

IFile - INTRODUZIONE ALL'UTILIZZO

Agosto 2013

by
isApp.it

SOMMARIO

Introduzione a IFile	5
Obiettivi	5
Argomenti trattati	5
Glossario dei termini	5
Conoscere IFile	7
Cos'è IFile.....	7
Migliorare le proprie applicazioni con IFile	7
I formati supportati	7
Come utilizzare IFile.....	8
Pre-Requisiti	8
Zend Framework.....	9
Encoding	10
SERVER.....	10
XPDF.....	10
ANTIWORD	10
PHP	11
Estensioni PHP	11
Funzioni PHP	12
Il "server check" di IFile	13
Installare IFile	14
Configurare IFile	14
La xpdfrc	28
Fields e custom fields	28
Default Fields.....	28
Tag ID3 degli MP3.....	30
Tag Exif dei JPEG	31
Utilizzare le interfacce di IFile.....	33
Lucene Interface	33
MySqli Inteface.....	34
IFile_Indexing_Interface	35

Metodo per aggiungere un campo personalizzato	35
Metodo per aggiungere un documento all'indice.....	36
Metodo per definire l'autocommit dei documenti	36
Metodo per commitare i documenti nell'indice	37
Metodo per recuperare del numero di documenti indicizzati	37
Metodo per cancellare un documento dall'indice	37
Metodo per cancellare tutti i documento dall'indice.....	38
Metodo per recuperare tutti i documento dall'indice	38
Metodo per recuperare dei campi personalizzati	39
Metodo per recuperare il campo di default di ricerca	39
Metodo per recuperare un documento dall'indice.....	39
Metodo per recuperare i fields dell'indice	40
Metodo per recuperare le info sul file da indicizzare.....	40
Metodo per recuperare il limite massimo di risultati	41
Metodo per verificare se esistono documenti cancellati	41
Metodo per verificare l'esistenza di un termine	41
Metodo per verificare se un documento e' cancellato	42
Metodo per recuperare il numero di documenti	42
Metodo per ottimizzare l'indice	42
Metodo per la ricerca dei termini	42
Metodo per la ricerca BOOLEAN dei termini.....	43
Metodo per la ricerca FUZZY dei termini	43
Metodo per la ricerca con la sintassi di lucene /mysql	44
Metodo per la ricerca di frasi	44
Metodo per la ricerca in un range di termini	45
Metodo per la ricerca con caratteri jolly (wildcard).....	45
Metodo per settare il campo di default di ricerca.....	46
Metodo per settare il file da indicizzare.....	46
Metodo per settare il limite massimo di risultati.....	46
Metodo per settare il tipo di ordinamento dei risultati	47
Metodo per ripristinare un documento cancellato.....	47
Esempi pratici	48
Indicizzare un singolo documento.....	50

Indicizzazione multipla	52
Indicizzazione manuale.....	55
Utilizzare i Custom Fields.....	59
Lavorare con l'indice.....	62
Ricerca all'interno di un indice.....	64
Estendere IFile	72
Realizzare un nuovo Adapter	72
Implementazione.....	72
Riferimenti On-line	75

INTRODUZIONE A IFILE

Questo documento è rivolto agli sviluppatori che vogliono imparare ad utilizzare IFile. Le basi necessarie per utilizzare i file sono: una buona conoscenza di PHP 5 ed una minima conoscenza di XML.

In questo documento imparerai ad utilizzare IFile. Verrai introdotto ai concetti principali di IFile, descritti anche con l'aiuto di semplici esempi pratici.

OBIETTIVI

Dopo aver terminato questo documento avrai imparato a:

- Capire cos'è IFile.
- Capire perché utilizzare IFile.
- Imparare ad utilizzare IFile.
- Imparare ad estendere IFile.

ARGOMENTI TRATTATI

In questo documento, verranno trattati i seguenti argomenti:

- Conoscere IFile
- Come utilizzare IFile
- Estendere IFile

GLOSSARIO DEI TERMINI

Exif: E' l'abbreviazione di "Exchangeable image file format" ed è una specifica per il formato di file immagine utilizzato dalle fotocamere digitali. La specifica utilizza i formati esistenti JPEG, TIFF Rev. 6.0, e RIFF, con l'aggiunta di specifiche etichette (tag) di metadati.

XPDF: è un lettore PDF libero e open source per X Window System e Motif.

ANTIWORD: è un lettore documenti Microsoft Word libero e open source, ed è disponibile per la maggior parte delle piattaforme informatiche

ID3: E' l'abbreviazione di "Identify an MP3", ed è il nome di un insieme di informazioni aggiuntive inserite in un file audio di tipo MP3.

IFile Adapter: script per il recupero e l'indicizzazione dei contenuti di un file.

CMS: Content Management System - sistema per la gestione dei contenuti.

Full-Text Search: Si riferisce ad una tecnica per la ricerca di contenuti.

Lucene: API gratuita ed open source per il reperimento di informazioni inizialmente implementata in Java da Doug Cutting. È supportata dall'Apache Software Foundation ed è resa disponibile con l'Apache License. Lucene è stata successivamente reimplementata in Perl, C#, C++, Python, Ruby e PHP.

Lucene Document: Un “documento” nella nomenclatura di Lucene non è altro che un insieme di Fields (campi) da indicizzare.

MySQL: è un Relational database management system (RDBMS), composto da un client con interfaccia a caratteri e un server, entrambi disponibili sia per sistemi Unix come GNU/Linux che per Windows.

PHP: Hypertext Preprocessor - linguaggio completo di scripting, flessibile e versatile, che può girare su qualsiasi server Web e su qualsiasi sistema operativo (Windows o Unix/Linux, ma anche Mac, AS/400, Novell, OS/2 e altri); consente di interagire con tipi diversi di database (MySQL, PostgreSQL, Sql Server, Oracle, SyBase, Access e altri).

Token: un token in Lucene è una ricorrenza di un “termine” testuale.

Zend Framework: insieme di librerie scritte in PHP 5 rilasciate dalla Zend.

XML: eXtensible Markup Language - metalinguaggio di markup, ovvero un linguaggio marcatore.

XSD: XML Schema Definition - linguaggio di descrizione del contenuto di un file XML.

CONOSCERE IFILE

- Cos'è IFile
- Migliorare le proprie applicazioni con IFile
- I formati supportati

COS'È IFILE

IFile nasce dall'esigenza di avere uno strumento che permetta l'indicizzazione dei contenuti testuali di un documento (DOC, PDF ...) e una rapida ricerca all'interno degli stessi. Scritto interamente in PHP 5, IFile si basa sulla piattaforma "Zend Framework" utilizzandone le librerie "Search Lucene" e semplificandone l'utilizzo, presentando una semplice interfaccia che ne permette una maggiore velocità e facilità d'uso.

La sua struttura semplice da estendere e da configurare, permette ad i IFile di avere potenzialità che vanno oltre la sola indicizzazione dei contenuti testuali di un documento.

Di fatto si potrebbe, mediante la creazione di Adapter particolari, utilizzarlo per indicizzare altri tipi di informazioni (come ad esempio i metadati di un file MP3) per poi permetterne una ricerca veloce e mirata.

MIGLIORARE LE PROPRIE APPLICAZIONI CON IFILE

IFile è stato pensato per tutte quelle applicazioni come i CMS (Content Managment System) che contengono una grande quantità di documenti nei formati DOC, PDF, EXCEL ma per i quali non è possibile effettuare delle ricerche sui loro contenuti. Di fatto la maggior parte dei CMS presentano un sistema di ricerca nei contenuti degli articoli, salvati nella maggior parte dei casi nella banca dati del sistema, ma non permettono una ricerca sui contenuti dei documenti.

IFile risolve questo problema valorizzando di molto le vostre applicazioni.

La sua versatilità e soprattutto la possibilità di utilizzarlo su qualsiasi architettura che si basi su PHP 5 ne permette l'utilizzo in una vasta quantità di domini e applicazioni.

I FORMATI SUPPORTATI

IFile nella versione 1.1 supporta per il recupero dei contenuti e l'indicizzazione una vasta quantità di formati:

- Rich Text Format (.rtf)
- Moving Picture Expert Group-1/2 Audio Layer 3 (.mp3)
- Tagged Image File Format (.TIFF)
- Joint Photographic Experts Group (.jpg - .jpeg)

- Microsoft Word 97-2000 (.doc)
- Microsoft Word 2003-2007 (.docx)
- Microsoft Excel 97-2000 (.xls)
- Microsoft Excel 2003-2007 (.xlsx)
- Microsoft PowerPoint 2003-2007 (.pptx)
- OpenOffice.org Writer (.ods)
- OpenOffice.org Calc (.odt)
- Adobe Portable Document Format (.pdf)
- File di testo (.txt)
- Web page (.htm - .html)
- eXtensible Markup Language (.xml)

Sarà comunque possibile estendere il numero dei formati già disponibili realizzando i propri Adapter.

Di seguito una descrizione delle proprietà degli Adapter.

Rich Text Format : IFile supporta la versione 1.5 ovvero, il processo potrebbe recuperare in modo non corretto il testo presente all'interno di un documento RTF generato con una versione successiva alla 1.5.

eXtensible Markup Language : IFile recupera solo il testo presente all'interno dei TAG e non quello presente negli attributi. Inoltre non vengono indicizzate le stringhe che definiscono il TAG stesso.

COME UTILIZZARE IFILE

- Pre-Requisiti
- Installare IFile
- Configurare IFile
- Field e custom field
- Utilizzare l'interfaccia di IFile
- Esempi pratici

PRE-REQUISITI

Prima di installare IFile sarà necessario verificare se il tuo ambiente presenta tutti i requisiti necessari per poter utilizzare la libreria senza incorrere in problemi. IFile è supportato dalla maggior parte dei Sistemi Operativi e pertanto non necessita una configurazione particolare di questi.

I requisiti necessari sono:

ZEND FRAMEWORK

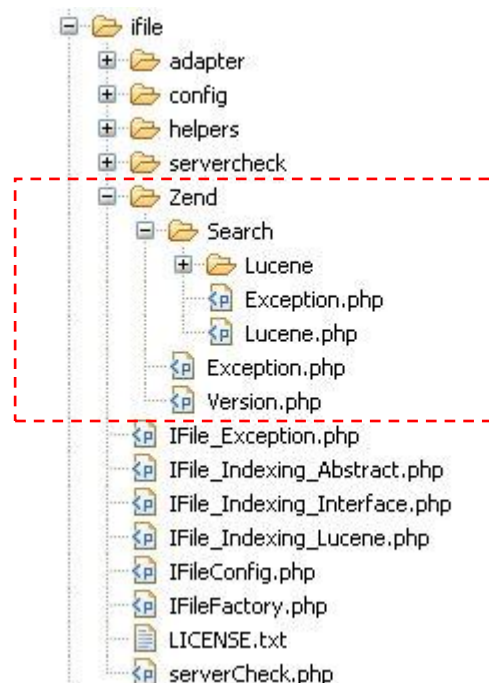
Deve essere presente ed utilizzabile la libreria di Zend: **Zend_Search_Lucene**.



IFile ad oggi non supporta Zend Framework 2.x

Le librerie di Zend devono essere installate come PEAR per poter essere utilizzate, ovvero devono trovarsi all'interno della **include_path** definita nel **php.ini**.

In caso questo non sia possibile si può includere la cartella di Zend Framework (nella figura sottostante sono presenti le cartelle ed i file necessari per il corretto funzionamento di IFile) all'interno del proprio progetto. Sarà in oltre necessario, durante l'esecuzione degli script, modificare l'**include_path**. Va ricordato che l'include path verrà ristabilito così com'è configurato nel **php.ini** al termine dello script.



Cartelle e file necessari di Zend Framework

Di seguito un esempio di come utilizzare la Zend Framework senza che questa sia presente nel PEAR di PHP.

```
<?php
/**
 * Lo script permette di utilizzare la libreria di zend_search_lucene
 * senza che questa si trovi installata sotto il PEAR di PHP
 */

$include_path = get_include_path();
// path dove e' stata copiata la libreria Zend_Search_Lucene
$include_path .= ";C:/xampp/htdocs/myproject/iframe";
set_include_path($include_path);
...
```

ENCODING

Il supporto **PCRE** per la gestione dei contenuti UTF8 deve essere abilitato. Il PCRE serve alla libreria “Zend Lucene” per la gestione e l’indicizzazione dei contenuti nell’encoding UTF8.

SERVER

Verifica se la versione di PHP utilizzata gira su 32bit o 64bit. Se PHP gira su server a 64bit, allora sarà necessario copiare l’eseguibile della XPDF a 64bit presente nella cartella

adapter/helpers/binaries/[linux/windows]/bin64/pdftotext

nella cartella

adapter/helpers/binaries/[linux/windows]

XPDF

Deve essere possibile eseguire la **XPDF**, file binario (Open Source) per il recupero dei contenuti di file PDF.



IFile permette di utilizzare la **XPDF** sia per sistemi a 32 che a 64bit solo per sistemi Windows e Linux. Per gli altri sistemi al momento è presente solo la versione a 32bit. Dalla versione di IFile 1.2 è possibile configurare eventuali XPDF installate sul server e non utilizzare quelle presenti nella libreria.

Se non si ha la necessità di indicizzare file PDF non sarà necessario che questo requisito sia presente.

ANTIWORD

Deve essere possibile eseguire **ANTIWORD**, file binario (Open Source) per il recupero dei contenuti di file DOC.



IFile, al momento, permette di utilizzare **ANTIWORD** “solo” per sistemi a 32 bit.

ANTIWORD non è esclusivamente necessario.

Di fatto IFile permette mediante configurazione, di utilizzare anche le librerie COM o eventualmente (processo di Default) utilizzare il recupero dei contenuti mediante librerie PHP.

Se non si ha la necessità di indicizzare file DOC non sarà necessario che questo requisito sia presente.



Ad oggi IFile gestisce ANTIWORD per sistemi Windows, Linux e OSX a 32 BIT.

PHP

La versione di **PHP** deve essere maggiore o uguale alla 5.1.0.

Questo perché alcune delle funzionalità utilizzate non sono supportate dalle versioni precedenti.

ESTENSIONI PHP

Devono essere installate le seguenti **estensioni** per la parserizzazione e gestione dei contenuti:

- libxml nella versione 2.6.0 o superiore
- dom
- SimpleXML
- mbstring
- zip
- zlib nella versione 1.0.9 o superiore
- iconv
- id3
- exif
- mysqli
- com_dotnet

libxml

Usato per la parserizzazione dei file in formato Office Open Xml e OpenDocument.

dom

Usato per la parserizzazione dei file in formato Office Open Xml e OpenDocument.

SimpleXml

Usato per la parserizzazione dei file in formato Office Open Xml e OpenDocument.

mbstring

Usato dalla librerie si Zend Search Lucene per la gestione dell'indicizzazione di contenuti multilingua.

zip

Usato per la parserizzazione dei file in formato Office Open Xml e OpenDocument.

ziplib

Usato per la parserizzazione dei file in formato Office Open Xml e OpenDocument.

iconv

Usato dalla librerie si Zend Search Lucene per la gestione dell'indicizzazione di contenuti multilingua.

Id3

Usato per la parserizzazione dei file in formato MP3.

Exif

Usato per la parserizzazione dei file in formato JPEG.

mysqli

Usato solo dall'interfaccia **MYSQLI** per l'indicizzazione mediante MySql

Com_dotnet

Usato solo per il recupero dei contenuti dei file DOC

*FUNZIONI PHP***popen**

Deve essere presente la funzione **popen** per l'esecuzione delle XPDF e di ANTIWOD.



Se PHP è abilitato in “**safe mode**”, la *popen* non può richiamare gli eseguibili della XPDF presenti nella cartella “*adapter/helpers/binaries/(so)*” se questo percorso non è definito nella “**safe_mode_exec_dir**” del file php.in

strip_tags

Deve essere presente la funzione **strip_tags** per la parserizzazione dei file XML.

IL “SERVER CHECK” DI IFILE

IFile mette a disposizione degli sviluppatori un meccanismo automatico di controllo per verificare che sull’ambiente su cui gira la libreria siano presenti tutti i requisiti.

Per il controllo sarà necessario richiamare lo script **serverCheck.php** presente nella root della libreria.

Lo script presenterà a video la lista dei requisiti necessari raggruppati per tipologie.

Se una delle componenti necessarie dovesse mancare o non essere aggiornata alla versione utile per il corretto funzionamento della libreria, verrà evidenziata in rosso. Per ogni componente è presente un riferimento al sito internet per il recupero di informazioni o una eventuale installazione o aggiornamento della stessa.

Component	Check	Requirements	Info	Use	Web Site
Zend Framework					
Zend Lucene	Exists	Version 1.10.1 or later	Zend Lucene is installed in C:\xampp\php\pear\ - Version 1.11.10	Used by Lucene and MySQL Interface	http://www.zend.com
Encoding					
PCRE	OK	Not defined	PCRE unicode support is enabled in PHP	Used by Zend Search Lucene Framework	http://www.php.net/manual/en/book.pcre.php
SERVER					
Server	32bit	Not defined	Note: If the OS is 64bit but running a 32 bit version of php, the check will return (32 bit)	Not defined	http://www.php.net/manual/en/install.php
XPDF					
XPDF Binaries File	Executable	CHMOD 0755	Permission XPDF Binaries File: 0777	Used only for PDF file parser	http://www.foolabs.com
PHP					
PHP Version	OK	Version 5.1.0 or later	Version installed is 5.2.4	Not defined	http://www.php.net
Extension					
libxslt	OK	Version 2.6.0 or later	Not check version	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	http://www.php.net/manual/en/book.libxslt.php
dom	OK	Not defined	Version installed is 20031129	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	http://www.php.net/manual/en/book.dom.php
SimpleXML	OK	Not defined	Version installed is 0.1	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	http://www.php.net/manual/en/book.simplexml.php
mbstring	OK	Not defined	Not check version	Used by Zend Search Lucene	http://www.php.net/manual/en/book.mbstring.php
ap	OK	Not defined	Version installed is 1.4.0	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	http://www.php.net/manual/en/class.apache.php
lib	OK	Version 1.0.9 or later	Version installed is 1.1	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	http://www.php.net/manual/en/book.lib.php
iconv	OK	Not defined	Not check version	Used by Zend Search Lucene	http://www.php.net/manual/en/book.iconv.php
id3	OK	Version 0.1 or later	Version installed is 0.1	Used for MP3 file parser	http://www.php.net/manual/en/book.id3.php
mysql	OK	Not defined	Version installed is 0.1	Used only for MySQL Interface	http://www.php.net/manual/en/book.mysql.php
exif	OK	Version 1.4 or later	Version installed is 1.4 \$Id: exif.c,v 1.173.2.5.2.20 2007/06/10 20:12:45 \$	Used for JPG file parser	http://www.php.net/manual/en/book.exif.php
Function					
popen	OK	Not defined	Function exists	Used only for PDF file parser	http://www.php.net/manual/en/function.popen.php
strip_tags	OK	Not defined	Function exists	Used only for XML file parser	http://php.net/manual/en/function.strip-tags.php

Figura 1 – Risultato della serverCheck.php

Per chi volesse integrare la presentazione dei requisiti necessari alla libreria all'interno della propria applicazione, IFile restituisce la lista degli oggetti ReportCheck.

Per istanziare l'oggetto LuceneServerCheck sarà necessario utilizzare la classe presente nel file `/servercheck/LuceneServerCheck.php`

Di seguito viene presentato un esempio di recupero della lista dei requisiti.

Con l'array degli oggetti ritornati, è possibile realizzare una propria presentazione del report di controllo sulla propria applicazione.

```
<?php
/**
 * Recupera l'array dei requisiti di IFile
 */

/** @see LuceneServerCheck */
require_once 'servercheck/LuceneServerCheck.php';
// istanzia la classe LuceneServerCheck
$serverCheck = LuceneServerCheck::getInstance();
// richiama il metodo per la verifica dei requisiti
$serverCheck->serverCheck();
// recupero dell'array degli oggetti di controllo
$reportCheck = $serverCheck->getReportCheck();
// stampa a video dei risultati
print_r($reportCheck);
```

INSTALLARE IFILE

Per installare IFile basta scaricare la libreria dal sito <http://ifile.isapp.it> che vi ridigerà sul sito di “SourceForge” dove è deploiato il codice sorgente ed i pacchetti di installazione, e decomprimere lo ZIP all'interno di una propria cartella di lavoro.

La libreria a questo punto è pronta per essere utilizzata.

CONFIGURARE IFILE

Questo è il capitolo più importante e andrebbe letto con molta attenzione per un utilizzo corretto della libreria. Una configurazione errata potrebbe provocarvi problemi nel processo di indicizzazione e ricerca dei vostri documenti.

La configurazione di IFile viene fatta mediante l'utilizzo di un file XML **IFileConfig.xml** presente nella cartella **/config** validato da un XSD **IFileConfig.xsd** che si trova nella stessa cartella.

Il file XML è così strutturato:

```
<ifile>
  <root-application>...</root-application>
  <table-name collation="..." >...</table-name>
  <timelimit>...</timelimit>
  <memorylimit>...</memorylimit>
  <duplicate>...</duplicate>
  <encoding>...</encoding>
  <doctotxt encoding="..." type="..." />
  <server bit="..." />
  <xpdf>
    <opw>...</opw>
    <pdftotext>
      <executable>...</executable>
      <xpdfrc>...</xpdfrc>
    </pdftotext>
    <pdfinfo>
      <executable>...</executable>
      <xpdfrc>...</xpdfrc>
    </pdfinfo>
  </xpdf>
  <zend-document>
    <fields>
      <field name="..." type="..." encoding="..." />
      ...
    </fields>
  </zend-document>
  <analyzer>
    <type>
      <default>...</default>
      <custom-default class="...">...</custom-default>
    </type>
    <filters>
      <stop-words>...</stop-words>
      <short-words>...</short-words>
      <custom-filters>
        <filter>...</filter>
        <filter>...</filter>
        ...
      </custom-filters>
    </filters>
  </analyzer>
</ifile>
```

Di seguito verranno descritti tutti gli elementi del file di configurazione. Ogni elemento viene definito con la seguente sintassi:

`<nomeTag> [obbligatorio][facoltativo] {occorrenze}`

Ifile [obbligatorio] {1}

Il tag `<ifile>` è il tag di root del file XML il quale all'interno presenta sei sezioni:

- root-application
- table-name
- timelimit
- memorylimit
- duplicate
- encoding
- analyzer

root-application [obbligatorio] {1}

Il tag `<root-application>` definisce il path della root dell'applicazione. Questo è molto utile da utilizzare in caso si debba spostare l'applicazione in ambienti diversi, permettendo così di avere path relativi dei file indicizzati. Il sistema verifica che il path inserito esista e sia corretto nell'ambiente in cui si sta eseguendo il processo di indicizzazione, altrimenti viene invocata un'eccezione di root inesistente.

Va ricordato che non va inserito l'ultimo separatore di delimitazione.

Esempio 1:

```
<root-application>/var/www/html/app</root-application>
```

Se si è configurata la root con il path sopra indicato ed il file si trova nella directory `/var/www/html/app/filePdf/mypdf.pdf`, IFile indicizzerà i seguenti fields (*per maggiori dettagli vedi capitolo: Fields e custom fields*):

- name: mypdf.pdf
- filemane: /var/www/html/app/filePdf/mypdf.pdf
- path: /filePdf/mypdf.pdf

Esempio 2:

```
<root-application>/var/www/html/newapp</root-application>
```

Se si è configurata la root con il path sopra indicato (diverso dal path in cui realmente si trova l'applicazione) ed il file si trova nella directory `/var/www/html/app/filePdf/mypdf.pdf`, IFile indicizzerà i seguenti field:

- name: mypdf.pdf
- filemane: /var/www/html/app/filePdf/mypdf.pdf
- path: /var/www/html/app/filePdf/mypdf.pdf

In questo caso il percorso del file (field:path) viene salvato come path assoluto e non relativo al path dell'applicazione.

table-name [opzionale] {1}

Il tag <table-name> definisce il nome della tabella se viene utilizzata l'interfaccia di Mysql per l'indicizzazione dei documenti.



Utilizzando l'interfaccia **MYSQL** non e' necessario configurare il TAG <analyzer> dato che la gestione e configurazione del processo di indicizzazione andrà effettuata direttamente su MySql.

Il sistema verifica l'esistenza della tabella, e se questa non esiste viene creata automaticamente con le caratteristiche di charset del Data Base.

Esempio:

```
<table-name>myTableInDB</table-name>
```

Il tag presenta l'attributo "collation" per la definizione della collation della Tabella. La collation definita, deve far parte della lista delle collation installate sul server MySql (SHOW COLLATION) e appartenente al charsert utilizzato, vedi tag <encoding>.

Attributo	Descrizione	Necessità
collation	Definizione della "collation" utilizzata nella tabella	Opzionale

timelimit [opzionale] {1}

Il tag <tilmelimit> definisce il tempo massimo di esecuzione del processo di indicizzazione del singolo documento.

Il tag accetta solo valori interi positivi.

Se non valorizzato, il time-limit è quello configurato nel php.ini.

Il valore minimo inseribile è di 180 secondi.

Esempio:

```
<timelimit>360</timelimit>
```



La modifica del “Timelimit” viene eseguita a runtime mediante la chiamata alla funzione di PHP “**set_time_limit**”. Pertanto se si vuole modificare questo valore va verificato che il proprio sistema (o servizio Hosting) permetta la modifica di questi valori mediante la funzione sopra descritta..

memorylimit [opzionale] {1}

Il tag `<memorylimit>` definisce la memoria massima che lo script può allocare durante l'esecuzione del processo di indicizzazione del singolo documento.

Il tag accetta solo valori interi positivi.

Se non valorizzato, il memory-limit è quello configurato nel `php.ini`.

Esempio:

```
<memorylimit>128</memorylimit>
```



La modifica della “memory limit” viene eseguita a runtime mediante la chiamata alla funzione di PHP “**ini_set**”. Pertanto se si vuole modificare questo valore va verificato che il proprio sistema (o servizio Hosting) permetta la modifica di questi valori mediante la funzione sopra descritta.

duplicate [opzionale] {1}

Il tag `<duplicate>` definisce la possibilità di avere documenti duplicati all'interno dell'indice.

Se settato a zero (0) o non presente, il sistema verifica che il contenuto del documento da indicizzare non sia già presente nell'indice. Se presente invoca una eccezione.

Altrimenti se settato a uno (1) il sistema indicizza tutti i documenti senza verificare l'esistenza del documento all'interno dell'indice.

La verifica viene fatta sul “**solo**” contenuto del documento.

Parametri permessi:

- 0
- 1

Esempio:

```
<duplicate>1</duplicate>
```

encoding [opzionale] {1}

Il tag `<encoding>` definisce il tipo di “charset encoding” in cui è stato scritto il documento. Di fatto la combinazione di questo tag con il tipo di Analyzer (descritto più avanti) definiscono il processo di conversione dei dati per una corretta indicizzazione. Ad oggi il “charset encoding” definito in fase di configurazione viene utilizzato non solo per l’indicizzazione del contenuto del documento ma per tutti i campi da indicizzare.

Se non valorizzato il sistema cerca di recuperare autonomamente il tipo di encoding di ogni campo.

I tipi di charset configurabili sono:

- UTF-8
- ASCII
- ISO-8859-1
- ISO-8859-15
- ISO-8859-2
- ISO-8859-7
- CP1256
- Windows-1252
- KOI8-R

E’ possibile estendere i charset integrandoli nel file XSD, nella parte dedicata alle tipologie di charset supportati:

```
<!-- encoding::encoding-type -->
```

Il tipo di encoding è mappato nel seguente modo per l’interfaccia MySQLi:

ascii

- ASCII

latin1

- ISO-8859-1
- ISO-8859-15
- Windows-1252

latin2

- ISO-8859-2

latin7

- ISO-8859-7

cp1256

- CP1256

utf8

- UTF-8



Un tipo di encoding errato potrebbe provocare una indicizzazione incompleta o non corretta dei campi da indicizzare.

Se si utilizza l'interfaccia “**LUCENE**”, va fatta molta attenzione al tipo di encoding che viene settato in funzione del tipo di Analyzer utilizzato.

Per le esperienze avute, andrebbe sempre configurato il TAG in funzione del tipo di encoding con il quale il documento è stato scritto.

Nel caso si utilizzi un Analyzer di tipo UTF-8 e si hanno file di cui non si conosce il tipo di charset, si potrebbe anche non configurare l'encoding (lasciare il TAG vuoto) dato che le librerie di Zend Lucene tenterebbero di forzare la codifica, ma questo potrebbe provocare, soprattutto per caratteri speciali (con un numero di byte maggiore di uno), una alterazione dei caratteri.

In altri casi il processo di tokenizzazione dei termini potrebbe non andare a buon fine e ritornare un insieme vuoto, pertanto se il sistema presenta una eccezione “**Corpo del documento vuoto**” vuol dire che il contenuto del documento non è stato tokenizzato e quindi si dovrebbe verificare l'encoding utilizzato.



La chiamata alla funzione “**iconv**” per la trasformazione del testo nel corretto encoding, da parte degli analyzer della libreria ZEND, potrebbe provocare un **NOTICE** di carattere illegale.

Questo tronca il contenuto al carattere illegale e pertanto il documento viene indicizzato in modo parziale.

Di seguito un link al forum di discussione:

<http://www.isapp.it/it/forum/supporto-ifile/15-field-qintrotextq-e-indicizzazioni-troncate.html>

Esempio:

```
<encoding>UTF-8</encoding>
```

doctotxt [opzionale] {1}

Il tag <doctotxt> è un tag vuoto e definisce il tipo, e l'encoding per il recupero del contenuto da file Microsoft Word.

Se non valorizzato il sistema definisce automaticamente dei parametri di default.

- type = PHP
- encoding = ""

Attributo	Descrizione	Necessità
type	Tipo di parser: - ANTIWORD - COM - PHP Non è possibile definire altri tipi.	Obbligatorio
encoding	Il tipo di encoding da utilizzare per il contenuto del documento è utilizzato solo per il type "ANTIWORD" ed equivale al nome senza estensione dei file .txt presenti nella cartella: adapter/helpers/binaries/resources/	opzionale

Se utilizzato il Type "ANTIWORD", il sistema cercherà di leggere i contenuti dei documenti Microsoft Word, mediante l'eseguibile ANTIWORD.

Esempio:

```
<doctotxt encoding="8859-1" type="ANTIWORD" />
```

server [opzionale] {1}

Il tag <server> è un tag vuoto e definisce il tipo di server su cui gira IFile.

Se non valorizzato il sistema definisce automaticamente dei parametri di default.

- bit = 32

Attributo	Descrizione	Necessità
bit	Tipo di server: - 32 - 64 Non è possibile definire altri tipi.	Opzionale

Esempio:

```
<server bit="64" />
```

xpdf [opzionale] {1}

Il tag <xpdf> permette la configurazione delle proprietà della XPDF

Il tag contiene una sezione:

- opw
- pdftotext
- pdfinfo

opw [opzionale] {1}

Il tag `<opw>` è il contenitore della “Password Utente” per leggere i contenuti di documenti PDF “Protetti”.

pdftotext [opzionale] {1}

Il tag `<pdftotext>` permette di definire un percorso diverso da quello impostato in IFile per l'utilizzo della “*pdftotext*” o del file di configurazione “*xpdfrc*”. Questo tag permette di utilizzare un eseguibile della “*pdftotext*” (o un file di configurazione della XPDF) diverso da quello presente in IFile. Questo a volte è necessario dato che alcuni server non riescono ad utilizzare la “*pdftotext*” presente in IFile e pertanto è necessario utilizzare quella installata sul server stesso.

Il tag contiene una sezione:

- executable
- xpdfrc

executable [opzionale] {1}

Il tag `<executable>` definisce il path “*assoluto*” dove è presente la “*pdftotext*” compreso dell'eseguibile stesso.

Esempio:

```
<executable>/var/user/xpdf/pdftotext</executable>
```

xpdfrc [opzionale] {1}

Il tag `<xpdfrc>` definisce il path “*assoluto*” dove è presente la “*xpdfrc*” compreso del file stesso.

Esempio:

```
<executable>/var/user/xpdf/config/xpdfrc</executable>
```

pdfinfo [opzionale] {1}

Il tag `<pdfinfo>` permette di definire un percorso diverso da quello impostato in IFile per l'utilizzo della “*pdfinfo*” o del file di configurazione “*xpdfrc*”. Questo tag permette di utilizzare un eseguibile della “*pdfinfo*” (o un file di configurazione della XPDF) diverso da quello presente in IFile. Questo a volte è necessario dato che alcuni server non riescono ad utilizzare la “*pdfinfo*” presente in IFile e pertanto è necessario utilizzare quella installata sul server stesso.

Il tag contiene una sezione:

- executable
- xpdfrc

executable [opzionale] {1}

Il tag `<executable>` definisce il path “*assoluto*” dove è presente la “pdinfo” compreso dell’eseguiibile stesso.

Esempio:

```
<executable>/var/user/xpdf/pdftotext</executable>
```

xpdfrc [opzionale] {1}

Il tag `<xpdfrc>` definisce il path “*assoluto*” dove è presente la “xpdfrc” compreso del file stesso.

Esempio:

```
<executable>/var/user/xpdf/config/xpdfrc</executable>
```

zend-document [opzionale] {1}

Il tag `<zend-document>` è il tag permette una configurazione manuale della Zend_Search_Lucene_Document .

Se non valorizzato il sistema definisce automaticamente dei parametri di default.

Il tag contiene due sezioni:

- fields

fields [opzionale] {1}

Il tag `<fields>` è il contenitore per la configurazione dei Field (campi) utilizzati da IFile, per l’indicizzazione (vedi capitolo: [Fields e custom fields](#)).

Il tag contiene una sezione:

- field

field [obbligatorio] {n}

Il tag `<field>` è un tag vuoto e definisce il tipo, il nome e l’encoding del field.

Attributo	Descrizione	Necessità
name	Nome del field (campo): <ul style="list-style-type: none">- name- extensionfile- path- filename- introtext- body	Obbligatorio

	<ul style="list-style-type: none"> - title - subject - description - creator - keywords - created - modified - pages <p>Non è possibile definire altri nomi.</p>	
type	Definizione del tipo di indicizzazione: <ul style="list-style-type: none"> - Keyword - UnIndexed - Binary - Text - UnStored 	Obbligatorio
encoding	Il tipo di encoding da utilizzare per il contenuto del field. <ul style="list-style-type: none"> - UTF-8 - ASCII - ISO8859-1 - ISO8859-15 - CP1256 - Windows-1252 	opzionale

Per maggiori dettagli sui tipi di indicizzazione si demanda al sito della Zend Framework:

<http://framework.zend.com/manual/en/zend.search.lucene.overview.html>

analyzer [opzionale] {1}

Il tag <analyzer> è il tag per la gestione del tipo di analizzatore del testo da indicizzare.



Il tipo di analyzer serve per la gestione dei documenti sia in fase di indicizzazione che in fase di ricerca solo se viene utilizzata l'interfaccia **LUCENE**. Il TAG risulta inutile se si utilizza una interfaccia diversa per l'indicizzazione, peranto si può anche non configurarlo. Se si configura un tipo di analyzer per indicizzare un documento e se ne configura un altro per la ricerca, i risultati potrebbero essere diversi da quelli attesi. Di fatto ***un analyzer dovrebbe essere sempre lo stesso per l'indice.***

Se non valorizzato il sistema definisce automaticamente come analyzer:

Utf8_CaseInsensitive (vedi tag type/default)

Il tag contiene due sezioni:

- type
- filters

type [opzionale] {1}

Il tag <type> definisce il tipo di analizzatore del testo da indicizzare.

Se non valorizzato il sistema definisce automaticamente come analyzer:

Utf8_CaseInsensitive (vedi tag default)

Il tag contiene due sezioni:

- default
- custom-default

Le due sezioni sono alternative ed esclusive, ovvero una esclude l'altra.

default [a scelta] {1}

Il tag <default> definisce il tipo di analizzatore del testo da indicizzare gestiti dalla libreria Zend_Search_lucene.

I tipi di analizzatori sono:

- Text
- TextNum
- Text_CaseInsensitive
- TextNum_CaseInsensitive
- Utf8
- Utf8Num

- Utf8_CaseInsensitive
- Utf8Num_CaseInsensitive

Per maggiori dettagli sul tipo di analyzer più appropriato alle proprie esigenze si demanda al sito della Zend Framework:

<http://framework.zend.com/manual/en/zend.search.lucene.extending.html>

Il tag è alternativo con il tag custom-default.

Esempio:

```
<default>Utf8Num</default>
```

custom-default [a scelta] {1}

Il tag `<custom-default>` definisce un tipo di analizzatore del testo personalizzato.

Ovvero è possibile definire delle proprie classi per analizzare e tokenizzare il testo. Il tag necessita del path assoluto ove risiede lo script PHP contenente la classe definita nell'attributo "class".

Attributo	Descrizione	Necessità
Class	Nome della classe da richiamare che estende la Zend_Search_Lucene_Analysis_Analyzer	Obbligatorio

Il tag è alternativo con il tag default.

Un esempio di analizzatore personalizzato è presente al sito:

<http://code.google.com/p/lucene-silverstripe-plugin/>

Esempio:

```
<custom-default  
class="StandardAnalyzer_Analyzer_Standard_English">  
/var/www/html/app/analyzer/StandardAnalyzer/Analyzer/Standard  
/English.php</custom-default>
```

filters [opzionale] {1}

Il tag `<filters>` è il tag per la gestione del processo di tokenizzazione definendo i filtri sui termini tokenizzati.



I filtri servono per la gestione dei documenti sia in fase di indicizzazione che in fase di ricerca. Se si configurano dei filtri per indicizzare un documento e se ne configurano altri per la ricerca, i risultati potrebbero essere diversi da quelli attesi. Di fatto i filtri dovrebbero essere sempre gli stessi per l'indice.

E' possibile non definire nessun tipo di filtro.

Il tag contiene tre sezioni:

- stop-words
- short-words
- custom-fields

stop-words [opzionale] {1}

Il tag `<stop-words>` definisce il file per l'eliminazione delle parole (termini tokenizzati) all'interno del testo da indicizzare.

Il tag richiede il path assoluto del file delle stop-words.

Il file deve contenere tutte le parole separate da un ritorno a capo.

Esempio:

```
<stop-words>/var/www/html/app/mystopwords.txt</stop-words>
```

short-words [opzionale] {1}

Il tag `<short-words>` definisce il limite minimo di caratteri del singolo termine. Pertanto settando un limite minimo di 3 caratteri per termine, il sistema non indicizzerà e non considererà nei processi di ricerca tutte quelle parole di due caratteri (come ad esempio: io, tu, li, il, ...).

Il sistema permette di definire un limite minimo che va da 2 (due) a 4 (quattro) caratteri.

E' possibile estendere il limite andando a definire nuovi limiti minimi nel file XSD.

Esempio:

```
<short-words>3</short-words>
```

custom-fields [opzionale] {1}

Il tag `<custom-filter>` definisce la gestione di eventuali filtri personalizzati.

E' permesso integrare più di un filtro.

Il tag contiene una sezione:

- filter

filter [opzionale] {n}

Il tag `<filter>` permette di configurare eventuali filtri sui termini.

Ovvero è possibile definire delle proprie classi per tokenizzare il testo. Il tag necessita del path assoluto ove risiede lo script PHP contenente la classe definita nell'attributo "class".

Attributo	Descrizione	Necessità
class	Nome della classe da richiamare che estende la Zend_Search_Lucene_Analysis_TokenFilter	Obbligatorio

Un esempio di analizzatore personalizzato è presente al sito:

<http://code.google.com/p/lucene-silverstripe-plugin/>

Esempio:

```
<filter  
class="StandardAnalyzer_Analysis_TokenFilter_EnglishStemmer">  
/var/www/html/app/analyzer/StandardAnalyzer/Analyzer/TokenFil  
ter/EnglishStemmer.php  
</custom-default>
```

LA XPDFRC

Dalla versione 1.1.4 di IFile è possibile configurare la xpdfrc

<http://linux.die.net/man/5/xpdfrc>

Il file, nominato "xpdfrc", non deve presentare nessun tipo di estensione e deve essere inserito nella cartella:

/ifile/adapters/helpers/binaries/xpdfrc

FIELDS E CUSTOM FIELDS

IFile indicizza di default i seguenti field:

DEFAULT FIELDS

Nome	Descrizione	Tipo	Necessità
root	Percorso assoluto dell'applicazione configurato nel file XML	UnIndexed	Obbligatorio
name	Nome del file compreso di estensione	Binary	Obbligatorio <i>solo se viene indicizzato un file mediante il</i>

			<i>processo automatico di IFile</i>
serchablename	Nome del file. Questo field è stato integrato per permettere una ricerca anche sul nome del file	UnStored	Obbligatorio solo se viene indicizzato un file mediante il processo automatico di IFile
key	Chiave (MD5 del contenuto del file) che identifica univocamente il file. Se si utilizza la configurazione di duplicazione dei contenuti (vedi configurazione TAG duplicate) allora si potrebbero avere più chiavi con lo stesso valore	Keyword	Obbligatorio solo se viene indicizzato un file mediante il processo automatico di IFile
path	Percorso relativo o assoluto in funzione della configurazione della root	Binary	Obbligatorio solo se viene indicizzato un file mediante il processo automatico di IFile
filename	Percorso assoluto del file indicizzato	Binary	Obbligatorio solo se viene indicizzato un file mediante il processo automatico di IFile
extensionfile	Estensione del file indicizzato	Keyword	Obbligatorio solo se viene indicizzato un file mediante il processo automatico di IFile
body	Contenuto del documento da	UnStored	Obbligatorio

	indicizzare		
introtext	Contiene i primi caratteri (circa 200) del corpo del contenuto	UnIndexed	Obbligatorio
title	Titolo del documento (recuperato dal documento se esiste)	Text	Facoltativo
subject	Oggetto del documento (recuperato dal documento se esiste)	Text	Facoltativo
description	Descrizione del documento (recuperato dal documento se esiste)	Text	Facoltativo
creator	Autore del documento (recuperato dal documento se esiste)	Text	Facoltativo
keywords	Parole chiave associate al documento (recuperato dal documento se esiste)	Keyword	Facoltativo
created	Data di creazione del documento (recuperato dal documento se esiste)	UnIndexed	Facoltativo
modified	Data di ultima modifica del documento (recuperato dal documento se esiste)	UnIndexed	Facoltativo
pages	Numero di pagine del documento	UnStored	Facoltativo

TAG ID3 DEGLI MP3

Se si indicizzano anche i file MP3, IFile aggiunge ai “default field” sopra descritti i seguenti field.

Per i file MP3 IFile compone il field “body” concatenando tutti i TAG ID3 recuperati del file con il carattere “spazio”.

Nome	Descrizione	Tipo	Necessità
encodedBy	Codificato da	Text	Facoltativo
track	Traccia	Text	Facoltativo
publisher	Pubblicato	Text	Facoltativo
partOfASet	Disco	Text	Facoltativo
bpm	Battiti al minuto	Text	Facoltativo
originalArtist	Artista originale	Text	Facoltativo
copyright	Diritto di copyright	Text	Facoltativo
band	Gruppo	Text	Facoltativo
genre	Genere	Text	Facoltativo

composer	Compositore	Text	Facoltativo
year	Anno	Text	Facoltativo
title	Titolo del brano	Text	Facoltativo
album	Album	Text	Facoltativo
artist	Cantante	Text	Facoltativo

TAG EXIF DEI JPEG

Se si indicizzano i file JPG o JPEG, IFile aggiunge ai “default field” sopra descritti i seguenti field.

Per i file JPG o JPEG IFile compone il field “body” concatenando i TAG evidenziati nella Tabella.

Legenda:

B: I campi contrassegnati sono utilizzati per creare il corpo (body) del documento per i file MP3.

Nome	Descrizione	Tipo	Necessità	B
FileSize	Dimensione in byte del file	Keyword	Facoltativo	
Height	Altezza in pixel dell'immagine	Keyword	Facoltativo	X
Width	Larghezza in pixel dell'immagine	Keyword	Facoltativo	X
IsColor	Flag che identifica se l'immagini e' a colori: <ul style="list-style-type: none"> • 0 -> no • 1 -> si 	Keyword	Facoltativo	
ApertureFNumber	Obiettivo	Keyword	Facoltativo	
UserComment	Commento dell'utente	Text	Facoltativo	X
ImageDescription	Descrizione dell'immagine	Text	Facoltativo	X
Orientation	Orientamento dell'immagine: <ul style="list-style-type: none"> • Left • Right • Top • Bottom • reserved 	Keyword	Facoltativo	
Make	Macchina utilizzata	Keyword	Facoltativo	X
Model	Modello della macchina	Keyword	Facoltativo	X

Software	Software installato o utilizzato per la generazione dell'immagine	Keyword	Facoltativo	X
Copyright	Copyright dell'immagine	UnIndexed	Facoltativo	
GPSLatitude	Informazioni della latitudine nel formato gradi/primi/secoli	Keyword	Facoltativo	
GPSLatitudeGoogle	Latitudine nel formato di Google Map	UnIndexed	Facoltativo	
GPSLongitude	Informazioni della longitudine nel formato gradi/primi/secoli	Keyword	Facoltativo	
GPSLongitudeGoogle	Longitudine nel formato di Google Map	UnIndexed	Facoltativo	
XResolution	Risoluzione X	Keyword	Facoltativo	
YResolution	Risoluzione Y	Keyword	Facoltativo	
DateTime	Data di creazione dell'immagine	Keyword	Facoltativo	
ExposureMode	Modalit' di esposizione <ul style="list-style-type: none"> • Auto exposure • Manual exposure • Auto bracket • None • reserved 	Keyword	Facoltativo	
ExposureTime	Tempo di esposizione	Keyword	Facoltativo	
SceneCaptureType	Tipo di scena <ul style="list-style-type: none"> • Standard • Landscape • Portrait • Night scene • reserved 	Keyword	Facoltativo	
LightSource	Sorgente di luce	Keyword	Facoltativo	

IFile permette anche di definire, oltre ai field sopra elencati, eventuali altri field personalizzati definendone il valore ed il tipo a loro associati.



Se un campo personalizzato (custom field) viene chiamato con uno dei nomi dei campi creati di default da IFile (default field) il valore del field creato automaticamente da IFile verrà sovrascritto da quello "custom".

Per un maggiore dettaglio sul tipo di field si può fare riferimento al seguente link:

<http://framework.zend.com/manual/en/zend.search.lucene.overview.html#zend.search.lucene.index-creation.understanding-field-types>

UTILIZZARE LE INTERFACCE DI IFILE

In questo capitolo verranno presentate le interfacce messe a disposizione dalla libreria IFile definendone punti di forza e punti di debolezza, per la gestione del processo di indicizzazione e ricerca dei documenti all'interno dell'indice.



Tutte le "interfacce" di IFile implementano l'interfaccia **IFile_Indexing_Interface** pertanto, a meno di piccolissimi cambiamenti, sarà molto semplice riutilizzare il codice già creato per una "interfaccia" per un'altra.

LUCENE INTERFACE

L'interfaccia si basa su Lucene e sulle librerie di Zend_Search_Lucene di Zend Framework.

Punti di forza:

- Possibilità di definire tipi diversi per l'indicizzazione di Filed e pertanto limitare la dimensione dell'indice.
- Possibilità di effettuare qualsiasi tipo di ricerca all'interno dell'indice.
- E' possibile configurare e implementare ogni tipo di analyzer a seconda delle proprie esigenze.
- Numero di filtri illimitato per il filtraggio dei token da indicizzare.

Punti di debolezza:

- Le dimensioni dell'indice gestito da Zend Search Lucene non può superare i 2GB di dimensione.
- Tempi di indicizzazione dei contenuti molto lenti in funzione del tipo di Analyzer e dalla quantità di filtri utilizzati.
- Tempi di risposta durante le ricerche molto lenti in funzione del tipo di Analyzer e dalla quantità di filtri utilizzati.

MYSQLI INTERFACCIA

L'interfaccia si basa sulla Full-Text di MySQL per engine MyISAM utilizzando l'estensione di mysqli di PHP.

A differenza dell'interfaccia **LUCENE**, la configurazione del processo di indicizzazione va fatta direttamente sul server MySQL.



Gli indici per l'interfaccia **MYSQLI**, a differenza dell'interfaccia **LUCENE** che partono dal numero zero (0), partono dal numero uno (1).

Ad esempio se si vuole modificare il limite minimo di caratteri che una parola (token) deve contenere per essere indicizzata (**short-words**), si dovrà agire sul file di configurazione di MySQL (my.cnf) ed aggiungere:

```
[mysqld]
ft_min_word_len = 3
```

ft boolean syntax	+ -><{}~*: " " &
ft max word len	84
ft min word len	3
ft query expansion limit	20
ft stopword file	{built-in}

Variabili server

Per maggiori informazioni visita:

<http://dev.mysql.com/doc/refman/5.5/en/fulltext-search.html>

Punti di forza:

- I tempi di indicizzazione dei contenuti ridotti.
- Tempi di risposta durante le chiamate di ricerca sono più rapidi.

Punti di debolezza:

- Tutti i field vengono creati come campi MyISAM all'interno del DB e definiti come campi Full-Text.

- Le dimensioni del DB possono crescere esponenzialmente dato che tutti i contenuti sono salvati.
- Alcuni tipi di ricerche non sono possibili (esempio RANGE) perché non supportate dall'engine.
- L'analyzer è unico e gestito dall'engine di MySql
- Numero di filtri limitati a quelli configurati su MySql

IFILE_INDEXING_INTERFACE

Di seguito verrà descritta l'interfaccia di IFile. Tutti i metodi si basano sull'interfaccia **LUCENE**.

La maggior parte dei metodi non dipendono dall'interfaccia. In caso un metodo abbia comportamenti diversi verrà evidenziato in riferimento all'interfaccia.

METODO PER AGGIUNGERE UN CAMPO PERSONALIZZATO

Permette di aggiungere uno o più campi personalizzati all'indicizzazione del documento.

Metodo:

void addCustomField (string \$field, string \$term, string \$type)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$field	Nome del campo	String	
\$term	Valore del campo	String	
\$type	Tipo di indicizzazione. I tipi possibili sono: <ul style="list-style-type: none">• FIELD_TYPE_KEYWORD• FIELD_TYPE_UNINDEXED• FIELD_TYPE_BINARY• FIELD_TYPE_TEXT• FIELD_TYPE_UNSTORED I tipi sono definiti come costanti della classe IFile_Indexing_Lucene	String	



Per l'interfaccia **MYSQLI** si sono definiti i type in questo modo:

FIELD_TYPE_KEYWORD:

FIELD_TYPE_TEXT:

FIELD_TYPE_UNSTORED:

Creazione di un campo di tipo "**TEXT**" indicizzato (**FULLTEXT**)

FIELD_TYPE_UNINDEXED:

Creazione di un campo di tipo "**TEXT**" NON indicizzato

FIELD_TYPE_BINARY:

Creazione di un campo di tipo "**BLOB**" NON indicizzato

Per maggiori dettagli sull'aggiunta di campi personalizzati, vedi "Esempi pratici: Utilizzare i custom fields".

METODO PER AGGIUNGERE UN DOCUMENTO ALL'INDICE

Permette di aggiungere un documento all'indice.

Metodo:

void addDocument ([\$doc = null])

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$doc	Parametro facoltativo. Il parametro dovrà essere passato se si vuole effettuare una indicizzazione manuale del documento.	Zend_Search_Lucene_Document	NULL

Per maggiori dettagli sull'aggiunta di un documento, vedi "Esempi pratici: Indicizzazione manuale - Indicizzare un singolo documento - Indicizzazione multipla".

METODO PER DEFINIRE L'AUTOCOMMIT DEI DOCUMENTI

Configura commit automatico dei documenti nel processo di indicizzazione.

Metodo:

void autoCommit (boolean \$autocommit)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$autocommit	Se true il sistema commetterà automaticamente ad ogni indicizzazione.	Boolean	



Per l'interfaccia **MYSQLI** l'auto-commit non è necessario dato che per letabelle MyISAM non esistono transazioni.

METODO PER COMMITARE I DOCUMENTI NELL'INDICE

Committa i documenti nel processo di indicizzazione .

Metodo:

void commit ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			



Per l'interfaccia **MYSQLI** il commit è automatico dato che per letabelle MyISAM non esistono transazioni.

METODO PER RECUPERARE DEL NUMERO DI DOCUMENTI INDICIZZATI

Ritorna il numero di documenti inseriti compresi quelli marcati come cancellati.

Metodo:

integer count ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

Per maggiori dettagli vedi "Esempi pratici: Lavorare con l'indice".

METODO PER CANCELLARE UN DOCUMENTO DALL'INDICE

Permette di cancellare un documento dall'indice.



Il metodo "delete" in realtà marca il documento come cancellato ma non lo cancella fisicamente. L'eliminazione fisica del documento viene fatta nel momento in cui l'indice viene ottimizzato.

Metodo:

void delete (integer \$id)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$id	ID del documento all'interno dell'indice	Integer	

Per maggiori dettagli vedi "Esempi pratici: Lavorare con l'indice".

METODO PER CANCELLARE TUTTI I DOCUMENTO DALL'INDICE

Cancella l'indice e ritorna il numero di documenti cancellati.

Se viene passato TRUE cancella solo tutti i documenti dall'indice e ritorna il numero di documenti cancellati altrimenti elimina completamente l'indice.

Metodo:

integer deleteAll ([bool \$doc = false])

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$doc	Se settato a TRUE cancella solo i documenti	boolean	False

METODO PER RECUPERARE TUTTI I DOCUMENTO DALL'INDICE

Ritorna un array contenente gli oggetti documento

Zend_Search_Lucene_Document di tutti i documenti indicizzati. Di default sono esclusi tutti i documenti segnati come cancellati. E' possibile recuperare anche solo una porzione dell'indice utilizzando i parametri offset e maxrow.

Metodo:

mixed getAllDocument ([boolean \$deleted = false], [integer \$offset = null], [integer \$maxrow = null])

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$deleted	Se settato a TRUE ritorna anche i documenti marcati come cancellati	boolean	False
\$offset	Definisce il valore di partenza per l'offset	Integer	null
\$maxrow	Numero di documenti che il metodo deve ritornare	Integer	null

Per maggiori dettagli vedi "Esempi pratici: Ricercare all'interno di un indice".

METODO PER RECUPERARE DEI CAMPI PERSONALIZZATI

Ritorna l'array dei campi personalizzati che si vuole aggiungere al documento.

Metodo:

array getCustomField ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

Per maggiori dettagli vedi "Esempi pratici: Utilizzare i custom field".

METODO PER RECUPERARE IL CAMPO DI DEFAULT DI RICERCA

Ritorna il campo (field) di default su cui ricercare i termini.

Metodo:

string getDefaultSearchField ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

Per maggiori dettagli vedi "Esempi pratici: Ricercare all'interno di un indice".

METODO PER RECUPERARE UN DOCUMENTO DALL'INDICE

Ritorna l'oggetto documento Zend_Search_Lucene_Document.

Ritorna un'eccezione Zend_Search_Lucene_Exception se l'id passato non è presente nel range degli id dell'indice.

Metodo:

Zend_Search_Lucene_Document getDocument (integer \$id)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$id	ID del documento all'interno dell'indice	integer	

Per maggiori dettagli vedi "Esempi pratici: Ricercare all'interno di un indice".

METODO PER RECUPERARE I FIELDS DELL'INDICE

Ritorna un array contenente tutti i fields (campi) definiti all'interno dell'indice.

Metodo:

array getFieldNames ([boolean \$indexed = false])

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$indexed	Se settato a TRUE Filtra l'array ritornando solo i fields (campi) che sono utilizzati come indici	Boolean	False

Per maggiori dettagli vedi "Esempi pratici: Lavorare con l'indice".

METODO PER RECUPERARE LE INFO SUL FILE DA INDICIZZARE

Ritorna un oggetto IFileInfoFile, contenente le informazioni sul file che si sta cercando di indicizzare in quel momento e che si è settato con il metodo setIndexFile().

Metodo:

IFileInfoFile getIndexFile ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

METODO PER RECUPERARE IL LIMITE MASSIMO DI RISULTATI

Ritorna il numero limite di risultati che possono ritornare da una ricerca.

Metodo:

integer getResultLimit ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

Per maggiori dettagli vedi “Esempi pratici: Lavorare con l’indice”.

METODO PER VERIFICARE SE ESISTONO DOCUMENTI CANCELLATI

Ritorna true se all’interno dell’indice esistono documenti cancellati.

Metodo:

boolean hasDeletions ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

Per maggiori dettagli vedi “Esempi pratici: Lavorare con l’indice”.

METODO PER VERIFICARE L’ESISTENZA DI UN TERMINE

Verifica se un determinato termine è presente all’interno dell’indice. Se settato il filed (campo) la verifica verrà fatta solo all’interno del field specificato.

Metodo:

boolean hasTerm (string \$term, [string \$field = null])

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$term	Termine (parola o token) da ricercare all’interno dell’indice	string	
\$field	Field (campo) in cui ricercare il termine. Se non settato il termine viene ricercato in tutto l’indice.	string	null

Per maggiori dettagli vedi “Esempi pratici: Lavorare con l’indice”.

METODO PER VERIFICARE SE UN DOCUMENTO E' CANCELLATO

Verifica se un documento è cancellato all'interno dell'indice, ovvero è stato marcato come cancellato con il metodo delete(\$id)..

Metodo:

boolean isDeleted (\$id)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$id	ID del documento all'interno dell'indice.	integer	

Per maggiori dettagli vedi "Esempi pratici: Lavorare con l'indice".

METODO PER RECUPERARE IL NUMERO DI DOCUMENTI

Ritorna il numero di documenti all'interno dell'indice senza considerare quelli marcati come cancellati.

Metodo:

integer numDocs ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

Per maggiori dettagli vedi "Esempi pratici: Lavorare con l'indice".

METODO PER OTTIMIZZARE L'INDICE

Permette di ottimizzare l'indice andando a cancellare definitivamente tutti i documenti marcati come cancellati e ricostruendo l'indice per migliorarne le performance.

Metodo:

void optimize ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

Per maggiori dettagli vedi "Esempi pratici: Lavorare con l'indice".

METODO PER LA RICERCA DEI TERMINI

Esegue la query di ricerca dei termini all'interno dell'indice. Il metodo necessita come parametro di un oggetto `IFileQueryRegistry`.
Ritorna un array di oggetti `Zend_Search_Lucene_Search_QueryHit` o un array vuoto in caso la query non presenta match dei termini ricercati all'interno dell'indice.

Metodo:

mixed query (IFileQueryRegistry \$query)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$query	Oggetto contenente i termini di ricerca	IFileQueryRegistry	

Per maggiori dettagli vedi "Esempi pratici: Ricercare all'interno di un indice".

METODO PER LA RICERCA BOOLEAN DEI TERMINI

Esegue la query di ricerca dei termini all'interno dell'indice potendoli ricercare in OR o AND. Il metodo necessita come parametro di un oggetto `IFileQueryRegistry` il quale deve a sua volta contenere oggetti `IFileQueryRegistry`.
Ritorna un array di oggetti `Zend_Search_Lucene_Search_QueryHit` o un array vuoto in caso la query non presenta match dei termini ricercati all'interno dell'indice.

Metodo:

mixed queryBoolean (IFileQueryRegistry \$query)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$query	Oggetto contenente i termini di ricerca	IFileQueryRegistry	

Per maggiori dettagli vedi "Esempi pratici: Ricercare all'interno di un indice".

METODO PER LA RICERCA FUZZY DEI TERMINI

Esegue la query di ricerca dei termini all'interno dell'indice potendoli ricercare con la metodologia fuzzy. E' possibile ricercare un solo termine nella ricerca fuzzy.
Ritorna un array di oggetti `Zend_Search_Lucene_Search_QueryHit` o un array vuoto in caso la query non presenta match dei termini ricercati all'interno dell'indice.

Metodo:

mixed queryFuzzy (IFileQueryRegistry \$query)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$query	Oggetto contenente i termini di ricerca	IFileQueryRegistry	

Per maggiori dettagli vedi “Esempi pratici: Ricercare all’interno di un indice”.

METODO PER LA RICERCA CON LA SINTASSI DI LUCENE /MYSQL

Esegue la query di ricerca dei termini utilizzando stringhe i ricerca. Questo metodo, per l’interfaccia **LUCENE** a tempi di risposta più lunghi rispetto agli altri metodi di ricerca e per questo andrebbe utilizzato solo se non è possibile ottenere lo stesso risultato utilizzando gli altri metodi.



In questo metodo è possibile utilizzare la sintassi di ricerca di Lucene per l’interfaccia **LUCENE**

<http://framework.zend.com/manual/en/zend.search.lucene.query-language.html>

e la sintassi di MySql per l’interfaccia **MYSQLI**

<http://dev.mysql.com/doc/refman/5.5/en/fulltext-natural-language.html> .

Metodo:

mixed queryParser (\$query)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$query	Stringa da ricercare all’interno dell’indice	string	

Per maggiori dettagli vedi “Esempi pratici: Ricercare all’interno di un indice”.

METODO PER LA RICERCA DI FRASI

Esegue la query per la ricerca di intere frasi all’interno dell’indice.

I fields (campi) devono essere gli stessi per tutti i termini settati nell’oggetto

IFileQueryRegistry, altrimenti viene generata una eccezione di tipo Zend_Search_Lucene_Exception.

Metodo:

mixed queryPhrase (IFileQueryRegistry \$query)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$query	Oggetto contenente i termini di ricerca	IFileQueryRegistry	

Per maggiori dettagli vedi “Esempi pratici: Ricercare all’interno di un indice”.

METODO PER LA RICERCA IN UN RANGE DI TERMINI

Esegue la query per la ricerca di un range di termini all’interno dell’indice.

Puo' essere ricercato solo un range di termini per lo stesso field nella ricerca (ovveto solo i termini di "From" e "To") altrimenti viene generata una eccezione di tipo IFile_Exception.

Metodo:

mixed queryRange (IFileQueryRegistry \$query)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$query	Oggetto contenente i termini di ricerca	IFileQueryRegistry	



Per l’interfaccia **MYSQLI**, il metodo invoca una eccezione.

Per maggiori dettagli vedi “Esempi pratici: Ricercare all’interno di un indice”.

METODO PER LA RICERCA CON CARATTERI JOLLY (WILDCARD)

Esegue la query per la ricerca di termini utilizzando caratteri jolly (wildcard).

Puo' essere ricercato solo un unico termine nella ricerca wildcard altrimenti viene generata una eccezione di tipo IFile_Exception.

Metodo:

mixed queryWildcard (IFileQueryRegistry \$query)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$query	Oggetto contenente i termini di ricerca	IFileQueryRegistry	

Per maggiori dettagli vedi “Esempi pratici: Ricercare all’interno di un indice”.

METODO PER SETTARE IL CAMPO DI DEFAULT DI RICERCA

Ritorna il campo (field) di default su cui ricercare i termini. Se il campo viene settato, quando si omette il nome del campo nelle query di ricerca il sistema ricercherà il termine all’interno del campo di default.

Metodo:

void setDefaultSearchField (string \$field)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$field	Field (campo) che verrà utilizzato come field di default nelle ricerche se il field è omesso	String	

Per maggiori dettagli vedi “Esempi pratici: Ricercare all’interno di un indice”.

METODO PER SETTARE IL FILE DA INDICIZZARE

Setta il file che deve essere processato per l’indicizzazione.

Metodo:

void setIndexFile (\$indexFile)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$indexFile	Percorso del file da indicizzare	string	

Per maggiori dettagli sull’indicizzazione di un documento vedi “Esempi pratici: Indicizzazione manuale - Indicizzare un singolo documento - Indicizzazione multipla”.

METODO PER SETTARE IL LIMITE MASSIMO DI RISULTATI

Setta il limite massimo di risultati che la ricerca può ritornare.

Metodo:

void setResultLimit (integer \$limit)

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$limit	Limite massimo di risultati per la ricerca	integer	

Per maggiori dettagli vedi “Esempi pratici: Ricercare all’interno di un indice”.

METODO PER SETTARE IL TIPO DI ORDINAMENTO DEI RISULTATI

Setta il tipo di ordinamento dei risultati che restituisce una query.

Metodo:

void setSort (string \$field, [string \$type = SORT_REGULAR], [string \$order = SORT_ASC])

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$field	Nome del Field (campo) su cui operare l'ordinamento	String	
\$type	Tipo di ordinamento: <ul style="list-style-type: none"> • SORT_REGULAR • SORT_NUMERIC • SORT_STRING 	Integer	SORT_REGULAR
\$order	Ordinamento ascendente o discendente: <ul style="list-style-type: none"> • SORT_ASC • SORT_DESC 	Integer	SORT_ASC

Per maggiori dettagli vedi “Esempi pratici: Ricercare all’interno di un indice”.

METODO PER RIPRISTINARE UN DOCUMENTO CANCELLATO

Il metodo non è ancora implementato dalle librerie Zend Search Lucene, pertanto non è implementato per l'interfaccia **LUCENE**.

Metodo:

void undeletedAll ()

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
Nessuno			

ESEMPI PRATICI

I meccanismi di indicizzazione possono essere molteplici, in questo capitolo verranno descritti la maggior parte dei possibili meccanismi di indicizzazione e ricerca utilizzando esempi pratici.

Tutti gli esempi sono presenti nel pacchetto “**IFile_Full_Package-1.1.5.zip**” scaricabile dal sito:

<http://sourceforge.net/projects/indexfile/>

Per la generazione degli esempi si è utilizzato XAMPP in ambiente Windows. Inoltre si è utilizzato Luke (<http://www.getopt.org/luke/>) per verificare l'indice per l'interfaccia **LUCENE** ed un qualsiasi tool per accedere a MySql per verificare i risultati dell'interfaccia **MYSQLI**.

Per gli esempi si sono processati i file nell'ordine sotto indicato realizzando così gli indici riportati nelle figure.

Configurazione:

La configurazione di IFile sotto riportata è una configurazione minimale necessaria per la sola esecuzione degli script di esempio.

Come si può vedere le configurazioni per entrambe le interfacce possono coesistere senza conflitti. In caso si utilizzi una delle interfacce si può configurare solo i tag necessari per l'interfaccia utilizzata.

In caso sia necessario riutilizzare lo stesso script per condizioni reali, andrà rivista la configurazione da parte dello sviluppatore (vedi capitolo: Configurare IFile).

```
<ifile>
  <root-application>C:\xampp\htdocs\IFile</root-application>
  <table-name>ifile_index</table-name>
  <!--timelimit--></timelimit-->
  <!--memorylimit--></memorylimit-->
  <duplicate>0</duplicate>
  <!--encoding--></encoding-->
  <!--doctotxt type="" encoding="" /-->
  <analyzer>
    <type>
      <default>Utf8_CaseInsensitive</default>
    </type>
    <!--filters>
      <stop-words></stop-words>
      <short-words></short-words>
      <custom-filters>
        <filter class=""></filter>
        <filter class=""></filter>
      </custom-filters>
```



```
</filters-->
</analyzer>
</ifile>
```

Definizione delle interfacce:

Di seguito viene descritta come vengono gestite le chiamate alle interfacce.

Questa parte di codice va integrata ad ogni codice di esempio.

LUCENE:

```
<?php

error_reporting(E_ALL);
// unico file da includere per l'utilizzo della libreria
require_once 'ifile/IFileFactory.php';

// Definizione della cartella degli indici
// E' importante ricordare che se si sta indicizzando per la prima volta
// la cartella non deve esistere. In caso la cartella esiste ed è vuota
// il sistema invocherà una eccezione
$index_path = 'ifile_index';

// Tutti gli errori vengono gestiti mediante l'invocazione di una eccezione
// pertanto e' possibile gestire gli stessi mediante l'utilizzo del try/catch
try {
    // istanza dell'oggetto IFileFactory
    $IFileFactory = IFileFactory::getInstance();
    // ritorna l'oggetto per l'indicizzazione del documento mediante Lucene
    $ifile = $IFileFactory->getIFileIndexing('lucene', $index_path);

    ...
}
```

MYSQLI:

```
<?php

error_reporting(E_ALL);
// unico file da includere per l'utilizzo della libreria
require_once 'sf_ifile/IFileFactory.php';

// creazione della connessione al DB con MySQLi
// IFile gestisce solo connessioni al DB mediante MySQLi
$connection = @new mysqli('localhost','root','', 'ifile', 3306);
// verifica che non si siano verificati errore di connessione
if (mysqli_connect_error()) {
    // blocca lo script
    die(" - ".mysqli_connect_error(). " - ");
}

// Tutti gli errori vengono gestiti mediante l'invocazione di una eccezione
// pertanto e' possibile gestire gli stessi mediante l'utilizzo del try/catch
try {
    // istanza dell'oggetto IFileFactory
    $IFileFactory = IFileFactory::getInstance();
}
```

```
// ritorna l'oggetto per l'indicizzazione del documento mediante MySQL
// E' importante ricordare che va sempre passata la connessione al DB
$file = $IFileFactory->getIFileIndexing('mysqli', $connection);
```

...

INDICIZZARE UN SINGOLO DOCUMENTO

Il codice seguente, presente nei file di esempio sotto indicati, descrive il processo per l'indicizzazione di un singolo documento.

Questo permette di indicizzare nuovi documenti all'interno di un indice già esistente, o di creare un nuovo indice in caso si stia indicizzando il primo documento.

Codice:

```
...
// File da indicizzare PDF
$file = 'myfile/ProvaFilePDF.pdf';
// setta il file da indicizzare
$file->setIndexFile($file);
// processa il documento e lo aggiunge all'indice
$file->addDocument();
// salva il documento nell'indice
$file ->commit();

echo "Documento ($file) indicizzato correttamente<br>";

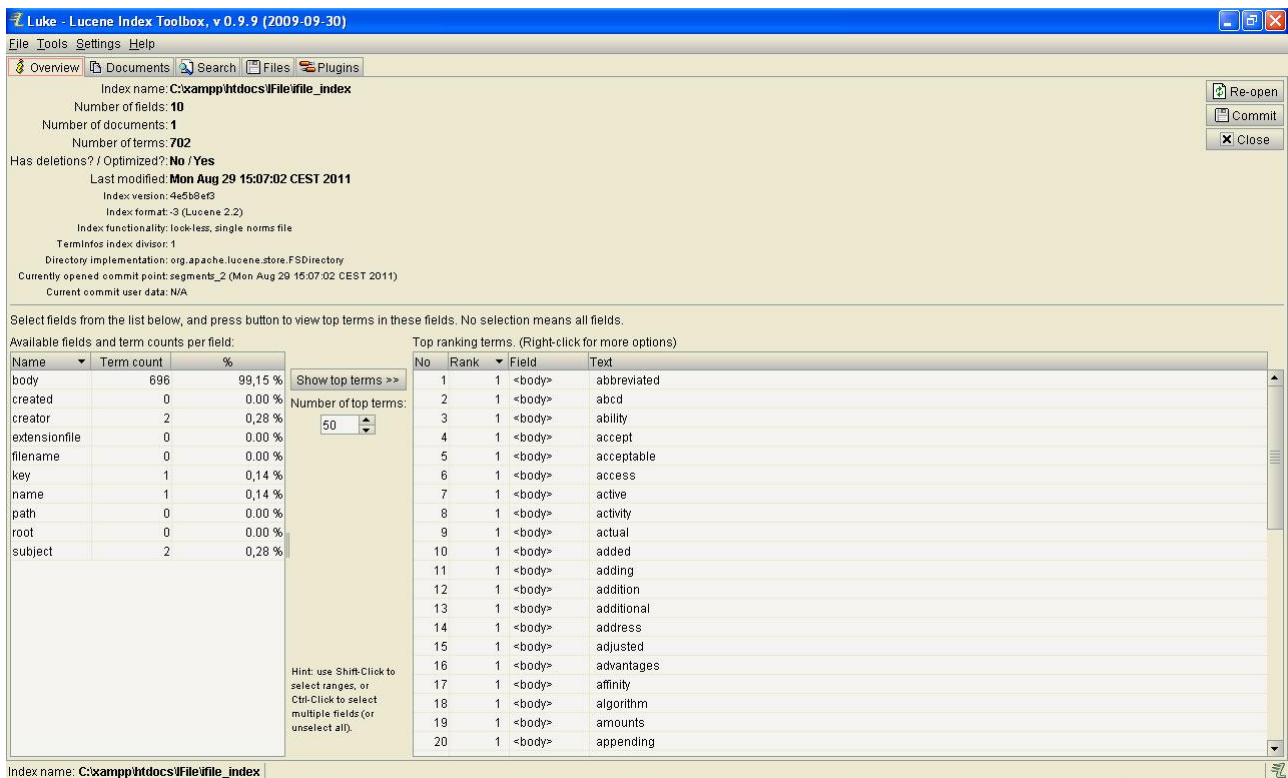
} catch (Exception $e) {
    echo "Errore ($file) ".$e->getMessage();
}
?>
```

Interfaccia LUCENE:

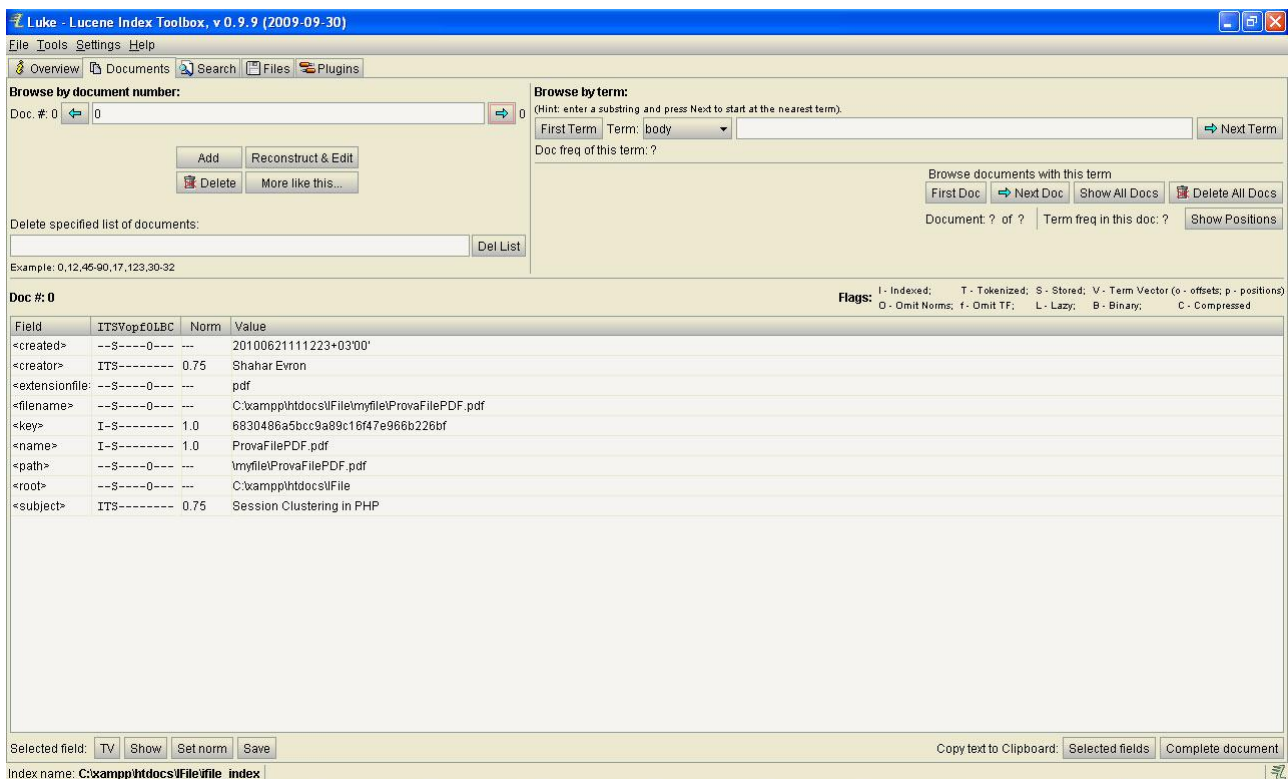
File di esempio:

IFile_Ex01_IndexOneFile.php

Risultato indicizzazione:



(Figura 1)



(Figura 2)

Interfaccia MYSQL:

File di esempio:

IFile_Mysql_Ex01_IndexOneFile.php

Risultato indicizzazione:

Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione
<input type="checkbox"/> id	int(22)			No		auto_increment	
<input type="checkbox"/> deleted	tinyint(4)			No	0		
<input type="checkbox"/> key	varchar(32)	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> body	longtext	latin1_general_ci		No			
<input type="checkbox"/> subject	text	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> creator	text	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> created	text	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> root	text	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> name	text	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> extensionfile	text	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> path	text	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> filename	text	latin1_general_ci		Sì	NULL		

Indici:					Spazio utilizzato		Statistiche righe	
Nome chiave	Tipo	Cardinalità	Azione	Campo	Tipo	Utilizzo	Istruzioni	Valore
PRIMARY	PRIMARY	1		id	Dati	33,716 B	Formato	dinamico
body	FULLTEXT	Nessuno		body	Indice	21,504 B	Collation	latin1_general_ci
key	FULLTEXT	Nessuno		key	Totale	55,220 B	Righe	1
subject	FULLTEXT	Nessuno		subject			Lunghezza riga	33,716
creator	FULLTEXT	Nessuno		creator			Dimensione riga	55,220 B
name	FULLTEXT	Nessuno		name			Prossimo Autoindex	2
Crea un indice su 1 columns <input type="button" value="Esegui"/>							Creazione	29 Ago, 2011 at 03:06 PM
							Ultimo cambiamento	29 Ago, 2011 at 03:06 PM
							Ultimo controllo	29 Ago, 2011 at 03:06 PM

(Figura 3)

Come si nota dalla figura 3, come IFile genera automaticamente i campi ID e DELETE all'interno della struttura dati.

query SQL:

```
SELECT *
FROM 'Ifile_Index'
LIMIT 0,30
```

☐ Profiling [Modifica](#) [Spiega SQL](#) [Crea il codice PHP](#) [Aggiorna](#)

Mostra: 30 righe a partire da 0

in modalità ☐ orizzontale ☒ e ripeti gli headers dopo 100 celle

	id	deleted	key	body	subject	creator	created	root	name	extensionfile	path	filename
<input type="checkbox"/>	1	0	6830486a5bcc9a89c1647e966b226bf	White Paper: Session Clustering in PHP	Session Clustering in PHP	Shahar Evron	20100621111223+03:00V	C:\xampp\htdocs\IFile	ProvaFilePDF.pdf	pdf	\myfile\ProvaFilePDF.pdf	C:\xampp\htdocs\IFile\myfile\ProvaFilePDF.pdf

(Figura 4)

INDICIZZAZIONE MULTIPLA

Il codice seguente, presente nei file di esempio sotto indicati, descrive il processo per l'indicizzazione di più documenti nello stesso processo. Questo può servire in caso si voglia effettuare una pre-indicizzazione di un repository già esistente per poi aggiungervi nuovi documenti.

Codice:

```
...
// directory dove sono presenti i file da indicizzare
```

```
$directory = "myfile";
// Lista dei file
$files = array();

// recupero i file dalla directory
if ($handle = opendir($directory)) {
    while ($file = readdir($handle)) {
        if (!is_dir("{ $directory}/{ $file}")) {
            if ($file != "." & $file != "..") {
                $files[] = "{ $directory}/{ $file}";
            }
        }
    }
}
closedir($handle);

foreach ($files as $file) {
    // questo try/catch permette di gestire eventuali errori nella
    // indicizzazione del singolo file e quindi continuare il ciclo
    // senza interrompere il processo di indicizzazione multipla
    try {
        // setta il file da indicizzare
        $ifile ->setIndexFile($file);
        // processa il documento e lo aggiunge all'indice
        $ifile ->addDocument();
        // salva il documento nell'indice
        $ifile ->commit();

        echo "Documento ($file) indicizzato correttamente<br>";
    } catch (Exception $e) {
        echo "Errore ($file) ".$e->getMessage(). "<br>";
    }
}

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
?>
```

Interfaccia LUCENE:

File di esempio:

IFile_Ex02_MultipleFile.php

Risultato indicizzazione:

Luke - Lucene Index Toolbox, v 0.9.9 (2009-09-30)

File Tools Settings Help

Overview Documents Search Files Plugins

Index name: C:\xampp\htdocs\file_index

Number of fields: 11
Number of documents: 4
Number of terms: 719

Has deletions? / Optimized? No / No

Last modified: Mon Aug 29 15:51:27 CEST 2011

Index version: 4e5b8e0
Index format: -3 (Lucene 2.2)
Index functionality: lock-less, single norms file
TermInfos index divisor: 1
Directory implementation: org.apache.lucene.store.FSDirectory
Currently opened commit point: segments_5 (Mon Aug 29 15:51:27 CEST 2011)
Current commit user data: N/A

Select fields from the list below, and press button to view top terms in these fields. No selection means all fields.

Available fields and term counts per field:

Name	Term count	%
body	707	98,33 %
created	0	0,00 %
creator	2	0,28 %
extensionfile	0	0,00 %
filename	0	0,00 %
key	4	0,56 %
modified	0	0,00 %
name	4	0,56 %
path	0	0,00 %
root	0	0,00 %
subject	2	0,28 %

Show top terms >>>

Number of top terms: 50

Hint: use Shift-Click to select ranges, or Ctrl-Click to select multiple fields (or unselect all).

Top ranking terms. (Right-click for more options)

No	Rank	Field	Text
1	3	<body>	file
2	2	<body>	prova
3	1	<body>	abcd
4	1	<body>	accept
5	1	<body>	acceptable
6	1	<body>	access
7	1	<body>	activity
8	1	<body>	advantages
9	1	<body>	actual
10	1	<body>	added
11	1	<body>	adding
12	1	<body>	addition
13	1	<body>	additional
14	1	<body>	address
15	1	<body>	attached
16	1	<body>	assume
17	1	<body>	affinity
18	1	<body>	algorithm
19	1	<body>	alias
20	1	<body>	amounts

Index name: C:\xampp\htdocs\file_index

(Figura 1)

Field	ITSVopE0LBC	Norm	Value
<created>	--S---0---	---	20100621111223+03'00'
<creator>	ITS-----	0.75	Shahar Evron
<extensionfile>	--S---0---	---	pdf
<filename>	--S---0---	---	C:\xampp\htdocs\file\myfile\ProvaFilePDF.pdf
<key>	I-S-----	1.0	6830486a5bcc9a89c16f47e966b226bf
<name>	I-S-----	1.0	ProvaFilePDF.pdf
<path>	--S---0---	---	\myfile\ProvaFilePDF.pdf
<root>	--S---0---	---	C:\xampp\htdocs\file
<subject>	ITS-----	0.75	Session Clustering in PHP
Field	ITSVopE0LBC	Norm	Value
<extensionfile>	--S---0---	---	doc
<filename>	--S---0---	---	C:\xampp\htdocs\file\myfile\ProvaFileDOC.doc
<key>	I-S-----	1.0	4b127a63360f998d791e71e0b7b4517a
<name>	I-S-----	1.0	ProvaFileDOC.doc
<path>	--S---0---	---	\myfile\ProvaFileDOC.doc
<root>	--S---0---	---	C:\xampp\htdocs\file
Field	ITSVopE0LBC	Norm	Value
<created>	--S---0---	---	2010-11-02 12:41:44Z
<extensionfile>	--S---0---	---	ods
<filename>	--S---0---	---	C:\xampp\htdocs\file\myfile\ProvaFileODS.ods
<key>	I-S-----	1.0	ae7fd1122810c12ed26625c0b4afb683
<name>	I-S-----	1.0	ProvaFileODS.ods
<path>	--S---0---	---	\myfile\ProvaFileODS.ods
<root>	--S---0---	---	C:\xampp\htdocs\file
Field	ITSVopE0LBC	Norm	Value
<extensionfile>	--S---0---	---	txt
<filename>	--S---0---	---	C:\xampp\htdocs\file\myfile\ProvaFileTXT.txt
<key>	I-S-----	1.0	69bd4caa95a1d74f0b25544a627df70
<modified>	--S---0---	---	19/07/2011 15:40:10.
<name>	I-S-----	1.0	ProvaFileTXT.txt
<path>	--S---0---	---	\myfile\ProvaFileTXT.txt
<root>	--S---0---	---	C:\xampp\htdocs\file

(Figura 2)

Interfaccia MYSQLI:

File di esempio:

IFile_Mysqlq_Ex02_MultipleFile.php

Risultato indicizzazione:

Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione							
<input type="checkbox"/> id	int(22)			No		auto_increment								
<input type="checkbox"/> deleted	tinyint(4)			No	0									
<input type="checkbox"/> key	varchar(32)	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> body	longtext	latin1_general_ci		No										
<input type="checkbox"/> subject	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> creator	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> created	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> root	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> name	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> extensionfile	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> path	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> filename	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/> modified	text	latin1_general_ci		Sì	NULL									

Indici: ?

Nome chiave	Tipo	Cardinalità	Azione	Campo
PRIMARY	PRIMARY	4		id
body	FULLTEXT	Nessuno		body
key	FULLTEXT	Nessuno		key
subject	FULLTEXT	Nessuno		subject
creator	FULLTEXT	Nessuno		creator
name	FULLTEXT	Nessuno		name

Crea un indice su columns

Spazio utilizzato

Tipo	Utilizzo
Dati	34,644 B
Indice	27,648 B
Totale	62,292 B

Statistiche righe

Istruzioni	Valore
Formato	dinamico
Collation	latin1_general_ci
Righe	4
Lunghezza riga ø	8,661
Dimensione riga ø	15,573 B
Prossimo Autoindex	5
Creazione	29 Ago, 2011 at 03:51 PM
Ultimo cambiamento	29 Ago, 2011 at 03:55 PM

(Figura 3)

	id	deleted	key	body	subject	creator	created	root	name	extensionfile	path	filename	modified
<input type="checkbox"/>	1	0	6830486a5bcc9a89c16f47e966b226bf	White Paper: Session Clustering in PHP	Session Clustering in PHP	Shahar Evron	20100621111223+03'00'	C:\xampp\htdocs\IFile	ProvaFilePDF.pdf	pdf	\myfile\ProvaFilePDF.pdf	C:\xampp\htdocs\IFile\myfile\ProvaFilePDF.pdf	NULL
<input type="checkbox"/>	2	0	4b127a63360f998d791e71e0b7b4517a	Prova indicizzazione file DOC.	NULL	NULL	NULL	C:\xampp\htdocs\IFile	ProvaFileDOC.doc	doc	\myfile\ProvaFileDOC.doc	C:\xampp\htdocs\IFile\myfile\ProvaFileDOC.doc	NULL
<input type="checkbox"/>	3	0	ae7fd1122810c12ed26625c0b4atb683	menutype name alias link to menu Nome 11 to...	NULL	NULL	2010-11-02 12:41:44Z	C:\xampp\htdocs\IFile	ProvaFileODS.ods	ods	\myfile\ProvaFileODS.ods	C:\xampp\htdocs\IFile\myfile\ProvaFileODS.ods	NULL
<input type="checkbox"/>	4	0	69bd4caa95a1df740b25544a627df70	prova file TXT	NULL	NULL	NULL	C:\xampp\htdocs\IFile	ProvaFileTXT.txt	txt	\myfile\ProvaFileTXT.txt	C:\xampp\htdocs\IFile\myfile\ProvaFileTXT.txt	19/07/2011 15:40:10

(Figura 4)

INDICIZZAZIONE MANUALE

Il codice seguente, presente nei file di esempio sotto indicati, descrive il processo per l'indicizzazione manuale di un documento.

Questo può servire in caso si voglia inserire contenuti che non sono esclusivamente presenti in file, ma si vuole creare un proprio indice in funzione di contenuti diversi.

Ad esempio potrebbe essere utilizzato per l'indicizzazione di contenuti presenti all'interno di un Data Base, o si potrebbe decidere di indicizzare le pagine web di un sito internet. Utilizzando una giusta struttura di fields, questo permetterebbe di utilizzare la potenza di IFile per l'indicizzazione e la

successiva ricerca dei contenuti presenti in strutture dove gli stessi sono difficilmente indicizzabili e ricercabili.

Codice:

```
...

// testo da indicizzare
$text = "Prova testo da indicizzare manualmente.
        Il testo potrebbe essere recuperato da un DB
        o da qualsiasi altra fonte.";

// Per creare un documento manualmente si possono utilizzare due
// processi differenti:
// 1. LuceneDataIndexBean: con l'utilizzo dell'oggetto e'
//    LuceneDataIndexBean e' possibile settare la maggior parte dei
//    fields utilizzati dalla libreria e farsi restituire un oggetto
//    Zend_Search_Lucene_Document
//    Questo permette di evitare di definire il tipo di fields e di
//    utilizzare la stessa struttura che utilizza IFile per
//    l'indicizzazione dei documenti.
//
// 2. Istanziando un oggetto Zend_Search_Lucene_Document
//    creando i field manualmente.

// require dello script per la creazione di un oggetto LuceneDataIndexBean
require_once
'C:/xampp/htdocs/IFile/iframe/adapters/beans/LuceneDataIndexBean.php';
// crea un'istanza di un oggetto LuceneDataIndexBean
$bean = new LuceneDataIndexBean();
// settaggio dei fields
// l'unico field obbligatorio è il field body in caso non esiste
// il sistema invoca una eccezione di body mancante
$bean->setBody($text);
$bean->setCreated('21 Luglio 2011');
$bean->setCreator('jumpaolo');
$bean->setKeywords("key01,key02");
$bean->setModified("22 Luglio 2011");
$bean->setSubject("Oggetto del documento da indicizzare");
$bean->setDescription("Breve descrizione del documento da indicizzare");
$bean->setTitle("Titolo del documento da indicizzare");
// recupero dell'oggetto Zend_Search_Lucene_Document dal bean
$doc = $bean->getLuceneDocument();

// 2. Utilizzare l'oggetto Zend_Search_Lucene_Document
// $doc = new Zend_Search_Lucene_Document();
// $doc->addField(Zend_Search_Lucene_Field::UnStored('body', $text));

// A questo punto e' possibile aggiungere altri fields indipendentemente
// dall'oggetto utilizzato per la creazione di un documento
$doc->addField(Zend_Search_Lucene_Field::Keyword('key', md5($text)));
$doc->addField(Zend_Search_Lucene_Field::Keyword('category', 'DB'));

// processa il documento e lo aggiunge all'indice
$Ifile->addDocument($doc);
// salva il documento nell'indice
$Ifile->commit();
```



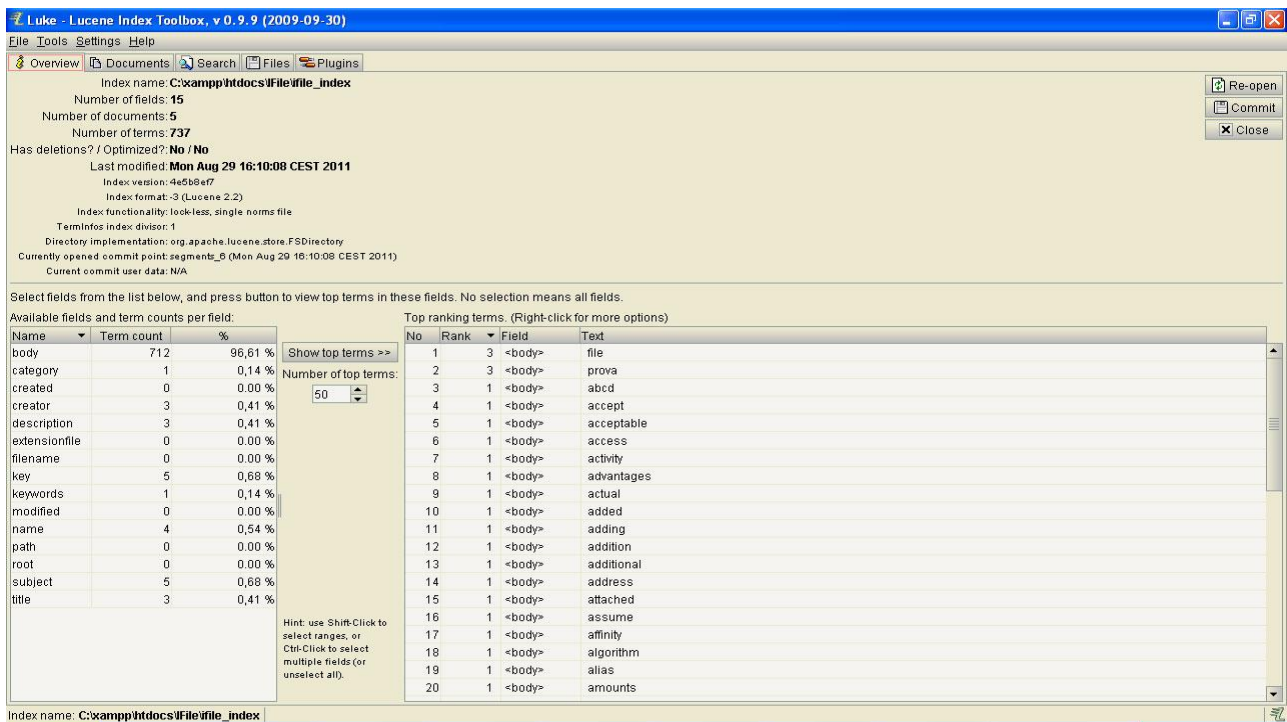
```
    echo "Documento Manuale: indicizzato correttamente<br>";  
} catch (Exception $e) {  
    echo "Errore: ".$e->getMessage();  
}  
?>
```

Interfaccia LUCENE:

File di esempio:

IFile_Ex03_IndexManual.php

Risultato indicizzazione:



Overview | Documents | Search | Files | Plugins

Index name: C:\xampp\htdocs\IFile\file_index
 Number of fields: 15
 Number of documents: 5
 Number of terms: 737
 Has deletions? / Optimized?: No / No
 Last modified: Mon Aug 29 16:10:08 CEST 2011
 Index version: 4e5b8ef7
 Index format: 3 (Lucene 2.2)
 Index functionality: lock-less, single norms file
 TermInfos index divisor: 1
 Directory implementation: org.apache.lucene.store.FSDirectory
 Currently opened commit point: segments_8 (Mon Aug 29 16:10:08 CEST 2011)
 Current commit user data: N/A

Select fields from the list below, and press button to view top terms in these fields. No selection means all fields.

Available fields and term counts per field:

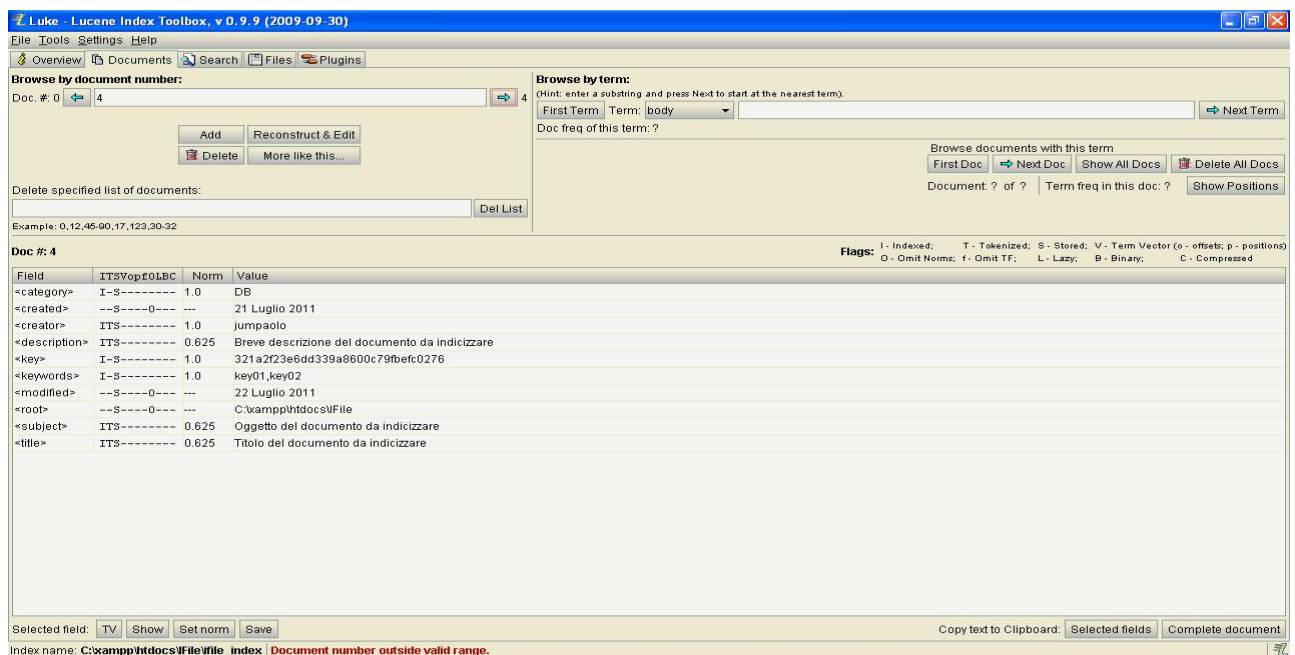
Name	Term count	%
body	712	96,61 %
category	1	0,14 %
created	0	0,00 %
creator	3	0,41 %
description	3	0,41 %
extensionfile	0	0,00 %
filename	0	0,00 %
key	5	0,68 %
keywords	1	0,14 %
modified	0	0,00 %
name	4	0,54 %
path	0	0,00 %
root	0	0,00 %
subject	5	0,68 %
title	3	0,41 %

Number of top terms: 50

Top ranking terms. (Right-click for more options)

No	Rank	Field	Text
1	3	<body>	file
2	3	<body>	prova
3	1	<body>	abcd
4	1	<body>	accept
5	1	<body>	acceptable
6	1	<body>	access
7	1	<body>	activity
8	1	<body>	advantages
9	1	<body>	actual
10	1	<body>	added
11	1	<body>	adding
12	1	<body>	addition
13	1	<body>	additional
14	1	<body>	address
15	1	<body>	attached
16	1	<body>	assume
17	1	<body>	affinity
18	1	<body>	algorithm
19	1	<body>	alias
20	1	<body>	amounts

(Figura 1)



Browse by document number:

Doc # 0 4

Buttons: Add, Reconstruct & Edit, Delete, More like this...

Delete specified list of documents: [] Del List

Example: 0,12,45-90,17,123,30-32

Browse by term:

(Hint: enter a subitting and press Next to start at the nearest term.)

First Term Term: body Next Term

Doc freq of this term: ?

Browse documents with this term

First Doc Next Doc Show All Docs Delete All Docs

Document: ? of ? Term freq in this doc: ? Show Positions

Doc #: 4

Field	ITSVopEOLBC	Norm	Value
<category>	I-S-----	1.0	DB
<created>	--S---0---	---	21 Luglio 2011
<creator>	ITS-----	1.0	jumpaolo
<description>	ITS-----	0.625	Breve descrizione del documento da indicizzare
<key>	I-S-----	1.0	321a2f23e6dd339a8600c79bfefc0276
<keywords>	I-S-----	1.0	key01, key02
<modified>	--S---0---	---	22 Luglio 2011
<root>	--S---0---	---	C:\xampp\htdocs\IFile
<subject>	ITS-----	0.625	Oggetto del documento da indicizzare
<title>	ITS-----	0.625	Titolo del documento da indicizzare

Selected field: TV Show Set norm Save

Copy text to Clipboard: Selected fields Complete document

Index name: C:\xampp\htdocs\IFile\file_index Document number outside valid range.

(Figura 2)

Interfaccia MySQLI:

File di esempio:

IFile_MySql_Ex03_IndexManual.php

Risultato indicizzazione:

	Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione							
<input type="checkbox"/>	id	int(22)			No		auto_increment								
<input type="checkbox"/>	deleted	tinyint(4)			No	0									
<input type="checkbox"/>	key	varchar(32)	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	body	longtext	latin1_general_ci		No										
<input type="checkbox"/>	subject	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	creator	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	created	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	root	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	name	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	extensionfile	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	path	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	filename	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	modified	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	title	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	description	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	keywords	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	category	text	latin1_general_ci		Sì	NULL									

Indici: ⓘ					Spazio utilizzato		Statistiche righe	
Nome chiave	Tipo	Cardinalità	Azione	Campo	Tipo	Utilizzo	Istruzioni	Valore
PRIMARY	PRIMARY	5		id	Dati	35,020 B	Formato	dinamico
body	FULLTEXT	Nessuno		body	Indice	31,744 B	Collation	latin1_general_ci
key	FULLTEXT	Nessuno		key	Totale	66,764 B	Righe	5
subject	FULLTEXT	Nessuno		subject			Lunghezza riga ø	7,004
creator	FULLTEXT	Nessuno		creator			Dimensione riga ø	13,353 B
name	FULLTEXT	Nessuno		name			Prossimo Autoindex	6
title	FULLTEXT	Nessuno		title			Creazione	29 Ago, 2011 at 04:21 PM
description	FULLTEXT	Nessuno		description			Ultimo cambiamento	29 Ago, 2011 at 04:21 PM
keywords	FULLTEXT	Nessuno		keywords				
category	FULLTEXT	Nessuno		category				

Crea un indice su columns

(Figura 3)

id	deleted	key	body	subject	creator	created	root	name	extensionfile	path	filename	modified	title	description	keywords	category
5	0	321a2023e6dd339a8600c79b0fc0276	Prova testo da indicizzare manualmente. Il t...	Oggetto del documento da indicizzare	jumpaolo	21 Luglio 2011	C:\xampp\htdocs\iFile	NULL	NULL	NULL	NULL	22 Luglio 2011	Titolo del documento da indicizzare	Breve descrizione del documento da indicizzare	key01,key02	DB

(Figura 4)

UTILIZZARE I CUSTOM FIELDS

Il codice seguente, presente nei file di esempio sotto indicati, descrive il processo di indicizzazione automatica di un file con l'integrazione di eventuali fields (campi) personalizzati al documento. Questo permette di aggiungere ai fields descritti nel capitolo "Fields e Custom Fields" nuovi fields che fanno parte del dominio applicativo del documento.

Codice:

```
...
// File da indicizzare PDF
$file = 'myfile/ProvaFileConCustomField.txt';
// setta il file da indicizzare
$file->setIndexFile($file);

// settaggio dei CustomField
$file->addCustomField('customfield_Keyword', 'mykeyword',
IFile_Indexing_Interface::FIELD_TYPE_KEYWORD);
$file->addCustomField('customfield_Text', 'mytext',
IFile_Indexing_Interface::FIELD_TYPE_TEXT);
$file->addCustomField('customfield_UnStored', 'mytextUnstored',
IFile_Indexing_Interface::FIELD_TYPE_UNSTORED);
$file->addCustomField('customfield_UnIndexed', 'mytextUnindexed',
IFile_Indexing_Interface::FIELD_TYPE_UNINDEXED);
$file->addCustomField('customfield_Binary', 'Binary',
IFile_Indexing_Interface::FIELD_TYPE_BINARY);

// ATTENZIONE:
// utilizzando un custom field con lo stesso nome dei field creati
// da IFile quest'ultimo, andra' a sovrascrivere il field creato da IFile.
// Esempio: questo field andra' a sovrascrivere il
// field "key" creato da IFile
$file->addCustomField('key', 'mykeyfile',
IFile_Indexing_Interface::FIELD_TYPE_KEYWORD);

// ritorna i custom field settati
$customFields = $file->getCustomField();
// processa il documento e lo aggiunge all'indice
$file->addDocument();
// salva il documento nell'indice
$file->commit();

echo "Documento ($file) indicizzato correttamente<br>";
echo "Campi personalizzati:<br>";
var_dump($customFields);

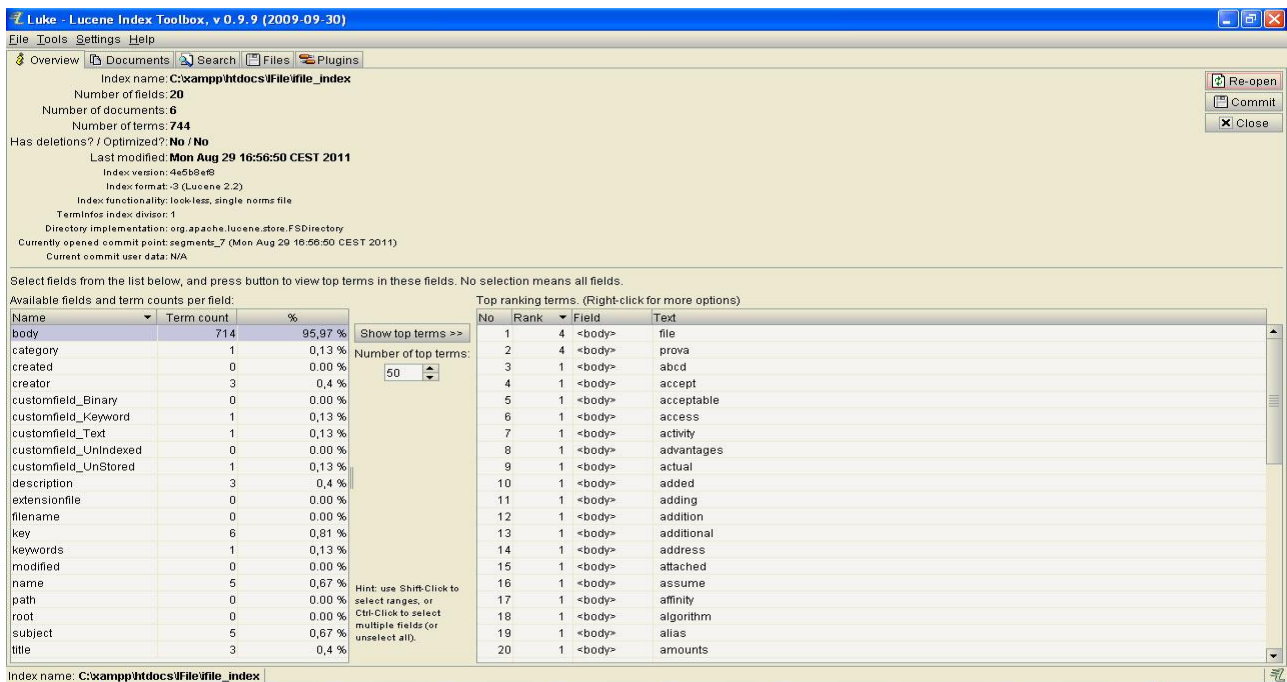
} catch (Exception $e) {
    echo "Errore ($file) ".$e->getMessage();
}
?>
```

Interfaccia LUCENE:

File di esempio:

IFile_Ex04_CustomFields.php

Risultato indicizzazione:



Overview

Index name: C:\xampp\htdocs\IFile\file_index

Number of fields: 20

Number of documents: 6

Number of terms: 744

Has deletions? / Optimized?: No / No

Last modified: Mon Aug 29 16:56:50 CEST 2011

Index version: 4u30e6

Index format: 3 (Lucene 2.2)

Index functionality: lookless, single norms file

TermInfos index divisor: 1

Directory implementation: org.apache.lucene.store.FSDirectory

Currently opened commit point: segments_7 (Mon Aug 29 16:56:50 CEST 2011)

Current commit user data: N/A

Select fields from the list below, and press button to view top terms in these fields. No selection means all fields.

Available fields and term counts per field:

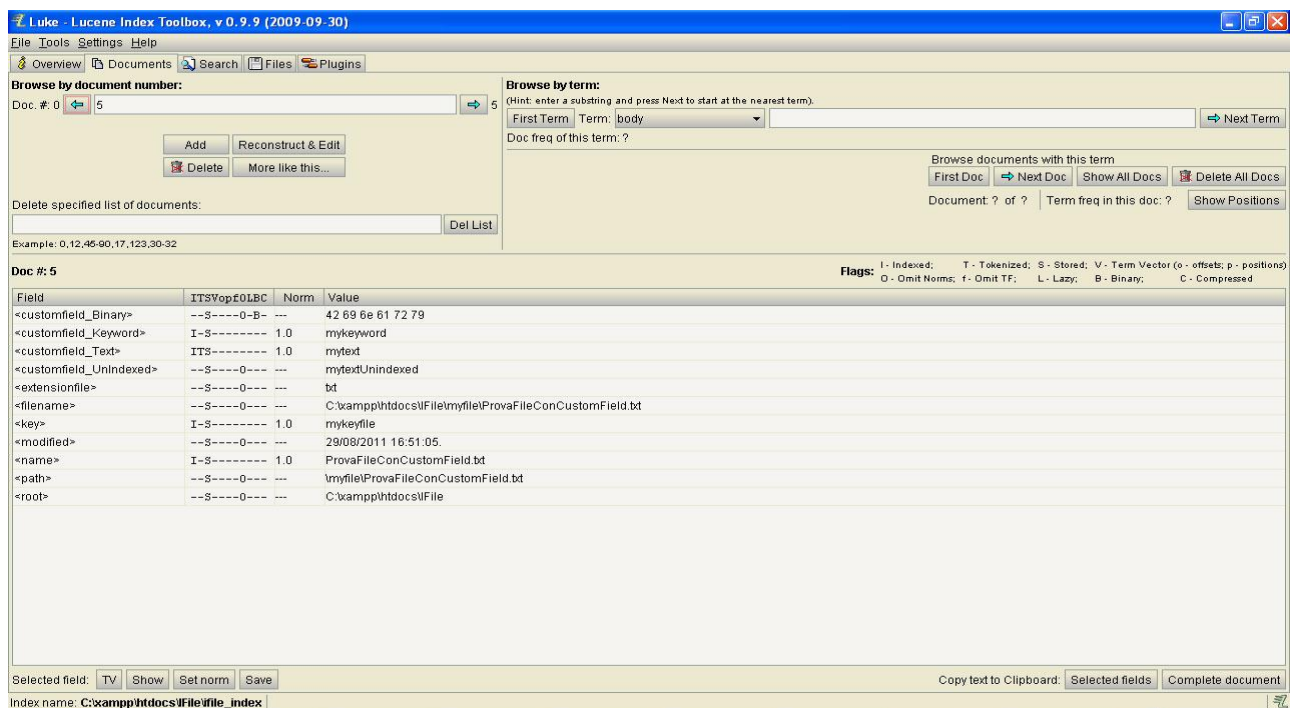
Name	Term count	%
body	714	95,97 %
category	1	0,13 %
created	0	0,00 %
creator	3	0,4 %
customfield_Binary	0	0,00 %
customfield_Keyword	1	0,13 %
customfield_Text	1	0,13 %
customfield_UnIndexed	0	0,00 %
customfield_UnStored	1	0,13 %
description	3	0,4 %
extensionfile	0	0,00 %
filename	0	0,00 %
key	6	0,81 %
keywords	1	0,13 %
modified	0	0,00 %
name	5	0,67 %
path	0	0,00 %
root	0	0,00 %
subject	5	0,67 %
title	3	0,4 %

Number of top terms: 50

Top ranking terms. (Right-click for more options)

No	Rank	Field	Text
1	4	<body>	file
2	4	<body>	prova
3	1	<body>	abcd
4	1	<body>	accept
5	1	<body>	acceptable
6	1	<body>	access
7	1	<body>	activity
8	1	<body>	advantages
9	1	<body>	actual
10	1	<body>	added
11	1	<body>	adding
12	1	<body>	addition
13	1	<body>	additional
14	1	<body>	address
15	1	<body>	attached
16	1	<body>	assume
17	1	<body>	affinity
18	1	<body>	algorithm
19	1	<body>	alias
20	1	<body>	amounts

(Figura 1)



Browse by document number:

Doc #: 0 5

Add Reconstruct & Edit

Delete More like this...

Delete specified list of documents:

Example: 0,12,45-90,17,123,30-32

Del List

Browse by term:

(Hint: enter a substring and press Next to start at the nearest term.)

First Term Term: body

Doc freq of this term: ?

Browse documents with this term

First Doc Next Doc Show All Docs Delete All Docs

Document: ? of ? Term freq in this doc: ? Show Positions

Doc #: 5

Field	ITSVopE0IBC	Norm	Value
<customfield_Binary>	--S---0-B---		42 69 6e 61 72 79
<customfield_Keyword>	I-S-----	1.0	mykeyword
<customfield_Text>	ITS-----	1.0	mytext
<customfield_UnIndexed>	--S---0---		mytextUnindexed
<extensionfile>	--S---0---		bt
<filename>	--S---0---		C:\xampp\htdocs\IFile\myfile\ProvaFileConCustomField.bt
<key>	I-S-----	1.0	mykeyfile
<modified>	--S---0---		29/08/2011 16:51:05
<name>	I-S-----	1.0	ProvaFileConCustomField.bt
<path>	--S---0---		\\myfile\ProvaFileConCustomField.bt
<root>	--S---0---		C:\xampp\htdocs\IFile

Selected field: TV Show Set norm Save

Copy text to Clipboard: Selected fields Complete document

Index name: C:\xampp\htdocs\IFile\file_index

(Figura 2)

Interfaccia MYSQLI:

File di esempio:

IFile_Mysql_Ex04_CustomFields.php

Risultato indicizzazione:

	Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione							
<input type="checkbox"/>	id	int(22)			No		auto_increment								
<input type="checkbox"/>	deleted	tinyint(4)			No	0									
<input type="checkbox"/>	key	varchar(32)	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	body	longtext	latin1_general_ci		No										
<input type="checkbox"/>	subject	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	creator	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	created	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	root	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	name	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	extensionfile	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	path	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	filename	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	modified	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	title	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	description	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	keywords	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	category	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	customfield_keyword	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	customfield_text	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	customfield_unstored	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	customfield_unindexed	text	latin1_general_ci		Sì	NULL									
<input type="checkbox"/>	customfield_binary	blob		BINARY	Sì	NULL									

Indici: ⓘ					Spazio utilizzato		Statistiche righe	
Nome chiave	Tipo	Cardinalità	Azione	Campo	Tipo	Utilizzo	Istruzioni	Valore
PRIMARY	PRIMARY	6		id	Dati	35,316 B	Formato	dinamico
body	FULLTEXT	Nessuno		body	Indice	35,840 B	Collation	latin1_general_ci
key	FULLTEXT	Nessuno		key	Totale	71,156 B	Righe	6
subject	FULLTEXT	Nessuno		subject			Lunghezza riga ø	5,886
creator	FULLTEXT	Nessuno		creator			Dimensione riga ø	11,859 B
name	FULLTEXT	Nessuno		name			Prossimo Autoindex	7
title	FULLTEXT	Nessuno		title			Creazione	29 Ago, 2011 at 04:56 PM
description	FULLTEXT	Nessuno		description			Ultimo cambiamento	29 Ago, 2011 at 04:56 PM
keywords	FULLTEXT	Nessuno		keywords				
category	FULLTEXT	Nessuno		category				
customfield_keyword	FULLTEXT	Nessuno		customfield_keyword				
customfield_text	FULLTEXT	Nessuno		customfield_text				
customfield_unstored	FULLTEXT	Nessuno		customfield_unstored				

Crea un indice su columns

(Figura 3)

id	deleted	key	body	subject	creator	created	text	name	extensionfile	path	filename	modified	title	description	keywords	category	customfield_keyword	customfield_text	customfield_unstored	customfield_unindexed	customfield_binary
1	0	mykeyfile	prova file con custom field TXT	NULL	NULL	NULL	C:\xampp\htdocs\ifile	Prova file con custom field.txt	txt	mykeyfile	C:\xampp\htdocs\ifile	29/08/2011 16:51:05	NULL	NULL	NULL	NULL	mykeyword	mytext	mytextunstored	mytextunindexed	mytextbinary

LAVORARE CON L'INDICE

Il codice seguente, presente nei file di esempio sotto indicati, descrive il meccanismo per interfacciarsi con l'indice creato.

Codice:

...

```
// TEST Cancellazione di un indice esistente
//echo "delete: ";
//echo $ifile->delete(1);

// TEST Cancellazione di un indice fuori dal range
// echo "delete (id no range): ";
// echo $ifile->delete(100000);

// TEST Ottimizzazione dell'indice
// echo "optimize: ";
// echo $ifile->optimize();

// TEST Numero di documenti (compresi quelli marcati come cancellati)
echo "count: ";
echo $ifile->count()."<br>";

// TEST Numero di documenti (senza quelli marcati come cancellati)
echo "numDocs: ";
echo $ifile->numDocs()."<br>";

// TEST Se ci sono documenti cancellati
echo "hasDeletions: ";
echo $ifile->hasDeletions()."<br>";

// TEST Se e' presente il termine ricercato
echo "hasTerm (Term:file, Field:body): ";
echo $ifile->hasTerm('file', 'body')."<br>";

// TEST Se e' presente il termine ricercato
echo "hasTerm (Term:file, Field:): ";
echo $ifile->hasTerm('file')."<br>";

// TEST Ritorna il field di default
echo "getDefaultSearchField: ";
echo $ifile->getDefaultSearchField()."<br>";

// TEST Setta il field di default
echo "setDefaultSearchField (field:body): ";
echo $ifile->setDefaultSearchField('body')."<br>";

// TEST Ritorna il field di default
echo "getDefaultSearchField: ";
echo $ifile->getDefaultSearchField()."<br>";

// TEST Se e' presente il termine ricercato
echo "hasTerm (Term:file, Field:) (setDefaultSearchField:body): ";
echo $ifile->hasTerm('file')."<br>";

// TEST Setta il field di default (non esistente)
echo "setDefaultSearchField: ";
echo $ifile->setDefaultSearchField('notexist')."<br>";

// TEST Se e' presente il termine ricercato
echo "hasTerm (Term:file, Field:) (setDefaultSearchField:not exist): ";
echo $ifile->hasTerm('file')."<br>";

// TEST Se il documento e' cancellato
echo "isDeleted (ID:1) : ";
echo $ifile->isDeleted(1)."<br>";
```

```
// TEST Riporta tutti i doc marcati come cancellati a non cancellati
// (NON IMPLEMENTATO per interfaccia LUCENE)
//echo "undeletedAll: ";
//echo $ifile->undeletedAll()."<br>";

// TEST Setta il limite massimo di risultati
echo "setResultLimit (10): ";
echo $ifile->setResultLimit(10)."<br>";

// TEST Il limite massimo di risultati
echo "getResultLimit: ";
echo $ifile->getResultLimit()."<br>";

// TEST Field dell'indice (tutti)
echo "getFieldNames: <pre>";
echo print_r($ifile->getFieldNames(), 1)."</pre><br>";

// TEST Field dell'indice (solo quelli indicizzati)
echo "getFieldNames (only indexed): <pre>";
echo print_r($ifile->getFieldNames(true), 1)."</pre><br>";

// TEST Recupero di un Documento dall'indice
echo "getDocument (ID:0): <pre>";
echo print_r($ifile->getDocument(0), 1)."</pre><br>";

// TEST Recupero di un Documento dall'indice (id fuori range)
//echo "getDocument (id no range): <pre>";
//echo print_r($ifile->getDocument(100), 1)."</pre><br>";

// TEST Recupero di tutti i Documento dall'indice senza quelli cancellati
echo "getAllDocument (without delete document): <pre>";
echo print_r($tot = $ifile->getAllDocument(), 1)."</pre><br>";
echo "Totale: ".$count($tot)."<br>";

// TEST Recupero di tutti i Documento dall'indice compresi quelli
cancellati
echo "getAllDocument (with delete document): <pre>";
echo print_r($tot = $ifile->getAllDocument(true), 1)."</pre><br>";
echo "Totale: ".$count($tot)."<br>";

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
?>
```

Interfaccia LUCENE:

File di esempio:

IFile_Ex05_UseIndex.php

Interfaccia MYSQLI:

File di esempio:

IFile_Mysql_Ex05_UseIndex.php

RICERCARE ALL'INTERNO DI UN INDICE

Il codice seguente, presente nei file di esempio sotto indicati, descrive con più esempi il meccanismo di ricerca all'interno di un indice.

Tutti i file richiedono la seguente funzione per la presentazione dei risultati:

```
/**
 * Stampa il risultato della ricerca.
 *
 * Questo metodo e' utilizzato come esempio per
 * presentare i risultati della ricerca.
 *
 * @param string $type
 * @param array $result_T
 * @return
 */
function printResult($type, $result) {

    echo "TIPO RICERCA: ".$type;
    if(!empty($result) && is_array($result)) {
        echo "<br>Result Search:<br>";
        foreach ($result as $hit) {
            try {
                $doc = $hit->getDocument();
                echo "File: ".$doc->name." - Chiave: ".$doc->key." -
Score: ".$hit->score."<br>";
            } catch (exception $e) {
                echo "File: - Chiave: ".$doc->key." - Score: ".$hit-
>score."<br>";
            }
        }
    } else {
        echo "<br>Ricerca vuota<br>";
    }

    echo "Fine stampa ricerca($type)<br><br>";
}
```

Codice:

Esempio 1.a - Utilizzo della ricerca: Parser di Lucene

Interfaccia LUCENE:

File di esempio:

IFile_Ex06_SearchParser.php

```
...
// TEST Query PARSER
// Per l'utilizzo del metodo di ricerca PARSER si può utilizzare
// la sintassi di ricerca di Lucene.
$search = "body:zend";
// definizione dell'ordinamento
$file->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Parser
$result = $file->queryParser($search);
```

```
// stampa dei risultati
printResult("Parser Query Lucene - order key:desc", $result);

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
```

Esempio 1.b – Utilizzo della ricerca: Parser di Mysql

Interfaccia MYSQLI:

File di esempio:

IFile_Mysql_Ex06_SearchParser.php

```
...
// TEST Query PARSE
// Per l'utilizzo del metodo di ricerca PARSE si può utilizzare
// la sintassi di ricerca per le FULLTEXT di MySQL
$search = "MATCH(body) AGAINST ('zend')";
// definizione dell'ordinamento
$file->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Parser
$result = $file->queryParser($search);
// stampa dei risultati
printResult("Parser Query MySQL - body:zend - order key:desc", $result);

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
```

Esempio 2 – Utilizzo della ricerca: Multi-Termine

Interfaccia LUCENE:

File di esempio:

IFile_Ex07_SearchMultiTerm.php

Interfaccia MYSQLI:

File di esempio:

IFile_Mysql_Ex07_SearchMultiTerm.php

```
...
// TEST Query Multi-termini

// 1. Ricerca di un singolo termine.
// Per effettuare una ricerca si utilizza l'oggetto IFileQueryRegistry
$fileQueryRegistry = new IFileQueryRegistry();
$fileQueryRegistry->setQuery('zend', 'body', IFileQuery::MATCH_REQUIRED);
```

```
// definizione dell'ordinamento
$file->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Multi-temine
$result = $file->query($fileQueryRegistry);
// stampa dei risultati
printResult("Ricerca Singolo termine (body:zend)", $result);

// 2. Ricerca piu' termini in AND.
// @TODO sembra ci sia un problema nella libreria LUCENE
// Zend\Search\Lucene\Search\Query\MultiTerm.php
// IL PROBLEMA È NATO DOPO L'OPTIMIZE
//
// In caso un solo termine non e' presente in tutti i documenti
// presenti nell'indice la ricerca risulta vuota
// Per effettuare una ricerca si utilizza l'oggetto IFileQueryRegistry
$fileQueryRegistry = new IFileQueryRegistry();
$fileQueryRegistry->setQuery('file', 'body', IFileQuery::MATCH_REQUIRED);
$fileQueryRegistry->setQuery('prova', 'body',
IFileQuery::MATCH_PROHIBITEN);

// definizione dell'ordinamento
$file->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Multi-temine
$result = $file->query($fileQueryRegistry);
// stampa dei risultati
printResult("Ricerca di piu' termini in AND (body:prova, body:file)",
$result);

// 3. Ricerca piu' termini in OR.
// Per effettuare una ricerca si utilizza l'oggetto IFileQueryRegistry
$fileQueryRegistry = new IFileQueryRegistry();
// se non valorizzato il parametro di match, il termine
// viene considerato come un termine opzionale nella ricerca
$fileQueryRegistry->setQuery('zend', 'body');
$fileQueryRegistry->setQuery('doc', 'body');
$fileQueryRegistry->setQuery('txt', 'body');
$fileQueryRegistry->setQuery('topmenu', 'body');
// chiamata al metodo di ricerca Multi-temine
$result = $file->query($fileQueryRegistry);
// stampa dei risultati
printResult("Ricerca di piu' termini in OR (body:zend, body:doc, body:txt,
body:topmenu)", $result);

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
```

Esempio 3 - Utilizzo della ricerca: Frase esatta

Interfaccia LUCENE:

File di esempio:

IFile_Ex08_SearchPhrase.php

Interfaccia MYSQLI:

File di esempio:

IFile_Mysql_Ex08_SearchPhrase.php

```
...
// TEST Query Phrase

// 1. Ricerca di una frase esatta
// Per effettuare una ricerca si utilizza l'oggetto IFileQueryRegistry
$fileQueryRegistry = new IFileQueryRegistry();
$fileQueryRegistry->setQuery('session clustering', 'body',
IFileQuery::MATCH_REQUIRED);
    // definizione dell'ordinamento
    $lucene->setSort('key', SORT_STRING, SORT_DESC);
    // chiamata al metodo di ricerca Phrase
    $result = $lucene->queryPhrase($fileQueryRegistry);
    // stampa dei risultati
    printResult("Ricerca per frase esatta", $result);

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
```

Esempio 4 - Utilizzo della ricerca: Fuzzy

Interfaccia LUCENE:

File di esempio:

IFile_Ex09_SearchFuzzy.php

Interfaccia MYSQLI:

File di esempio:

IFile_Mysql_Ex09_SearchFuzzy.php

```
...
// TEST Query Fuzzy

// 1. Ricerca di un termine fuzzy
// Per effettuare una ricerca si utilizza l'oggetto IFileQueryRegistry
$fileQueryRegistry = new IFileQueryRegistry();
$fileQueryRegistry->setQuery('zend', 'body', null);
    // definizione dell'ordinamento
    $lucene->setSort('key', SORT_STRING, SORT_DESC);
    // chiamata al metodo di ricerca Fuzzy
    $result = $lucene->queryFuzzy($fileQueryRegistry);
    // stampa dei risultati
    printResult("Ricerca un termine fuzzy (body:zend)", $result);

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
```

Esempio 5 – Utilizzo della ricerca: Boolean

Interfaccia LUCENE:

File di esempio:

IFile_Ex10_SearchBoolean.php

Interfaccia MYSQLI:

File di esempio:

IFile_Mysql_Ex10_SearchBoolean.php

```
...
// TEST Query Boolean

// 1. Ricerca di gruppi di termini in AND
// Per effettuare una ricerca si utilizzano piu' oggetti1
IFileQueryRegistry
// Primo gruppo
$fileQueryRegistry_01 = new IFileQueryRegistry();
$fileQueryRegistry_01->setQuery('prova', 'body',
IFileQuery::MATCH_REQUIRED);
$fileQueryRegistry_01->setQuery('txt', 'body',
IFileQuery::MATCH_REQUIRED);
// Secondo gruppo
$fileQueryRegistry_02 = new IFileQueryRegistry();
$fileQueryRegistry_02->setQuery('doc', 'body',
IFileQuery::MATCH_REQUIRED);
// Definizione delle query
$fileQueryRegistry = new IFileQueryRegistry();
$fileQueryRegistry->setQuery($fileQueryRegistry_01, null,
IFileQuery::MATCH_REQUIRED);
$fileQueryRegistry->setQuery($fileQueryRegistry_02, null,
IFileQuery::MATCH_OPTIONAL);
// definizione dell'ordinamento
$file->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Boolean
$result = $file->queryBoolean($fileQueryRegistry);
// stampa dei risultati
printResult("Ricerca un termine boolean", $result);

// 2. Ricerca di gruppi di termini in OR
// Per effettuare una ricerca si utilizzano piu' oggetti1
IFileQueryRegistry
// Primo gruppo
$fileQueryRegistry_01 = new IFileQueryRegistry();
$fileQueryRegistry_01->setQuery('prova', 'body',
IFileQuery::MATCH_REQUIRED);
$fileQueryRegistry_01->setQuery('txt', 'body',
IFileQuery::MATCH_REQUIRED);
// Secondo gruppo
$fileQueryRegistry_02 = new IFileQueryRegistry();
```

```
$ifileQueryRegistry_02->setQuery('doc', 'body',
IFileQuery::MATCH_REQUIRED);
// Definizione delle query
$ifileQueryRegistry = new IFileQueryRegistry();
$ifileQueryRegistry->setQuery($ifileQueryRegistry_01, null,
IFileQuery::MATCH_OPTIONAL);
$ifileQueryRegistry->setQuery($ifileQueryRegistry_02, null,
IFileQuery::MATCH_OPTIONAL);
// definizione dell'ordinamento
$ifile->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Boolean
$result = $ifile->queryBoolean($ifileQueryRegistry);
// stampa dei risultati
printResult("Ricerca un termine boolean", $result);

// 3. Ricerca di gruppi di termini in OR
// Per effettuare una ricerca si utilizzano piu' oggetti1
IFileQueryRegistry
// Primo gruppo
$ifileQueryRegistry_01 = new IFileQueryRegistry();
$ifileQueryRegistry_01->setQuery('prova', 'body',
IFileQuery::MATCH_REQUIRED);
$ifileQueryRegistry_01->setQuery('txt', 'body',
IFileQuery::MATCH_REQUIRED);
// Secondo gruppo
$ifileQueryRegistry_02 = new IFileQueryRegistry();
$ifileQueryRegistry_02->setQuery('doc', 'body',
IFileQuery::MATCH_REQUIRED);
// Definizione delle query
$ifileQueryRegistry = new IFileQueryRegistry();
$ifileQueryRegistry->setQuery($ifileQueryRegistry_01, null,
IFileQuery::MATCH_REQUIRED);
$ifileQueryRegistry->setQuery($ifileQueryRegistry_02, null,
IFileQuery::MATCH_PROHIBITEN);
// definizione dell'ordinamento
$ifile->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Boolean
$result = $ifile->queryBoolean($ifileQueryRegistry);
// stampa dei risultati
printResult("3 Ricerca un termine boolean", $result);

// 4. Ricerca di gruppi di termini in OR
// Per effettuare una ricerca si utilizzano piu' oggetti1
IFileQueryRegistry
// Primo gruppo
$ifileQueryRegistry_01 = new IFileQueryRegistry();
$ifileQueryRegistry_01->setQuery('prova', 'body',
IFileQuery::MATCH_REQUIRED);
$ifileQueryRegistry_01->setQuery('txt', 'body',
IFileQuery::MATCH_REQUIRED);
// Secondo gruppo
$ifileQueryRegistry_02 = new IFileQueryRegistry();
$ifileQueryRegistry_02->setQuery('doc', 'body',
IFileQuery::MATCH_REQUIRED);
// Definizione delle query
$ifileQueryRegistry = new IFileQueryRegistry();
$ifileQueryRegistry->setQuery($ifileQueryRegistry_01, null,
IFileQuery::MATCH_OPTIONAL);
```

```
$ifileQueryRegistry->setQuery($ifileQueryRegistry_02, null,
IFileQuery::MATCH_PROHIBITEN);
// definizione dell'ordinamento
$ifile->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Boolean
$result = $ifile->queryBoolean($ifileQueryRegistry);
// stampa dei risultati
printResult("4 Ricerca un termine boolean", $result);

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
```

Esempio 6 – Utilizzo della ricerca: Wildcard

Interfaccia LUCENE:

File di esempio:

IFile_Ex11_SearchWildcard.php

Interfaccia MYSQLI:

File di esempio:

IFile_Mysql_Ex11_SearchWildcard.php

```
...
// TEST Query Wildcard

// 1. Ricerca di termini con l'utilizzo di Wildcard
// Per effettuare una ricerca si utilizza l'oggetto IFileQueryRegistry
$ifileQueryRegistry = new IFileQueryRegistry();
$ifileQueryRegistry->setQuery('cluster*', 'body',
IFileQuery::MATCH_REQUIRED);
// definizione dell'ordinamento
$ifile->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Wildcard
$result = $ifile->queryWildcard($ifileQueryRegistry);
// stampa dei risultati
printResult("Ricerca un termine Wildcard", $result);

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}
```

Esempio 7 – Utilizzo della ricerca: Range

Interfaccia LUCENE:

File di esempio:

IFile_Ex12_SearchRange.php

Interfaccia MySQLI:

File di esempio:

Nessuno. Metodo non implementato.

```
...
// TEST Query Range

// 1. Ricerca di termini con l'utilizzo di Range
// Per effettuare una ricerca si utilizza l'oggetto IFileQueryRegistry
$fileQueryRegistry = new IFileQueryRegistry();
$fileQueryRegistry->setQuery('a', 'body');
$fileQueryRegistry->setQuery('c', 'body');
// definizione dell'ordinamento
$file->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Range
$result = $file->queryRange($fileQueryRegistry);
// stampa dei risultati
printResult("Ricerca un termine Range", $result);

} catch (Exception $e) {
    echo "Errore: ".$e->getMessage();
}

}
```

ESTENDERE IFILE

- Realizzare un nuovo Adapter

REALIZZARE UN NUOVO ADAPTER

IFile permette la realizzazione dei nuovi Adapter, o meglio classi, per la parserizzazione di file oltre a quelli già presenti.

Di seguito verranno descritte le linee guida per implementare un nuovo Adapter.

IMPLEMENTAZIONE

Per poter implementare un nuovo adapter bastano tre semplici passi:

Passo 1 – Creazione file PHP:

Creare un nuovo file PHP nella cartella:

/adapter/

con il nome del file così strutturato:

Adapter_Search_Lucene_Document_<EXT>.php

Dove al valore “<EXT>” si dovrà inserire l’estensione, in MAIUSCOLO, del file che si vuole parserizzare. Ad esempio se si vuole creare un Adapter per i file RTF basterà creare un nuovo file vuoto PHP:

```
/adapter/ Adapter_Search_Lucene_Document_RTF.php
```

Passo 2 – La classe:

Dopo aver creato il file andrà inclusa la classe astratta:

```
/adapter/ Adapter_Search_Lucene_Document_Abstarct.php
```

e crea la classe con lo stesso nome del file (senza .php) che estende la classe sopradescritta.

```
<?php
/**
 * IFile framework
 *
 * @category   IndexingFile
 * @package    ifile
 * @subpackage adapter
 * @author     xxx
 * @copyright
 * @license    GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999
 * @version    1.0 Adapter_Search_Lucene_Document_RTF.php 2011-01-12 11:41:09
 */

/** Adapter_Search_Lucene_Document_Abstract */
require_once 'Adapter_Search_Lucene_Document_Abstract.php';

/**
 * Adapter per il recupero del contenuto dei file RTF
 *
 * @category   IndexingFile
 * @package    ifile
 * @subpackage adapter
 * @author     xxx
 * @copyright
 * @license    GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999
 */
class Adapter_Search_Lucene_Document_RTF extends
Adapter_Search_Lucene_Document_Abstract
{
    public function __construct() {
        parent::__construct();
    }
}

...

...
```

Passo 3 – Implementare il metodo loadParserFile:

A questo punto basterà implementare il metodo dell'interfaccia `Adapter_Search_Lucene_Document_Interface` "**loadParseFile**".
Il metodo deve restituire un oggetto `Zend_Search_Lucene_Document`.

Va ricordato che l'oggetto `Zend_Search_Lucene_Document` deve ritornare un field "body" valorizzato per non mandare in eccezione il processo di indicizzazione.

```
/**
 * Ritorna un oggetto Zend_Search_Lucene_Document
 *
 * Implementa il metodo dell'interfaccia
 * Adapter_Search_Lucene_Document_Interface
 *
 * @return Zend_Search_Lucene_Document
 */
public function loadParserFile()
{
    // creazione del Bean
    $this->indexValues = new LuceneDataIndexBean();
    // recupero il contenuto dal file
    $data = mioMetodoRecuperaContenutoFile();

    if ($data === '') {
        require_once 'Adapter_Search_Lucene_Exception.php';
        throw new Adapter_Search_Lucene_Exception(' Empty body ');
    }

    $this->indexValues->setBody($data);

    return $this->indexValues->getLuceneDocument();
}
```

RIFERIMENTI ON-LINE

<http://ifile.isapp.it>

<http://forum.isapp.it>

<http://sourceforge.net/projects/indexfile/>

<http://lucene.apache.org/java/docs/index.html>

<http://www.getopt.org/luke/>

<http://framework.zend.com/manual/en/zend.search.lucene.html>

<http://framework.zend.com/apidoc/core/>

<http://dev.mysql.com/doc/refman/5.5/en/fulltext-search.html>

<http://www.exif.org/Exif2-2.PDF>

<http://www.foolabs.com/xpdf/>

<http://www.winfield.demon.nl/>

<http://linux.die.net/man/5/xpdfrc>