

---

# IFile - INTRODUZIONE ALL'UTILIZZO

**Gennaio 2012**

---

## 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.....	8
Encoding .....	10
XPDF.....	10
PHP .....	10
Estensioni PHP .....	10
Funzioni PHP .....	11
Il “server check” di IFile .....	11
Installare IFile .....	12
Configurare IFile .....	12
Fields e custom fields .....	20
Default Fields.....	20
Tag ID3 degli MP3.....	22
Tag Exif dei JPEG .....	22
Utilizzare le interfacce di IFile.....	24
Lucene Interface .....	25
MySqli Inteface .....	25
IFile_Indexing_Interface .....	26
Metodo per aggiungere un campo personalizzato .....	26
Metodo per aggiungere un documento all'indice.....	27
Metodo per definire l'autocommit dei documenti .....	28

---

Metodo per committare i documenti nell'indice .....	28
Metodo per recuperare del numero di documenti indicizzati .....	28
Metodo per cancellare un documento dall'indice .....	29
Metodo per recuperare tutti i documento dall'indice .....	29
Metodo per recuperare dei campi personalizzati .....	30
Metodo per recuperare il campo di default di ricerca .....	30
Metodo per recuperare un documento dall'indice.....	30
Metodo per recuperare i fields dell'indice .....	30
Metodo per recuperare le info sul file da indicizzare.....	31
Metodo per recuperare il limite massimo di risultati .....	31
Metodo per verificare se esistono documenti cancellati .....	31
Metodo per verificare l'esistenza di un termine .....	32
Metodo per verificare se un documento e' cancellato .....	32
Metodo per recuperare il numero di documenti .....	32
Metodo per ottimizzare l'indice .....	33
Metodo per la ricerca dei termini .....	33
Metodo per la ricerca BOOLEAN dei termini.....	33
Metodo per la ricerca FUZZY dei termini .....	34
Metodo per la ricerca con la sintassi di lucene /mysql .....	34
Metodo per la ricerca di frasi .....	35
Metodo per la ricerca in un range di termini .....	35
Metodo per la ricerca con caratteri jolly (wildcard).....	36
Metodo per settare il campo di default di ricerca.....	36
Metodo per settare il file da indicizzare.....	37
Metodo per settare il limite massimo di risultati.....	37
Metodo per settare il tipo di ordinamento dei risultati .....	37
Metodo per ripristinare un documento cancellato.....	38
Esempi pratici .....	38
Indicizzare un singolo documento.....	40
Indicizzazione multipla .....	43
Indicizzazione manuale.....	46
Utilizzare i Custom Fields.....	49
Lavorare con l'indice.....	52

---

Ricerca all'interno di un indice.....	54
Estendere IFile .....	62
Realizzare un nuovo Adapter .....	62
Implementazione.....	62
Riferimenti On-line .....	65

---

## 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 e verrai introdotto ai concetti principali di IFile, descritti anche con l'aiuto di 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.

**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 di 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 Management System) che hanno la necessità di effettuare ricerche nei contenuti degli articoli, salvati nella maggior parte dei casi nella banca dati del sistema, e nei documenti in formato DOC, PDF, EXCEL che la piattaforma CMS gestisce dove sono presenti la maggior parte dei contenuti ma che sono difficilmente ricercabili.

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)
- **M**oving **P**icture Expert Group-1/2 Audio Layer **3** (.mp3)
- **J**oint **P**hotographic **E**xperts **G**roup (.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.

***I requisiti necessari sono:***

### *ZEND FRAMEWORK*

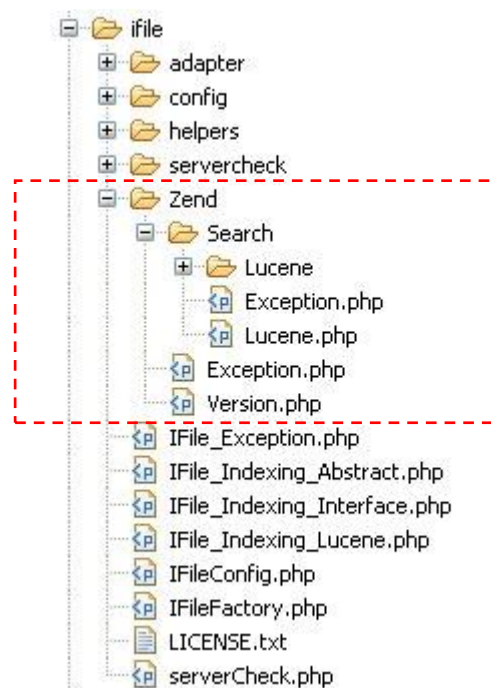
Deve essere presente ed utilizzabile la libreria di Zend: ***Zend\_Search\_Lucene***.





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.

### *XPDF*

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

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

### *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

### **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 libreria 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 libreria 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

**FUNZIONI PHP**

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

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 aggiornata alla versione utile per il corretto funzionamento della libreria, verrà evidenziata in rosso (*vedi figura*). 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
<b>Zend Framework</b>					
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 MySQLi Interface	<a href="http://www.zend.com">http://www.zend.com</a>
<b>Encoding</b>					
PCRE	OK	Not defined	PCRE unicode support is enabled in PHP	Used by Zend Search Lucene Framework	<a href="http://www.php.net/manual/en/book.pcre.php">http://www.php.net/manual/en/book.pcre.php</a>
<b>XPDF</b>					
XPDF Binaries File	Executable	CHMOD 0755	For more information visit web site	Used only for PDF file parser	<a href="http://www.foo labs.com">http://www.foo labs.com</a>
<b>PHP</b>					
PHP Version	OK	Version 5.1.0 or later	Version installed is 5.2.4	Not defined	<a href="http://www.php.net">http://www.php.net</a>
<b>Extension</b>					
libxml	OK	Version 2.6.0 or later	Not check version	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	<a href="http://www.php.net/manual/en/book.libxml.php">http://www.php.net/manual/en/book.libxml.php</a>
dom	OK	Not defined	Version installed is 20031129	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	<a href="http://www.php.net/manual/en/book.dom.php">http://www.php.net/manual/en/book.dom.php</a>
SimpleXML	OK	Not defined	Version installed is 0.1	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	<a href="http://www.php.net/manual/en/book.simplexml.php">http://www.php.net/manual/en/book.simplexml.php</a>
mbstring	OK	Not defined	Not check version	Used by Zend Search Lucene	<a href="http://www.php.net/manual/en/book.mbstring.php">http://www.php.net/manual/en/book.mbstring.php</a>
zip	OK	Not defined	Version installed is 1.4.0	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	<a href="http://www.php.net/manual/en/class.ziparchive.php">http://www.php.net/manual/en/class.ziparchive.php</a>
zlib	OK	Version 1.0.9 or later	Version installed is 1.1	Used for Office Open Xml (OOXML) and OpenDocument (ODF) file parser	<a href="http://www.php.net/manual/en/book.zlib.php">http://www.php.net/manual/en/book.zlib.php</a>
iconv	OK	Not defined	Not check version	Used by Zend Search Lucene	<a href="http://www.php.net/manual/en/book.iconv.php">http://www.php.net/manual/en/book.iconv.php</a>
id3	OK	Version 0.1 or later	Version installed is 0.1	Used for MP3 file parser	<a href="http://www.php.net/manual/en/book.id3.php">http://www.php.net/manual/en/book.id3.php</a>
mysqli	OK	Not defined	Version installed is 0.1	Used only for MySQL Interface	<a href="http://www.php.net/manual/en/book.mysqli.php">http://www.php.net/manual/en/book.mysqli.php</a>
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 ilias Exp \$	Used for JPG file parser	<a href="http://www.php.net/manual/en/book.exif.php">http://www.php.net/manual/en/book.exif.php</a>
<b>Function</b>					
popen	OK	Not defined	Function exists	Used only for PDF file parser	<a href="http://www.php.net/manual/en/function.popen.php">http://www.php.net/manual/en/function.popen.php</a>
strip_tags	OK	Not defined	Function exists	Used only for XML file parser	<a href="http://php.net/manual/en/function.strip-tags.php">http://php.net/manual/en/function.strip-tags.php</a>

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 includere la classe presente nel file /servercheck/LuceneServerCheck.php

Il codice sottostante descrive il modo per recuperare la lista:

```

/** 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);

```

Con l'array degli oggetti ora è possibile realizzare una propria presentazione del report di controllo.

## INSTALLARE IFILE

Per installare IFile basta scaricare la libreria dal sito

<https://sourceforge.net/projects/indexfile/> e decomprimerla all'interno della propria cartella di lavoro.

La libreria a questo punto è pronta per essere utilizzata.

## CONFIGURARE IFILE

Questa è la parte più importante per l'utilizzo corretto della libreria. Una configurazione errata potrebbe provarvi problemi nel processo di

---

indicizzazione. Pertanto questo è il capito più importante e andrebbe letto con molta attenzione.

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>...</table-name>
  <timelimit>...</timelimit>
  <memorylimit>...</memorylimit>
  <duplicate>...</duplicate>
  <encoding>...</encoding>
  <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 una 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 field (*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 è 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>
```

### **timelimit [opzionale] {1}**

Il tag `<timelimit>` 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>
```

### **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>
```

### **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 non verifica l'esistenza del documento all'interno dell'indice.

La verifica viene fatta sul 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
- CP1256
- Windows-1252

E' possibile estendere i charset andando a definire nel file XSD altri tipi.



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

Va fatta molta attenzione al tipo di encoding che viene settato in funzione del tipo di Analyzer utilizzato.

Per le esperienze avute, in caso non si abbia una vera conoscenza del tipo di encoding del documento (ovvero nella maggior parte dei casi), conviene non configurare il tag.

Di fatto nel caso si utilizzi un Analyzer di tipo UTF-8 e si hanno file codificati in UTF-8 non andrebbe configurato l'encoding dato che le librerie di Zend Lucene tenterebbero di forzare la codifica e pertanto si potrebbe avere, 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 che il contenuto del documento non è stato tokenizzato verificate anche 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.



---

Esempio:

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

### **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, pertanto 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/TokenFilter/EnglishStemmer.php
</custom-default>
```

### 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	Keyword	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	Keyword	Obbligatorio solo se viene indicizzato un file mediante il processo automatico

	duplicate) allora si potrebbero avere più chiavi con lo stesso valore		<i>di IFile</i>
path	Percorso relativo o assoluto in funzione della configurazione della root	UnIndexed	Obbligatorio <i>solo se viene indicizzato un file mediante il processo automatico di IFile</i>
filename	Percorso assoluto del file indicizzato	UnIndexed	Obbligatorio <i>solo se viene indicizzato un file mediante il processo automatico di IFile</i>
extensionfile	Estensione del file indicizzato	UnIndexed	Obbligatorio <i>solo se viene indicizzato un file mediante il processo automatico di IFile</i>
body	Contenuto del documento da indicizzare	UnStored	Obbligatorio
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

### *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	Keyword	Facoltativo	<b>X</b>

	dell'immagine			
Width	Larghezza in pixel dell'immagine	Keyword	Facoltativo	<b>X</b>
IsColor	Flag che identifica se l'immagini e' a colori: <ul style="list-style-type: none"> <li>• 0 -&gt; no</li> <li>• 1 -&gt; si</li> </ul>	Keyword	Facoltativo	
ApertureFNumber	Obiettivo	Keyword	Facoltativo	
UserComment	Commento dell'utente	Text	Facoltativo	<b>X</b>
ImageDescription	Descrizione dell'immagine	Text	Facoltativo	<b>X</b>
Orientation	Orientamento dell'immagine: <ul style="list-style-type: none"> <li>• Left</li> <li>• Right</li> <li>• Top</li> <li>• Bottom</li> <li>• reserved</li> </ul>	Keyword	Facoltativo	
Make	Macchina utilizzata	Keyword	Facoltativo	<b>X</b>
Model	Modello della macchina	Keyword	Facoltativo	<b>X</b>
Software	Software installato o utilizzato per la generazione dell'immagine	Keyword	Facoltativo	<b>X</b>
Copyright	Copyright dell'immagine	UnIndexed	Facoltativo	
GPSLatitude	Informazioni della latitudine nel formato gradi/primi/secondi	Keyword	Facoltativo	
GPSLatitudeGoogle	Latitudine nel formato di Google Map	UnIndexed	Facoltativo	
GPSLongitude	Informazioni della longitudine nel formato gradi/primi/secondi	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"> <li>• Auto exposure</li> <li>• Manual exposure</li> <li>• Auto bracket</li> <li>• None</li> <li>• reserved</li> </ul>	Keyword	Facoltativo	
ExposureTime	Tempo di esposizione	Keyword	Facoltativo	
SceneCaptureType	Tipo di scena <ul style="list-style-type: none"> <li>• Standard</li> <li>• Landscape</li> <li>• Portrait</li> <li>• Night scene</li> <li>• reserved</li> </ul>	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 di quest'ultimo verrà sovrascritto.

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 dei tipi di Analyzer e dalla quantità di filtri utilizzati.
- Tempi di risposta durante le ricerche molto lenti in funzione dei tipi di Analyzer e dalla quantità di filtri utilizzati.

### *MYSQLI INTEFACE*

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
```

<b>ft boolean syntax</b>	+ -><()~*:""€
<b>ft max word len</b>	84
<b>ft min word len</b>	3
<b>ft query expansion limit</b>	20
<b>ft stopword file</b>	(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"> <li>• FIELD_TYPE_KEYWORD</li> <li>• FIELD_TYPE_UNINDEXED</li> <li>• FIELD_TYPE_BINARY</li> <li>• FIELD_TYPE_TEXT</li> <li>• FIELD_TYPE_UNSTORED</li> </ul> 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 le tabelle 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 le tabelle 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 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.

Metodo:

## array getAllDocument ([boolean \$deleted = false])

Descrizione dei parametri di input :

Parametro	Descrizione	Tipo	Default
\$deleted	Se settato a TRUE ritorna anche i documenti marcati come cancellati	boolean	False

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 <b>TRUE</b> 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 field (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 (ovvero 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).  
Può 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	String	

	come field di default nelle ricerche se il field è omesso		
--	---	--	--

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"> <li>• SORT_REGULAR</li> <li>• SORT_NUMERIC</li> <li>• SORT_STRING</li> </ul>	Integer	SORT_REGULAR
\$order	Ordinamento ascendente o discendente: <ul style="list-style-type: none"> <li>• SORT_ASC</li> <li>• SORT_DESC</li> </ul>	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.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-->
  <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
$conconnection = @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
    $ifile = $IFileFactory->getIFileIndexing('mysqli', $conconnection);

    ...

```

### **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
$ifile->setIndexFile($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();
}
?>

```

### Interfaccia LUCENE:

File di esempio:

**IFile\_Ex01\_IndexOneFile.php**

Risultato indicizzazione:

The screenshot shows the Luke - Lucene Index Toolbox v 0.9.9 interface. The main window displays the following information:

- Index name: C:\xampp\htdocs\IFile\file\_index
- Number of fields: 10
- Number of documents: 1
- Number of terms: 702
- Has deletions? / Optimized?: No / Yes
- Last modified: Mon Aug 29 15:07:02 CEST 2011
- Index version: 4e5b8ef3
- 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\_2 (Mon Aug 29 15:07:02 CEST 2011)
- Current commit user data: N/A

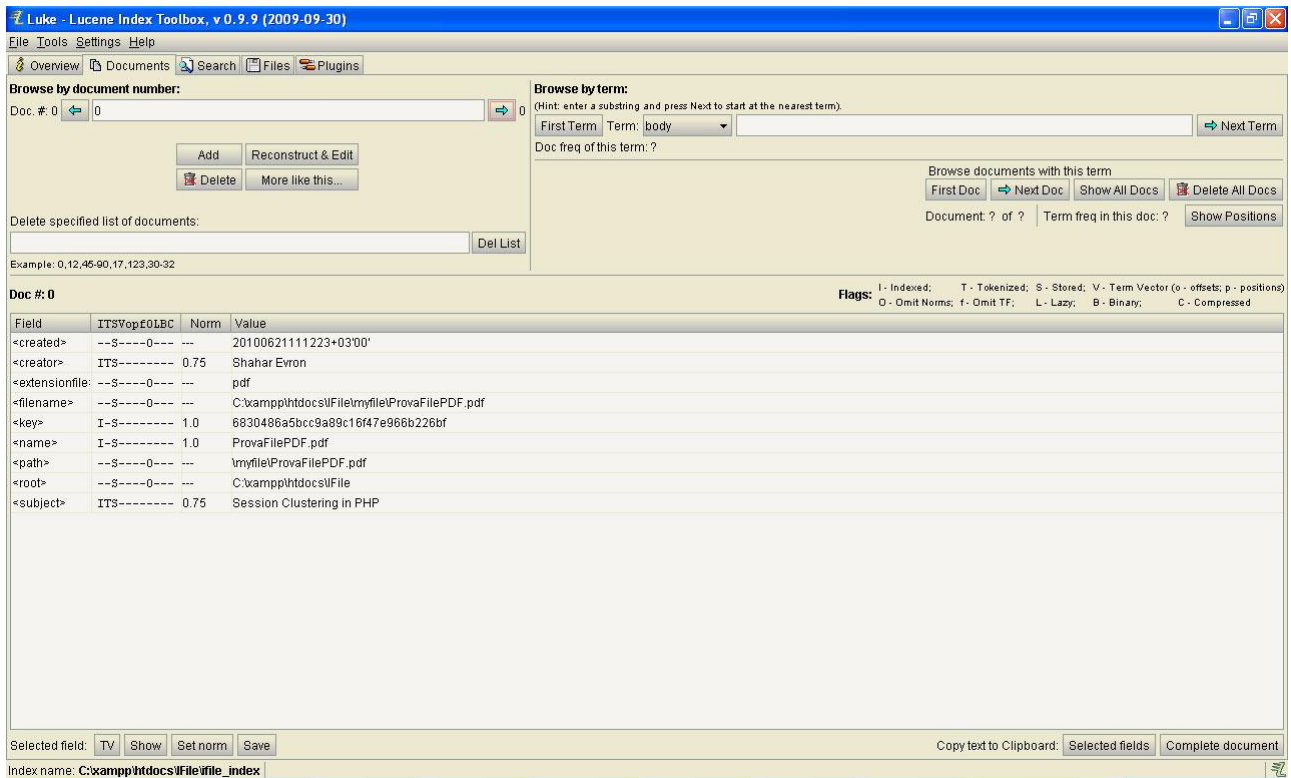
Below the statistics, there is a section for selecting fields and viewing top terms. The "Available fields and term counts per field" table is as follows:

Name	Term count	%
body	696	99,15 %
created	0	0,00 %
creator	2	0,28 %
extensionfile	0	0,00 %
filename	0	0,00 %
key	1	0,14 %
name	1	0,14 %
path	0	0,00 %
root	0	0,00 %
subject	2	0,28 %

The "Top ranking terms" table is as follows:

No	Rank	Field	Text
1	1	<body>	abbreviated
2	1	<body>	abcd
3	1	<body>	ability
4	1	<body>	accept
5	1	<body>	acceptable
6	1	<body>	access
7	1	<body>	active
8	1	<body>	activity
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>	adjusted
16	1	<body>	advantages
17	1	<body>	affinity
18	1	<body>	algorithm
19	1	<body>	amounts
20	1	<body>	appending

(Figura 1)



(Figura 2)

## Interfaccia MYSQLI:

File di esempio:

**IFile\_Mysql\_Ex01\_IndexOneFile.php**

Risultato indicizzazione:

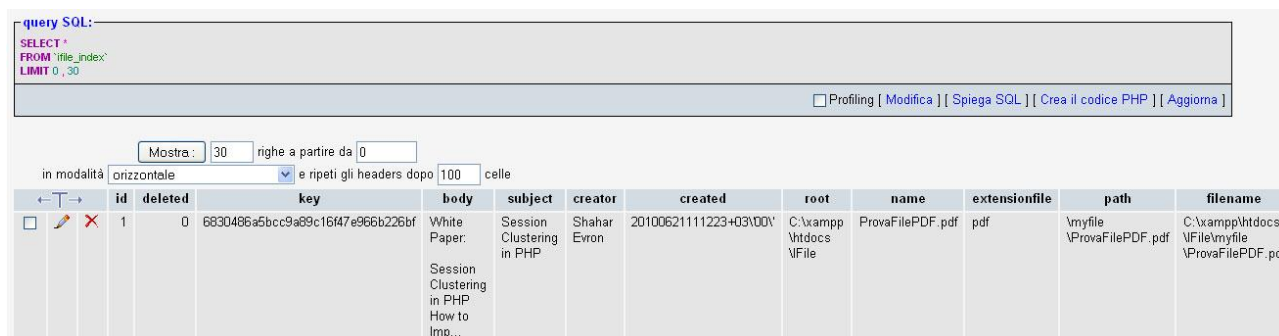
Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione
<input type="checkbox"/> id	int(22)			No		auto_increment	
<input type="checkbox"/> deleted	tinyint(4)			No	0		
<input type="checkbox"/> key	varchar(32)	latin1_general_ci		Sì	NULL		
<input type="checkbox"/> body	longtext	latin1_general_ci		No			
<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: 0					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 0			33,716
creator	FULLTEXT	Nessuno		creator	Dimensione riga 0			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.



The screenshot shows a database query interface. At the top, there is a text area containing the following SQL query:

```
query SQL:
SELECT *
FROM 'file_index'
LIMIT 0, 30
```

Below the query area, there are several controls: a checkbox for 'Profiling', and buttons for 'Modifica', 'Spiega SQL', 'Crea il codice PHP', and 'Aggiorna'. Below these are input fields for 'Mostra: 30 righe a partire da 0' and 'e ripeti gli headers dopo 100 celle'. The main part of the screenshot is a table with the following columns: id, deleted, key, body, subject, creator, created, root, name, extensionfile, path, and filename. The table contains one row of data:

id	deleted	key	body	subject	creator	created	root	name	extensionfile	path	filename
1	0	6830486a5bcc9a89c16f47e966b226bf	White Paper: Session Clustering in PHP	Session Clustering in PHP	Shahar Evron	20100621111223+03:00\	C:\xampp\htdocs\myfile	ProvaFilePDF.pdf	pdf	\myfile	C:\xampp\htdocs\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();
}
?>

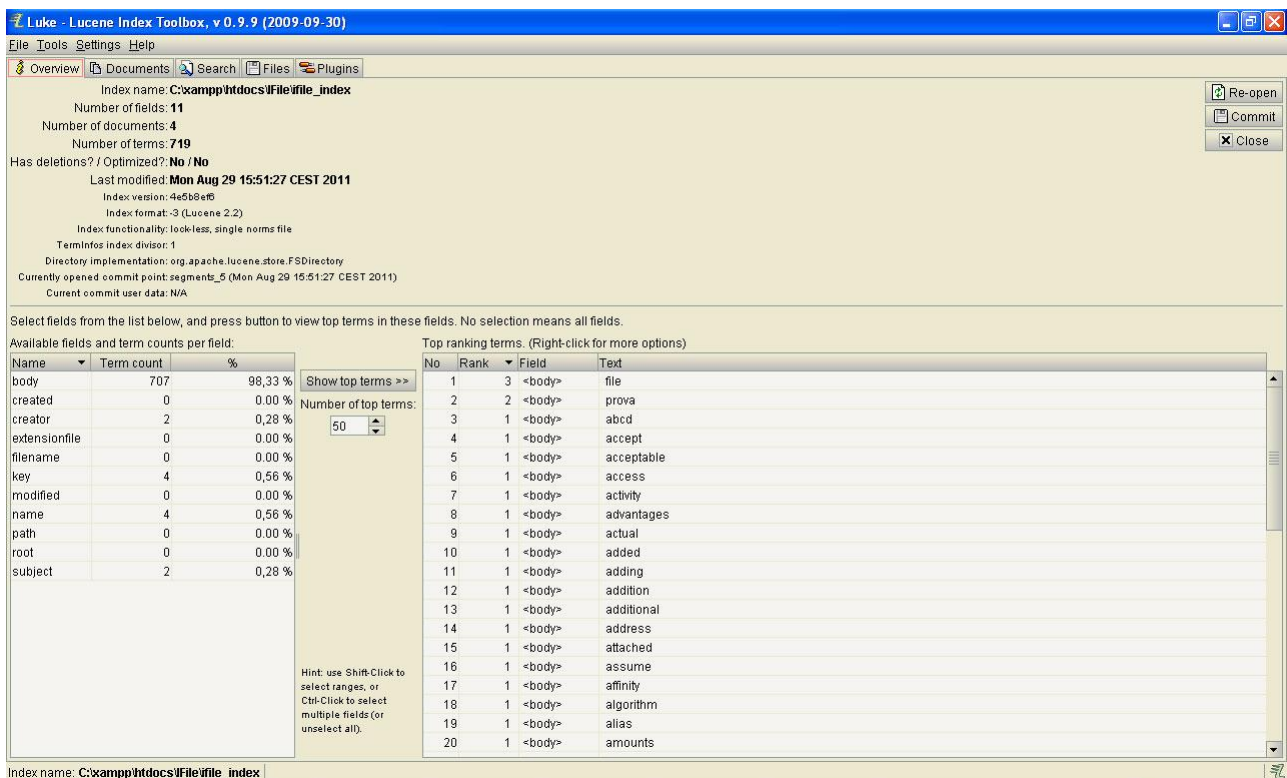
```

### Interfaccia LUCENE:

File di esempio:

**IFile\_Ex02\_MultipleFile.php**

Risultato indicizzazione:



(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	ae7fd1122810c12ed26625c0b4af683
<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---		bt
<filename>	--S----0---		C:\xampp\htdocs\file\myfile\ProvaFileTXT.bt
<key>	I-S-----	1.0	69bd4caa95a1df74f0b25544a627df0
<modified>	--S----0---		19/07/2011 15:40:10.
<name>	I-S-----	1.0	ProvaFileTXT.bt
<path>	--S----0---		\myfile\ProvaFileTXT.bt
<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		Si	NULL		
<input type="checkbox"/> body	longtext	latin1_general_ci		No			
<input type="checkbox"/> subject	text	latin1_general_ci		Si	NULL		
<input type="checkbox"/> creator	text	latin1_general_ci		Si	NULL		
<input type="checkbox"/> created	text	latin1_general_ci		Si	NULL		
<input type="checkbox"/> root	text	latin1_general_ci		Si	NULL		
<input type="checkbox"/> name	text	latin1_general_ci		Si	NULL		
<input type="checkbox"/> extensionfile	text	latin1_general_ci		Si	NULL		
<input type="checkbox"/> path	text	latin1_general_ci		Si	NULL		
<input type="checkbox"/> filename	text	latin1_general_ci		Si	NULL		
<input type="checkbox"/> modified	text	latin1_general_ci		Si	NULL		

Indici: 0					Spazio utilizzato			Statistiche righe		
Nome chiave	Tipo	Cardinalità	Azione	Campo	Tipo	Utilizzo	Istruzioni	Valore		
PRIMARY	PRIMARY	4		id	Dati	34,644 B	Formato	dinamico		
body	FULLTEXT	Nessuno		body	Indice	27,648 B	Collation	latin1_general_ci		
key	FULLTEXT	Nessuno		key	Totale	62,292 B	Righe	4		
subject	FULLTEXT	Nessuno		subject	Lunghezza riga 0			8,661		
creator	FULLTEXT	Nessuno		creator	Dimensione riga 0			15,573 B		
name	FULLTEXT	Nessuno		name	Prossimo Autoindex			5		
Crea un indice su 1 columns <input type="button" value="Esegui"/>								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	6830486a5bcc9a89c1647e966b226bf	White Paper: Session Clustering in PHP	Session Clustering in PHP	Shahar Evron	20100621111223+03V00V	C:\xampp\htdocs\IFile	ProvaFilePDF.pdf	pdf	\myfile\ProvaFilePDF.pdf	C:\xampp\htdocs\IFile\myfile\ProvaFilePDF.pdf	NULL
<input type="checkbox"/>	2	0	4b127a63360f98d791e71e0b7b4517a	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	ae7fd1122810c12ed26625c0b4afb683	menutype name alias link topmenu 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	69bd4caa95a1df740b26544a627df70	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/ifile/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
$file->addDocument($doc);
// salva il documento nell'indice
$file->commit();

echo "Documento Manuale: indicizzato correttamente<br>";

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

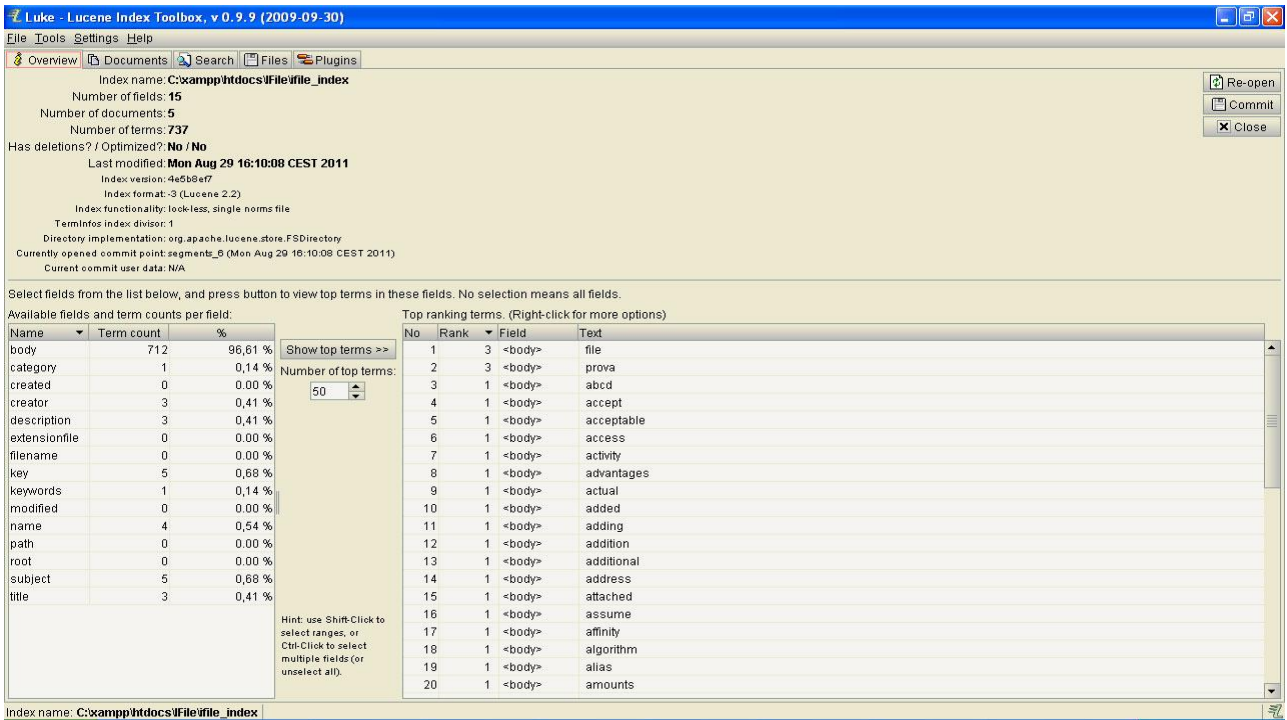
```

### ***Interfaccia LUCENE:***

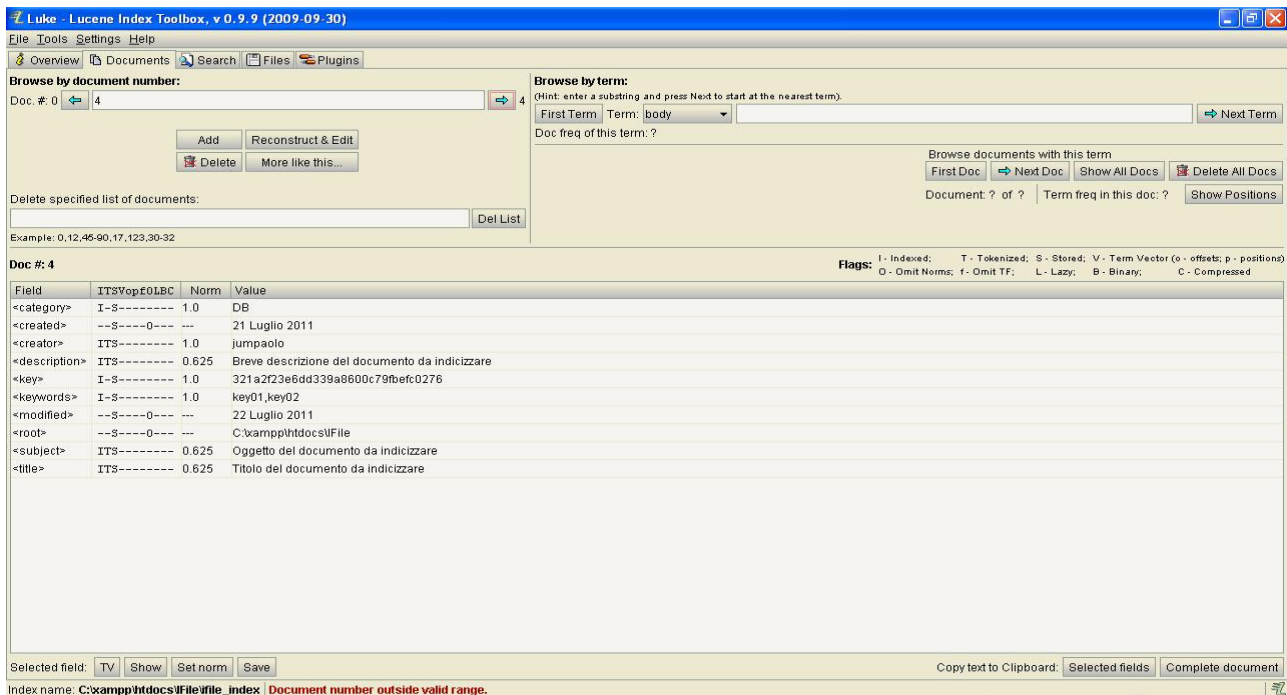
*File di esempio:*

**IFile\_Ex03\_IndexManual.php**

*Risultato indicizzazione:*



(Figura 1)



(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 0	7,004
creator	FULLTEXT	Nessuno		creator			Dimensione riga 0	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	321a2f23e6dd339a6600c79b6efc0276	Prova testo da indicizzare manualmente. Il t...	Oggetto del documento da indicizzare	jumpaolo	21 Luglio 2011	C:\xampp\htdocs\NFile	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();
}
?>

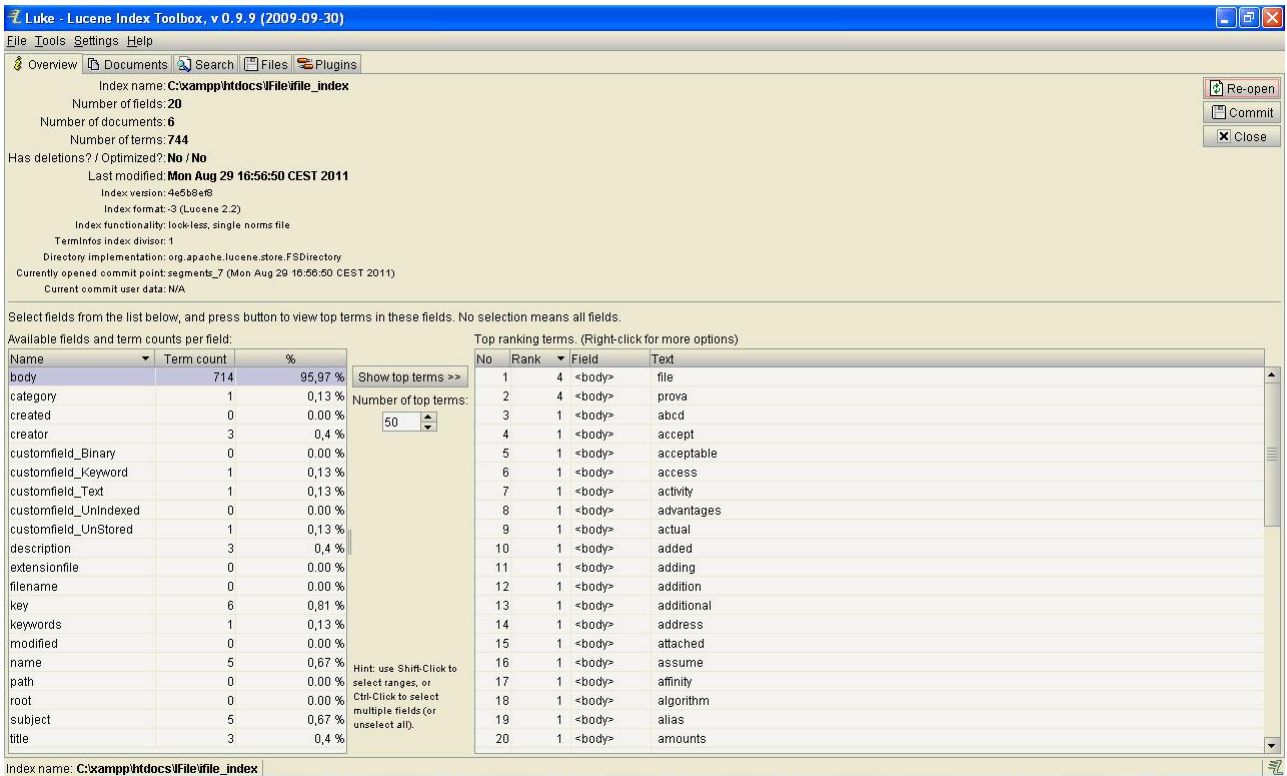
```

### ***Interfaccia LUCENE:***

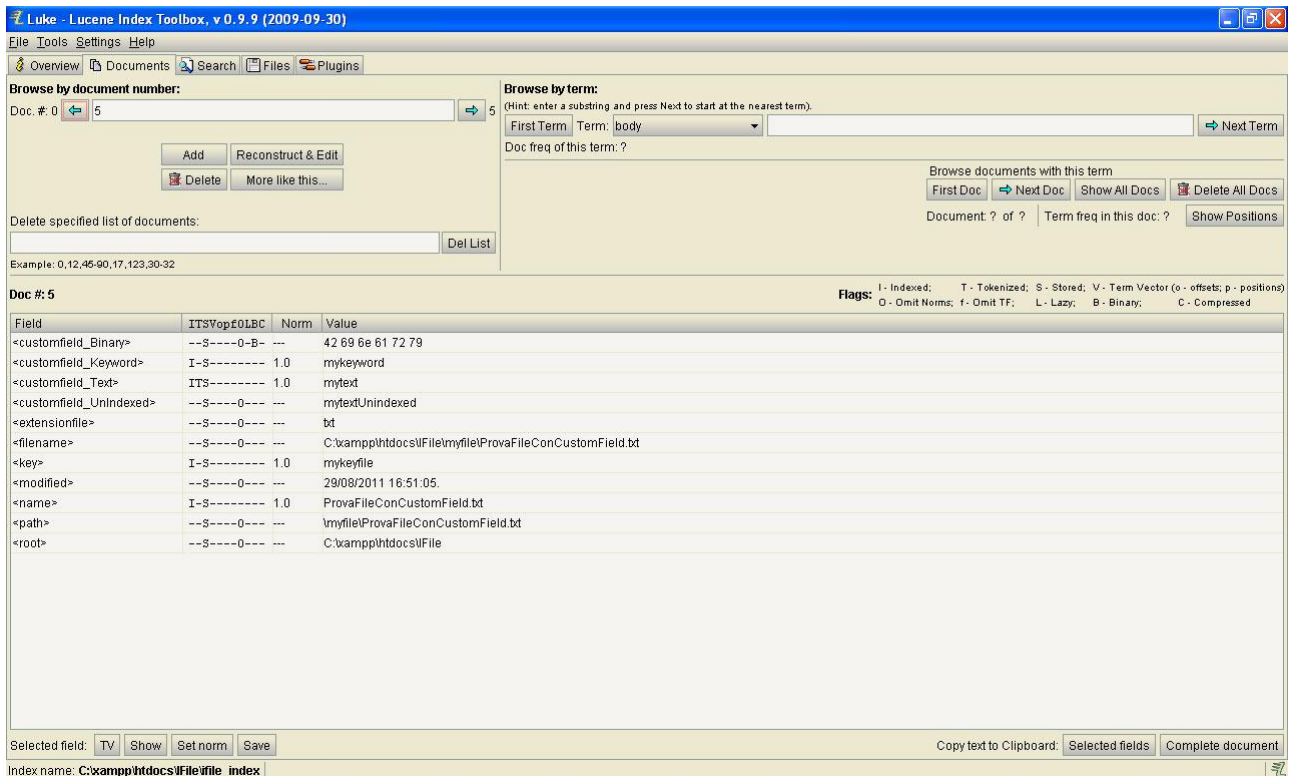
*File di esempio:*

**IFile\_Ex04\_CustomFields.php**

*Risultato indicizzazione:*



(Figura 1)



(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 0			5,896
creator	FULLTEXT	Nessuno		creator	Dimensione riga 0			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 1 columns

(Figura 3)

id	deleted	key	body	subject	creator	created	root	name	extensionfile	path	filename	modified	title	description	keywords	category	customfield_keyword	customfield_text	customfield_unstored	customfield_unindexed	customfield_binary
1	0	mykeyfile	prima file con custom field 100	NULL	NULL	NULL	C:\xampp\htdocs\IFile	PrimoFileConCustomField.txt	txt	\\myfile	PrimoFileConCustomField.txt	29/08/2011 16:51:05	NULL	NULL	NULL	NULL	mykeyword	mytext	mydataUnstored	mydataUnindexed	mydataBinary

### 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 PARSER
// Per l'utilizzo del metodo di ricerca PARSER 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-termini
$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
    $ifileQueryRegistry = new IFileQueryRegistry();
    $ifileQueryRegistry->setQuery('session clustering', 'body',
IFileQuery::MATCH_REQUIRED);
    // definizione dell'ordinamento
    $lucene->setSort('key', SORT_STRING, SORT_DESC);
    // chiamata al metodo di ricerca Phrase
    $result = $lucene->queryPhrase($ifileQueryRegistry);
    // 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
$ifileQueryRegistry = new IFileQueryRegistry();
$ifileQueryRegistry->setQuery('zend', 'body', null);
    // definizione dell'ordinamento
    $ifile->setSort('key', SORT_STRING, SORT_DESC);
    // chiamata al metodo di ricerca Fuzzy
    $result = $ifile->queryFuzzy($ifileQueryRegistry);
    // 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();
$fileQueryRegistry_02->setQuery('doc', 'body',
IFileQuery::MATCH_REQUIRED);
// Definizione delle query
$fileQueryRegistry = new IFileQueryRegistry();
$fileQueryRegistry->setQuery($fileQueryRegistry_01, null,
IFileQuery::MATCH_OPTIONAL);
$fileQueryRegistry->setQuery($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);

    // 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
$fileQueryRegistry = new IFileQueryRegistry();
$fileQueryRegistry->setQuery('cluster*', 'body',
IFileQuery::MATCH_REQUIRED);
// definizione dell'ordinamento
$file->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Wildcard
$result = $file->queryWildcard($fileQueryRegistry);
// 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');
```

---

```
$ifileQueryRegistry->setQuery('c', 'body');
// definizione dell'ordinamento
$ifile->setSort('key', SORT_STRING, SORT_DESC);
// chiamata al metodo di ricerca Range
$result = $ifile->queryRange($ifileQueryRegistry);
// 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
 */

/** Adatpter_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 Adatpter\_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
 * Adatpter_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://sourceforge.net/projects/indexfile/files/Document/1.1>

<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>