|  |
| --- |
| Advent Software, Inc. |
| Tamale Server Refactor Investigation |
| Summarized by Tamale Developers |

|  |
| --- |
| Advent Software, Inc.  12/26/2013 |

Contents

[Lucene 2](#_Toc375904839)

[Version 2](#_Toc375904840)

[Document Organization 2](#_Toc375904841)

[Contents for Entity 2](#_Toc375904842)

[Contents for Contact 2](#_Toc375904843)

[Contents for Entry 2](#_Toc375904844)

[Contents for Notification 3](#_Toc375904845)

[Index Organization 3](#_Toc375904846)

[How Index Is Setup 3](#_Toc375904847)

[How Index Is Updated 3](#_Toc375904848)

[How Index Is Searched 4](#_Toc375904849)

[How Index & Search Performance Is Measured 5](#_Toc375904850)

[Overall Class Diagram 5](#_Toc375904851)

[Refactor Suggestion 5](#_Toc375904852)

# Lucene

## Version

Lucene core 2.4.1 is used in Tamale. Latest version is 4.6.0.

## Document Organization

Search engine only indexes entity, contact, entry, and notification. They all created as a Lucene document via TDocument.class; typically following fields are included in a document.

|  |  |  |
| --- | --- | --- |
|  | **Store** | **Tokenized** |
| **Contents** | YES | YES |
| **EntityRefs** | YES | NO |
| **ID** | YES | NO |
| **SEQNUM** | YES | NO |
| **DATE** | YES | NO |
| **TYPE** | YES | NO |

### Contents for Entity

Includes following metadata, stored in a single field: name, short name, entity type name, alias, corporate email, corporate address, corporate phone, corporate fax, corporate officer names.

### Contents for Contact

Includes following metadata, stored in a single field: entity content, business address, business fax, business phone, company, home address, home address, home phone, job title, mobile phone, primary email, secondary email, IM handle.

### Contents for Entry

Items are stored in multi value fields in entry content.

#### Common

Entry type, primary metadata (entity metadata: short name, long name; contact metadata: contact name, company name), secondary metadata (source metadata, submitter metadata).

#### Note & Event

Plain body and entry version title.

#### Attachment

Entry version title, file extension, entry version tile + file extension.

Search engine uses perl scripts to extra text in pdf/msg/doc/ppt/xls/html/rtf files and save into new files under /home/tsdata/search/txtxt, and add them to content.

### Contents for Notification

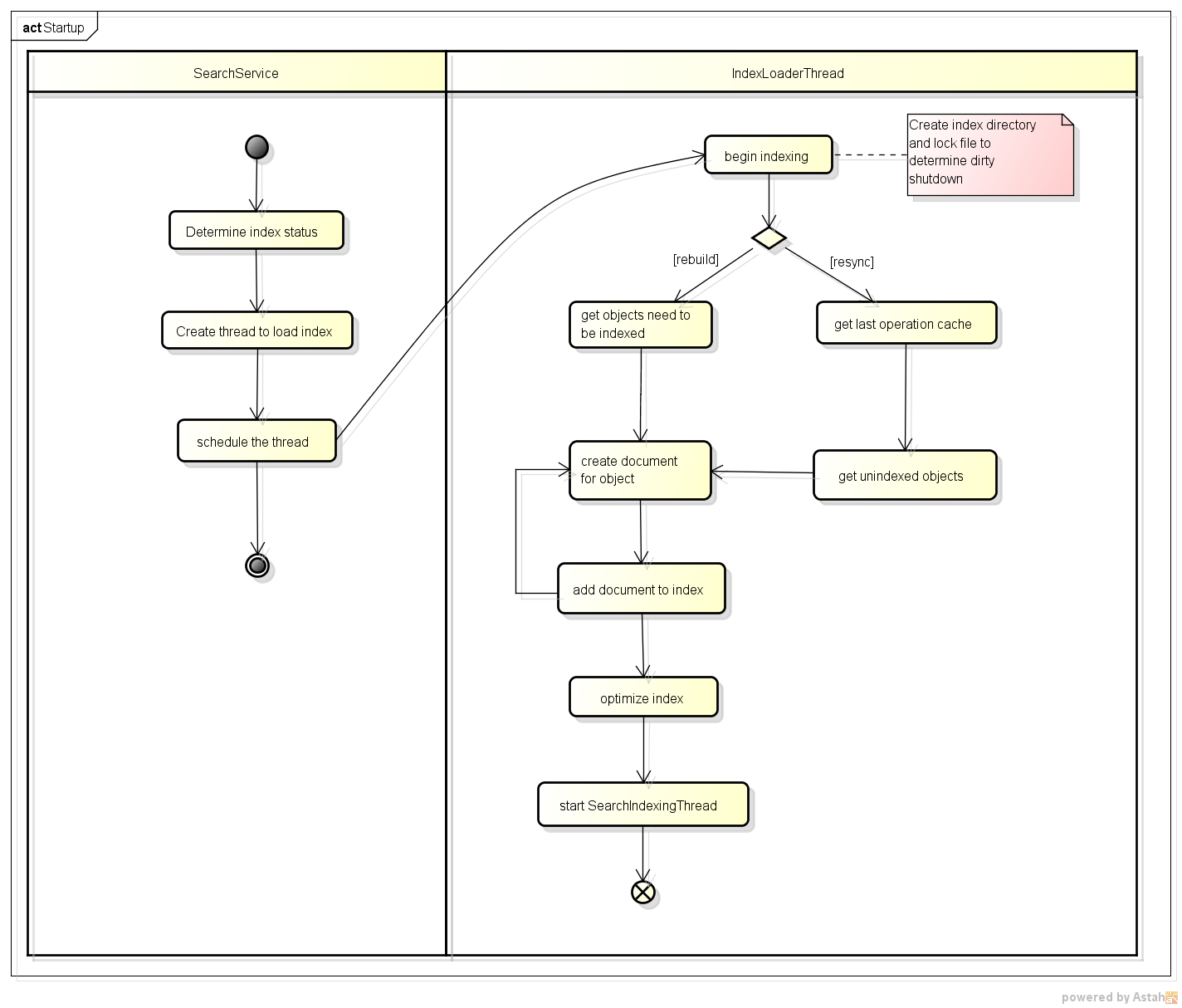
Items are stored in multi value fields in notification content. Include all entity names, primary metadata, notification file (same as attachment), subject, and source, attached file name, attached file extension, and attached file content.

## Index Organization

All documents are contained in a single index on file system with standard analyzer (both index and search) provided by Lucene, and max field length is set to 1,000,000.

## How Index Is Setup

When tamale server starts up, search engine will check the index status to determine whether to rebuild or resync the index. A simple activity diagram is as follows.

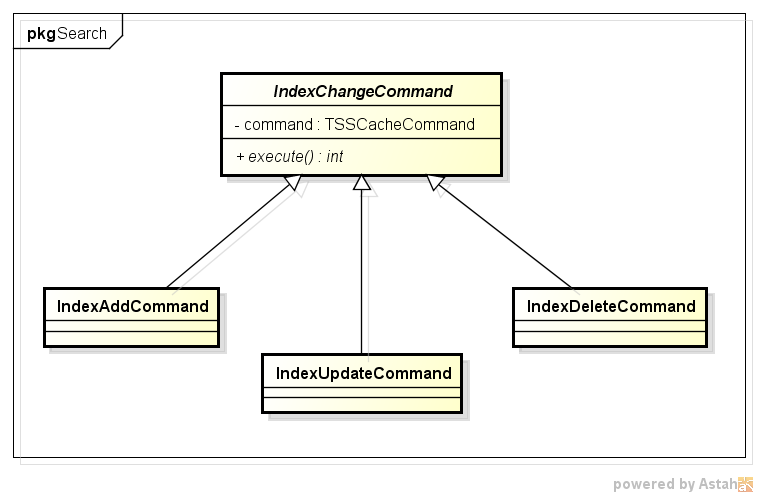


## How Index Is Updated

Sounds like a producer-consumer model, but is a little different from previous version. Queue manager maintain a command queue, and is responsible for adding add/update/delete command into it, and another thread dequeues command from queue manager and execute it. If there is any document added/updated/deleted, it will optimize the index.

Refer to following diagrams.





## How Index Is Searched

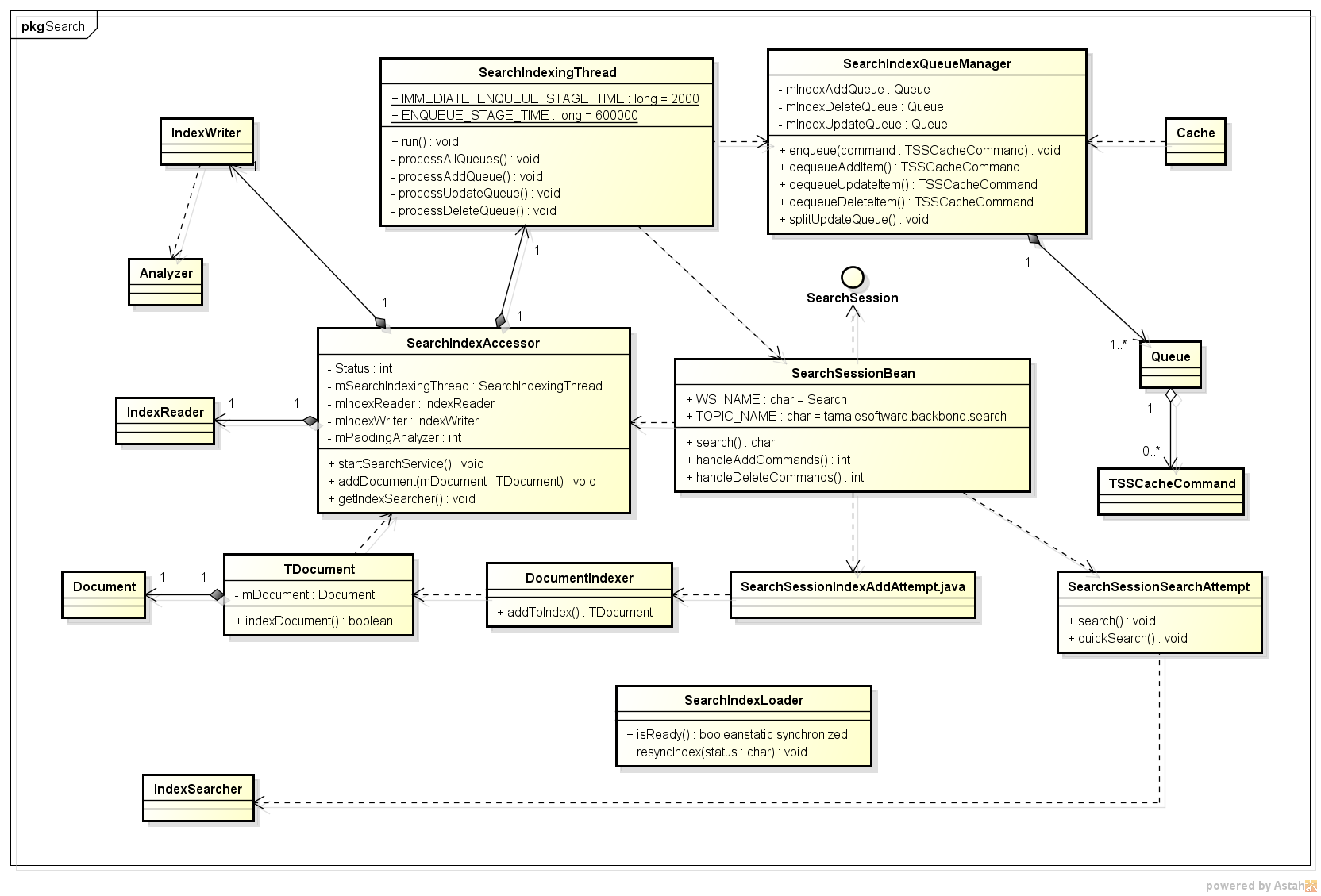
Before Lucene query is built, special character in input search string is replaced with space, since they may influence search result. These characters include: ‘+’, ‘-’, ‘&&’, ‘||’, ‘!’, ‘(’, ‘)’, ‘{’, ‘}’, ‘[’, ‘]’, ‘^’, ‘~’, ‘?’, ‘:’, ‘\’, ‘\*’.

Max hit count parameter is read from system configuration file. The built out query is a combination of QueryAnalyzer, BooleanQuery and TermQuery of Lucene with different SHOULD/MUST/SHOULD\_NOT/MUST\_NOT… occurs. QueryAnalyzer is performed on each token analyzed by java.util.StringTokenizer. TermQuery is performed on condition in search request, e.g. to search a specific type or not. All the queries are connected with BooleanQuery to build the final query.

## How Index & Search Performance Is Measured

This is simply done by logging the time which a specific operation is cost. But it introduces many non-business concerns.

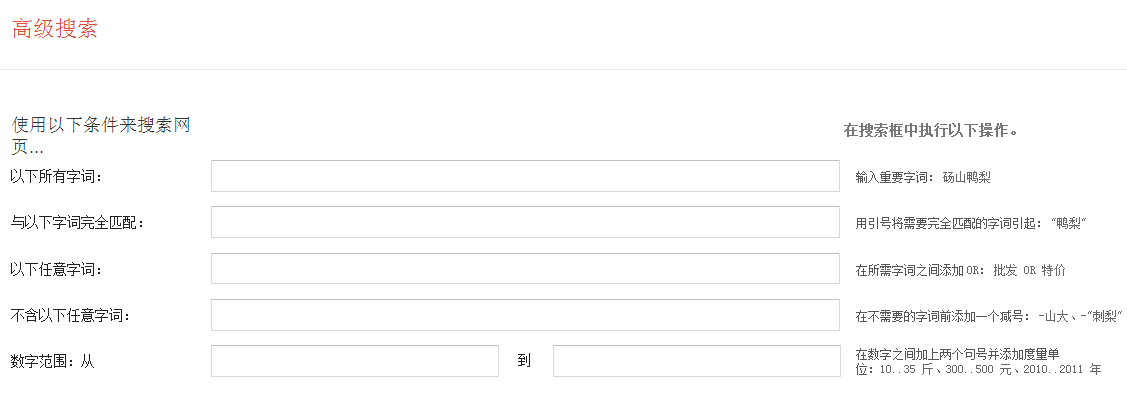
## Overall Class Diagram



## Refactor Suggestion

* SearchIndexAccessor and SearchService, SearchIndexingThread and SearchIndexQueueManager have too much coupling. Consider redefining their functionalities.
* Lots of deprecated Lucene methods are used. Consider upgrading Lucene to a newer stable version and avoid using deprecated APIs.
* Consider defining more specific Lucene document structure, since single TDocument class is too general. Also, consider adding more specific fields in document instead of putting all metadata in a single content string, just like what is done for entity right now.
* Consider supporting office 2007+ format attachment file. The recommended Lucene plugin Tika could be taken into account to extra text from attachments.
* Consider between single index and multi indexes. E.g. to create different index for different data types. And consider not optimizing index each time, since it is an expensive operation.
* Consider adding advanced search function to narrow down search scope.





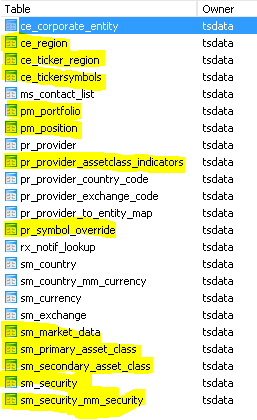
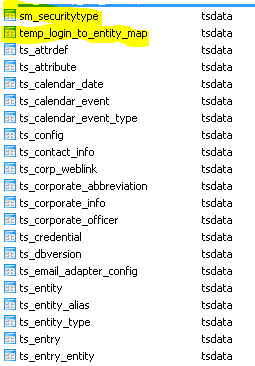
* Consider how to choose better analyzer for Chinese automatically.

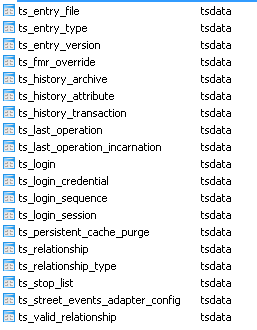
# Persistence

A high level database structure could be found here:



All tables in database are list as below:



The highlighted tables are not referenced in jboss.

## Refactor Suggestion

* Consider removing “ON UPDATE CASCADE ON DELETE CASCADE DEFERRABLE INITIALLY DEFERRED” related setting for better database practice.
* Compared with pure JDBC code, to meet our performance requirement, persistence framework is much simpler. Also we can do some optimization when batching creation, e.g. in data loader.
* History related data (ts\_history\_attribute, ts\_history\_transaction, ts\_history\_archive) is inserted but is not consumed anywhere. We can provide function to search history transaction in admin page.