EXT: Indexed Search Engine

Extension Key: **indexed_search**

The content of this document is related to TYPO3

- a GNU/GPL CMS/Framework available from www.typo3.com

Table of Contents

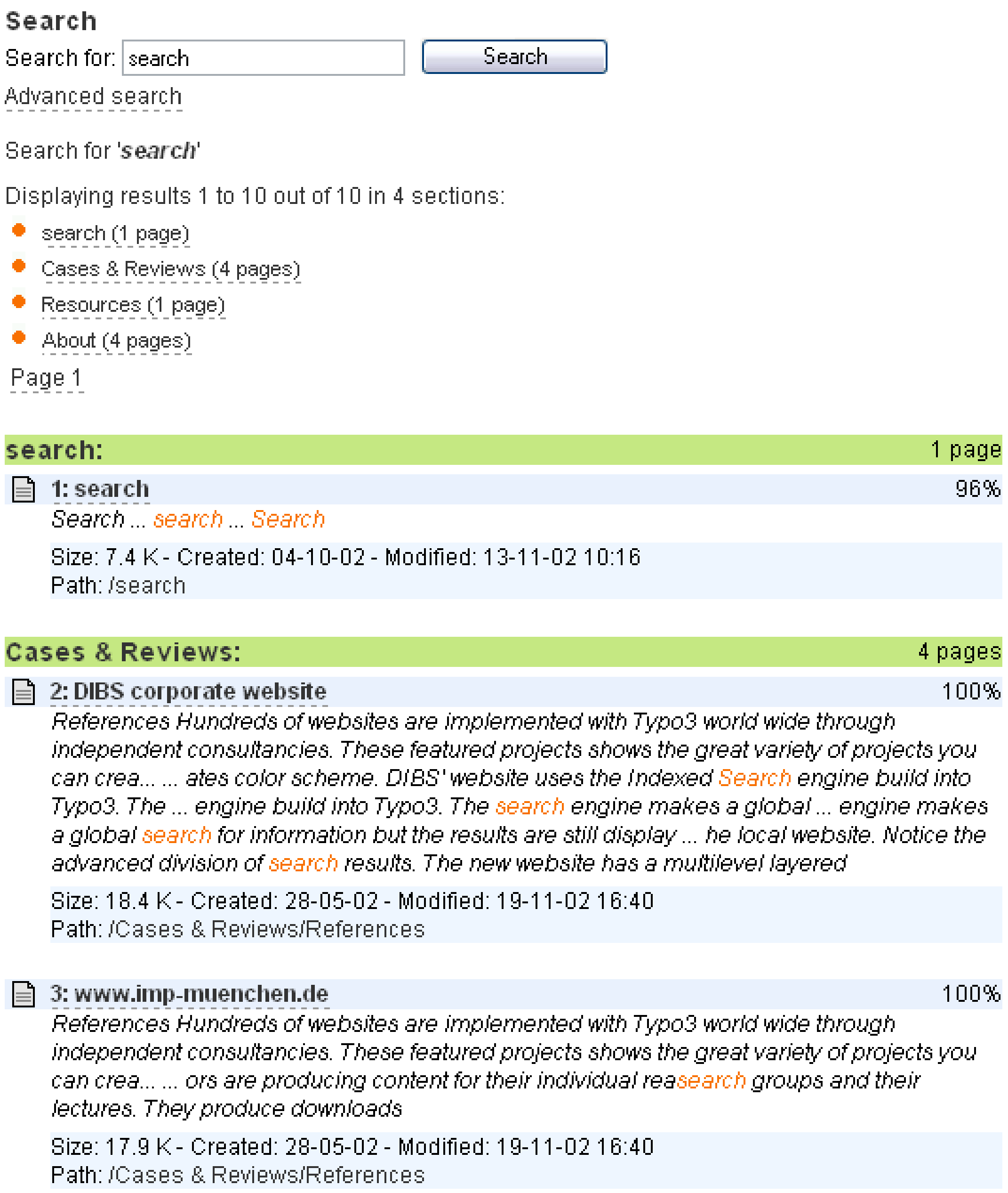
Introduction

What does it do?

The Indexed Search Engine provides two major elements to TYPO3:

1. **Indexing:** An indexing engine which indexes TYPO3 pages on-the-fly as they are rendered by TYPO3's frontend. Indexing a page means that all words from the page (or specifically defined areas on the page) are registered, counted, weighted and finally inserted into a database table of words. Then another table will be filled with relation records between the word table and the page. This is the basic idea.
2. **Searching:** A plugin you can insert on your website which allows website users to search for information on your website. By searching the plugin first looks in the word-table if the word exist and if it does all pages which has a relation to that word will be considered for the search result display. The search results are ordered based on factors like where on the page the word was found or the frequency of the word on the page.

This is an example of how the search interface on a website looks:



Search

Search for: search [Search]

Advanced search

Search for '*search*'

Displaying results 1 to 10 out of 10 in 4 sections:

- search (1 page)
- Cases & Reviews (4 pages)
- Resources (1 page)
- About (4 pages)

Page 1

search: 1 page

1: search 96%

Search ... search ... Search

Size: 7.4 K - Created: 04-10-02 - Modified: 13-11-02 10:16
Path: /search

Cases & Reviews: 4 pages

2: DIBS corporate website 100%

References Hundreds of websites are implemented with Typo3 world wide through independent consultancies. These featured projects shows the great variety of projects you can crea... ... ates color scheme. DIBS' website uses the Indexed Search engine build into Typo3. The ... engine build into Typo3. The search engine makes a global ... engine makes a global search for information but the results are still display ... he local website. Notice the advanced division of search results. The new website has a multilevel layered

Size: 18.4 K - Created: 28-05-02 - Modified: 19-11-02 16:40
Path: /Cases & Reviews/References

3: www.imp-muenchen.de 100%

References Hundreds of websites are implemented with Typo3 world wide through independent consultancies. These featured projects shows the great variety of projects you can crea... ... ors are producing content for their individual reasearch groups and their lectures. They produce downloads

Size: 17.9 K - Created: 28-05-02 - Modified: 19-11-02 16:40
Path: /Cases & Reviews/References

Features of the indexer

The indexing engine has several features:

- HTML data priority: 1) <title>-data 2) <meta-keywords>, 3) <meta-description>, 4) <body>
- Indexing external files: Text formats like html and txt and doc, pdf by external programs (catdoc / pdftotext)
- Wordcounting and frequency used to rate results
- Exact, partially or metaphone search
- Searching freely for sentences (non-indexed).
- NOT case-sensitive in any ways though.

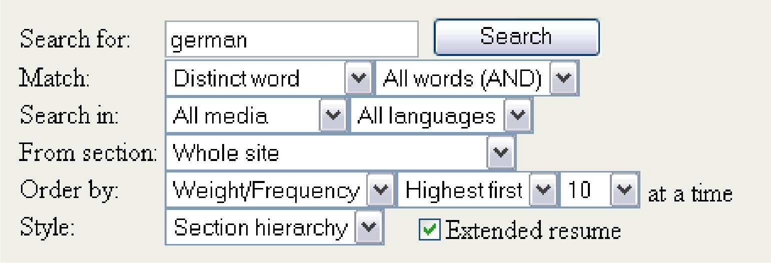
Features of the search frontend (the plugin)

The search interface has several options for advanced searching. Any of those can be disabled and/or preset with default values:

- Searching whole word, part of word, sounds like, sentence
- Logical AND and OR search including syntactical recognition of AND, OR and NOT as logical keywords. Furthermore sentences encapsulated in quotes will be recognized.

- Searching can be targeted at specific media, for instance searching only indexed PDF files, HTML-files, Word-files, TYPO3-pages or everything
- The engine is language sensitive based on the multiple-language feature of TYPO3's CMS frontend.
- Searching can be performed in specific sections of the website.
- Results can be sorted descending or ascending and ordered by word frequency, weight, location relative to page top, page modification date, page title, etc.
- The display of search results can be intelligently divided into sections based on the internal page hierarchy. Thus results are primarily grouped by relation, then by hit-relevance.

This shows the full range of default options for “advanced search”:



Search for: german

Match: Distinct word, All words (AND)

Search in: All media, All languages

From section: Whole site

Order by: Weight/Frequency, Highest first, 10 at a time

Style: Section hierarchy, ☑ Extended resume

Warning

The search frontend plugin is optimized for features, *not* speed. Especially it will be slow on a website with many pages in the page tree because it traverses the whole tree each time to build a list of accessible pages. However you can circumvent this by modifications to the search plugin so it does not check page access based on the id-list. But then you loose that feature of course. Can't have both.

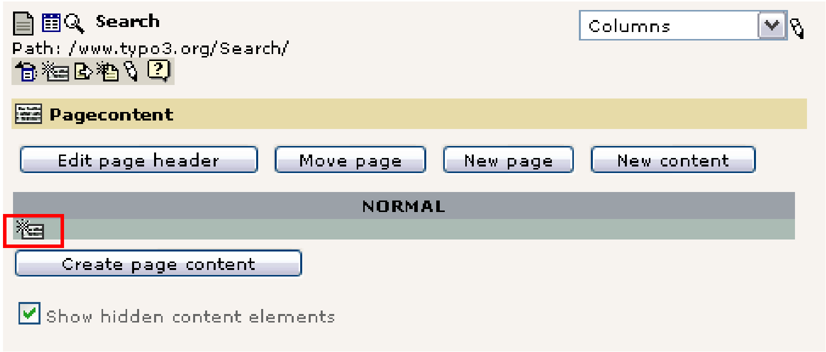
In any case; The indexing of pages and searching the indexed information are two different processes and therefore you can easily use another frontend plugin for making searches in the same data for whatever reason you might have for discarding the default search plugin.

User manual

Adding the search plugin to a page

That is really easy:

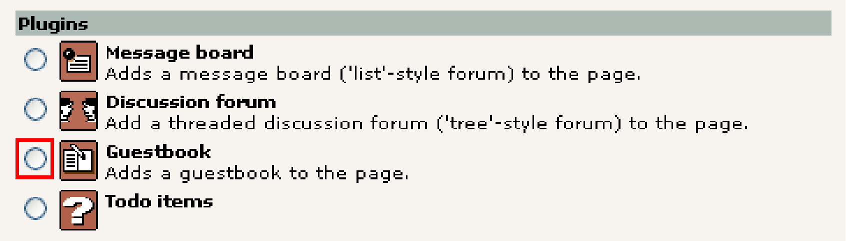
1. Create a page called “Search” or something like that. This is where the search box will appear.
2. Then create a new content element on that page. From the Web>Page module you can do it like this:



Search

Path: /www.typo3.org/Search/

3. Then select some plugin-type if you can. It doesn't matter if it's a guestbook or forum. Or if no plugins are available, just select a “Regular text element” as in the top of the page.



Plugins

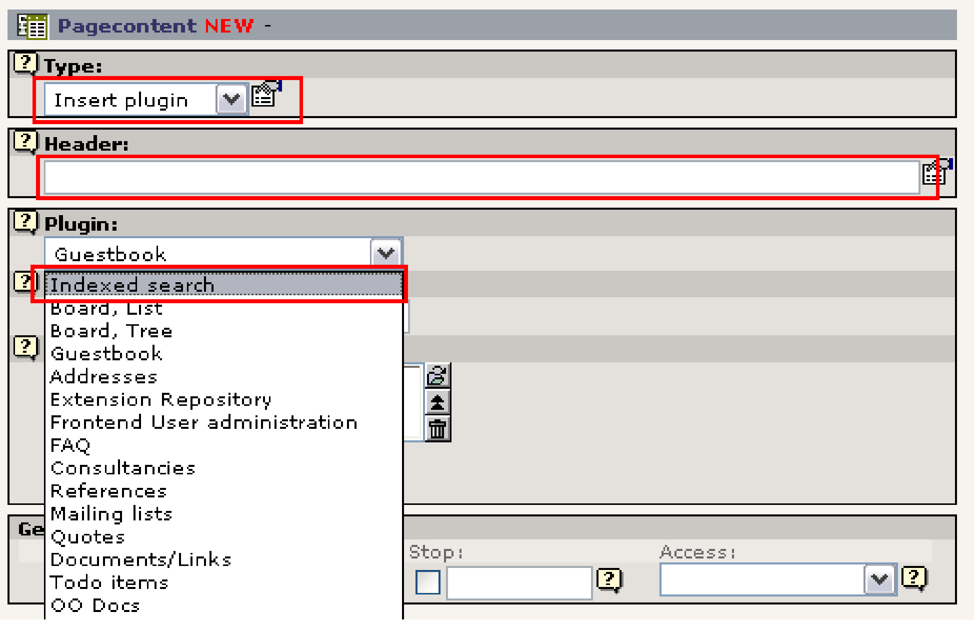
○ **Message board**
Adds a message board ('list'-style forum) to the page.

○ **Discussion forum**
Add a threaded discussion forum ('tree'-style forum) to the page.

○ **Guestbook**
Adds a guestbook to the page.

○ **Todo items**

4. Then make sure “Insert plugin” is selected (if not, select it and save the element, then you'll see the form below), enter a title and select the “Plugin” type to be “Indexed search”:



Pagecontent NEW -

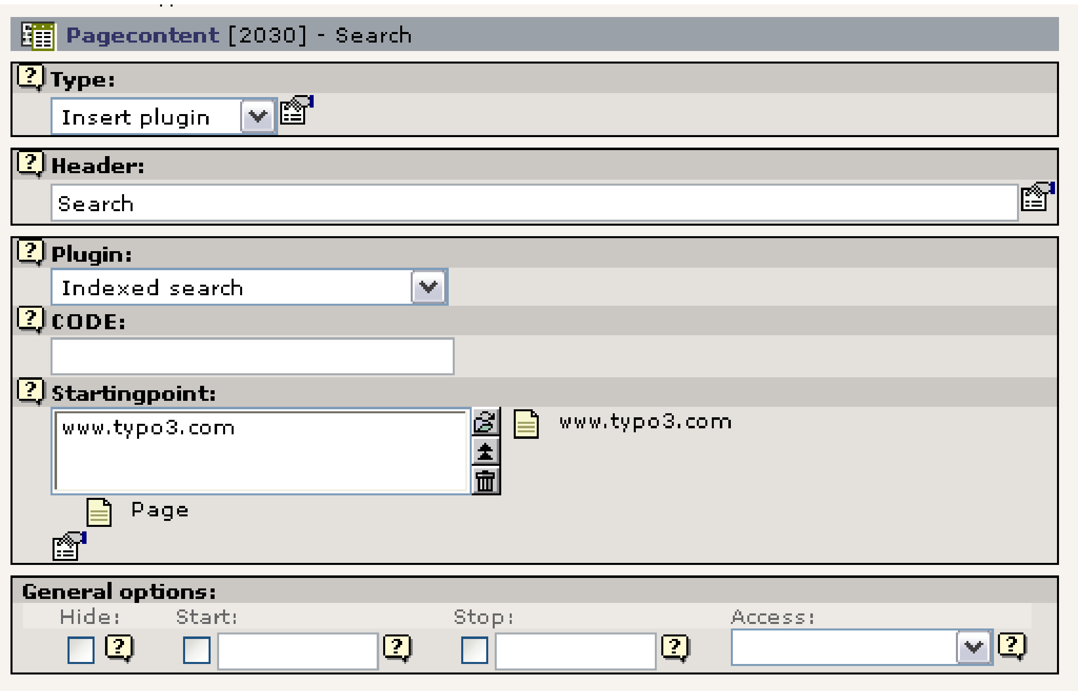
Type: Insert plugin

Header:

Plugin: Guestbook

Indexed search
Board, List
Board, Tree
Guestbook
Addresses
Extension Repository
Frontend User administration
FAQ
Consultancies
References
Mailing lists
Quotes
Documents/Links
Todo items
OO Docs

5. Then select the root page of your website as the “Starting point” of the plugin content element:



Pagecontent [2030] - Search

Type: Insert plugin

Header: Search

Plugin: Indexed search

CODE:

Startingpoint: www.typo3.com

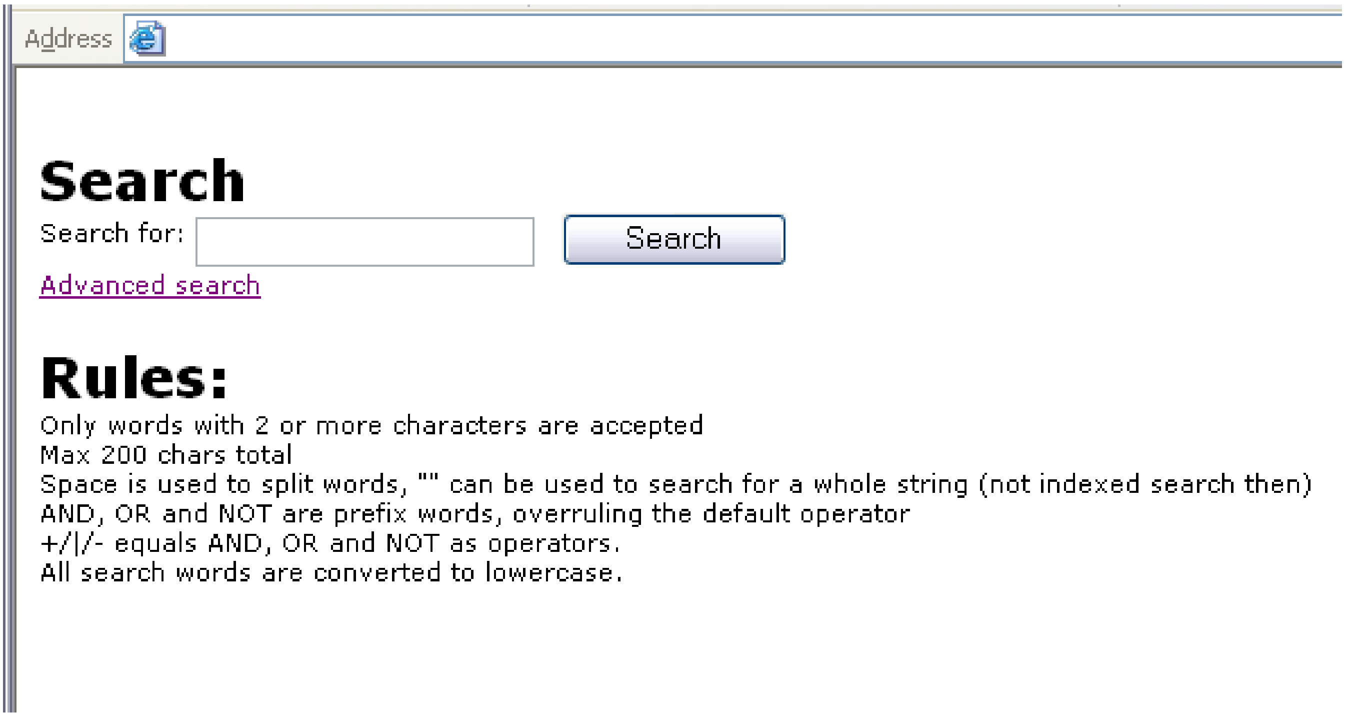
www.typo3.com

Page

General options:

Hide: □ Start: □ Stop: □ Access:

And that's it. Your frontend should now look like this:



Rules:

Only words with 2 or more characters are accepted
Max 200 chars total
Space is used to split words, "" can be used to search for a whole string (not indexed search then)
AND, OR and NOT are prefix words, overruling the default operator
+/|/- equals AND, OR and NOT as operators.
All search words are converted to lowercase.

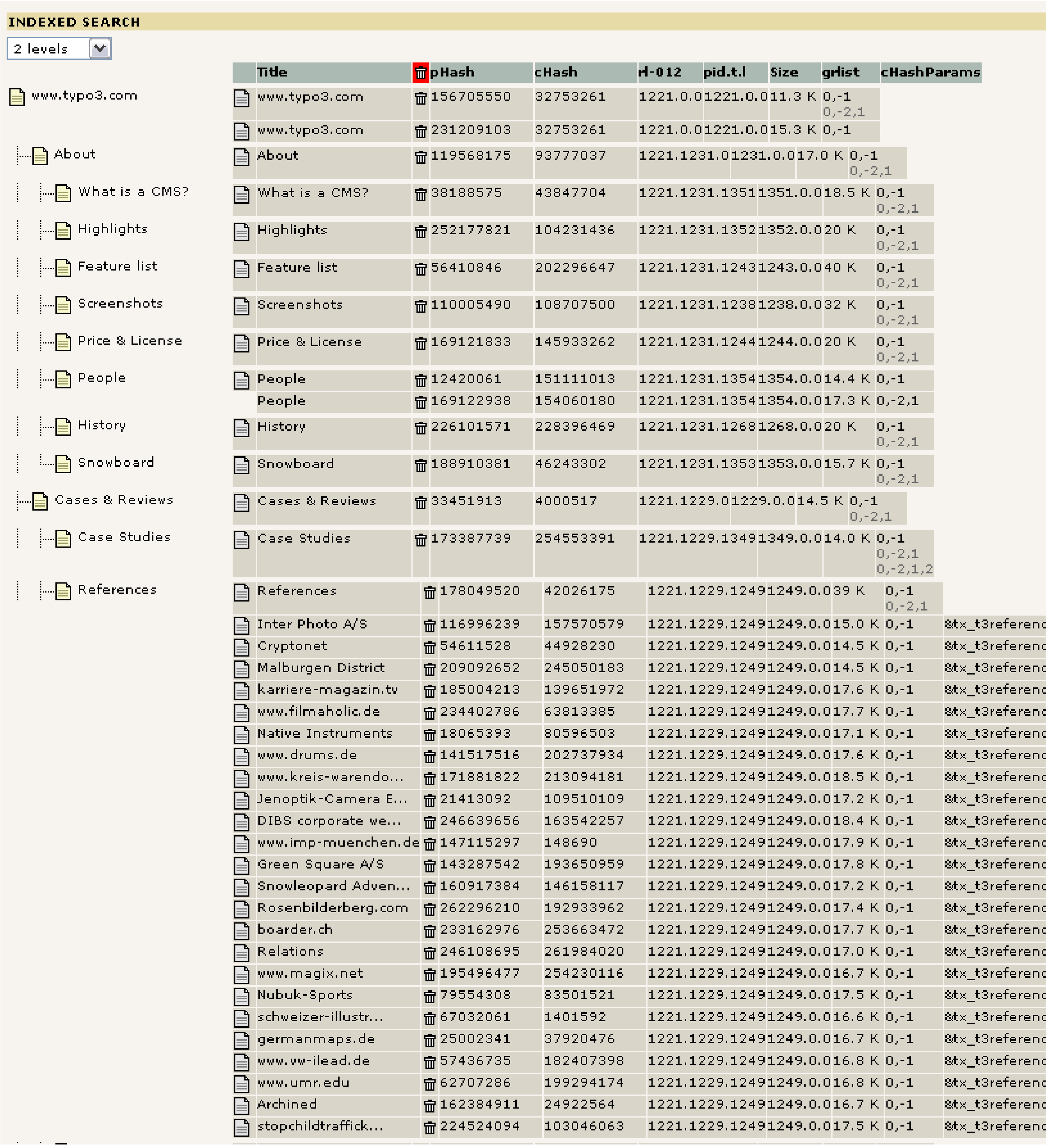
The styles are most likely different from this, but that is controlled by the developer having administration access to the system.

Adminstration

Monitoring indexed content

The Indexed Search extension adds two backend modules, one as a global database-wide statistics module and a page specific analysis module.

In the Web>Info module you can see an overview of how many instances are indexed per TYPO3-page. Look at this image:

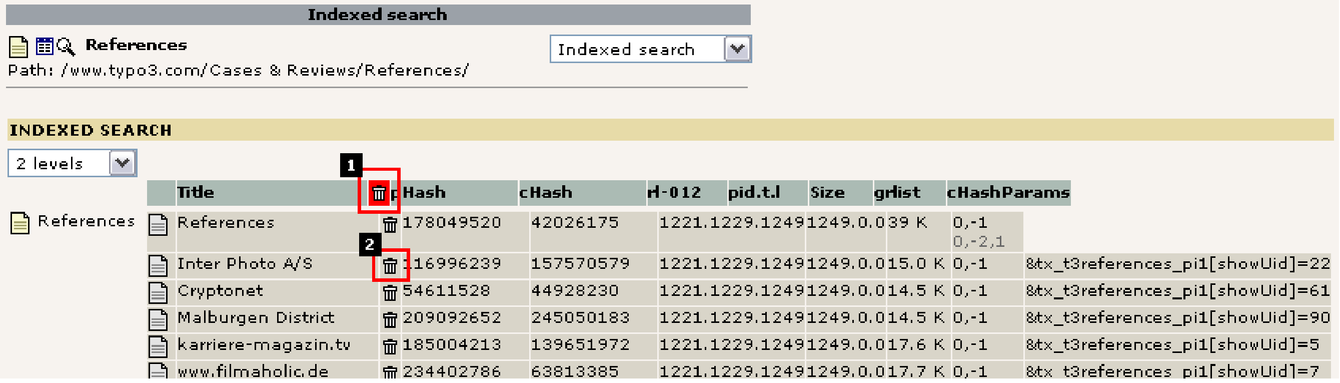


Title	pHash	cHash	rl-012	pid.t.l	Size	grlist	cHashParams
www.typo3.com	156705550	32753261	1221.0.0	1221.0.0	11.3 K	0,-1 0,-2,1	
www.typo3.com	231209103	32753261	1221.0.0	1221.0.0	15.3 K	0,-1	
About	119568175	93777037	1221.1231.0	1231.0.0	17.0 K	0,-1 0,-2,1	
What is a CMS?	38188575	43847704	1221.1231.1351	1351.0.0	18.5 K	0,-1 0,-2,1	
Highlights	252177821	104231436	1221.1231.1352	1352.0.0	20 K	0,-1 0,-2,1	
Feature list	56410846	202296647	1221.1231.1243	1243.0.0	40 K	0,-1 0,-2,1	
Screenshots	110005490	108707500	1221.1231.1238	1238.0.0	32 K	0,-1 0,-2,1	
Price & License	169121833	145933262	1221.1231.1244	1244.0.0	20 K	0,-1 0,-2,1	
People	12420061	151111013	1221.1231.1354	1354.0.0	14.4 K	0,-1	
People	169122938	154060180	1221.1231.1354	1354.0.0	17.3 K	0,-2,1	
History	226101571	228396469	1221.1231.1268	1268.0.0	20 K	0,-1 0,-2,1	
Snowboard	188910381	46243302	1221.1231.1353	1353.0.0	15.7 K	0,-1 0,-2,1	
Cases & Reviews	33451913	4000517	1221.1229.0	1229.0.0	14.5 K	0,-1 0,-2,1	
Case Studies	173387739	254553391	1221.1229.1349	1349.0.0	14.0 K	0,-1 0,-2,1 0,-2,1,2	
References	178049520	42026175	1221.1229.1249	1249.0.0	39 K	0,-1 0,-2,1	
Inter Photo A/S	116996239	157570579	1221.1229.1249	1249.0.0	15.0 K	0,-1	&tx_t3referenc
Cryptonet	54611528	44928230	1221.1229.1249	1249.0.0	14.5 K	0,-1	&tx_t3referenc
Malburgen District	209092652	245050183	1221.1229.1249	1249.0.0	14.5 K	0,-1	&tx_t3referenc
karriere-magazin.tv	185004213	139651972	1221.1229.1249	1249.0.0	17.6 K	0,-1	&tx_t3referenc
www.filmaholic.de	234402786	63813385	1221.1229.1249	1249.0.0	17.7 K	0,-1	&tx_t3referenc
Native Instruments	18065393	80596503	1221.1229.1249	1249.0.0	17.1 K	0,-1	&tx_t3referenc
www.drums.de	141517516	202737934	1221.1229.1249	1249.0.0	17.6 K	0,-1	&tx_t3referenc
www.kreis-warendo...	171881822	213094181	1221.1229.1249	1249.0.0	18.5 K	0,-1	&tx_t3referenc
Jenoptik-Camera E...	21413092	109510109	1221.1229.1249	1249.0.0	17.2 K	0,-1	&tx_t3referenc
DIBS corporate we...	246639656	163542257	1221.1229.1249	1249.0.0	18.4 K	0,-1	&tx_t3referenc
www.imp-muenchen.de	147115297	148690	1221.1229.1249	1249.0.0	17.9 K	0,-1	&tx_t3referenc
Green Square A/S	143287542	193650959	1221.1229.1249	1249.0.0	17.8 K	0,-1	&tx_t3referenc
Snowleopard Adven...	160917384	146158117	1221.1229.1249	1249.0.0	17.2 K	0,-1	&tx_t3referenc
Rosenbilderberg.com	262296210	192933962	1221.1229.1249	1249.0.0	17.4 K	0,-1	&tx_t3referenc
boarder.ch	233162976	253663472	1221.1229.1249	1249.0.0	17.7 K	0,-1	&tx_t3referenc
Relations	246108695	261984020	1221.1229.1249	1249.0.0	17.0 K	0,-1	&tx_t3referenc
www.magix.net	195496477	254230116	1221.1229.1249	1249.0.0	16.7 K	0,-1	&tx_t3referenc
Nubuk-Sports	79554308	83501521	1221.1229.1249	1249.0.0	17.5 K	0,-1	&tx_t3referenc
schweizer-illustr...	67032061	1401592	1221.1229.1249	1249.0.0	16.6 K	0,-1	&tx_t3referenc
germanmaps.de	25002341	37920476	1221.1229.1249	1249.0.0	16.7 K	0,-1	&tx_t3referenc
www.vw-ilead.de	57436735	182407398	1221.1229.1249	1249.0.0	16.8 K	0,-1	&tx_t3referenc
www.umr.edu	62707286	199294174	1221.1229.1249	1249.0.0	16.8 K	0,-1	&tx_t3referenc
Archined	162384911	24922564	1221.1229.1249	1249.0.0	16.7 K	0,-1	&tx_t3referenc
stopchildtraffick...	224524094	103046063	1221.1229.1249	1249.0.0	17.5 K	0,-1	&tx_t3referenc

As you can see most pages here are indexed only one time. However a few are indexed twice. This can happen for several reasons and here the reason is most likely due to a user login or something related.

The most interesting occurence is the page “References” which has more than 20 indexed instances available. The reason is that this page holds multiple cached views due to some parameters which are used by a plugin on that page. Each instance will be searchable as a unique search result.

Now imagine that you want to clear out all those instances of the “References” page to let them be re-indexed when viewed again; Simply click the page “References” in the page tree to the left. Then you see this:



Indexed search

References

Path: /www.typo3.com/Cases & Reviews/References/

Indexed search

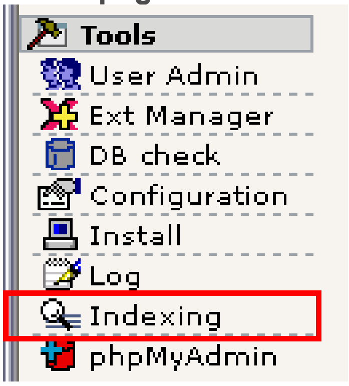
INDEXED SEARCH

2 levels

	Title		pHash	cHash	rl-012	pid.t.l	Size	grlist	cHashParams
References	References		178049520	42026175	1221.1229.1249	1249.0.0	39 K	0,-1 0,-2,1	
	Inter Photo A/S		116996239	157570579	1221.1229.1249	1249.0.0	15.0 K	0,-1	&tx_t3references_pi1[showUid]=22
	Cryptonet		54611528	44928230	1221.1229.1249	1249.0.0	14.5 K	0,-1	&tx_t3references_pi1[showUid]=61
	Malburgen District		209092652	245050183	1221.1229.1249	1249.0.0	14.5 K	0,-1	&tx_t3references_pi1[showUid]=90
	karriere-magazin.tv		185004213	139651972	1221.1229.1249	1249.0.0	17.6 K	0,-1	&tx_t3references_pi1[showUid]=5
	www.filmaholic.de		234402786	63813385	1221.1229.1249	1249.0.0	17.7 K	0,-1	&tx_t3references_pi1[showUid]=7

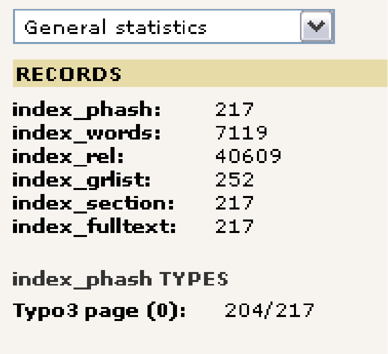
You can either click the red garbage bin (1) in order to clear all listed instances or alternatively pick out single instances by clicking the local garbage bin (2).

Monitoring the global picture of indexed pages



By the Tools>Indexing module you can get statistics about the indexing engine. Currently they are sparse and very roughly presented. This view needs some more work to be friendly and really useful.

“General statistics”



General statistics

RECORDS

index_phash:	217
index_words:	7119
index_rel:	40609
index_grlist:	252
index_section:	217
index_fulltext:	217

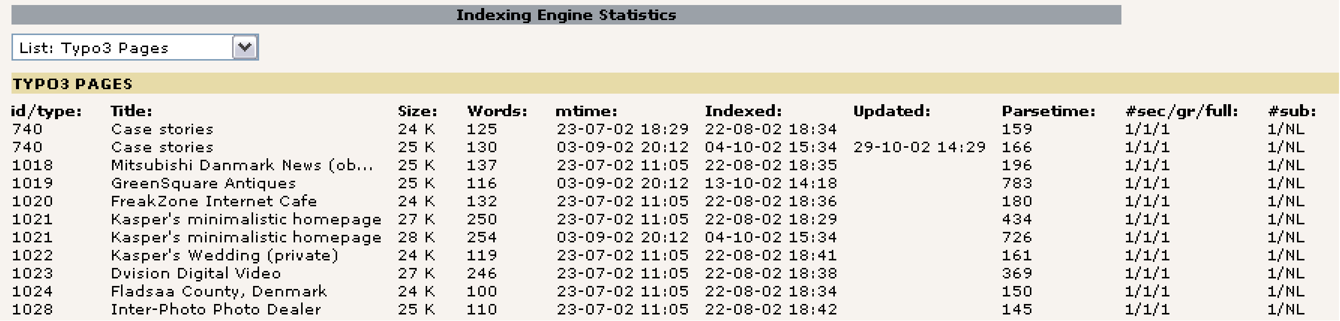
index_phash TYPES

Typo3 page (0):	204/217

This shows that 217 pages are indexed, comprising 7000+ words and using 40.000 records in the relation table to glue things together.

“List: TYPO3 Pages”

This view shows a list of indexed pages with all the technical details:



Indexing Engine Statistics

List: Typo3 Pages

TYPO3 PAGES

id/type:	Title:	Size:	Words:	mtime:	Indexed:	Updated:	Parsetime:	#sec/gr/full:	#sub:
740	Case stories	24 K	125	23-07-02 18:29	22-08-02 18:34		159	1/1/1	1/NL
740	Case stories	25 K	130	03-09-02 20:12	04-10-02 15:34	29-10-02 14:29	166	1/1/1	1/NL
1018	Mitsubishi Danmark News (ob...	25 K	137	23-07-02 11:05	22-08-02 18:35		196	1/1/1	1/NL
1019	GreenSquare Antiques	25 K	116	03-09-02 20:12	13-10-02 14:18		783	1/1/1	1/NL
1020	FreakZone Internet Cafe	24 K	132	23-07-02 11:05	22-08-02 18:36		180	1/1/1	1/NL
1021	Kasper's minimalistic homepage	27 K	250	23-07-02 11:05	22-08-02 18:29		434	1/1/1	1/NL
1021	Kasper's minimalistic homepage	28 K	254	03-09-02 20:12	04-10-02 15:34		726	1/1/1	1/NL
1022	Kasper's Wedding (private)	24 K	119	23-07-02 11:05	22-08-02 18:41		161	1/1/1	1/NL
1023	Dvision Digital Video	27 K	246	23-07-02 11:05	22-08-02 18:38		369	1/1/1	1/NL
1024	Fladsaa County, Denmark	24 K	100	23-07-02 11:05	22-08-02 18:34		150	1/1/1	1/NL
1028	Inter-Photo Photo Dealer	25 K	110	23-07-02 11:05	22-08-02 18:42		145	1/1/1	1/NL

Configuration

General

The most basic requirement for the search engine to work is that pages are getting indexed. That will not happen by just installing the plugin! You will have to set up in TypoScript that a certain page should be indexed. That is needed for several good reasons. First of all not all sites in a TYPO3 database might need indexing. So therefore we disable it on a per-site basis. Secondly a single site may have frames and in that case we need only index the page-object which actually shows the page content.

Lets say that you have a PAGE object called "page" (that is pretty typical), then you will have to set this config-option:

```
page.config.index_enable = 1
```

When this option is set you should begin to see your pages being indexed when they are shown next time. Remember that only cached pages are indexed!

This is documented in TSref in the CONFIG section. Please look there for further options. For instance indexing of external media can also be enabled there.

Languages

The plugin supports all system languages in TYPO3. Translation is done using the typo3.org tools.

If you want to use eg. danish language that will automatically be used if this option is set in your template (the value is the internal language key):

```
config.language = dk
```

TypoScript

[Still missing the major parts here. Just use the object browser for now since that includes all options]

Property:	Data type:	Description:	Default:
specConfs.[pid]	-	“specConfs” is an array of objects with properties that can customize certain behaviours of the display of a result row depending on it's position in the rootline. For instance you can define that all results which links to pages in a branch from page id 123 should have another page icon displayed. Of you can add a suffix to the class names so you can style that section differently. **Examples:** If a page “Contact” is found in a search for “address” and that “Contact” page is in the rootline “Frontpage [ID=23] > About us [ID=45] > Contact [ID=77]” then you should set the pid value to either “77” or “45”. If “45” then all subpages including the “About us” page will have similar configuration. If the pid value is set to 0 (zero) it will apply to all pages. Please see the options below.	
specConfs.[pid].pageIcon	->IMAGE cObject	Alternative page icon.	
specConfs.[pid].CSSsuffix	string	A string that will be appended to the class-names of all the class-attributes used within the result row presentation. The prefix will be like this: **Example:** If “...CSSsuffix = doc” then eg. the class name “tx-indexedsearch-title” will be “tx-indexedsearch-title-doc”	

Property:	Data type:	Description:	Default:
search.rootPidList	list of int	A list of integer which should be root-pages to search from. Thus you can search multiple branches of the page tree by setting this property to a list of page id numbers. If this value is set to less than zero (eg. -1) searching will happen in ALL of the page tree with no regard to branches at all. Notice that by “root-page” we mean a website root defined by a TypoScript Template! If you just want to search in branches of your site, use the possibility of searching in levels.	The current root-page id
search.detect_sys_domain_records	boolean	If set, then the search results are linked to the proper domains where they are found.	
search.detect_sys_domain_records.target	string	Target for external URLs.	

[tsref:plugin.tx_indexedsearch]

Technical details

HTML content

HTML content is weighted by the indexing engine in this order:

1. <title>-data
2. <meta-keywords>
3. <meta-description>
4. <body>

In addition you can insert markers as HTML comments which define which part of the body-text to include or exclude in the indexing:

The marker is **<!--TYPO3SEARCH_begin-->** or **<!--TYPO3SEARCH_end-->**

Rules:

1. If there is no marker at all, everything is included.
2. If the first found marker is an “end” marker, the previous content until that point is included and the preceeding code until next “begin” marker is excluded.
3. If the first found marker is a “begin” marker, the previous content until that point is excluded and preceeding content until next “end” marker is included.

Use of hashes

The hashes used are md5 hashes where the first 7 chars are converted into an integer which is used as the hash in the database. This is done in order to save space in the database, thus using only 4 bytes and not a varchar of 32 bytes. It's estimated that a hash of 7 chars (32) is sufficient (originally 8, but at some point PHP changed behavior with hexdec-function so that where originally a 32 bit value was input half the values would be negative, they were suddenly positive all of them. That would require a similar change of the fields in the database. To cut it simple, the length was reduced to 7, all being positive then).

How pages are indexed

First of all a page must be cachable. For pages where the cache is disabled, no indexing will occur.

The “phash” is a unique identification of a “page” with regard to the indexer. So an entry in the index_phash table equals 1 resultrow in the search-results (called a phash-row).

A phash is a combination of the page-id, type, sys_language id, gr_list, MP and the cHash parameters of the page (function setT3Hashes()). If the phash is made for EXTERNAL media (item_type > 0) then it's a combination of the absolute filename hashes with any “subpage” indication, for instance if a PDF-document is splitted into subsections.

So for external media there is one phash-row for each file (except PDF-files where there may be more). But for TYPO3-pages there can be more phash-rows matching one single page. Obviously the type-parameter would normally always be only one, namely the type-number of the content page. And the cHash may be of importance for the result as well with regard to plugins using that. For instance a message board may make pages cachable by using the cHash params. If so, each cached page will also be indexed. Thus many phash-rows for a single page-id.

But the most tricky reason for having multiple phash-rows for a single TYPO3-page id is if the gr_list is set! This works like this: If a page has *exactly* the same content both with and without logins, then it's stored only once! If the page-content differs whether a user is logged in or not - it may even do so based on the fe_groups! - then it's indexed as many times as the content differs. The phash is of course different, but the phash_grouping value is the same.

The table index_grlist will always hold one record per phash-row (of item_type=0, that is TYPO3 pages). But it may also hold many more records. These point to the phash-row in question in the case of other gr_list combinations which actually had the

SAME content - and thus refers to the same phash-row.

External media

External media (pdf, doc, html, txt) is tricky. External media is always detected as links to local files in the content of a TYPO3 page which is being indexed. But external media can the linked to from more than one page. So the index_section table may hold many entries for a single external phash-record, one for each position it's found. Also it's important to notice that external media is only indexed or updated if a “parent” TYPO3 page is re-indexed. Only then will the links to the external files be found. In a searching operation external media will be listed only once (grouping by phash), but say two TYPO3 pages are linking to the document, then only one of them will be shown as the path where the link can be found. However if both TYPO3 pages are not available, then the document will not be shown.

Handling extendToSubpages - or not

In the searching plugin there are two ways of searching with respect to accessible pages.

1) join_pages=1: If set, then the final result rows are joined with the pages table. This will make sure that no enableFields-hidden (but NOT extendToSubpages) pages are selected. And it will also make sure to search ALL pages within the rl0 of the index_section table. But extendToSubpages will NOT be taken into account!

2) join_pages=0 (default): Then a long list of page-ids are selected first and after that the final result-rows are selected but without joining the pages-table. This will work with a limited number of page-ids (which means most sites). And it makes sure that any extendToSubpages-hidden pages are NOT selected along with enableFields-hidden pages. BUT it will also prevent pages down the branch of a “php_tree_stop” from being selected as well.

Access restricted pages

A TYPO3 page will always be available in the search result only if there is access to the page. This is secured in the final result query. Whether extendToSubpages is taken into account depends on the join_pages-flag (see above). But the page will only be listed if the user has access.

However a page may be indexed more than once if the content differ from usergroup to usergroup or just without login. Still the result display will display only one occurrence, because similar pages (determined based on phash_grouping) will be detected.

The tricky scenario:

Say that a page has a content element with some secret information visible for only one usergroup. The page as a whole will be visible for all users. The page will be indexed twice - both without login and with login because page content differs. The problem is that if a search is conducted and matching one of the secret words in the access restricted section, then the page will be in the search result even if the user is not logged in!

The best solution to this problem is to allow the result to be listed anyway, but then HIDE the resume if the index_grlist table cannot confirm positively that the combination of usergroups of the user has access to the result. So the result is there, but no resume shown (The resume might contain hidden text).

External media

Equally for external media they are linked from a TYPO3 page. When an external media is selected we can be sure that the page linking to it can be selected. But we cannot be sure that the link was in a section accessible for the user. Similarly we should make a lookup in the index_grlist table selecting the phash/gr_list by the phash_t3-value of the section record for the search-result. If this is not available we should not display a link to the document and not show resume, but rather link to the page, from which the user can see the real link to the document.

Note:

These tricky scenarios exist only if the content on a page differs based on login. It does not affect situations with access restriction to the page as a whole. A general lesson from this is to reduce the number of hidden content elements! Instead use hidden pages. Better, more reliable.

Analysing the indexed data

The indexer is constructed to work with TYPO3's page structure. Opposite to a crawler which simply indexes all the pages it can find, the TYPO3 indexer MUST take the following into account:

- Only cached pages can be indexed.
  Pages with dynamic content - such as search pages etc - should supply their own search engine for lookup in specific tables. Another option is to selectively allow certain of those “dynamic” pages to be cached anyways (see the cHashParams concept used by some plugins)
- Pages in more than one language must be indexed separately as “different pages”.
- Pages with messageboards may have multiple indexed versions based on what is displayed on the page: The overview or a single messageboard item? This is determined by the cHashParams value.
- Pages with access restricted to must be observed!
- Because pages can contain different content whether a user is logged in or not and even based on which groups he is a member of, a single page (identified by the combination of id/type/language/cHashParams) may even be available in more

than one indexed version based on the user-groups. But while the same page may have different content based on the user-groups (and so must be indexed once for each) such pages may just as well present the SAME content regardless of usergroups! This is the very most tricky thing.

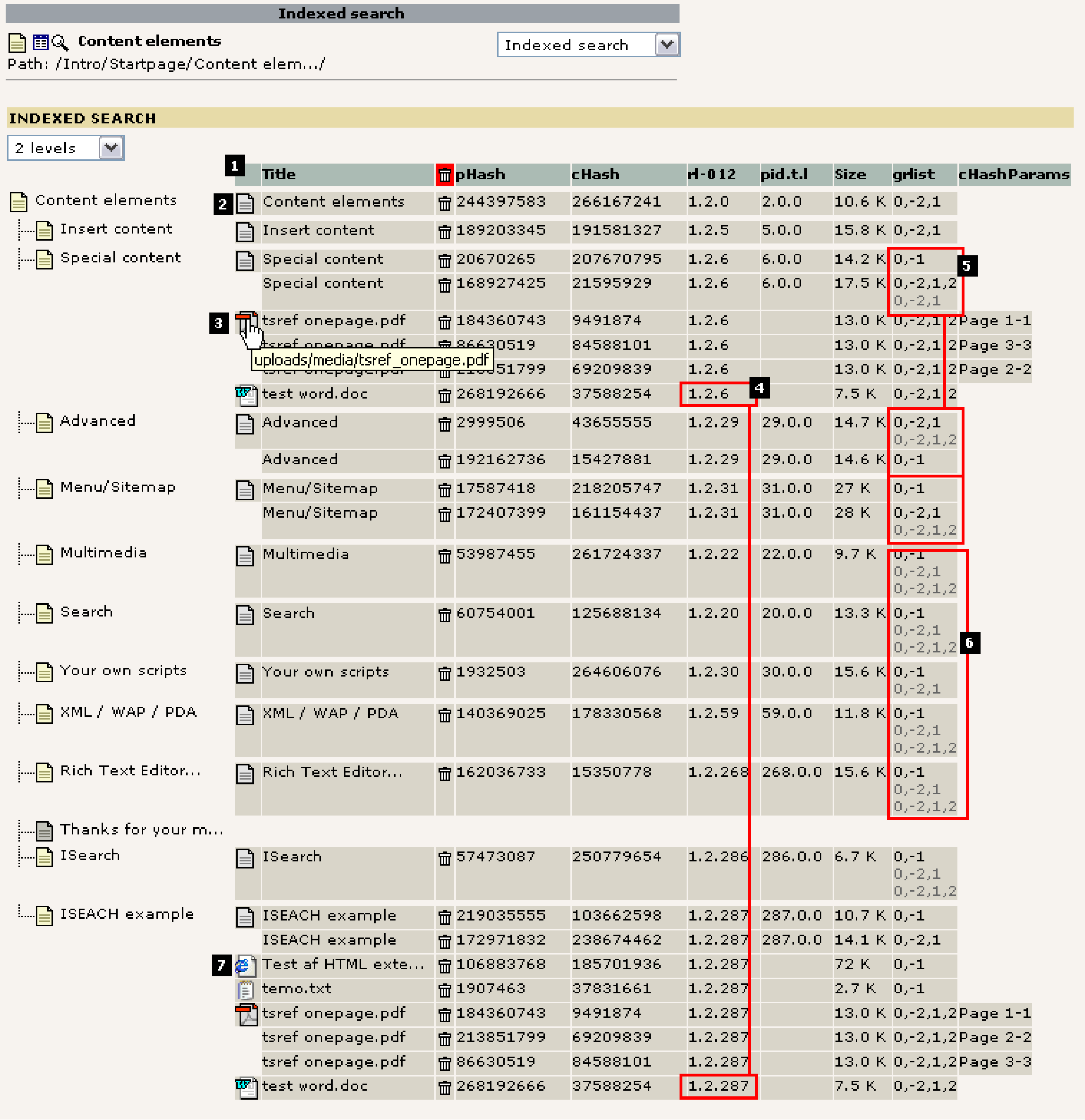
Understanding these complex scenarios...

The best thing to do is to grab an example. Please refer to the picture below while reading the bulletlist here:

1. The overview in general shows **one line per "phash-row"** (a single row from the index_phash table). Such a row represents a single hit in a searching session. In other words, each line with grayish background in the overview may be a search-hit. The columns of these rows are:
    - Title: The search-result title.
    - [icon]: Click here to remove the indexed information for this entry (will be re-indexed on the next hit).
    - pHash: The "id" of the search row. The hash is calculated based on id/type/language/MP/cHashParams/gr_list of the page when indexed. For external media this is based on filepath/page-interval (for PDF's only)
    - cHash: Calculated based on the actual content which was indexed.
    - rl-012: This is the rootline ids for level 0,1,2. Used when searching in certain sections. For instance a search-operation may select all pages with rl1=123 which will result in a search within pages which exist ONLY in the branch of the website where the level1-page has uid=123.
    - pid.t.l: This is the page-id / type-number / sys_language uid
    - Size: How many bytes the indexed page consumed
    - grlist: This is the gr_list of the user which initiated the indexing operation.
    - cHashParams: Additional parameters which are identifying the page in addition to the id/type number which usually does that.
2. The page "Content elements" has one indexed version. The page-id of the root-page is "1" and the page on level-1 in the rootline had the uid "2". Notice how all subpages to "Content elements" has the exact same rl0 and rl1 value. Where the page "Content elements" does NOT have a value for rl2 so does all the subpages (because they ARE the level 2 themselves!). Furthermore the page has the page-id "2", a type-value of "0" and is indexed with the default language "0". The size was 10.6 KB and the user who initiated the indexing operation was a member of the groups 0,-2,1 (which is effectively fe_group "1", because 0 and -2 is pseudogroups).
3. On the page "Special content" there must have been a link to a local PDF and Word file, since those two are indexed in relation to this page. The PDF-file is located in the path "uploads/media/tsref_onepage.pdf" relative to the website. Notice that the PDF file is actually indexed three times, one time per page. This is of course configurable. Each indexed section of the PDF-file has the potential to show up as a search-result row of course (because the phash is different per indexed part). The whole point with this is that a large PDF file might contain so much information that it might match all too many search-queries. So breaking a PDF-file down into smaller parts makes it possible for us to indicate exactly WHERE in the PDF-file the search word was found!
4. Looking at the word file (and the PDF-file as well) we see that they are found on BOTH the page "Special content" and on the page "ISEARCH example". But looking at the phash values (for the word-file it is "268192666") it is the SAME value in both cases. So this means, that the Word and PDF file is indexed only once - when it is first discovered! Later when another page is indexed and a link to the same document appears, then the document is not indexed as another document, but rather an entry in the index_section table is made indicating that this result row is also found available (linked to) from another page/section.
   Say you are doing a search in the section from "Content elements" and outwards in the page tree. The word-document is matched in the search, but it will appear only once in the search result. Now, if one of the two pages where the Word document was either hidden or access restricted the word-document would still be matched (because one of the pages is accessible for the user). But if BOTH pages with the link to the word document is not accessible for the user doing the search, then the word document will not be included in the search result.
5. Here we can see that the pages "Special content", "Advanced" and "Menu/Sitemap" is indexed twice each. The reason is that those three pages has had different content depending on whether or not a user was logged in!
   In the case of the page "Special content" the reason is that the page contained a content element which was visible for users which was a member of group number #1. Therefore the page was different in the two cases.
   The page "Advanced" has a user-login form and that form looks different whether a user is logged in or not.
   Finally the page "Menu/Sitemap" apparently changed. There reason was that this page includes a sitemap and that sitemap displayed some extra pages when the logged in users hit the page and so the content was not the same as without login.
   Another thing which is interesting is that two different users must have visited those pages. We can see that because the page "Special content" was apparently indexed with the usergroup combination "1,2". Later another user hit the page but only a member of group "1". However the page content was the SAME. And because those two users saw the very same page, it was not indexed a third time, but it was instead noted down that a user with membership of only group "1" did also see this same page. That comparison was based on the cHash (contentHash) which is a hash-value based on the actual content being indexed. So when the user with group "1" only came to the page, the indexer engine realize that the page as it looked has already been indexed because another phash-row with that content hash was already available.
6. These pages does not contain any tricks it appears. According to the grlist's both users with membership of group "1,2" and group "1" only *as well* as surfers who did not at all login ("0,-1" is the pseudo-group for no login) as visited the page. And because only one indexed version exist the page must have had the same content to present all users regardless of

their login-status.
The reason why the page “Your own scripts” does not contain a grlist value “0,-2,1,2” as the others do is simply because no user with that combination of usergroups has ever visited the page!

7. txt and html documents can also be indexed as external media. In the case of HTML-documents the documents <title> is detected and used.



Path: /Intro/Startpage/Content elem.../

INDEXED SEARCH

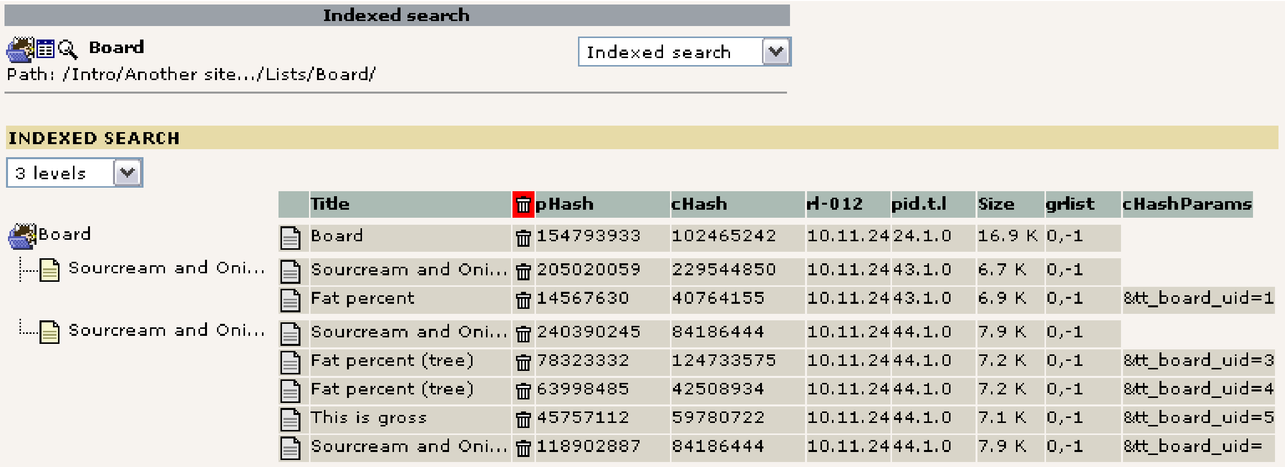
2 levels

- Content elements
  - Insert content
  - Special content
  - Advanced
  - Menu/Sitemap
  - Multimedia
  - Search
  - Your own scripts
  - XML / WAP / PDA
  - Rich Text Editor...
  - Thanks for your m...
  - ISearch
  - ISEACH example

Title	pHash	cHash	rl-012	pid.t.l	Size	grlist	cHashParams
Content elements	244397583	266167241	1.2.0	2.0.0	10.6 K	0,-2,1	
Insert content	189203345	191581327	1.2.5	5.0.0	15.8 K	0,-2,1	
Special content	20670265	207670795	1.2.6	6.0.0	14.2 K	0,-1	
Special content	168927425	21595929	1.2.6	6.0.0	17.5 K	0,-2,1,2 0,-2,1	
tsref onepage.pdf	184360743	9491874	1.2.6		13.0 K	0,-2,1,2	Page 1-1
tsref onepage.pdf	86630519	84588101	1.2.6		13.0 K	0,-2,1,2	Page 3-3
tsref onepage.pdf	213851799	69209839	1.2.6		13.0 K	0,-2,1,2	Page 2-2
test word.doc	268192666	37588254	1.2.6		7.5 K	0,-2,1,2	
Advanced	2999506	43655555	1.2.29	29.0.0	14.7 K	0,-2,1 0,-2,1,2	
Advanced	192162736	15427881	1.2.29	29.0.0	14.6 K	0,-1	
Menu/Sitemap	17587418	218205747	1.2.31	31.0.0	27 K	0,-1	
Menu/Sitemap	172407399	161154437	1.2.31	31.0.0	28 K	0,-2,1 0,-2,1,2	
Multimedia	53987455	261724337	1.2.22	22.0.0	9.7 K	0,-1 0,-2,1 0,-2,1,2	
Search	60754001	125688134	1.2.20	20.0.0	13.3 K	0,-1 0,-2,1 0,-2,1,2	
Your own scripts	1932503	264606076	1.2.30	30.0.0	15.6 K	0,-1 0,-2,1	
XML / WAP / PDA	140369025	178330568	1.2.59	59.0.0	11.8 K	0,-1 0,-2,1 0,-2,1,2	
Rich Text Editor...	162036733	15350778	1.2.268	268.0.0	15.6 K	0,-1 0,-2,1 0,-2,1,2	
ISearch	57473087	250779654	1.2.286	286.0.0	6.7 K	0,-1 0,-2,1 0,-2,1,2	
ISEACH example	219035555	103662598	1.2.287	287.0.0	10.7 K	0,-1	
ISEACH example	172971832	238674462	1.2.287	287.0.0	14.1 K	0,-2,1	
Test af HTML exte...	106883768	185701936	1.2.287		72 K	0,-1	
temo.txt	1907463	37831661	1.2.287		2.7 K	0,-1	
tsref onepage.pdf	184360743	9491874	1.2.287		13.0 K	0,-2,1,2	Page 1-1
tsref onepage.pdf	213851799	69209839	1.2.287		13.0 K	0,-2,1,2	Page 2-2
tsref onepage.pdf	86630519	84588101	1.2.287		13.0 K	0,-2,1,2	Page 3-3
test word.doc	268192666	37588254	1.2.287		7.5 K	0,-2,1,2	

uploads/media/tsref_onepage.pdf

On the image below we are looking at another scenario. In this case the cHashParams is obviously used by the plugin “tt_board”. The plugin has been constructed so intelligently that it links to the messages in the message board without disabling the normal page-cache but rather sending the tt_board_uid parameter along with a so called “cHash”. If this is combined correctly the caching engine allows the page to be cached. Not only does this mean a quicker display of pages in the message board - it also means we can index the page!



Indexed search

Board

Path: /Intro/Another site.../Lists/Board/

INDEXED SEARCH

3 levels

	Title	pHash	cHash	rl-012	pid.t.l	Size	grlist	cHashParams
Board	Board	154793933	102465242	10.11.24	24.1.0	16.9 K	0,-1	
Sourcream and Oni...	Sourcream and Oni...	205020059	229544850	10.11.24	43.1.0	6.7 K	0,-1	
	Fat percent	14567630	40764155	10.11.24	43.1.0	6.9 K	0,-1	&tt_board_uid=1
Sourcream and Oni...	Sourcream and Oni...	240390245	84186444	10.11.24	44.1.0	7.9 K	0,-1	
	Fat percent (tree)	78323332	124733575	10.11.24	44.1.0	7.2 K	0,-1	&tt_board_uid=3
	Fat percent (tree)	63998485	42508934	10.11.24	44.1.0	7.2 K	0,-1	&tt_board_uid=4
	This is gross	45757112	59780722	10.11.24	44.1.0	7.1 K	0,-1	&tt_board_uid=5
	Sourcream and Oni...	118902887	84186444	10.11.24	44.1.0	7.9 K	0,-1	&tt_board_uid=

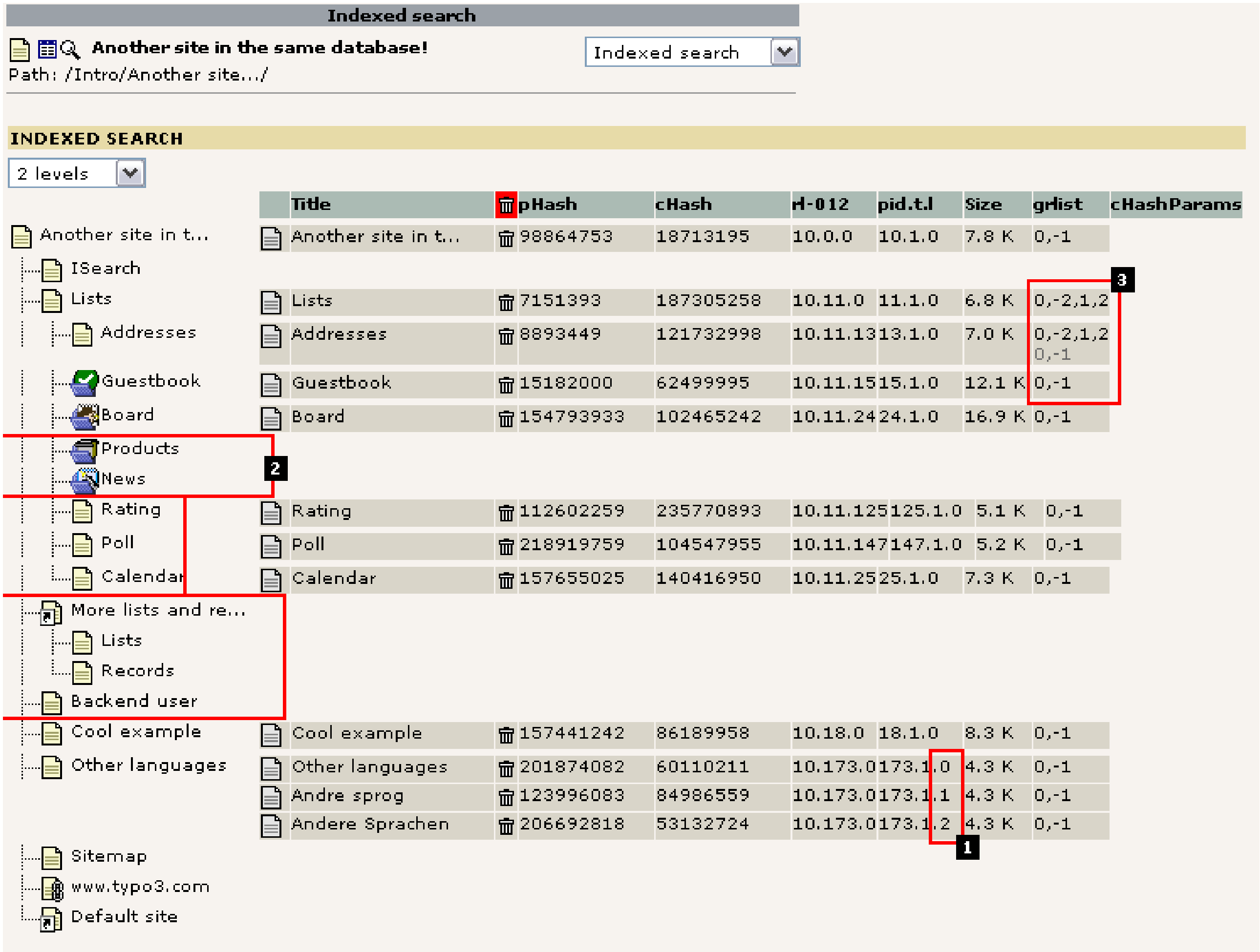
As you see the main board page showing the list of messages/threads ("Sourcream and Oni...") is indexed without any values for the parameter tt_board_uid (the cHashParams field is blank). Then it has also been indexed one time for each display of a message. In a search result any of these five rows may appear as an independent result row - after all they are to be regarded as a single page with unique content, despite sharing the same page-id!

Another interesting thing is that while the main page has inherited the page title for the search-result ("Sourcream and ...") each of the indexed pages with a message has got another title - namely the subject line of the message shown! Thus a search matching three of these five pages will not shown three similar page-titles but a unique page title relative to the actual content on the page. It is the tt_board plugin that sets the page-title itself by an API-call.

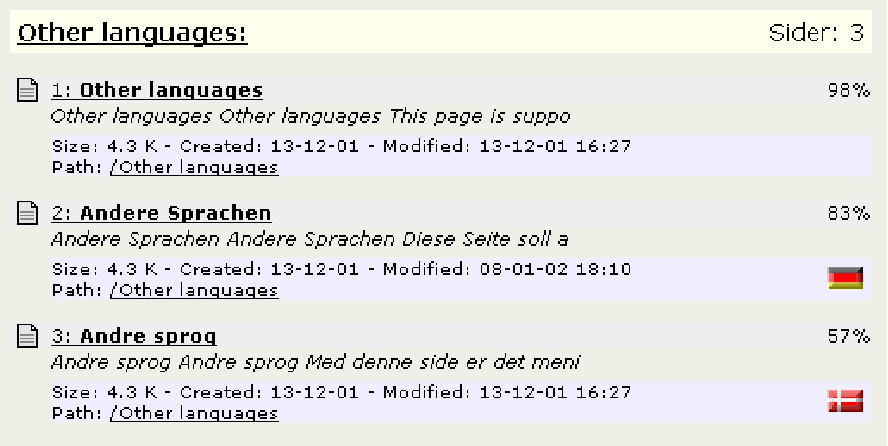
The only glitch here is that the tt_board plugin has falsely allowed the main page to be cached twice. See the first and last phash-row. The last row has got the parameter "&tt_board_uid= " sent and the tt_board plugin should not have allowed that! Because looking at the content hash of the first and last we realize that it's the SAME hash (84186444) and therefore the SAME content! However being two separate result rows they will both be displayed in the search result as separate hits. The responsibility for this lies with the plugin. However such occurrences can be automatically filtered out during the search result display. But it's better to avoid this kind of stuff.

The last example below has three main issues to discuss:

1. The page "Other languages" is apparently available in three languages. Which ones are not possible to determine unless we know the value from the sys_languages table. In this case the default language (zero - 0) is english and the language with id 1 and id 2 is danish and german versions of the page.
   When a search is conducted each page may turn up as a result page but with a little flag telling if the page was found in another language than the main language on the website (see second illustration hereafter)
2. If there is no phash-rows found for a page this can mean three things:
   1. Either the page is not cached. In this case both the tt_products and tt_news plugins apparently disables the caching of the page thereby disabling any indexing of the pages. Searching in news and products must be done with a searching function looking up directly in the news and products tables.
   2. In the case with other pages the reason may be that the pages has never been visited and therefore not indexed yet! Indexing of pages in TYPO3 happens during the rendering of the page - there is currently no "crawler" to assist this job.
   3. Finally the reason for a page not being indexed can be the combination of 1 and 2: That the page has never been visited. And if it was visited, the cache would have been disabled.
3. These numbers just tells us that:
   - the page "Lists" was indexed once by a user with membership of group 1 and 2.
   - the page "Addresses" was also indexed by a user with membership of group 1 and 2 but has since been visited by a user without login. Both instances yielded a similar page and it was therefore not indexed twice.
     This raises the question about the page "Lists": Is that access-restricted for users without login or has a user without login just never visited that page since no "0,-1" grlist has been detected? Both could be the answer. On pages which has access-restriction (or a whole section in an intranet) such pages would obviously not have been indexed by no-login users. However in this case nothing indicates that the page should be hidden for non-login users and so we must conclude that the page has simply not yet been visited by a no-login user - otherwise it would look like the page "Addresses" having also the "0,-1" list detected.
   - The "Guestbook" page was indexed by a user without login only.



Title	pHash	cHash	rl-012	pid.t.l	Size	grlist	cHashParams
Another site in t...	98864753	18713195	10.0.0	10.1.0	7.8 K	0,-1	
Lists	7151393	187305258	10.11.0	11.1.0	6.8 K	0,-2,1,2	
Addresses	8893449	121732998	10.11.13	13.1.0	7.0 K	0,-2,1,2 0,-1	
Guestbook	15182000	62499995	10.11.15	15.1.0	12.1 K	0,-1	
Board	154793933	102465242	10.11.24	24.1.0	16.9 K	0,-1	
Rating	112602259	235770893	10.11.125	125.1.0	5.1 K	0,-1	
Poll	218919759	104547955	10.11.147	147.1.0	5.2 K	0,-1	
Calendar	157655025	140416950	10.11.25	25.1.0	7.3 K	0,-1	
Cool example	157441242	86189958	10.18.0	18.1.0	8.3 K	0,-1	
Other languages	201874082	60110211	10.173.0	173.1.0	4.3 K	0,-1	
Andre sprog	123996083	84986559	10.173.0	173.1.1	4.3 K	0,-1	
Andere Sprachen	206692818	53132724	10.173.0	173.1.2	4.3 K	0,-1	



Other languages: Sider: 3

1: Other languages 98%
Other languages Other languages This page is suppo
Size: 4.3 K - Created: 13-12-01 - Modified: 13-12-01 16:27
Path: /Other languages

2: Andere Sprachen 83%
Andere Sprachen Andere Sprachen Diese Seite soll a
Size: 4.3 K - Created: 13-12-01 - Modified: 08-01-02 18:10
Path: /Other languages

3: Andre sprog 57%
Andre sprog Andre sprog Med denne side er det meni
Size: 4.3 K - Created: 13-12-01 - Modified: 13-12-01 16:27
Path: /Other languages

Illustration 1A seach result showing how localized versions of a page are displayed.

Database Tables

index_phash

This table contains references to TYPO3 pages or external documents. The fields are like this:

phash	7md5/int hash. It's an integer based on a 7-char md5-hash. This is a unique representation of the 'page' indexed. For TYPO3 pages this is a serialization of id,type,gr_list (see later), MP and cHashParams (which enables 'subcaching' with extra parameters). This concept is also used for TYPO3 caching (although the caching hash includes the all-array and thus takes the template into account, which this hash does not! It's expected that template changes through conditions would not seriously alter the page content) For external media this is a serialization of 1) unique filename id, 2) any subpage indication (parallel to cHashParams). gr_list is NOT taken into consideration here!
phash_grouping	7md5/int hash. This is a non-unique hash exactly like phash, but WITHOUT the gr_list and (in addition) for external media without subpage indication. Thus this field will indicate a 'unique' page (or file) while this page may exist twice or more due to gr_list. Use this field to GROUP BY the search so you get only one hit per page when selecting with gr_list in mind. Currently a seach result does not either group or limit by this, but rather the result display may group the result into logical units.
item_mtime	Modification time: For TYPO3 pages: the SYS_LASTCHANGED value For external media: The filemtime() value. Depending on config, if mtime hasn't changed compared to this value the file/page is not indexed again.
tstamp	time stamp of the indexing operation. You can configure min/max ages which are checked with this timestamp. A min-age defines how long an indexed page must be indexed before it's reconsidered to index it again. A max-age defines an absolute point at which re-indexing will occur (unless the content has not changed according to an md5-hash)
cHashParams	The cHashParams. For TYPO3 pages: These are used to re-generate the actual url of the TYPO3 page in question For files this is an empty array. Not used.
item_type	An integer indicating the content type, 0 is TYPO3 pages 1- external files like pdf (2), doc (3), html (1), txt (4) and so on. See the class.indexer.php file
item_title	Title: For TYPO3 pages, the page title For files, the basename of the file (no path)
item_description	Short description of the item. Top information on the page. Used in search result.
data_page_id	For TYPO3 pages: The id
data_page_type	For TYPO3 pages: The type
data_filename	For external files: The filepath (relative) or URL (not used yet)
contentHash	md5 hash of the content indexed. Before reindexing this is compared with the content to be indexed and if it matches there is obviously no need for reindexing.
crdate	The creation date of the INDEXING - not the page/file! (see item_crdate)
parsetime	The parsetime of the indexing operation.
sys_language_uid	Will contain the value of GLOBALS["TSFE"]->sys_language_uid, which tells us the language of the page indexed.
item_crdate	The creation date. For files only the modification date can be read from the files, so here it will be the filemtime().
gr_list	Contains the gr_list of the user initiating the indexing of the document.

index_section

Points out the section where an entry in index_phash belongs.

phash	The phash of the indexed document.
phash_t3	The phash of the "parent" TYPO3 page of the indexed document. If the "document" being indexed is a TYPO3 page, then phash and phash_t3 are the same. But if the document is an external file (PDF, Word etc) which are found as a LINK on a TYPO3 page, then this phash_t3 points to the phash of that TYPO3 page. Normally it goes like this when indexing: 1) The TYPO3 document is indexed (this has a phash-value of course), then 2) if any external files are found on the page, they are indexed as well AND their phash_t3 will become the phash of the TYPO3 page they were on. The significance of this value is that indexed external files may have more than one record in "index_section" (with the same phash), a record for each parent page where a link to the document was found! There are details about this in the section of this document that describes the complexities of indexing pages.
rl0	The id of the root-page of the site.
rl1	The id of the level-1 page (if any) of the indexed page.
rl2	The id of the level-2 page (if any) of the indexed page.
page_id	The page id of the indexed page.
uniqid	This is just an autoincremented unique, primary key. Generally not used (i think)

index_fulltext

For free text searching, eg with a sentence, in all content: title, description, keywords, body

phash	The phash of the indexed document.
fulltextdata	The total content stripped for any HTML codes.

Currently the MySQL FULLTEXT search is not used (something with MATCH ... AGAINST), but this will be added in the future.

index_grlist

This table will hold records related to a phash-row. Records in this table confirms that certain gr_lists would actually share the same content as represented by phash-row - even though the phash-row may be indexed under another login. The table is used during result-display to positively confirm if the current user may see the resume (which otherwise might contain secret info). Please see discussion far above.

index_words, index_rel

Words-table and word-relation table. Almost self-explanatory. For the index_rel table some fields require explanation:

count	Number of occurrences on the page
first	How close to the top (low number is better)
freq	Frequency (please see source for the calculations. This is converted from some floating point to an integer)
flags	Bits, which describes the weight of the words: 8th bit (128) = word found in title, 7th bit (64) = word found in keywords, 6th bit (32) = word found in description, Last 5 bits are not used yet, but if used they will enter the weight hierarchy. The result rows are ordered by this value if the "Weight/Frequency" sorting is selected. Thus results with a hit in the title, keywords or description are ranked higher in the result list.

Known problems

- Currently the extension is under observation because instances of heavy server load/unstability has been reported. It is not yet clear if THIS extension has anything to do with. So it's only under suspicion at this point until further data has been collected. But for now it is adviced to be careful with the application of the extension for mission critical, high-load environments.

- It's still uncertain how performance is under heavy load conditions and when MANY pages are indexed. Currently benchmarks has been done only up to 2000 pages indexed/approx. 400.000 relation records. It is probably that some parts has to be optimized for such scenarios.