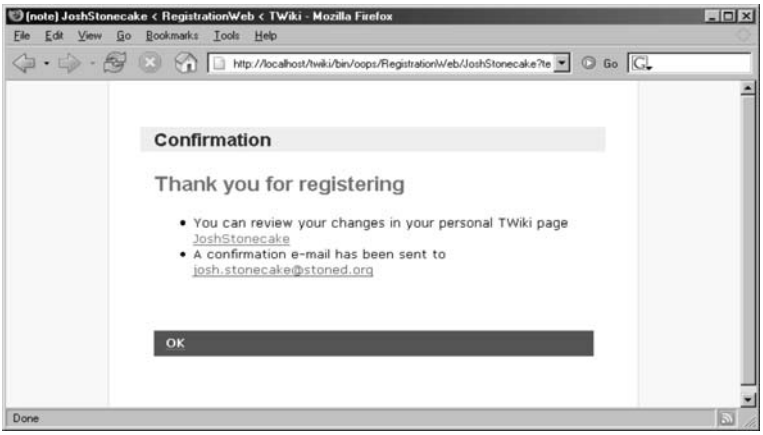


Fig. 20.3

You can edit the content of the confirmation email in the same way in the file *registernotify.tmpl*.

The registration is thus complete. We will return to this topic in the next chapter under "Checking in," within the context of participant administration.

21 Executing and Documenting an Event

Readers
Authors
WikiAdmins
WebAdmins

21.1 In the Event Office

Crisis management

With the start of the conference, the organizers have their hands full to ensure that the lectures, workshops and podiums run smoothly. In the process, they have set up the usual event office on site. Despite all of the preparation, improvisation and continual crisis management are now the order of the day. People cannot find their rooms, there is no change left in the cash box, one of the beamers has disappeared, and one of the lecturers has forgotten to bring the stone slabs she needs for her presentation. Since people at the conference will be speaking directly with each other, the computer takes a back seat as a means of communication. However, we would still like to point out a few aspects of TWiki that can be of assistance in executing your event.

Bulletin board

At first, the wiki is the central bulletin board for organizers and participants alike. The event office, for instance, can announce room changes via the wiki by projecting the program on a wall with a beamer. Participants can find out about the current event program at home or on the go. The current nature of this function makes wiki not only a publication medium, but also the central organizational medium. Any assistant can easily maintain an overview of room assignments with a corresponding EditTable plugin.

Note: Remember that additional assistants – such as during check-in – should be introduced to the software in a timely manner.

Workshop TWikis

After the workshops, participants can immediately supplement texts and publish their results. For instance, in "open space" conferences, short summaries are written directly after an event, which are posted in a central location. This makes it easier to pick up on ideas from other events. Furthermore, it offers the possibility for participants to centrally collect addresses and exchange ideas via the wiki.

21.2 Participant Administration/Registration Status

Checking in

The bottleneck of a conference is the check-in. Participants must definitively register, pay the fee and receive conference materials.

Adjusting the RegistrationWeb

To be able to complete as many of the formalities of the registration office as possible in TWiki, let us have another look at our *RegistrationWeb*. As a reminder, the registration form and user pages of registered participants are located on that web. It will now be expanded by three more pages: *InternalAdministration*, *Receipt* and *ParticipantList*. They primarily serve to take the load off of employees in registering the expected rush of visitors on the first day as quickly as possible while still keeping an overview of participant numbers and finances – because nothing is more frustrating for visitors than standing in line for hours right at the start of the conference all because registration is not optimally coordinated.

21.2.1 The *InternalAdministration* Page

Internal Administration

First we will generate the page *InternalAdministration.* It represents the central contact point from which all activities originate during the check-in procedure. We recommend simply generating a link to this page in the left side menu bar of the *RegistrationWeb*.

Identification

The first step is entering visitors in the data management. The page *InternalAdministration* must now be prepared for two scenarios:

1. A visitor is not yet registered and must thus be entered in the TWiki system with all of his or her data.
2. A participant – in our case Mr. Stonecake – has already registered.

For the first case, the page contains a link to conference registration, where an employee can enter the new person's data in the registra-

tion form at check-in. He or she then returns to the *InternalAdministration* page.

Search form

If a visitor has already registered, he has to be found in TWiki. To do this, a formula with a search function is generated that already contains a couple of parameters that you otherwise would have to generate from scratch in a normal search, such as the setting of the `scope` parameter, which affects a search of the page title as well as within the page itself.

The form, with a text box for entering the search item and the subsequent search function, looks like this:

Wiki

```
<form name="Check1"                                    ↵
      action="%SCRIPTURL%/                             ↵
              view%SCRIPTSUFFIX%/                      ↵
              %WEB%/%TOPIC%"                           ↵
      method="get">
  <input type="text"                                   ↵
         name="query"                                  ↵
         value="%URLPARAM{"query"}%">
  <input type="submit">
</form>

%SEARCH{search="%URLPARAM{"query"}%"                   ↵
        scope="both"                                   ↵
        type="keyword"                                 ↵
        noheader="on"                                  ↵
        web="RegistrationWeb"                          ↵
        nosearch="on"                                  ↵
        excludetopic="%TOPIC%,                         ↵
                      ParticipantList"                 ↵
}%
```

If, during check-in, Mr. Stonecake's name is entered in the search, a link to his user page should appear.

Receipt

The next section of the page leads to the receipt that is to be printed out for every participant. Of course, you can simply generate a link to this spot. However, the receipt should already be filled out with the individual details of the participant, including name, address and fee paid, which is why the link must already include the identity of the participant, so to speak, using the `%URLPARAM{"query"}%` variables. Technically, this is done using another form with a text box in which the TWiki name of the visitor should be entered:

Wiki

```
<form name="Check2"                                    ↵
      action="%SCRIPTURL%/                             ↵
              view%SCRIPTSUFFIX%/                      ↵
              %WEB%/Receipt"                           ↵
      method="get">
  <input type="text"                                   ↵
         name="query"                                  ↵
         value="%URLPARAM{"query"}%">
  <input type="submit"                                 ↵
         value="Generate receipt">
</form>
```

Further links

Before getting into what exactly happens on the receipt page, let us have a brief look at the rest of the *InternalAdministration* page:

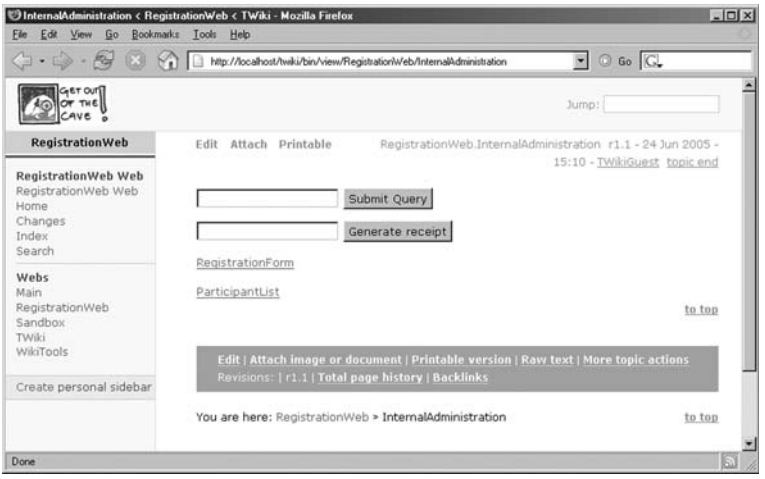
Wiki

```
[[RegistrationForm]]
[[ParticipantList]]
```

As mentioned above, the first link is primarily required in cases where the visitor is not yet registered in TWiki. The second link leads to the *ParticipantLink* page, where employees are to enter all visitors that have checked in and paid. This gives conference management an overview of which visitors have actually arrived at the conference and how much money should be in the cash box. We will get into more detail about this page later.

This completes the *InternalAdministration* page. The page view would look like this:

Fig. 21.1



21.2.2 Generating Receipts

The *Receipt* page – as we have mentioned – already has access to a Wiki name, via the data in the URL, and thus to the user page of the respective participant. The receipt is now to be filled out in the right spots with the content from the user page, specifically the address of the user on the upper left, his or her last name in the salutation, and the conference fee that was paid in the text portion. This is done using the `%SEARCH%`- variable with the corresponding regular expressions. You now have the opportunity to use the skills you learned in Chapter 13 or simply take on the three expressions. Directly under the heading "Receipt," however, in the same table row, the logo of the conference is entered which then appears on the right side in page view.

Wiki

```
<table border=0 width="100%">
<tr>
  <td>---++ Receipt</td>
  <td align="right">
    %PUBURL%/%TWIKIWEB%/                    ↵
    TWikiLogos/wiki-logo-roll.jpg
  </td>
</tr>
</table>

%SEARCH{search="%URLPARAM{"query"}%"        ↵
        scope="topic"                       ↵
        web="RegistrationWeb"               ↵
        noheader="on"                       ↵
        nosearch="on"                       ↵
        nototal="on"                        ↵
        format="                            ↵
$pattern(.*?\*.*?First name:\s*([^\n\r]+).*) ↵
$pattern(.*?\*.*?\sName:\s*([^\n\r]+).*)%BR% ↵
$pattern(.*?\*.*?Street:\s*([^\n\r]+).*)%BR% ↵
$pattern(.*?\*.*?Zip code:\s*([^\n\r]+).*)   ↵
$pattern(.*?\*.*?Town:\s*([^\n\r]+).*)"      ↵
}%


Dear
%SEARCH{search="%URLPARAM{"query"}%"        ↵
```

```
        scope="topic"                          ↵
        web="RegistrationWeb"                  ↵
        noheader="on"                          ↵
        nosearch="on"                          ↵
        nototal="on"                           ↵
        format="                               ↵
$pattern(.*?\*.*?First name:\s*([^\n\r]+).*) ↵
$pattern(.*?\*.*?\sName:\s*([^\n\r]+).*)"    ↵
}%,

In accordance with your registration, we
hereby charge you a participation fee:

   * conference fee:
%SEARCH{search="%URLPARAM{"query"}%"        ↵
        scope="topic"                          ↵
        web="RegistrationWeb"                  ↵
        noheader="on"                          ↵
        nosearch="on"                          ↵
        nototal="on"                           ↵
        format="                               ↵
$pattern(.*?\*.*?fee:\s*([^\n\r]+).*)"       ↵
}%

to participate in the conference "Get out of
the Cave" from June 15-17.

Best regards,

Your Conference Manager, Harry Hunter
```

Tip: In order to avoid superfluous navigation elements from being included on the printout, you can apply the *Plain* skin to the Receipt page (see Chapter 16.1). This skin is set so that none of the elements before and after the page text are printed.

Mr. Stonecake receives the following receipt after he has paid:

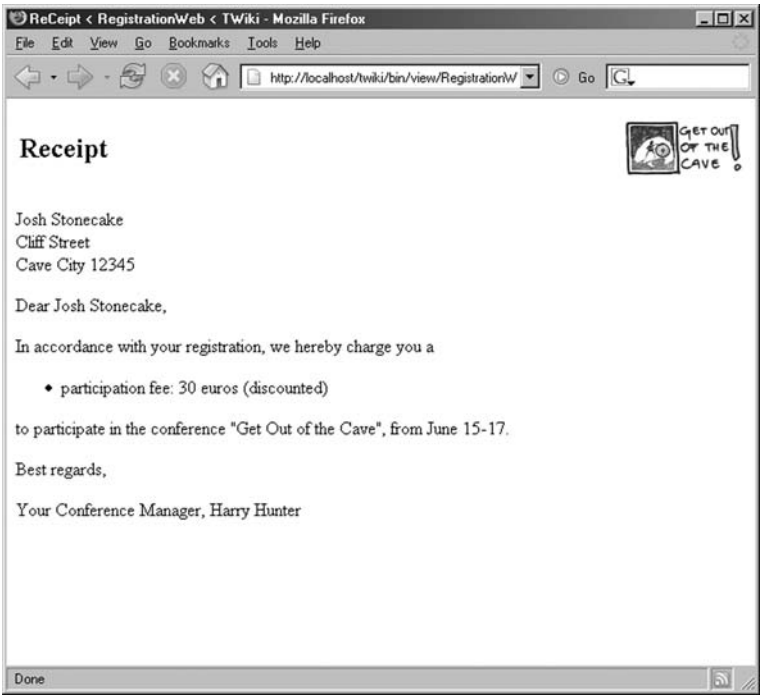


Fig. 21.2

21.2.3 Participant List

The last page that we are adding is the *ParticpantList*. Its primary purpose is to provide an internal overview of the visitors that have already checked in, by entering their names in a table. In addition, the fee paid is to be recorded, to facilitate checking the status of the cash box.

The EditTable plugin can be of great assistance (see above). Using it, we can design a user-friendly form that also makes the employees' work easier. It backlinks to the *InternalAdministration* page, and might look like this:

Wiki

```
%EDITTABLE{header="|*No*|*Last name*|                ↵
                   *First name*|*fee|"               ↵
           format="|row, -1| text,25 |                ↵
                  text,25 |select,1,35,50 |" ↵
           changerows=on                             ↵
}%

[[InternalAdministration]]
```

Here you see the table after a few visitors have arrived:

Fig. 21.3

No	Last name	First name	*fee
1	Stonecake	Josh	50
2	Venus	Geri	35
3	Woodlawn	Mark	35
4	Berry	Louis	50

Save table | Quiet save | Add row | Delete last row | Cancel

21.3 Lectures with the Presentation Plugin (SlideShow)

Presentations and lectures

TWiki offers simple assistance for the creation of presentations. If Harry Hunter would like to give a salutatory speech for the Planning group at the conference, he can make use of his SlideShow plugin, which, similar to Microsoft Power Point, enables him to structure content and prepare it for a presentation.

Mode of operation

The SlideShow plugin is included in the standard installation. It is not a separate program interface: Each page or part of a page divided by headings can be turned into a presentation. Of course, the plugin cannot be compared to professional presentation software that includes animation and other multimedia effects. However, it is an easy way to present content in TWiki for a lecture.

To prepare a presentation, write the content of the presentation on a page and divide it by headings. Each heading and the text following it is displayed as a slide. The format of the slides is based on a slide template that you can format at will and fill with contents as well as navigation buttons.

21.3.1 Syntax

You begin the presentation in edit mode with the variables

Wiki

```
%SLIDESHOWSTART%
```

Then the slide content of the presentation is entered. The slide title should be identified as a second-level heading using `---++`. You can also use headings of other levels, but they should be used consis-

tently for each slide of the presentation. The text following a respective heading is interpreted as slide content. Usually, it is structured in unnumbered lists, but you can also integrate tables and graphic images.

If you wish to add a comment within the presentation, start a slide with a third-level heading and call it "Comments." The text underneath it and up to the next slide will be interpreted as a comment, and will not appear in the presentation view.

You can define the end of a presentation with

```
%SLIDESHOWEND%
```

Wiki

After you save your work, you will then see the slide breakdown of your presentation.

Harry Hunter's draft for the presentation of his salutatory speech looks as follows in the edit view:

```
%SLIDESHOWSTART{                                    ↵
     template="ConferenceLaunch"}%
---++++ Conference: Prospects for the Future ↵

in the Neolithic Age
%BR%
<center>
by Harry Hunter, Conference Manager
</center>

---++++ Salutation
   * Promotion of a "Neolithic                      ↵
     Revolution"
   * Conference as a first step?
   * Status of the debate: Doubt as to
whether a new era is even possible.
   * Example: sedentariness

---++++ Sedentariness: Cons
   * Humans are not intended for agriculture
   * Health concerns
   * Humans will become soft
   * Shamans prophesize the decline of              ↵
     culture

---++++ Sedentariness: Pros
<table width=100% border=0>
```

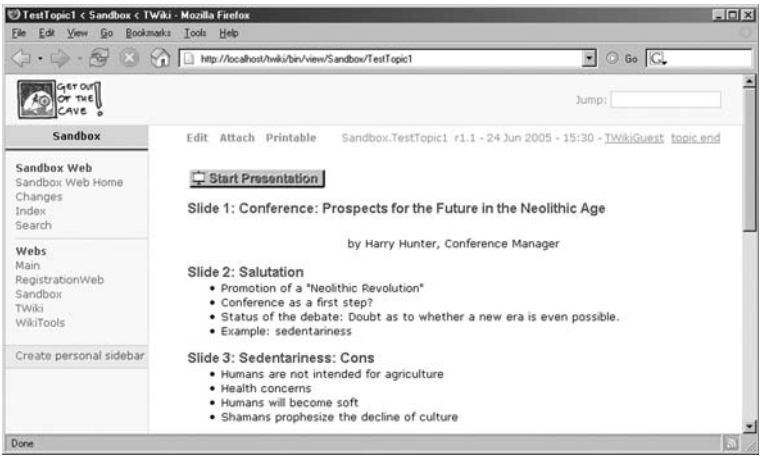
Wiki

```
<tr>
  <td>
    %PUBURL%/Main/DiaGram/                    ↵
    _ChartPlugin_bar_graincultivation.png
  </td>
</tr>
</table>
   * Agriculture creates jobs
   * Better nutrition
   * More variety on the table
   * More time for family

---++++ Conference Program
   * Workshops
   * Cultural Program

---++++ Finally: Technical tips
   * Events Office
   * Room changes: How do I find my          ↵
     "cave"?
%SLIDESHOWEND%
```

As you may have noticed, Harry Hunter has integrated an image, specifically a diagram, on the third slide. You can find out how to generate diagrams in Chapter 21.4.

Let us get back to the presentation: Before each slide title, the number of the slide is indicated. In addition, a button is inserted that starts the presentation:

Fig. 21.4



When you press the button, TWiki begins your presentation: It accesses a template and inserts the slide title and contents you have provided. If you have not yet created your own template, the TWiki presentation template is used, which is defined in the topic *SlideshowPlugin*. Using the included navigation buttons, you can jump back and forth from slide to slide.

You can exit the presentation button via the slides with the button **End Presentation**.

21.3.2 Your Own Template

You most likely do not want to employ the TWiki test template for your presentation, but would rather use the opportunity to create your own layout. It basically does not matter where you define your template: You can generate a separate topic for it, or insert the template on the page in which your presentation is also located. However, there should only be one template per topic, since it is integrated via the topic title.

Surround the template with the variables `%STARTINCLUDE%` and `%STOPINCLUDE%`. Between them, use an HTML table that defines the design of your slide. Thus, for instance, you can define the first row as your slide title by inserting the logo and the placeholder `%SLIDETITLE%`.

The following variables are available for the template:

Tab. 21.1

Variable	Significance
`%SLIDETITLE%`	Slide title.
`%SLIDETEXT%`	Slide text.
`%SLIDENUM%`	Current slide number.
`%SLIDEMAX%`	Number of the last slide.
`%SLIDENAV%`	Navigation buttons for the “first,” “previous,” and “next” slide.
`%SLIDENAVALL%`	Navigation buttons for the “first,” “previous,” “next” and “last” slide.
`%SLIDENAVFIRST%`	Navigation button “First slide.”
`%SLIDENAVPREV%`	Navigation button “Previous slide.”
`%SLIDENAVNEXT%`	Navigation button “Next slide.”
`%SLIDENAVLAST%`	Navigation button “Last slide.”

Of course you can also utilize all other design options of HTML tables, such as background color, text alignment, and so on.

On the other hand, it is easiest to alter the standard template on the page *SlideShowPlugin* or copy and edit it. That is what Harry Hunter has opted to do. However, there are a few settings that need to be defined on the copy of the template, in addition to the new logo: For instance, the graphic file *clearpixel.jpg* is integrated in the standard template, to which the altered template will continue to want to be able to access. To allow this, the following path to the topic *SlideShowPlugin* in *TWiki* web must be set:

Wiki

```
<img src="%PUBURL%/TWiki/                    ↵
  SlideShowPlugin/clearpixel.gif">
```

Here you can see the complete template of the salutatory speech on the page *ConferenceLaunch*:

Wiki

```
%STARTINCLUDE%
<table width="100%" border="0"               ↵
       cellspacing="0" cellpadding="0">
<tr bgcolor="#003399">
  <td valign="middle" width="2%"></td>
  <td valign="middle">
    <font size="+3" color="#FFFFFF">
      %SLIDETITLE%
    </font>
  </td>
  <td align="right" valign="middle">
    <img src="%PUBURL%/TWiki/TWikiLogos      ↵
              /wiki-logo-roll.jpg"           ↵
         border="0"                          ↵
         alt="Get out of the Cave" />
     
  </td>
</tr>
</table>
<table width="100%" border="0"               ↵
       cellspacing="0" cellpadding="3">
<tr bgcolor="#FFFF99">
  <td width="1">
    <img src="%PUBURL%/TWiki/SlideShowPlugin ↵
              /clearpixel.gif"               ↵
         width="1" height="480" alt="" />
  </td>
  <td valign="top">
    %BR%
```

```
    <font size="+2" color="#003399">
      %SLIDETEXT%
    </font>
  </td>
</tr>
</table>
<table width="100%" border="0"                    ↵
       cellspacing="0" cellpadding="0">
<tr bgcolor="#FFFFCC">
  <td valign="middle">
    %SLIDENAVALL% 
  </td>
  <td valign="middle" align="right">
    <font size="-1" color="#666666">
      Slide %SLIDENUM% of %SLIDEMAX%
    </font>
  </td>
  <td valign="middle" align="right"></td>
</tr>
</table>
%STOPINCLUDE%
```

The following image results in the page view:

Fig. 21.5

There are now two ways to allocate the template to a presentation. You can define it as a standard template in the variable `%TEMPLATE%` on the *TWikiDrawPlugin* page, e.g. with

Wiki

```
Set TEMPLATE=ConferenceLaunch
```

In this case, the template is automatically allocated to every presentation in the TWiki.

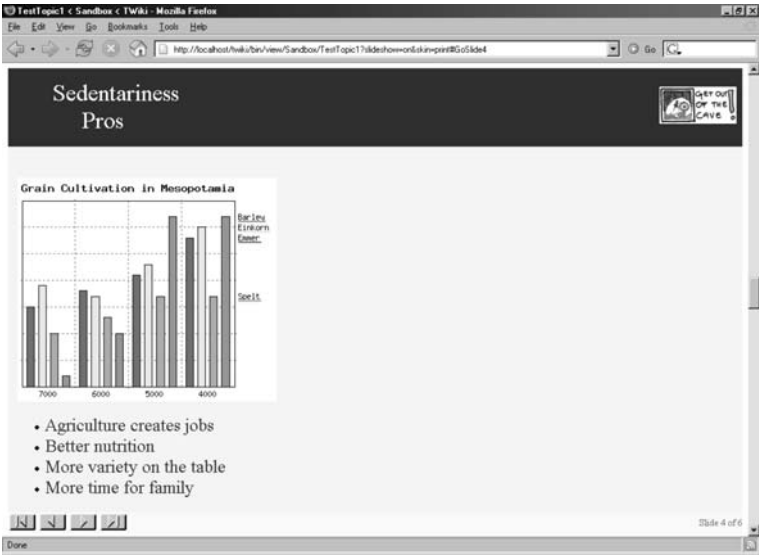
Alternatively, you can integrate the template once in a single presentation by transferring it to the variable `%SLIDESHOWSTART%` as a parameter. In our example, we need to add the following:

Wiki

```
%SLIDESHOWSTART {template=
   "ConferenceLaunch"}%
```

The third slide of the launch speech looks as follows after we integrate the template:

Fig. 21.6



Default settings

Before we leave the realm of presentations, let us have a brief look at the default settings of the plugin, which you can view and edit directly on the Plugin page:

Tab. 21.2

Variable	Significance
`%TEMPLATE%`	Defines the standard template for all presentations in TWiki Example: `Set TEMPLATE = ConferenceLaunch`

`%HIDECOMMENTS%`	Activates or deactivates the option of including comments in the presentation Example: `Set HIDECOMMENTS = on`

21.4 Generating Visuals for Statistics using Diagrams (Chart Plugin)

Statistics

Diagrams are irreplaceable in providing visualization. The depiction of trends and statistics are essential in projects, including our conference. Perhaps you wish to portray the number of visitors over the course of the conference, for instance. Now that we are nearing the end of our Wiki project, we would like to present the Chart Plugin. It can be used to generate very simple diagrams in the wiki.

Mode of operation

This plugin provides a function that is often familiar from spreadsheets: The program generates a diagram based on the identified database.

In the TWiki, the database consists of a table that by default should be in the same topic. However, there is also the option of citing tables from other pages. There are five different types of diagrams that can be selected, which can also be combined with each other. Furthermore, it is possible to set several diagram parameters, such as the color combinations. The generated diagram is created in jpeg or png format. All diagrams on a page are located in the open folder of the corresponding web. Thus, others also have access to the diagrams.

Note: For the Chart plugin, you will also need the additional Perl module GD. You also have to install it retroactively with CPAN. GD, in turn, requires the graphic library gdlib to be in your system. You can obtain further information at http://www.boutell.com/gd/.

21.4.1 Syntax

To generate a diagram on a page, insert the variable `%CHART{...}%` in the appropriate spot. The diagram type, data source and diagram attributes are to be defined inside the curly brackets.

A few parameters, such as `xlabel` , expect information on the table ranges. This information corresponds to the data with which you are already familiar from our discussion on spreadsheets: a cell is defined by a row and column entry, e.g. `R1:C1` for the cell in the first row and first column. A table range, in turn, is defined by two cells; e.g. `R2:C1..R5:C7` refers to the corresponding twenty-eight cells.

Here is a brief example:

Fig. 21.7

	Hunters	Gatherers	Shamanes
Day 1	40	59	11
Day2	32	62	9

The source text of the above is as follows:

Wiki

```
%TABLE{name="Example_1"}%
| | Hunters | Gatherers | Shamans |
| Day 1 | 40 | 59 | 11 |
| Day 2 | 32 | 62 | 9 |
```

Table plugin

The reason that we make reference to the source code again here is the `%TABLE%` variable in the first row that belongs to the so-called Table plugin, which we must mention here. This plugin is part of the standard installation, and enables formatting and sorting parameters, such as a background color, to be defined in a table using pipe syntax. However, it also enables you to name the table, which is vital to generating our diagram, so that the desired database can be distinctly named in the `table` attribute. That is why only the parameter `name` is relevant here, that we have allocated with “Example_1”. We will not go into more detail about further options of the plugin.

We would now like to portray the data from the “Example_1” diagram as a bar graph. The code line for this procedure is as follows:

Wiki

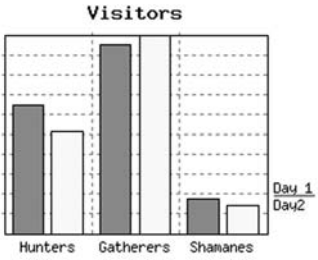
```
%CHART{ type="bar" name="Bar_1"                ↵
  table="Example_1" data="R2:C2..R3:C4"        ↵
  xaxis="R1:C2..R1:C4" legend="R2:C1..R3:C1" ↵
  ymin="0" width="225" title="Visitors"        ↵
  height="200"
}%
```

It is used to generate the following graph, *Bar_1.png* , which is saved in

```
%PUBURL%/%WEB%/%TOPIC%/                                    ↵
   _ChartPlugin_<type>_<name>.png .
```

Wiki

Fig. 21.8



The list of possible parameters for the diagram is extensive. However, the only necessary ones are *name*, *table* and *data*.

Tab. 21.3

%CHART%-Parameter	**Description**
`type`	Primary diagram type ■ "area" = area diagram ■ "bar" = bar graph ■ "line" = line graph ■ "scatter" = scatter diagram ■ "combo" = combination diagram
`subtype`	Diagram types of the remaining data series for the primary type combo; possible combinations: `area, bar, line, point, pline`. Area and bar graphs should not be mixed in a single diagram.
`scale`	Linear or semi-logarithmic.
`name`	Name that distinctly identifies the diagram.
`web`	Web in which the topic with the base table is located.
`topic`	Topic containing the base table.
`table`	Base table
`title`	Diagram title; appears above the diagram.
`xlabel`	X-axis label.
`ylabel`	Y-axis label.
`data`	Value range of the table that is to be depicted.
`defaultdata`	Default value if a cell is empty.
`xaxis`	Values of the X-axis; is indicated as a cell range.

`xaxisangle`	Angle at which the values of the X-axis are shown; O means horizontal, and all other numbers are vertical.
`yaxis`	Values of the Y-axis; “off” means they are not displayed.
`ymin`	Minimum shown on the Y-axis.
`ymax`	Maximum shown on the Y-axis.
`xgrid`	Grid lines on the X-axis level: ■ “on”: solid lines ■ “off”: no lines ■ “dot”: dotted lines
`ygrid`	Grid lines on the Y-axis level; values as in `xgrid`**.**
`numygrids`	Number of grid lines on the Y-axis level.
`numxgrids`	Number of grid lines on the X-axis level.
`datalabel`	Individual data points are displayed: ■ “on”: Values are displayed. ■ “off”: Values are not displayed. ■ “box”: Values are displayed in the box. ■ “off, off, box”: Every third value is displayed.
`legend`	Legend displayed on the right of the diagram; is defined via a table range.
`width`	Width of the diagram in pixels.
`height`	Height of the diagram in pixels.
`colors`	Colors used for the data rows.
`bgcolor`	Background color of the diagram.
`gridcolor`	Color of the grid lines.
`linewidth`	Line width of the curve in pixels.
`pointsize`	Width of a data point in pixels (in the direction of the X as well as the Y-axis).

21.4.2 Sample Table

The following diagrams have been generated with the data of this table:

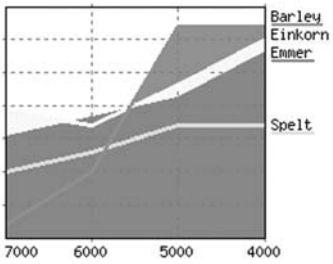
	7000	6000	5000	4000
Emmer	20	23	26	33
Einkorn	24	22	28	35
Spelt	15	18	22	22
Barley	7	15	37	37

Fig. 21.9

Here, you can see the Chart plugin in action:

Area diagram:

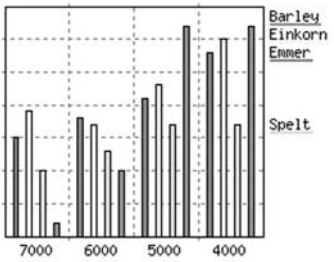
```
%CHART{ type="area"
name="Grain_
cultivation_area" ta-
ble="Cultivation"
data="R2:C2..R5:C5"
xaxis="R1:C2..R1:C6"
legend="R2:C1..R5:C1"
width="250"
height="200" }%
```



Tab. 21.4

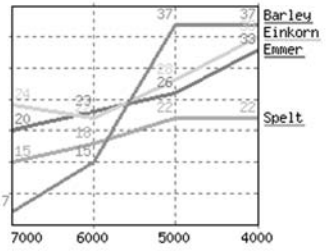
Bar diagram:

```
%CHART{ type="bar"
name="Grain_cultivati
on_bar" ta-
ble="Cultivation"
data="R2:C2..R5:C5"
xaxis="R1:C2..R1:C5"
legend="R2:C1..R5:C1"
width="250"
height="200" }%
```



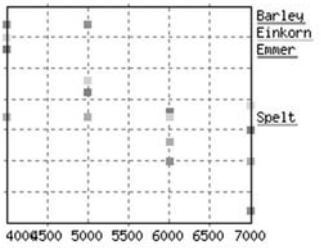
Line graph:

```
%CHART{ type="bar"
name="Grain_cultivati
on_line" ta-
ble="Cultivation"
data="R2:C2..R5:C5"
xaxis="R1:C2..R1:C5"
legend="R2:C1..R5:C1"
width="250" datala-
bel=”on” height="200"
}%
```



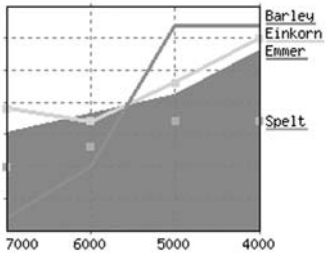
Scatter diagram:

```
%CHART{ type="bar"
name="Grain_cultivati
on_scatter" ta-
ble="Cultivation"
data="R2:C2..R5:C5"
xaxis="R1:C2..R1:C5"
legend="R2:C1..R5:C1"
width="250"
height="200" }%
```



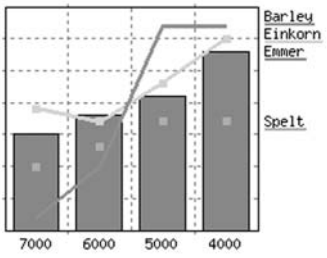
Combo diagram I:

```
%CHART{ type="combo"
sub-
type="area,pline,poin
t,line"
name="Grain_cultivati
on_combo_one" ta-
ble="Cultivation"
data="R2:C2..R5:C5"
xaxis="R1:C2..R1:C5"
legend="R2:C1..R5:C1"
width="250"
height="200" }%
```



Combo diagram II:

```
%CHART{ type="combo"
sub-
type="bar,pline,point
,line"
name="Grain_cultivati
on_combo_two"
R5:C5"
xaxis="R1:C2..R1:C5"
legend="R2:C1..R5:C1"
width="250"
height="200" }
```



In the event that the necessary parameters are not defined, an error message is displayed instead of the diagram.

Default settings

Many of the variables that you can use to design a standard diagram in the default settings (Plugin page) should be familiar to you from the parameters of the `%CHART%` variables: `TYPE`, `WIDTH`, `HEIGHT`, `AREA_COLORS`, `LINE_COLORS`, `BGCOLOR`,

GRIDCOLOR, NUMYGRIDS, DEFAULTDATA, SCALE, LINEWIDTH, POINTSIZE. In addition, you will also find specific options here for the bar graph:

Variable	Significance
%BARLEADING SPACE%	Number of pixels before the first bar. Example: Set BARLEADINGSPACE = 6
%BARTRAILING SPACE%	Number of pixels after the last bar. Example: Set BARTRAILINGSPACE = 6
%BARSPACE%	Number of pixels between the bars. Example: Set BARSPACE = 5

Tab. 21.5

21.5 Final Steps: Feedback and Documentation

Feedback

The post-processing of projects is often neglected. However, it is very important with regard to future projects to pool experiences. The aspects that went well and those that could have gone better should be documented. Positive feedback and an honest inventory provide motivation for new tasks. Unexpressed and unresolved conflicts, on the other hand, have a debilitating effect. If a round of feedback is missing, the project is not complete.

For the group itself, the evaluation of the project has a significant function. Especially in the final, stressful phases, misunderstandings can crop up, or tension may build between participants that was never able to be addressed.

It is once again important that insights can be gleaned from the consequences and earnest improvement for future projects can be expected. It is not uncommon for final reports of projects to land, unread, in the binder, which is a source of further frustration.

Evaluation

Set up a separate page for feedback on which both positive and negative feedback can be collected. This can be done on an anonymous basis in a separate evaluation meeting. The discussion should clarify the following questions: Were the goals reached? Why or why not? What should be done differently next time, and what proved to be successful?

Archiving

Finally, the wiki must be modified for the documentation, if necessary, so that outsiders can also quickly find the information they seek (overview pages). This can, in some circumstances, be very time-consuming. In the ideal case, the wiki will already contain all necessary content. Through the history pages, the wiki documents the development of the project in a detailed manner. However, it is

possible that some content still needs to be submitted: Perhaps photographs of the posters, as well as work results and reports need to be worked in. You must decide which areas are to be closed and which are required for continuation of the project.

Note: Be sure to protect sensitive data, such as participant lists and user topics, from unauthorized access.

Photo gallery

A photo gallery that reflects the atmosphere of the project adds a nice as well as popular finishing touch to the project.

V. Tools with a Future

22 Tools with a Future? Technical and Social Outlook with Wikis

Readers
Authors
WikiAdmis
WebAdmis

22.1 Technical Outlook

22.1.1 Wiki Editor?

To enable an easy introduction to the world of wikis and create a comfortable working atmosphere, wiki syntax was made as simple as possible.

However, there has been increasing discussion of late regarding the development of special WYSIWYG editors, designed to make working with wikis even more convenient. "Yet wikis have also rekindled interest in efficient browser editors. Browser producers and third-party suppliers have recognized the demand, including with regard to content management systems, very widely popular in the commercial environment, which are not as open as wikis, but offer similar functions," says Möller.[1]

Mode of operation

The advantage of an editor is that text areas can be highlighted and, with a click of the mouse, directly formatted. The user is no longer expected to remember complicated formatting commands. Significant symbols assist in the process. The editor inserts the layout commands in the proper places. WYSIWYG editors go even one step farther. They enable you to see right away how a change will effect the overall look of the document, as is the case with standard

[1] Möller 2003.

word processing programs, without requiring the detour through saving the file and viewing the edited version.

Necessity of editors

In the meantime, simple editors can be found all over the Internet. For example, you have seen one model in *MediaWiki*, the software that helps you design a page with a toolbar with which you can perform common text formats (bold, italics, etc.) using mouse clicks. The introduction of editors was a giant leap in the realm of user friendliness. Even if purist programmers are not enthusiastic, such aids are urgently needed for the average user.

There are a few web-based HTML editors today that offer a WYSIWYG display. We have seen once such editor for wiki pages in *TWiki*, in the form of the Kupu Editor. However, a wiki poses special challenges to an editor. While normal HTML documents are generally independent, that is, they contain all information pertaining to the design of the page, a wiki page must first be brought into its actual form with the aid of a wiki script (identification of links, replacement of variables, etc.). Since such a script runs on the server, a permanent exchange between server and client would be necessary. This goes beyond the current technical possibilities of the Net. Creative solutions are needed to meet this challenge. Kupu, for instance, directly displays static parts, such as text format, and marks others, such as links or variables, in a certain color to indicate that they are subject to a further processing step. The editor also compares the page to the document structure existing at the start of the editing procedure. Thus, Kupu knows which pages exist in *TWiki*.

With regard to real wiki editors, there is still need for action. An ideal solution would be the development of a common editor for all wikis.

22.1.2 Standardization

MediaWiki und Wikipedia als Standard?

One great drawback to date has been that every wiki clone has its own formatting commands that the user has to learn anew every time. That is why efforts are underway to standardize wiki syntax. However, which program should serve as the standard? Wales argues in favor of *MediaWiki*, since it has become the most popular wiki, thanks to Wikipedia. Many people are familiar with its formatting commands and the names of its most important functions. And yet, of course, popularity does not necessarily indicate quality.

CamelCase

People are of two minds especially with reference to CamelCase. Some see in it a genuine, typical wiki aspect. The defenders of CamelCase swear by its simplicity with which associations and links

can be generated. Others, including Jimmy Wales, reject it for aesthetic reasons and complain that it continually causes links to be unintentionally set.[2] They have voiced the general insight that the syntax should differ from normal language to the extent that unintentional layout definitions can be avoided. This requires cooperation between several Wikians. One approach is a page in UseMod which documents the heated battle for standardization suggestions.[3]

Interwiki

Exchange between wikis would be considerably easier with a standard. At the moment, the transfer of data from one wiki clone to another without data loss is coupled with substantial difficulties. Even if the simplest formatting commands can be easily exchanged between themselves, the devil, as we know, is in the details. It poses a certain risk, since one cannot be sure how long individual wiki clones will be developed further and adapted to the current state of the art. There is also the danger that data will become "trapped" in an obsolete system. If the wiki community would at least agree on a single transfer format for data, some of the insecurity in using wikis could be eliminated.

22.1.3 Projects of the Wikimedia Group

Wikimedia Foundation

Many Wikians are at work on projects of the new Wikimedia Foundation. Thus, in the medium term, that foundation will be an important source for control points for the evaluation, distribution and use of wiki technology. Its content focus lies in the area of public reference works: the construction of a library of free textbooks, study and teaching materials (Wikibooks), a freely accessible dictionary (Wiktionary), a collection of free texts (Wikisource), quotes (Wikiquote) or biological species (Wikispecies). This area is expected to continue to grow considerably.

Wikimedia Commons

Images, text, video, music as free downloads

One remarkable project of the Wikimedia Foundation is Wikimedia Commons. It is designed to centrally store images, videos, music and texts for all Wikimedia projects. Images stored there and available for downloading must be in the public domain or be subject to the GNU FDL.

An international communal and multi-language data administration project firstly has the advantage that images no longer need to

[2] Wales in a personal conversation on Dec. 28, 2004.

[3] http://www.usemod.com/cgi-bin/mb.pl?WikiMarkupStandard, 15 Jan 2005.

be individually loaded for each Wikipedia system. Since they are free, they can also be used beyond the boundaries of that system. In addition, we can assume that the data is not subject to copyright laws, which would eliminate the need for extensive research into the holder of that copyright. The Wikimedia Commons project is very strict with regard to the license status of individual images. Information on the license must always be supplied with every image, it source quoted and, if the image stems from a Wikipedia project, the person who originally made the image available to Wikipedia must also be named.

Perspectives

Beyond the Wikipedia group, this project could become the largest international database with free image and text material affiliated with the efforts of the Free Software Initiative. Gallery pages can be generated for images with a particular theme and linked through categories. This makes it easy to find material on a certain topic and utilize the available documents. The wiki principle applies here. A large web catalogue is cooperatively generated and optimized according to the search criteria of users. Results of complex research can be made available to others. This greatly facilities finding relevant data. On the other hand, the opposite could also occur, and users could end up destroying each other's order.

Wikinews

News

With Wikinews, the Wikimedia Group has begun a new and challenging project. Only two weeks after the English version was released, a German version also appeared on the Net, in December of 2004. With this project, the wiki community is venturing an attempt to create a daily, updated news overview. Whether or not this newest project from the Wikipedia environment, still in the testing stage, will succeed, is a question that is also being closely followed by its initiators.

Its goal is to establish an international network of "citizen journalists", as they are called, who report on large and small events on site or put together articles from other sources. Especially in countries with limited freedom of the press, Wikinews could thus become an alternative news source. In this regard, the project is certainly comparable with the media network Indymedia. However, while Indymedia often represents a decidedly political standpoint, the angle at Wikinews will continue to be the "Neutral Point of View" (see below): Opinions and assumptions must be identified as such and clearly attributed to those holding that opinion. The entire source material upon which an article is based is to be made available online along with the article itself, so that each reader can form his or her own opinion as to how reliable a news item is.

Problems

The Wiki community initially had to rack their brains over producing their own news. In contrast to Wikipedia, news cannot be corrected in a subsequent communication process; it must be of good quality at the outset.[4] Thus, a type of editing system was introduced according to which an article must first be confirmed by other users before it is cleared as 'published'. However, this 'clearance' is not a technical process: All articles will continue to be accessible to everyone, except that clearance as published has a status approaching a quality assessment and provides indications as to the seriousness of a news item.

Experiments

The Wikinews concept should be considered an experiment that just may push the wiki principle to its limits. If and under what circumstances "citizen journalists" are really capable of tapping the same or even better sources than the established media remains to be seen. Critics fear that Wikinews could become a rumor mill, because effective editing controls are missing. Yet experience with Wikipedia, which in the beginning was met with suspicion, gives us cause for hope.

The demonstrative distinction between it and Indymedia, and its proclaimed self-conception as "citizen's journalism" lead to the suspicion that the conception was in no way as "neutral" as one might like to think. Krüger also has his doubts: "Wikipedia is a project of the Net-educated upper middle class. And Wikinews will also orient itself with regard to theme, content and language more towards the typical *Zeit* than the average *Bildzeitung* reader. The pretense of wanting to report about 'what is happening in the world around us' will likely be reduced to a kind of reporting which reflects only a class-specific, selective section of current world affairs – but which is probably very well-founded."[5]

Wikinews will also have to face this problem.

Wikidata

Database

The most technically challenging project is Wikidata. Its goal is to be able to design and edit data, including structured data, in the wiki way. We have already struggled with this type of data in the conference registration process detailed in Section IV of this book. Even simple tasks, such as displaying an address from a wiki page, require a great deal of effort and complicated procedure.

---

[4] Patalong announced skepticism on the side of the established media for 2004.

[5] Krüger 2004. Regarding the German periodicals mentioned, *Zeit* is a respected national weekly newspaper, and *Bildzeitung* is a national daily tabloid notorious for its sensationalism.

Structured data offers several advantages. Searches and editing are made considerably easier. Even automatic handling of the data is possible, since it can be more easily transferred in computer-readable format than as free wiki code. As soon as information can be structured in a table, entire new application options open up, such as allowing towns to be shown on a map, as well as many others.

Technical challenges

The development and design of a database is generally very complex. To remain true to the wiki philosophy, we must have methods of easily generating and structuring data. A data structure editor is in planning, whose fields are administered with the respective data type (e.g. text, numbers) and limits (e.g. unique, max. length), and that allows relationships between fields (e.g. parents, brothers) to be established. Once there is a template for structured data, a wiki syntax must be developed for the page view and edit mode with which form fields can be edited. Last but not least, history, diff views and the recent changes will be available on the field level. All of this is still in the conception phase.

Social challenges

New challenges will emerge for the wiki community with this project. One problem is that a change to the data structure has extensive consequences and reproduces itself throughout the system. This means a rollback is no longer possible without further effort. Also, how would one deal with vandalism? One answer would be strict user administration in which only administrators have the right to change the data structure. Whether or not the wiki would then lose its character as an open, easy and transparent system remains to be seen.

Potential

However, there is great potential in Wikidata. New dimensions of working with wikis emerge. Applications that require structured information can be based on Wikidata (calculations, graphics, simulations) and thus tap other formats and modes of display for the wiki world.

22.2 Wikis as an Engine for Social Change?

Media revolution?

What do all of these technical developments mean? Are we witnesses to a "secret media revolution"? Are web logs, wikis and free software changing the world from behind? These questions stem from a book of the same title by Erik Möller. It presents a difficult problem: What is the relationship between technology and social progress? Something seems to have gone out of control with wikis.

Defenders of free software and Wikipedians see great benefit in it. Others fear uncontrollable developments.

Fundamental question of social principles

It seems to be impossible to speak of wikis without being immediately confronted with fundamental problems: objectivity and partiality, intrinsic and extrinsic motivation, central and decentralized control mechanisms, hierarchical and antiauthoritarian order structures. And, last but not least: the question of other lifestyles and other property relationships. The debate about wikis gains energy because it is taking place right in the middle of a phase of accelerated social change. It is true that, in the case of wikis, as in that of free software, we are talking superficially about the user of the new information and communication technologies; yet the debate cannot be comprehended without the fundamental question of social principles thus brought into the discussion.

22.2.1 Wiki, a Democratic Medium?

Let us begin with the wiki as a medium. Is it "inherently democratic" (Cunningham)? Have, with wikis, irretrievable developments been initiated?

Technical utopias

With regard to technology and progress, we are confronted with deterministic ideas. Since the beginning of the "scientific age" (Brecht), people have been trying in a variety of ways to establish a unilateral relationship between historical rise and fall and technical innovations.[6] As we know, in the 19th century, industrialization was equated with social progress. At the start of the 21st century, too, technical utopias have been booming. The refrigerator that orders the food itself is touted as a step toward a life of luxury and unlimited consumption. Especially in political discourse, we witness technical determinism, such as when the "sustainability" of a country is described as dependent on its "technical edge" or its "innovative ability." The widespread hope of the 1990's that the Internet would be a technical lever for the democratization and revitalization of the public sector also tended toward a technical utopia.[7]

Pessimistic cultural view

In contrast, there is a pessimistic cultural tradition of technological determinism, in which technical developments are seen as an engine of social decline. Technical development is perceived as something of which people are losing control. As seen from this standpoint, once a path has been taken, there is no turning back. This ar-

[6] Degele provides an introduction to the relationship between technology and society (2002).
[7] For the history, see Rilling 2004.

gument has a powerful example in nuclear technology, which forces people to deal with radioactive waste today based on yesterday's decisions. In the case of information technology and biotechnology, some see the same mechanism at work, although their consequences lie on another level.[8]

Democratic use of a medium

The various forms of technical determinism have in common the notion that social change is no longer subject to human decision, but rather stems from the logic of technical development. In places where information technology was seen as a threat or guarantor of an open society, this view was consistently met with criticism. In the early 1970s, Enzensberger, with regard to the mass media of television and radio, argued that the utilization of media makes a difference in the evaluation of those media.[9] In his *Baukasten zu einer Theorie der Medien (Constituents of a Theory of the Media)*, he offers further suggestions for an emancipatory use of the media that can also be translated to the interactive use of the Internet: decentralized programs, every recipient a potential sender, mobilization of the masses, interaction of participants, feedback, political learning processes, collective production, and social control through self-organization. Similarly, Rilling believes: "The Internet offers no room for decisions, but places for political debates, communication and the forming of opinions."[10] Wikis are programs that, for the first time, radically expand the possibilities of the Internet as a large-scale technical system for "democratic use."

Practical problem

In this regard, the initial developmental phase was less about social change and more about creating a simple communication platform for decentralized software development. Cunningham's concept that uncomplicated and non-hierarchical cooperation is part of successful software development was included in the development. If he had thought that software development would not be possible without authoritarian, central control, the program code would have developed differently. Instead, the result is a program whose advanced use would only become apparent with time. Thus, what Brecht said about radio applies to wikis: "It was not the public that had been waiting for radio, but radio that had been waiting for the public." [11]

---

[8] Among the pessimistic cultural groups of technical determinism also certainly include those forms that express themselves in daily language, such as when people speak of the detrimental effect of "the media" on the social behavior of children, etc.

[9] See Enzensberger 1970, 173.

[10] Rilling 2004.

[11] Brecht 1930/1967, 128.

Combination of various, contradictory objectives

In our case, this means that wiki technology has been taken up once again, but it has been further developed from different viewpoints. The examples of MediaWiki and TWiki demonstrate that, in the process, various and at times contradictory objectives are continually flowing into the projects. In these many instances, the need for open communication and flat hierarchies with practical demands collude in developing a free encyclopedia or a project tool with potential marketability. The respective work processes influence the participants as well as the society as a whole.

Learning process

According to Rauter, the relationship between humans and tools can be seen as a dialectical learning process: "People exist because they make tools. The tools that man develops not only decide whether man lives, but also how he lives. The experiences he has when he manufactures and uses tools change him. His changes flow into the tools. The tools affect how people deal with each other."[12] This also applies to the development of wiki software.

Social structuring

That means that the development of tools is embedded in their culture and society, with its specific decision-making structures, and is closely related to the goals, experiences and assumptions of their "creators." The form and content of the developmental process, however, also serve to "structure" society. With them, their power ratio is also confirmed or altered – for instance, when democratic negotiation structures are able to prevail in the developmental phase. When an object is completed, a work and planning process comes to an end that has taken on material form. [13]

During the development of the first wikis, ideas came into play that were "inherently democratic." However, a democratizing effect through wikis, even though the tool opens new opportunities in this vein, is not necessarily guaranteed. On the contrary, wikis are already being employed in companies today without having caused any fundamental changes.

Dual use

Once again, the same thing that can be said about the ambivalent social function of science and technology also applies to wikis. Technical inventions and their use are without doubt of extraordinary importance to the development of human societies. Tools have provided people with room to take action, since they have enabled

[12] Rauter 1977, 10.

[13] This is especially apparent in the development of the atomic bomb. For it, people were mobilized and means committed to a scientific industrial large-scale project. State and economic institutions are created. The hope of participating scientists for securing peace certainly play a role, but ultimately, the democratic counterforces were not strong enough to assert an alternative peace policy or democratic decision structures – which then would have had the respective legal and institutional effects.

natural limitations to be surmounted. Humans built utilities that enable them to better control nature and humankind, in every sense of the word. Thus, perspectives have opened up. Relationships have become identifiable and the future plannable. The liberating power of technology could also transform into chains of liberation and into the instrumentalization of humans. This predicament has become known as the dual use problem.

In summary, we can say that for the assessment of whether a tool has any socially beneficial effects depends, *firstly*, on the question of which goals its creation was based upon and with which goals it is used; *secondly*, this must be seen before the backdrop of the extent to which social circumstances are cemented or accordingly changed. This also applies to wikis.

22.2.2 Ownership

Wiki: free software

A hot topic, not only for the wiki community, is the significance of licenses and free software. Generally, digitalization and the world wide web as a world wide copy machine have rekindled the fundamental debate on the role and function of ownership, primarily that of intellectual property (copyright, patents, trademark protection). The wiki communities also needed to face this issue, for the simple fact that, with the publication of wiki software and every further wiki clone, a fundamental decision regarding their conditions for use had to be made. Does a legal entity have the right to appropriate the software or parts of the software, and for what purposes? Many wikis today are subject to GNU-GPL regulations, and are thus free software programs that may not be sold.[14] Thus, wiki programmers are among the defenders of free software.

Free content

However, we are confronted with exploitation rights not only with regard to software, but also to the content level of wiki pages.[15] To whom do the texts and images belong? When is a text a text? Who is legally responsible? Research into whether image material is part of the public domain is a very time-consuming effort at Wikipedia. After all, the site strives to guarantee that Wikipedia truly remains a "free content" encyclopedia. This means that it supports a copyright which allows anyone to copy, alter or otherwise use text

[14] We already discussed licensing in Chapter 1.5.2. The best introduction to the history and issues of free software is provided by Grassmuck 2002.
[15] See Möller 2005, 171f.

for any purpose.[16] The original or altered version may be freely distributed. The opposite also applies: If you contribute content to Wikipedia, that text is automatically subject to this license, and you cannot retroactively assert copyright claims.[17]

Wave of privatization

Making available freely-accessible knowledge stands in deliberate contrast to the current trend toward the privatization of public goods; a trend that spares no area of society: from the hollowing out of the public character of companies to the privatization of water supply or education. The debate on the ability to patent gene sequences or software is also part of the issue.

Academies and monasteries

Partisanship for the reclamation of commons (Allmende) and public goods from the culture of the private are the order of the day. With regard to open source codes, Himanen speaks metaphorically of the "academy" in which scientists make their work available to others so that it can be utilized, tested and further developed.[18] In contrast, he metaphorically refers to closed models that withhold information and demonstrate authoritarian relationships as "monasteries." However, critical social theorists question whether a simply ethical critique is perhaps insufficient. The bone of contention is whether a call to "social responsibility" paired with "competitive ability" is trying to bring together two diametrically opposed principles.

Property Rights Regime

Once again, valid social control models – not the moral lapses of individuals such as Bill Gates – need to be investigated. The privatization of commons is the prerequisite to be able to market these goods – in this case, software. To make information products commercially utilizable, a shortage must be created. In the case of software, this means withholding the source text and limiting it through licensing and patents. Software is subject to a property rights regime. This results in an uneven fight, with attempts of infiltration, criminalization and rejection and the development of new strategies and counter-strategies.[19] There is doubt as to whether this can be traced

---

[16] The GNU Free Documentation License (GFDL), which governs Wikipedia articles, is the most common license for "free" content. The FreeBSD Documentation License, on the other hand, is not a copyleft license. Furthermore, Creative Commons releases licenses which however do not ensure "free content" for all.

[17] As a private individual or firm, you can subject your texts to a free content license (copyleft). You can find the GFDL on the Internet at: http://www.gnu.org/copyleft/fdl.html. Links to further licenses such as OpenContent License or the GPL can be found at: http://www.reference.com/Dir/Computers/Software/Licensing/Open_Source_Licenses.

[18] Himanen 2001, 91ff.

[19] For an introduction to the discussion, see e.g. Meretz 2003 and Nuss 2003.

back to a basic anthropological constant ("That's just the way people are") or to seemingly "objective" necessities that result from an economic constitution that eliminates or controls existential fundamentals of competitors.

High hopes in free software

The free software community represents a counterdraft to the commercialization of information products – especially as an alternative to Unix. Free software functions here as an alternative model that generates the hope that the expansion of the information technology-based leading sector can be able to be seriously blocked by making available stable and free software. These hopes have only been partially realized. The property rights regime was in no way questioned by the demand for freedom of information. Several forms of ownership can coexist. Free goods can be utilized very well as a component of an overall product; for instance, in the case of free software, through support and books.

Wikimedia projects

In the attempt to transform digital goods into commodities through copyright, patents and trademarks, the music industry is leading the way, in its fight against "bootleg" copies with their motto: "downloading is theft." On the opposite side, with the Wikimedia group, a significant player is growing that is turning against the marketing of software patents, image, film and text rights with its free content by pursuing the goal of building up enclaves of digital commons.[20] The Wiki Commons project has already been mentioned, as has Wikibooks, whose focus is the provision of cost-free textbooks whose commercial alternatives must be obtained at great cost. This is done in the hope of expanding access even to classes with less disposable income.

A nerve struck?

To what extent a nerve of a developing high-tech capitalism has been struck with the blockade of privatization tendencies in information technology is debatable. In his book *The Age of Access: The New Culture of Hypercapitalism, Where All Life is a Paid-For Experience* (2000), Jeremy Rifkin sees the question of "access" as the successor to the ownership question. In contrast, others have raised the objection that the "old economy" and its accompanying ownership relationships, over which the new technologies are simply superimposed, are thus being masked.[21] Furthermore, in various instances, the argument has been voiced that a greater danger is growing from the property rights regime through the direct contact between producers and consumers than through the provision of cost-free services.

---

[20] In addition to others, such as the *Free Software Foundation*, *Creative Commons* or the hacker culture.

[21] e.g. Haug 2003, 67f.

Legal political initiatives

The "key legal political invention" (Rilling) of the free software culture is the GPL license. It serves as the basis of further free content licenses, such as FDL or Creative Commons. Using such licenses, and based on the respective national legal systems, an attempt is being made to utilize the legal framework to secure open goods and commons. In this regard, however, the discussion cannot stop at the topic of digital and intellectual property. The question will be posed with regard to the overall concept of civil ownership. When attempting to answer the question, it could help to remember that the issue of ownership does not focus on the relationship between a person and a thing, but rather the relationship between people with reference to a thing.

22.2.3 Forms of Work

Alternative organisation models

It has been hinted at various occasions that the wiki philosophy inspires one to rethink the future organization of work. Centrally planned, controlled and regulated organizational models are countered with those that are decentralized and self-organized. One motive for this is that the tendency to subject work processes to bureaucratic controls (for instance in the course of quality or issue management) limits room for design and suffocates any intrinsic motivation. As early as the introduction of Linux, the free software movement not only confronted Windows with a new operating system, but also with a decentralized and democratic organizational philosophy. Alternatives are being sought to heteronymous and estranged work relationships, which ultimately also affect one's entire lifestyle, private life and the organization of one's free time. Some of the defenders of free software see here the budding of a new, non-capitalistic social order and hope that the production methods of programmers can be carried over to other social realms.
In this regard, wikis are one in a series of free software success stories. It is being proven that, as in the case of other software projects, free work forms can be employed to surmount even complex problems. In the long run, their products and services will even demonstrate a high degree of permanence, because more people are involved in the development process right from the start.

Counter-concepts?

The extent to which Linux and the wiki philosophy are already examples of true counter-concepts thus remains to be seen. The Net world is currently an anarchistic one, yet it also reproduces existing control models. During the Linux debate, Nuss already argued that

the decentralized work form could also assist less free production paradigms:

"Their [free software's, R.H.] production method is based on public knowledge, cooperation, flat hierarchy, flexibility, international networking, and largely unpaid work, usually without contractual obligations. The production model of open source and free software has been serving as a role model for industry for quite some time now."[22]

Self-responsible control processes in companies

Large private companies are also searching for solutions to their structural crises and for ways to be more flexible on the market. However, centrally regulated control models prove too rigid at an early stage. Employees have been independently compensating for the resulting problems for ages. They have been counteracting errors, faulty planning and malfunctions. "Without the targeted deviation from plans and instructions, company specifications cannot be fulfilled."[23] This means that companies, too, have been dependent upon the autonomous actions of their employees all along. Some see the solution to all problems in the promotion of employee self-organization. The goal is to make work enjoyable again, motivate volunteerism or include customers in development processes. However, this is coupled with unsecured jobs and a consolidation of working hours through – computer-assisted – optimization. Corporate planning security is shifted to the detriment of personnel planning security. Yet the ability to plan is a prerequisite of human self-actualization.

Social ingenuity

The wiki communities are propagating their idea that work organization "from below" can work by trying it out, and with a good measure of success. Their work organization model differs from the conventional pattern through a certain openness for results: The responsibility for planning and execution of the work process is completely transferred to the producers. The wiki communities, with their actions, promote social ingenuity. Their attempts in initiating large wiki projects not only have an external effect, but are also a form of self-enlightenment in such experiments.

[22] Nuss 2003, 154. Candeias develops the thought farther (2004, 179ff.).
[23] Candeias 2004, 179.

22.2.4 Objectivity

Credibility problem

Open systems are often met with mistrust. Forgoing an intermediate (quality) control causes many to doubt the credibility of the information, which is less a product of the philosophical dimension discussed above. Mistrust is directed at the question as to whether or not the masses even have enough knowledge to produce something trustworthy. In addition, it is difficult to verify from whom the information originates. Are the postmodern theorists correct when they speak of a disappearance of the author? On the Internet, it is easy to take on another identity – one need only think of the "sock puppets" mentioned at the start of this book. Furthermore, in cooperatively written texts, individual authors often can no longer be discerned. Also, after the previous discussion on the political dimension of the wiki phenomenon: Is this not in fact a process of manipulation and counter-manipulation? We need to take this question seriously. How can objectivity and quality be achieved?[24]

Motivations

Let us begin with the motivations of wiki authors. The problem of vandalism has already been described. Destructive practices of anonymous users are a lasting problem of open systems. However, it is interesting to examine the other reasons why authors refrain from wanting to be named.

Appreciation and work forms

Similar processes can be observed in wikis as have been discussed with regard to the free software community and critical information technology. In open processes, authors do not disappear – quite the opposite is true. Within the communities, many discover an appreciation and work forms that they seldom find in professional life. The Wikimedia projects, as the free software initiatives before them, enable self-determined work in a sensible project directed at the good of the community. Recognition and the fact that one's own contributions are noticed and discussed, motivate participants in a project to continue working. However, this also means that most Wikipedia texts in fact are not created anonymously. On the contrary – as we will see when observing the statistics – a group of people forms that knows each other and establishes trust in each other. This is generally applicable: In the evaluation of open systems, one must not forget the concrete social relationships behind them.

[24] To what extent the traditional mass media in their current form are "objective" would extend beyond the scope of this book.

Quality system

Good will alone is not enough. That is why wiki systems must develop assessment possibilities and conventions that promote verification and trackability. Once again, Wikipedia, for which a package of such measures has been developed, serves as a positive example. Initially, many people look at the articles in the encyclopedia. Ideally, they would immediately edit any discrepancies. Then admins check the edits of articles. The recent changes function offers a quick overview. In addition, frequently vandalized pages are placed under special supervision. This permanent effort is the price of an open system. Furthermore, conventions regarding language, methodology and balance of content have been stipulated, for example, with regard to the extent to which every opinion should be cited in an article.

NPOV

This is the point where the Wikipedia community conducts its debate on the relationship between objectivity and partiality, which equally applies when dealing with every other medium. We are talking about the debate on the neutral point of view (NPOV). Basically, it states that articles in Wikipedia are to take a neutral viewpoint into account.[25] This initially means a minimum of quality measures and the adherence to certain conventions: objective language, a reporting perspective and indirect speech. For instance, the observation "Picasso was the greatest painter of the 20th century" becomes "In the art world, Picasso is considered to be one of the greatest painters of the 20th century."

A variety of positions

Then there is the representation of a variety of positions. An article "should fairly represent all sides of a dispute."[26] The English version of Wikipedia stresses that an objective standpoint is not possible. One motivation for this phenomenon is the desire not to simply declare truths. Readers are encouraged to make their own decisions and form their own opinions. However, this point of view can often unintentionally lead to relativizations, such as the following:

"**Holocaust denial** is the claim that the mainstream historical version of the Holocaust is either highly exaggerated or completely falsified. Experts, witnesses, and historians almost universally regard Holocaust denial as untrue. Holocaust deniers prefer to be called

---

[25] "To whatever extent possible, encyclopedic writing should steer clear of taking any particular stance **other than** the stance of the neutral point of view. The neutral point of view attempts to present ideas and facts in such a fashion that both supporters and opponents can agree. Of course, 100% agreement is not possible; there are ideologues in the world who will not concede to any presentation other than a forceful statement of their own point of view." http://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view, 10 Jun 2005.

[26] http://en.wikipedia.org/wiki/NPOV, 15. Jan 2005.

Holocaust revisionists. Most people contend that the latter term is misleading."[27]

Absolute neutrality impossible

Among Wikipedia authors, there are various opinions regarding NPOV. A few consider neutral, objective portrayals to be possible. For others, neutral points of view, even in an encyclopedia, are impossible: "We can only seek a type of writing that is agreeable to essentially rational people who may differ on particular points" (Jimmy Wales).

If science used to claim to be objective, a new perception has taken hold in the last century in which every statement is initially selective.[28] A view is already present in the selection of a topic, the terminology, explanatory patterns and models. We know that, historically, dictionaries have always been an expression of the knowledge of a certain time and class (as a rule, the educated bourgeoisie).

Critical theory

Critical theory has tried to handle this dilemma, among other ways, by indicating that the subjective person creates objective relationships. The theory stresses an attainment of consciousness that is as objective as possible from the contradictions of the relationships. To put it very simply, the difference between what is and what could be results in a standard with which individual practices and ideas can be "measured" – a viewpoint that can also be employed within the context of publications in wikis. The discussion on objectivity could benefit from considering critical theory.

So what do we do to obtain as much objectivity as possible?

Partiality

1. Whether we like it or not, subjectivity is prerequisite to objective consciousness. Objectivity is not a freedom from values. Those wanting to be objective must have a point of view and be able to say where they want to go. Only then can statements be discussed. The important factors are verifiability, logical argumentation, and measuring arguments with proclaimed goals and their observance.

Sociographics

2. Striving toward objectivity is always also a form of self-education. For instance, this means asking who uses a piece of wiki software; because even if the new media tend toward bursting traditional social classes, use of the Internet, up to now, has been characterized by considerable social inequal-

[27] http://en.wikipedia.org/wiki/Holocaust_denial, 15. Jan 2005.

[28] Extreme postmodernists take this notion as far as relativism, or find that everything is constructed.

ity.[29] This is no different in wikis, at least not as of yet. According to Wales, 10% of the users of Wikipedia perform 80% of the edits, and 5% perform 66% of the edits. Thus, despite the openness of the system, core groups of authors have crystallized. Research into this phenomenon is still in its initial stages, and has yet to be intensified. Under consideration of these circumstances, it is better to estimate which projects are operated by which social groups and which attitudes they represent in the process. What is their social and historical position from which their actions originate? Using this requirement, we must clarify to what extent partial or total interest can be represented. This requires additional knowledge of the sociology of group processes.

Power gradients in wikis

3. As we have already demonstrated in our model of levels in the first section of this book, wikis, too, are internally not authority-free zones. There are various power relationships that are related to access rights, for instance. In addition, real power relationships (ownership, employment relationships, capitalistic order structures, etc.) intersect with the functions within the wiki community. This fact, too, must also be taken into account.

22.2.5 Progress?

Problematic progress

In the 20th century at the latest, progress has transformed into a problematic term. For example, Walter Benjamin writes: "Marx says the revolutions are the locomotives of world history. But perhaps they are something different altogether. Perhaps revolutions are the hand of humankind riding in the train grabbing for the emergency brake."[30] His criticism is directed at estranged lifestyles that bank on historical change taking place with or without the intervention of humans. His utopia is very similar to that of many free software developers. It is about the crystallization of a truly self-determining lifestyle. This means that the idea of what progress is must continually be revised and agreed upon. In the process, the emphases of social labor shift.

[29] Regarding the inequality of the social user profile and power relationships on the Net, see e.g. Rilling 2004.
[30] Benjamin GS, I, 1231.

Initial steps

In the example of the wiki communities, we see that the first step is a very small one. Enzensberger remarks that every socialist strategy of the media must aspire to eliminate the isolation of individual participants in social learning and production processes. "This is not possible without self-organization of the participants. This is the political core of the media issue."[31] The Wikipedia project appears to be re-figuring the conditions of change. One important focus is the mutual acceptance of corresponding persons or opponents as reasonable – and we would like to add non-competitive – entities. The enormous difficulties coupled with this process cannot be ignored. One could perhaps speak of progress when wikis become a truly collaborative and integrative medium extending beyond certain social groups. But if there is to be collaboration, then with what goal?

Beginning

Wikis are fascinating tools, but not cure-alls. They must continually be filled with new ideas in order to be able to advance toward the distant goal of free development of the individual as a condition for the free development of the collective. In the wiki world, this has already been realized today in several small areas.

[31] Enzensberger 1970, 169.

Appendix

A Installations in TWiki

Cygwin

To manually install Cygwin from the Internet, proceed as follows:

- Start the file setup.exe on the CD in the *cygwin* directory.
- Select “Install from Internet.”
- You then have to indicate the installation directory.
- Make sure that the “Default Text File Type” is set to “Unix.” Cygwin should also be set up for all users.
- Enter the directory in which the installation files are to be stored. You now must supply some information regarding your Internet connection and then select a server from the list from which the source files are to be retrieved.
- You will see a selection window in which you can indicate the components to be installed. A few files are already marked, but you will also have to mark a few manually. The following table shows the programs which you will need for the setup and operation of TWiki described here and which thus need to be marked for selection. You can tell that an entry is selected by the fact that, instead of “n/a”, a version number will appear in the line.

Name	sel.	Function
Archive		
unzip	-	Unpack .zip files.
Base		
bash	+	Command line interpreter under UNIX.
coreutils	+	Here: tools for editing text files.
diffutils	+	Finds differences between files.

grep	+	Searches for specific patterns in character strings and files.
gzip	+	GNU compression utility.
tar	+	GNU archiving utility.
Devel		
binutils	-	GNU assembler and linker.
gcc	-	C Compiler – Necessary for the installation of additional Perl components.
make	-	Make Tool – Necessary for the installation of additional Perl components.
pcre	-	Perl library of regular expressions.
rcs	-	Versioning software.
Editors		
nano	-	Simple text editor.
Interpreters		
perl	-	Interpreter for the Perl script language.
Libs		
w32api	+	Access to Windows functions.
Net		
ncftp	-	FTP program – Necessary for the installation of additional Perl components.
Web		
wget	-	For downloading files from the Internet – an alternative to ncftp.

- When you have finished, click on **Continue**. Now the necessary files will be copied and set up in the installation directory.
- Finally, you can indicate whether you would like to have an icon placed on your desktop and in your start menu. This completes the automatic installation of Cygwin.

Perl

Unfortunately, some Perl installations lack modules required by TWiki or its plugins for smooth operation. Thus, they must be subsequently installed. The program CPAN will help you do so, which must be configured after you have opened it for the first time.

Caution: CPAN requires an Internet connection for the installation of the new modules!

Under Cygwin, you first must enter the following lines:

```
export TEMP=/c/temp
```
Shell

Now you can start CPAN in Cygwin, as in Linux, with

```
cpan
```
Shell

When you first open CPAN, it must be configured. That means you will need to answer a host of questions that can all be confirmed with <ENTER>, until you are asked to indicate your area of the world. Enter the number of your continent, e.g. for "North America," and in the next question, your country, e.g. "USA." Now you will see a list of servers that you can select by entering one or more numbers separated by a space. Now you should see the `cpan>` prompt.

The program is ready, and you can set up the missing modules using the `install` command. TWiki, for example, requires the following three packages:

```
install Net::SMTP
install Digest::SHA1
install MIME::Base64
```
Shell

Perl is then equipped for all eventualities of TWiki operation. You can leave CPAN with the command `exit`.

Shebang

The Perl script files must be adapted so that they refer to the Perl interpreter. This allows the web server to "know" what to do with the file. Under UNIX, it is possible to indicate the type of file or program with which the script is to be executed at the beginning of a document, in the so-called shebang line. In our case, it must refer to the Perl interpreter. The TWiki scripts are set up by default such that they refer to the standard Perl path under Linux. Thus, under Windows, you will most likely have to adjust this. You can use the Cygwin shell to do so. The method shown is also valid for Linux.

First, switch to the *bin* directory and create a backup for the event that something does not function properly:

```
cd /twiki/bin
mkdir .backup
cp * .backup
```
Shell

With the command

Shell
```
head -1 view
```

you can display the current shebang line of the *view* scripts. The following Perl command changes the lines of all files in the directory. The Perl path is the entire path in Unix notation, including the name of the program, i.e. "perl." In our example, it is *c:/cygwin/bin/perl*.

Shell
```
perl -pi~ -e                                          ↵
 's;#!/usr/bin/perl;#!perlpath;' *[a-z]
```

If you now display the first line again, you should see the new path. Otherwise, delete the files and retrieve your backup from *.backup* to try again:

Shell
```
rm *
cp .backup/* .
```

If the changes have been successful, you can delete the backup files and directory again:

Shell
```
rm .backup/*
rmdir .backup
```

Changing *RCS* Owners

To manually adapt the RCS locks in the Cygwin or bash shell, first switch to the data directory and make a backup of the files:

Shell
```
cd /twiki/data
tar czvf all.tar.gz */*
```

Now change the entry for the owner using the following Perl command (put it all on one line):

Shell
```
perl -pi~~~ -e 'NR <= 10 &&                           ↵
  s/nobody:/username:/ ' */*,v
```

To be on the safe side, let us check to make sure everything has been changed. If you do not receive a message at this point, the operation was a success. In that case, you can delete the old files.

```
grep 'strict;$ ' */*,v | grep -v username
rm */*~~~
```
Shell

Should something have gone wrong, you can restore the former status by copying the backup back over. This is done using

```
tar xzvf all.tar.gz
```
Shell

Otherwise, delete the backup file with

```
rm all.tar.gz
```
Shell

When we open *testenv* again, we will still receive a warning, but this time we can happily ignore it.

Glossary

Add-on. An independent program that performs additional services for a certain piece of software. → Plugin.

Administrator (Admin). Software user with special rights. Has the task of installing, maintaining and updating the software and of establishing the conditions that normal users need to perform their daily business with the software.

ASCII. Refers to a standard of code for characters on a computer. ASCII text often stands for simple, non-formatted text. ASCII art is the depiction of drawings using simple text. Requires that the text be rendered in a non-proportional font.

Authoring tool. An application with which various data, such as text, graphics, sound or animation, can be assembled to create multimedia content.

Back end. Refers to that portion of a software application that runs on the server and administers the data. In comparison, the program that displays the data to the client is known as the front end. The back end area is only accessible to administrators.

Backlink. A link on another page that leads to the current page.

Blog. ("Weblog"). A type of web-based diary in which the blogger can quickly publish articles. These entries are displayed in a list, beginning with the newest entry. Other users can usually comment on but not alter entries.

CamelCase. Special practice of writing words together to form a link in wikis. A word begins with a capital letter and has at least one

other capital letter in the middle. Usually, several words are simply joined together, e.g. “WikiWord” or “WhatIsCamelCase.”

Clone. In wiki terminology, the not exactly precise designation for all programs that are based on Cunningham's *WikiWikiWeb*.

Community. The group of people who work and interact together in a wiki.

Content Management System (CMS). Software for the storage, administration and search for documents or content. Web-based CMSs are often used to jointly create a homepage.

Copyleft. License model that guarantees the possibility of freely copying, distributing and editing a work provided that the free license is distributed along with the work.

Creative Commons. Free license that regulates the use of a document at various levels. The author has the option of stipulating that his name be cited, as well as of permitting commercial use and prohibiting, allowing or generally permitting alteration if the license is observed (“share alike”). Creative Commons is thus not necessarily the same as copyleft. The license text is available in three versions: a short version for the layperson, the legally accurate long version and a machine-readable version (e.g. for search engines). This text has been translated into several languages.

CSS. Cascading Style Sheet. A formalism that describes the way in which an HTML file looks. With CSS, layout and formats of a website can be centrally administered with little effort.

Diff. A function in wiki that displays the differences between two → Versions of a page.

Edit War. Two or more users repeatedly delete each other's articles from the wiki.

Escape character. A character that “masks” a subsequent character. It is used to make a character that normally has a specific significance to the computer be treated as a normal character.

Extreme Programming. (“XP”). Programming technique in which two people always work on the same computer and the program parts are integrated into a running test system at very short intervals.

This enables closely customer-oriented programming. The wiki was originally developed to meet the demands of XP.

FDL. ("Free Documentation License"). A license for documents emanating from the software license → GPL. It guarantees that the use of the document adheres to → Copyleft. The user is obligated to name everyone involved in the generation of the document and to distribute the document exclusively in connection with the FDL. The license text must be displayed or printed along with the document, which is problematic especially for smaller documents. In addition, the FDL is only available in the English language. An alternative is the → Creative Commons license.

Flame War. Two or more users engage in the exchange of senseless and insulting name-calling in a forum such as a wiki, or others.

Front end. That part of a web-based piece of software that is accessible to all users. The opposite of → Back end.

GNU GPL. ("General Public License"). Software license that established the → Copyleft principle. Guarantees that programs subject to this license may be copied, distributed and passed on under the condition that the license is not altered. For documents, the → FDL was developed as an offshoot.

Groupware. Software that supports collaboration in a group. This includes such tasks as setting appointments, producing collective notes and having collective access to documents, task management, and of course communication possibilities as well.

Hash. The symbol "#".

History. List of all → Versions of a page. Using it, the history of edits to that page can be tracked.

Hypertext. Instead of a single long text, several text parts are connected via links.

Implementation. Concrete execution of a program in a programming language.

License. The granting and regulation of user rights to a work or document by the author. The licensing of documents is a very new phenomenon that is closely linked to the digital existence of docu-

ments and their ability to be copied. Innovations are expected in this area in the coming years. → Copyleft, Creative Commons, FDL

Maintainer. Wiki user with special technical access rights. Makes sure that wiki content is maintained, conventions are followed and the climate in the → Community is and remains pleasant.

Mindmap. Associative depiction of ideas. Various keywords are ordered around a central theme and connected with lines. These keywords can, in turn, serve as starting points for further associations.

Newsfeed. Computer-readable summary of a website that is queried at regular intervals by a reading program and prepared for the user. The reader is thus kept abreast of any changes. → RSS.

Open edit. A text that anyone can alter such that the alterations appear in the original text.

Open source. The source text of a program is freely accessible and may be altered and distributed by anyone.

Open space. Model of a group workshop in which the management of participants is kept to a minimum, and great trust is placed in the powers of self-control of the group itself.

Open text. An existing text that can be used freely by anyone. The original, however, remains unaltered.

Parent page. In TWiki, the page from which another page is generated. It can also be edited manually in order to establish a hierarchy.

Pipe. The symbol "|".

Plugin. A program component which can be coupled to an existing piece of software in order to provide it with supplementary functions. → Add-on.

Production mode. Refers to the use of software "in real life." The opposite of this is the test mode.

RecentChanges. A wiki page that lists the most recent changes to a page.

Revision. TWiki term for → Version.

Rollback. (also “revert”). Restoration of a previous → Version of a page.

Root directory. Starting point (root) of a directory structure.

RSS. Standard for → Newsfeeds. Is supported by several → RecentChanges pages in wikis.

SandBox. Test page in wiki on which anyone can learn how to use the software.

Script. A small program that is usually in source text and can be executed with the aid of an “interpretation program” (interpreter).

Shebang line. The first line of →Scripts that indicates in Unix the language in which the script is written and/or what program is required to execute it.

Syntax. Notation type or convention, in this case, for computer files. It must be observed exactly so that the computer can “understand” what is meant.

Sysop. → Administrator.

Tag. A type of label with which information is indicated about a certain area of a document. In HTML, tags are used to indicate formats.

Template. Layout prototype into which further content is integrated by the software (e.g. by a wiki).

Topic. TWiki term for a wiki page.

Troll. A troublemaker in a wiki who tries to gain attention by posting senseless or provocative articles.

Vandal. Deletes or destroys pages in a wiki with a destructive intent. → Troll.

Version. The status of a page at a particular time. With every edit, a new version is generated; however, old versions still remain accessible (usually via a version number).

Web. In TWiki, an enclosed area in which the → Topics are located. Otherwise, it is an abbreviation for the World Wide Web (WWW).

Web host. The service provider who makes storage space and the address for your homepage available.

Workflow. Precisely defined flow of work steps to execute a certain task. Serves to coordinate cooperative work.

WYSIWYG editor. "What you see is what you get." Refers to editors with which text can be edited directly in its correctly formatted form. Thus, while you are entering text, you can already see what the text will look like in print or on the display.

Bibliography

Aronsson, Lars (2002): *Operation of a Large Scale. General Purpose Wiki Website. Experience from susning.nu's first nine months in service.* http://aronsson.se/wikipaper.html, Nov. 7, 2004.

Benjamin, Walter (1972-1989): *Gesammelte Schriften*, 7 vol., Suhrkamp, Frankfurt a. M. (GS)

Brecht, Bertolt (1930/1967): *Der Rundfunk als Kommunikationsapparat*, in: Gesammelte Werke, vol. 18, Frankfurt a. M., 127-134.

Candeias, Mario (2004): *Neoliberalismus – Hochtechnologie – Hegemonie. Grundrisse einer transnationalen kapitalistischen Produktions- und Lebensweise. Eine Kritik*, Argument, Hamburg.

Dauscher, Ulrich (1998): *Moderationsmethode und Zukunftswerkstatt*, 2nd edition, Luchterhand, Neuwied/Kriftel/Berlin.

Degele, Nina (2002): *Einführung in die Techniksoziologie* (=UTB Vol. 2288), Fink, Munich.

Ebersbach, Anja und Glaser, Markus (2004): *Towards Emancipatory Use of a Medium: The Wik*i, in: International Journal of Information Ethics 2/2004, http://container.zkm.de/ijie/ijie/no002/ijie_002_09_ebersbach.pdf

Enzensberger, Hans-Magnus (1970): *Baukasten zu einer Theorie der Medien*, in: Kursbuch 20, 159-186.

Ferber, Reginald (2003): *Information Retrieval*, Dpunkt, Heidelberg.

Grassmuck, Volker (2002): *Freie Software. Zwischen Privat- und Gemeineigentum*, Bundeszentrale für politische Bildung, Bonn.

Haug, Wolfgang Fritz (2003): *High-Tech-Kapitalismus. Analyse zu Produktionsweise, Arbeit, Sexualität, Krieg und Hegemonie*, Argument, Hamburg.

Himanen, Pekka (2001): *Die Hacker-Ethik und der Geist des Informationszeitalters*, Riemann, Munich.

IBM (2003): *History Flow. Visualizing Dynamic, Evolving Documents and the Interactions of Multiple Collaborating Authors: a Preliminary Report*, http://www.research.ibm.com/history, Jan. 15, 2005.

Kluge, Friedrich (2002): *Etymologisches Wörterbuch der deutschen Sprache*, ed.: Elmar Sebold, 24th edition, Berlin/New York.

Krüger, Alfred (2004): *Wikinews geht an den Start*, telepolis Dec. 1 12, 2004, http://www.heise.de/tp/r4/artikel/18/18928/1.htm, Jan. 15, 2005.

Kuhlen, Rainer (1991): *Hypertext*, Springer, Berlin.

Kuhlen, Rainer (2004): *Wenn Autoren und ihre Werke Kollaborateure werden – was ändert sich dann? Oder: wenn Kommunikation ein Recht, gar ein Menschenrecht wird – was ändert sich dann?*, in: C. Bieber; C. Leggewie (ed.): Interaktivität – ein transdisziplinärer Schlüsselbegriff, Campus, Frankfurt. As a PDF on the Internet at: http://www.inf-wiss.uni-konstanz.de/People/RK/-publikationen.html, Jan. 15, 2005.

Leuf, Bo/Cunningham, Ward (2001): *The Wiki Way. Quick Collaboration on the Web*, Addison Wesley, Boston.

Maleh, Carol (2000): *Open Space: Effektiv arbeiten mit großen Gruppen*, Beltz, Weinheim/Basel.

Marx-Engels-Werke, Berlin/GDR, 1957ff. (MEW).

Möller, Erik (2003): *Bearbeiten Sie diesen Text. Tanz der Gehirne*, http://www.humanist.de/erik/tdg, Nov. 9, 2004.

Möller, Erik (2005): *Die heimliche Medienrevolution. Wie Weblogs, Wikis und freie Software die Welt verändern*, Heise, Hannover.

Meretz, Stefan (2003): *Freie Software - Ideen für eine andere Gesellschaft*, in: Anja Ebersbach, Richard Heigl, Thomas Schnakenberg (ed.): Missing Link. Fragen an die Informationsgesellschaft, Universitätsverlag Regensburg, Regensburg, 99-134.

Nielson, Jakob (1995): *Multimedia and Hypertext*, AP Professional, Boston.

Nuss, Sabine (2003): *Download ist Diebstahl? Eigentum in einer digitalen Welt*, in: Ebersbach, Anja/Heigl, Richard/Schnakenberg, Thomas (ed.): Missing Link. Fragen an die Informationsgesellschaft, Universitätsverlag Regensburg, Regensburg, 135-158.

Patalong, Frank: *Wiki-News. Und noch ein Blog...*, Spiegel online Dec. 6, 2004.

Petri, Katrina (2000): *Open Space Technology*, in: Roswitha Königswiese/Marion Keil (ed.): Das Feuer großer Gruppen. Konzepte, Designs, Praxisbeispiele für Großveranstaltungen, Klett-Cotta, Stuttgart, 146-163.

Portny, Stanley E. (2001): *Projektmanagement für Dummies*, mitp-Verlag, Bonn.

Rauter, E. A. (1977): *Vom Faustkeil zur Fabrik. Warum die Werkzeuge den Menschen und die Menschen die Werkzeuge verändern*, Weismann, Munich.

Raymond, Eric (1999): *Das Erfolgsgeheimnis von Linux. Die Kathedrale und der Basar*, translated from American English by R. Gantar, http://www.phone-soft.com/RaymondCathedralBazaar/catb_g.0.html#toc15, Nov. 28, 2004.

Rifkin, Jeremy (2000): *The Age of Access: The New Culture of Hypercapitalism, Where All Life is a Paid-For Experience*. Putnam, New York.

Rilling, Rainer (2004): *Internet* (=Stichwort für das HKWM, Vol. 6/II), http://www.rainer-rilling.de/texte/inkrit-internet.html, Dec. 10, 2004.

Schwall, Johannes (2003): *The Wiki Phenomenon*, http://www.schwall.de/thewikiphenomenon/index.php, Jan. 15, 2005.

Weizenbaum, Joseph (1977): *Die Macht der Computer und die Ohnmacht der Vernunft*, Suhrkamp, Frankfurt.

Index

S

T

U

V

W

X