

SPH  
IP Tracking

Version 1.7

Solution Architecture

Last Revised 23 January 2009

LUN ZHONG

TIM HILL

# Table of Contents

[Table of Contents 2](#_Toc220993836)

[Figures 4](#_Toc220993837)

[Document Details 5](#_Toc220993838)

[Revision History 5](#_Toc220993839)

[References 5](#_Toc220993840)

[Scope 6](#_Toc220993841)

[Location 6](#_Toc220993842)

[1 Introduction 7](#_Toc220993843)

[1.1 Project Overview 7](#_Toc220993844)

[2 Requirements and Constraints 8](#_Toc220993845)

[2.1 Crawling Requirements 8](#_Toc220993846)

[2.2 Search Requirements 9](#_Toc220993847)

[2.3 Product and Platform Details 11](#_Toc220993848)

[2.4 Custom Component Development 11](#_Toc220993849)

[2.5 External Product Dependencies 11](#_Toc220993850)

[2.6 Performance 12](#_Toc220993851)

[2.7 Assumptions 12](#_Toc220993852)

[3 System Design 13](#_Toc220993853)

[3.1 System Architecture 13](#_Toc220993854)

[3.2 Logical Architecture 14](#_Toc220993855)

[3.3 Crawler Management 15](#_Toc220993856)

[3.3.1 Start New Crawling Cycle 15](#_Toc220993857)

[3.3.2 Resume Crawling 17](#_Toc220993858)

[3.3.3 Stop/Pause Crawling 17](#_Toc220993859)

[3.3.4 Update Crawler Configuration 17](#_Toc220993860)

[3.3.5 Crawling Statistics 17](#_Toc220993861)

[3.4 Sources and Data Feeds 17](#_Toc220993862)

[3.5 Collections 18](#_Toc220993863)

[3.6 Document Processing 18](#_Toc220993864)

[3.6.1 NewslinkSimilarFinder 22](#_Toc220993865)

[3.6.2 SimilarityComparer 22](#_Toc220993866)

[3.7 Queries and Updates 23](#_Toc220993867)

[3.7.1 Queries 23](#_Toc220993868)

[3.7.2 Updates 23](#_Toc220993869)

[4 Environments and Infrastructure 25](#_Toc220993870)

[4.1 Remote Access 25](#_Toc220993871)

[4.2 Production Environment 25](#_Toc220993872)

[4.3 Production Infrastructure 25](#_Toc220993873)

[5 Open issues 27](#_Toc220993874)

# Figures

[Figure 1 System Architecture 3](#_Toc220473857)

[Figure 2 Logical Architecture 3](#_Toc220473858)

# Document Details

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Comment** | **Author** |
| 1.0 | 10-11-2008 | Initial draft | Lun Zhong |
| 1.1 | 18-11-2008 | Completed extra sections. Logical architecture changes. | Tim Hill |
| 1.2 | 20-11-2008 | Incorporated changes. | Lun Zhong |
| 1.3 | 21-11-2008 | Incorporated remaining changes and reviewed. | Tim Hill |
| 1.6 | 1-12-2008 | Incorporated SPH feedback | Tim Hill |
| 1.7 | 22-01-2009 | Incorporated the final pipeline/stages changes | Lun Zhong |

## References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Title** | **Author** | **Document** |
| 1.0 | 22-05-2008 | SoW | Alex Chua | *SPH Newslink Intellectual Property Policing Statement of work.doc* |
| 1.0 | 22-10-2008 | Requirements | Lun Zhong | *SPH\_IP\_Tracking\_BusinessRequirements\_v1.0.docx* |

## Scope

This document describes a mutually agreed-upon approach to the delivery and deployment of an ESP solution at Singapore Press Holdings Limited (SPH).

In no way does this document construct or dictate contractual obligations of either FAST or SPH. All such obligations are contained within the Professional Services Agreement (*PSA*) and Statement of Work (SOW) referenced above.

The deliverables of this project are as follows:

* FAST to provide *Enterprise Crawler* installation, production environment deployment, appropriate specifications, design, development and technical documentation.
* FAST to provide *ESP 5* installation, production environment deployment, appropriate specifications, design, development and technical documentation.

According to the SOW, FAST will also provide a measurable amount of technical support for the front-end application development.

This is a working document and should be updated throughout the project in order that it contains an accurate representation of the solution delivered.

## Location

All FAST project documentation is stored on the file system on FAST CFS and all customized code delivered by FAST is stored in CVS.

# Introduction

## Project Overview

Main board-listed **Singapore Press Holdings Limited** (**SPH**) is the leading media organization in Singapore, with one of the region's most advanced printing assets. It is licensed to publish 14 newspapers in four languages in Singapore and together these have a circulation of more than one million copies a day. SPH also publishes and distributes more than 90 periodicals in Singapore and the region. In addition, it owns a 40 percent stake in MediaCorp Press Pte. Ltd, which publishes free newspaper Today. Every day, SPH newspapers are read by 2.7 million individuals, or 83 percent of Singaporeans above 15 years old.

The focus of **IP Tracking** is to help SPH combat misuse of intellectual property by third parties, and ensures that their right to have content appropriately credited is respected. The backend (crawler and ESP) should be able to identify and log SPH text content on the Internet. The frontend application should provide general infringement reports and be a general tool for SPH staff investigating and searching for infringements.

The table below lists the key members in the project:

| **Company** | **Name** | **Role** | **Phone** | **Email** |
| --- | --- | --- | --- | --- |
| SPH | **Ben Lim** | SPH Project Manager | 63191758 | limcwb@sph.com.sg |
| SPH | **Xian Zhihai** | Developer | 63191023 | zhihai@sph.com.sg |
| SPH | **Cai Qing** | Developer | 63192354 | caiqing@sph.com.sg |
| FAST | **Cindy** | FAST Project Manager | +61 289183208 | cindysch@microsoft.com |
| FAST | **Lun Zhong** | FAST Technical Architect | 96477009 | lunz@microsoft.com |
| FAST | **Tim Hill** | FAST Engineer |  | thill@microsoft.com |

# Requirements and Constraints

This section captures crawling and search requirements in this project. It identifies ESP feature or ESP-related development activities needed to meet these requirements.

## Crawling Requirements

Table 1 Crawling requirements and how to fulfill them

| **#** | | **Requirement specification** | **Fulfillment** | **Comment** |
| --- | --- | --- | --- | --- |
| 2.1 | Able to start up & shut down the crawler. | | Command line & XML-RPC | Front end trigger to start, resume, stop crawling, refer to section 3.3. |
| 2.3 | Crawling configuration generated from database:   * Start URLs * Start URL Priority (Urgent/High/Medium(default)/Low) * Level[[1]](#footnote-3) (default: 4) * etc. | | Crawler configuration  Command line & XML-RPC | Front end application to create the configuration from the Candidate Site database and update the crawler. |
| 2.4 | Delete or add web sites at run time | | Command line & XML-RPC |  |
| 2.5 | Provide crawling cycle statistics. | | XML-RPC and cron job to collect statistics periodically. | Insert the statistics into the Candidate Site database |
| 2.6 | Relevant Content Detection: remove the “noisy” content, such as banner, navigation, etc. | | * Custom document processing pipeline. * Crawler configuration. | * Several content removal schemes will be tested. * The crawler can also follow links from non-html content, such as Flash and Javascript style links. |

## 

## Search Requirements

Table 2 Search requirements and how to fulfill them

| **#** | | **Requirement specification** | **Fulfillment** | **Comment** |
| --- | --- | --- | --- | --- |
| 3.1 | Content input:   * Suspected content crawled from public web sites. * Search must still be available while indexing. | | Supported by ESP |  |
| 3.2 | Update index: every crawling cycle | | Supported by ESP |  |
| 3.3 | Suspected Content Detection:   * Content mapping with *Newslink* ESP index * Detail Content matching to be specified by % | | Custom document processing pipeline. | % match will be a score from 0 to 100 (*similarityscore* in index profile). |
| 4.1 | Search suspected content by:   * Site Info (Web Site or Company Name) * Publication * Article Headline * Status   Order by:   * Web Site * Company Name * Publication * Article Headline * Match % | | Supported by FAST Search API and a series of custom document processing stages, including:   * NewslinkSimilarFinder * SimilarityComparer |  |
| 4.2 | Result:   * Web Site * Company Name * Publication * Article Headline * Source URL * Match % * Status: new(default), infringement, non-infringement, uncertain | | FAST Search API |  |
| 5.3 | Infringement Validation:   * Pop up both suspected content and *Newslink* content side by side * User can update the status, i.e. infringement to non-infringement and vice versa. | | FAST Search & Content API |  |

## Product and Platform Details

The following ESP components will be used for this deployment:

* ESP 5.2
* Enterprise Crawler 6.7

The platform for this deployment will be Sun Solaris 10 running on Sun Sparc hardware. The API will run on JDK 1.6 and the web server will be Sun Java System Web Server.

## Custom Component Development

The following ESP-related components/document processors will be developed for this solution:

* Customizations to existing extractor and vectorization stages
  + For improving the document vector sent to the NewslinkSimilarFinder.
* SPHPersonExtractor
  + Extraction of person names from the web pages crawled.
* SPHKeywordExtractor
  + Extraction of newspaper names from the web pages crawled.
* NGram
  + Compute the ngram data of the web pages crawled
* Spotsig
  + Compute the spotsig data of the web pages crawled
* NewslinkSimFinder
  + A high level comparison of a document against all *Newslink* documents to return a list of similar documents.
* SimScore
  + A low level (one at a time) comparison of each of the identified similar documents that gives a similarity score to each of the similar documents and puts the maximum score against the current document.
* SiteInfo
  + To add meta data to candidate documents to improve searching (e.g. company, siteid).

## External Product Dependencies

* *Newslink* will need to be reindexed. See section 3.6.2.
* Newslink will need to have its entity extraction method modified and its index profile and some configuration files changed.
* The existing Newslink ESP system must be able to handle a QPS of a rate equal to the document feeding rate of the IP Tracking ESP system[[2]](#footnote-4).
* The IP Tracker ESP system needs user access to the Candidate Site database.

## Performance

This ESP solution must meet the following performance criteria:

Table 1 Sizing requirements

| **Item** | **Values for 1st version** | **Values for 2nd version** |
| --- | --- | --- |
| **Crawler: Number of documents** | 20 Million |  |
| **Crawler: Average document size** | 10 KB |  |
| **Crawler: Total document size** | 250 GB |  |
| **Index: Number of documents** | 200,000 |  |
| **Estimated index size** | 1GB |  |
| **Indexing latency** | - |  |
| **Continuous or batch updates  (add, delete, update)** | Continuous |  |
| **Update rate** | - |  |
| **QPS Average** | << 1 QPS |  |
| **QPS Peak** | << 1 QPS |  |

In addition, the existing Newslink ESP system must be able to handle a QPS of a rate equal to the update rate of the IP Tracking ESP system.

## Assumptions

* There is a limitation of around 20million documents for a single crawler node.
* Crawler configuration updates may not take effect immediately but depends on the crawler status and what kind of changes.
* The QPS ability of the existing *Newslink* search is sufficient to handle the additional QPS requirements.
* Existing *Newslink* search is able to be reindexed during development, which may mean that search is unavailable for certain periods.
* Existing *Newslink* search must be available during development.
* Side-by-side comparison of newslink and web articles should be done via the source document, not the cached index version.

# System Design

## System Architecture

Figure 1 System Architecture shows the whole system architecture:



Figure 1 [System](#_Figures) Architecture

## Logical Architecture

Figure 2 Logical Architecture shows the logical architecture from an ESP perspective, including content sources, connectors, pipelines and search clients:



Figure 2 Logical Architecture

## Crawler Management

The crawler will crawl a list of websites, beginning at a list of start URIs, and following links recursively for one complete ‘cycle’. At the end of the cycle, the crawler will start over, rereading the list of start URIs. Existing pages will be recrawled for updates, new pages will be added to the index, and pages that have been removed or are not seen in several cycles will eventually be deleted from the index.

Table 4 Crawler XML-RPC API

| **Method** | **Parameter** | **Response** | **Description** |
| --- | --- | --- | --- |
| **CollectionAdd** | Config | Errcode, Message | This method adds/updates a crawler configuration |
| **CollectionDelete** | Collection | Errcode, Message | This method deletes the specified collection from the crawler database. |
| **CollectionSuspend** | Collection | Errcode, Message | This method suspends crawling of the specified collection. |
| **CollectionResume** | Collection | Errcode, Message | This method resumes crawling of a previously suspended collection. |
| **CollectionDeleteSite** | Collection, Site | Errcode, Message | This method issues a delete command of the specified site to the crawler, which will be implemented within the next crawling cycle. |
| **AddUris** | Collection, Urgent, Uris | Errcode, Message | This method adds one or more URIs to the front of their respective work queues. |
| **CollectionGetStatisticsXml** | Collection | XML | This method gets statistics for the collection. |
| **CollectionGetSiteStatisticsXml** | Collection,  Site | XML | This method gets statistics for a site in the collection. |

The subsections below detail the commands that will be run by the equivalent XML-RPC calls in the table above.

### Start New Crawling Cycle

To start a new crawling cycle:

1. Lock the candidate site info table in database
2. Stop statistics cron job
3. Run and collect crawler statistics
4. Generate and upload new crawler configuration
5. Start new crawling cycle with new crawler configuration
6. Start statistics cron job
7. Unlock the candidate site info table in database

To add/update a crawler configuration (step 4):

# crawleradmin -f <file>

To restart a crawler cycle (step 5):

# crawleradmin --refetch <collection>

### Resume Crawling

To resume crawling, just make the XML-RPC call or

# crawleradmin –s <collection>

### Stop/Pause Crawling

To stop/pause crawling, just make the XML-RPC call or

# crawleradmin –r <collection>

### Update Crawler Configuration

To update crawler configuration:

1. Lock the candidate site info table in database
2. Fetch/re-fetch the sites marked as Urgent immediately
3. Delete the sites marked as *Active=false*
4. Update crawler with a new crawler configuration
5. Unlock the candidate site info table in database

To force crawler to fetch sites immediately:

# crawleradmin --force –adduri <collection>:<site>  
# crawleradmin --force -addurifile <collection>:<file>

To delete a site:

# crawleradmin --deletesite <collection>:<web site>

### Crawling Statistics

The crawling statistics will be collected and inserted into database every hour.

Collection statistics for current cycle:

# crawleradmin --collstats <collection> --cycle

Collection statistics for all cycles:

# crawleradmin --collstats <collection> --cycle all

Site statistics for current cycle:

# crawleradmin --sitestats <collection>:<web site> --cycle

Site statistics for all cycles:

# crawleradmin --sitestats <collection>:<web site> --cycle all

## Sources and Data Feeds

There is only one data source: Singapore web sites

* Content format: HTML
* Access method: HTTP
* Seed list: provided by SPH (the front-end application provides a GUI interface to manage each piece of candidate site information).
* Crawling restrictions:

|  |  |
| --- | --- |
| **Option** | **Value** |
| Request rate | 60 secs |
| Refresh interval | 144000 mins/100 days |
| MIME type | text/html, text/plain |
| Refresh mode | Scratch |
| Crawl mode | Level |
| Max levels | 5 |
| Follow cross-site URIs | No |
| Focused language | English |
| Work Queue Priority | levels: 3 default: 2  start\_uri\_pri: 2  pop\_scheme: pri put\_scheme: include – insert a list of sites with *high* crawling priority into priority=1 section, a list of sites with *low* crawling priority into priority=3 section. |

Some of the above parameters can be adjusted accordingly after monitoring the actual crawling.

* An update will be manually triggered by the front-end application. See section 3.3.1.
* The request rate is the rate at which the crawler will fetch each web page *per webserver*. This setting is so as not to overload the webservers being crawled and has an impact on how long a complete crawl will take.

## Collections

There is only one collection: **localweb**

## Document Processing

Table 4 Document Processing Pipeline

| **Stage** | | **Type** | | **Description** | |
| --- | --- | --- | --- | --- | --- |
| **DocInit** | | Default | | Initialize document elements. Set size element to the size of the external document. | |
| **DocumentRetriever** | | Default | | Retrieve external document. If the data element is set, the external document is retrieved from this element. Alternatively the external document is retrieved from the URL supplied in the *getpath* element. | |
| **URLProcessor** | | Default | | The URL of the document (*url* attribute) and urls of redirects to and duplicates of the document are concatenated into the *urls* attribute. | |
| **FormatConverter** | | Default | | Detects the format of the data element (MIME type if found). | |
| **SimpleConverter** | | Default | | If data contains HTML or plain text, this stage copies the content into the html element. Plain text content is mapped to a simple HTML representation. | |
| **LanguageAndEncodingDetector** | | Default | | Automatic language and encoding detection. | |
| **LanguageDetectorTextParts** | | Default | | Automatically detect and assign language to a section of text. | |
| **EncodingNormalizer** | | Default | | Encoding is normalized to UTF-8. | |
| **FastHTMLParser** | | Default | | The HTML parser extracts visible content and initialize the body, title and metadata elements. It also extracts metadata from HTML header and place those in corresponding meta\_\* elements. | |
| **TeaserGenerator** | | Default | | The Teaser generator summarizing the document and creates a static teaser (document summary). | |
| **Tokenizer** | | Default | | Performs language-specific tokenization (normalization) including removal of special characters and accents. The Tokenization is configurable.  Typical input: title body  Typical output: elemtitle elembody | |
| **PersonExtractorWhiteListSpecific**  **PersonExtractorWhiteListAny**  **PersonExtractor1**  **PersonExtractor2** | | Modified | | Person name entity extraction.  Input: title, body  Output: personnames | |
| **SPHPersonExtractor** | | Modified | | Person name entity extraction based on SPH dictionary  Input: title, body  Output: personnames | |
| **CompanyExtractorWhiteListSpecific**  **CompanyExtractorWhiteListAny**  **CompanyExtractor1**  **CompanyExtractor2** | | Modified | | Company name entity extraction.  Input: title, body  Output: companies | |
| **LocationExtractorWhiteListSpecific**  **LocationExtractorWhiteListAny**  **LocationExtractor1** | | Modified | | Geographical location entity extraction.  Input: title, body  Output: locations | |
| **TimeExtractor** | | Default | | Date/time entity extraction.  Input: title, body  Output: time | |
| **MeasurementExtractor** | | Default | | Extracts entities that is numerically quantified, e.g., technical specifications or other types of measurements.  Input: title, body  Output: measurement | |
| **UppercaseExtractor** | | Default | | Extracts other entities, based on capitalization/punctuation heuristics.  Input: title, body  Output: uppercaseentities | |
| **SPHKeywordExtractor** | | New | | SPH newspaper name and special entity extraction.  Input: title, body  Output: newspapers | |
| **SPHEntitiesVectorizer** | | Modified | | Creates a document vector based on strings/entities extracted  Input: personnames\_raw,   companies\_raw,   locations\_raw,   time\_raw,  measurement\_raw,   uppercaseentities\_raw,   newspaper\_raw  Output: docvector | |
| **Vectorizer** | | Default | | Appends to the document vector a series of noun phrases used to distinguish the individual document. The resulting document vector will also include the entities included by the entity extractor.  Input: title, body  Output: docvector | |
| **ConceptSuppressor** | | Default | | Suppresses the set of concept strings, normally extracted to the docvector\_navigator, to avoid excessive memory usage for very large document collections. | |
| **SPH5gram** | | New | | Compute n-gram  Input: textattribute  Output: ngram | |
| **SPHSpotsig** | | New | | Compute Spotsig  Input: textattribute  Output: spotsig | |
| **SPHSimFinder** | | New | | Find similar documents from Newslink and store into similardoclist  Output: simfinder\_resultset | |
| **SPHSimScore** | | New | | Compute a similarity score by comparing the web page against each doc in similardoclist and store the result in candidates as (url,score) tuples. It also puts the maximum score in a separate field to allow user filtering on score. If the document is extremely short, then a comparison will not be made.  Input: simfinder\_resultset  Output: candidates  similarityscore | |
| **DropNonCandidates** | | New | | Drop documents for which the similarity score is below a certain threshold.  Input: similarityscore | |
| **SiteInfo** | | New | | Assign values from database to site info fields  Input: url  Output: site  sitecompany  starturi  crawlcycle  crawlpriority | |
| **Lemmatizer** | | Default | | Performs language-specific lemmatization for non-CJK languages. | |
| **DateTimeNormalizer**  **DateTimeSelector**  **MapperTransformer** | | Default | | These stages are used to normalize datetime elements, select date/time source for freshness boosting and mapping numeric values to corresponding data types in the index. | |
| **RankTuner** | | Default | | This stage handles rank tuning (Boost & Blocks) settings from the Search Business Center (SBC). | |
| **FIXMLGenerator** | | Default | | Generates the internal pre-index representation of the documents (FIXML). | |
| **RTSOutput** | | Default | | Output stage for routing to Indexer nodes. | |

### NewslinkSimilarFinder

This stage finds a number of similar Newslink documents to be compared in more detail in the SimilarityComparer stage. The number of potential similar documents will be limited to a customizable value, typically 100.

The quality of this stage depends on careful identification and extraction of key words and phrases (entities) in the documents. This is done by earlier \*Extractor\* stages.

SPH can provide the following list in order to improve entity extraction:

* Person names
* Company names
* Locations
* Other keywords: newspaper names and others specific to SPH.

1. *Newslink* index has to contain the same extractors and EntitiesVectorizer in order to match highly similar *Newslink* articles.

### SimilarityComparer

The similarity comparison is done for each Newslink document identified in the earlier SPHSimFinder stage. The flow is as follows:



The SimScore is a number between 0 and 100, equivalent to a percentage match.

## Queries and Updates

### Queries

ESP provides Java content and search APIs. To select suspected documents the query is similar to the following:

**and(string(“a b c”, mode=”simpleall”), status:string(“x y z”, mode=”or”))**

For example:

* List candidates for newslink article
  + **and(string(“newslinkID?1234”, mode=”phrase”, linguistics=”off”), similiarityscore:>60, status:string(“new updated”, mode=”or”))**
* Show cached version of document
  + **url:string(“http://ripoffnews.com/?id=1234”, mode=”phrase”, linguistincs=”off”)**
* Show all new candidates with a score above 80
  + **and(status:new, similarityscore:>80)**

In this way, the *List of Suspects* and *Cached HTML* screens can be generated.

### Updates

The buttons to update status will use the ESP Content API to submit a partial update of the document to the search engine.

An overview of how to run an update using the Content API is shown below:

IDocumentFeederFactory documentFeederFactory = null;

IDocumentFeeder documentFeeder = null;

collection = "sgweb";

/\* Set up feeder \*/

try {

documentFeederFactory = DocumentFeederFactory.newInstance();

documentFeeder = documentFeederFactory.createDocumentFeeder(collection, null);

} catch (FactoryException e) {

System.out.println("Could not create IDocumentFeeder " + e);

return;

}

/\* Create partial document with changes \*/

IDocument document = DocumentFactory.newDocument(**url**);

document.addElement(DocumentFactory.newString("status", **updatedStatus**));

/\* Submit: update existing document with new status field \*/

documentFeeder.updateDocument(document);

documentFeeder.waitForCompletion();

IDocumentFeederStatus status = documentFeeder.getStatusReport();

/\* Handle failed submits – eg retry/report \*/

handleErrors(status);

/\* Close connections \*/

documentFeeder.deactivate();

# Environments and Infrastructure

1. There are no dev or staging ESP environments available for this project. Changes to the system design may require down-time on the production IP Tracking system.

## Remote Access

Remote access is not available. SPH provide access points at SPH ITD office. Login details can be found in the appendix.

## Production Environment



Figure 3 Production Environment

## Production Infrastructure

| **Item** | **Comments** |
| --- | --- |
| **Platform** | Sun V490 |
| **OS** | Solaris 10 |
| **CPU** | 2 dual core CPUs |
| **RAM** | 8 GB |
| **Crawler Data Storage** | External 250 GB storage |
| **Index Data Storage** | Internal 144G storage |

FAST recommends a separate physical partition for the ESP index, with fast (preferably 15K RPM) disks. Using a dedicated part of the internal storage for the index and installation, and the external drives for the crawler data storage would be suitable.

FAST will document recommended housekeeping tasks to help prevent log files from taking up too much space.

# Open issues

* Crawler Management and Statistics

SPH has to integrate and experiment with the unofficial XML-RPC interface. If it does not work well an alternative solution is to use command line tools.

* Crawling Priority  
    
  The priority is defined in the include session of the crawler configuration. The default priority is 2 and we only need to list down the sites with *high* (priority=1) and *low* (priority=3) priority. We do not expect a big number of these due to crawler limitations. *Urgent* priority (requirement 2.3) is defined in a different parameter.
* Format Converter  
    
  There are different internet documents – text, pdf, word, excel, flash, and various multimedia. In order to support additional formats besides plain HTML, SPH can consider additional converters from FAST – FlashConverter, PDFConverter, SearchExportConverter, etc., refer to the *FAST ESP Configuration Guide* for more details. In the current IP Tracking scope, only HTML/text is supported.
* Similarity   
    
  SimilarityComparer parameters such as thresholds can be tuned by testing some sample data. As part of the implementation, FAST will do some tests on a training set of documents provided by SPH.
* *Newslink* search engine  
    
  Changes will need to be made to the existing *Newslink* search engine in order to enable entity extraction. These amendments and updates will need to be done so that there’s minimal downtime and impact on the existing production Newslink system.
* Display of cached text  
    
  Cached text will be in simple text format without paragraph boundaries and potentially not easily readable, however it needs to be provided in some form, in case the infringing content is removed or unavailable when the user manually compares the sites. The typical side-by-side comparison of documents should be done via the source URLs rather than cached text.

1. Index Profile

<?xml version="1.0"?>

<!DOCTYPE index-profile SYSTEM "index-profile-5.0.1.dtd">

<index-profile name="default">

<!-- Flat fields. -->

<field-list sort-xnear-stop-word-threshold="2E8">

<!-- Standard fields. -->

<field name="title" fullsort="yes" result="dynamic" fallback-ref="title" lemmatize="yes">

<vectorize default="10:0"/>

</field>

<field name="body" max-index-size="8192" max-result-size="1024" fallback-ref="teaser" result="dynamic" index="no" lemmatize="yes">

<vectorize default="5:5" alternative="{ko,zh,szh,tzh}:5:0"/>

</field>

<field name="teaser" index="no"/>

<field name="headings" tokenize="auto" lemmatize="yes"/>

<field name="description" element-name="meta\_description" tokenize="auto" result="no"/>

<field name="anchortext" tokenize="auto" result="no"/>

<field name="keywords" element-name="meta\_keywords" tokenize="auto" result="no"/>

<field name="contenttype" element-name="mime"/>

<field name="format" boundary-match="yes"/>

<field name="language"/>

<field name="languages" separator=";"/>

<field name="charset"/>

<field name="urls" index="no"/>

<field name="url" index="no"/>

<field name="domain" element-name="url.domain"/>

<field name="tld" element-name="url.tld" result="no"/>

<field name="path" element-name="url.path" result="no"/>

<field name="urlkeywords" element-name="url.keywords" result="no"/>

<field name="crawltime" type="datetime" fullsort="yes"/>

<field name="processingtime" type="datetime" fullsort="yes"/>

<field name="docdatetime" type="datetime" fullsort="yes"/>

<field name="size" type="int32" fullsort="yes"/>

<field name="docvector" index="no"/>

<!-- Entity extraction placeholders. -->

<field name="personnames" separator=";"/>

<field name="companies" separator=";"/>

<field name="locations" separator=";"/>

<field name="concepts" separator=";" element-name="docvector\_navigator"/>

<!-- Newslink Fields -->

<field name="similardocslist" separator=";"/>

<field name="candidates" separator=";"/>

<field name="similarityscore" type="float"/>

<field name="infringementid"/>

<field name="headline"/>

<field name="publication"/>

<field name="publicationdate"/>

<field name="status"/>

<!-- Site Info Fields -->

<field name="siteid"/>

<field name="sitecompany"/>

<field name="crawlpriority" type="uint32"/>

</field-list>

<!-- Scope fields. -->

<scope-field-list>

<scope-field name="xml" result="dynamic" lemmas="yes"/>

</scope-field-list>

<!-- Standard composite field. -->

<composite-field name="ip" default="yes" query-tokenize="auto" lemmas="yes">

<field-ref name="sitecompany" level="4" field-separation-length="256"/>

<field-ref name="urlkeywords" level="3" field-separation-length="256"/>

<field-ref name="domain" level="3" field-separation-length="256"/>

<field-ref name="title" level="2" field-separation-length="256"/>

<field-ref name="keywords" level="2" field-separation-length="256"/>

</composite-field>

<composite-field name="news" query-tokenize="auto" lemmas="yes">

<field-ref name="title" level="4" field-separation-length="256"/>

<field-ref name="body" level="1" field-separation-length="256"/>

<field-ref name="sitecompany" level="3" field-separation-length="256"/>

<field-ref name="urlkeywords" level="3" field-separation-length="256"/>

<field-ref name="keywords" level="3" field-separation-length="256"/>

<field-ref name="anchortext" type="external" level="5" field-separation-length="256"/>

</composite-field>

<!-- Rank profiles. -->

<rank-profile name="iptracking" rank-model="news" default="yes"

stop-word-threshold="2E6"

position-stop-word-threshold="2E7">

<authority weight="0"/>

<freshness weight="200" field-ref="crawltime" auto="yes"/>

<composite-rank composite-field-ref="ip">

<proximity weight="50" />

<context weight="50">

<field-weight field-ref="sitecompany" value="50"/>

<field-weight field-ref="urlkeywords" value="30"/>

<field-weight field-ref="domain" value="30"/>

<field-weight field-ref="title" value="10"/>

<field-weight field-ref="keywords" value="10"/>

</context>

</composite-rank>

</rank-profile>

<rank-profile name="news" rank-model="news" default="no"

stop-word-threshold="2E6"

position-stop-word-threshold="2E7">

<authority weight="50" field-ref="anchortext"/>

<freshness weight="10" field-ref="docdatetime" auto="yes"/>

<composite-rank composite-field-ref="news">

<proximity weight="50" />

<context weight="50">

<field-weight field-ref="title" value="50"/>

<field-weight field-ref="body" value="10"/>

<field-weight field-ref="sitecompany" value="40"/>

<field-weight field-ref="urlkeywords" value="40"/>

<field-weight field-ref="keywords" value="30"/>

</context>

</composite-rank>

</rank-profile>

<!-- Result processing specifications.-->

<result-specification>

<!-- Navigators. -->

<numeric-navigator name="crawldatenavigator" display="CrawlingDate" intervals="4" resolution="1440">

<field-ref name="crawltime"/>

<range-label type="first" format="Before %.10s"/>

<range-label type="middle" format="Between %.10s and %.10s"/>

<range-label type="last" format="%.10s or after"/>

</numeric-navigator>

<string-navigator name="domainnavigator" display="Domain">

<field-ref name="domain"/>

</string-navigator>

<string-navigator name="companynavigator" display="Company">

<field-ref name="sitecompany"/>

</string-navigator>

</result-specification>

</index-profile>

1. Crawler XML Configuration Template

<?xml version="1.0" encoding="utf-8"?>

<CrawlerConfig>

<DomainSpecification name="localweb">

<attrib name="accept\_compression" type="boolean"> yes </attrib>

<attrib name="allowed\_schemes" type="list-string">

<member> http </member>

<member> https </member>

</attrib>

<attrib name="allowed\_types" type="list-string">

<member> text/html </member>

<member> text/plain </member>

</attrib>

<section name="cachesize">

<attrib name="aliases" type="integer"> 1048576 </attrib>

<attrib name="pp" type="integer"> 1048576 </attrib>

<attrib name="pp\_pending" type="integer"> 131072 </attrib>

<attrib name="routetab" type="integer"> 1048576 </attrib>

</section>

<attrib name="check\_meta\_robots" type="boolean"> yes </attrib>

<section name="crawlmode">

<attrib name="fwdlinks" type="boolean"> no </attrib>

<attrib name="fwdredirects" type="boolean"> no </attrib>

<attrib name="mode" type="string"> LEVEL </attrib>

<attrib name="reset\_level" type="boolean"> yes </attrib>

</section>

<attrib name="csum\_cut\_off" type="integer"> 0 </attrib>

<attrib name="cut\_off" type="integer"> 5000000 </attrib>

<attrib name="dbswitch" type="integer"> 5 </attrib>

<attrib name="dbswitch\_delete" type="boolean"> no </attrib>

<attrib name="delay" type="real"> 5.0 </attrib>

<attrib name="domain\_clustering" type="boolean"> no </attrib>

<attrib name="enable\_flash" type="boolean"> no </attrib>

<attrib name="enforce\_delay\_per\_ip" type="boolean"> yes </attrib>

<attrib name="exclude\_exts" type="list-string">

<member> .jpg </member>

<member> .jpeg </member>

<member> .ico </member>

<member> .tif </member>

<member> .png </member>

<member> .bmp </member>

<member> .gif </member>

<member> .wmf </member>

<member> .avi </member>

<member> .mpg </member>

<member> .wmv </member>

<member> .wma </member>

<member> .ram </member>

<member> .asx </member>

<member> .asf </member>

<member> .mp3 </member>

<member> .wav </member>

<member> .ogg </member>

<member> .ra </member>

<member> .aac </member>

<member> .m4a </member>

<member> .zip </member>

<member> .gz </member>

<member> .vmarc </member>

<member> .z </member>

<member> .tar </member>

<member> .iso </member>

<member> .img </member>

<member> .rpm </member>

<member> .cab </member>

<member> .rar </member>

<member> .ace </member>

<member> .hqx </member>

<member> .swf </member>

<member> .exe </member>

<member> .java </member>

<member> .jar </member>

<member> .prz </member>

<member> .wrl </member>

<member> .midr </member>

<member> .css </member>

<member> .ps </member>

<member> .ttf </member>

<member> .mso </member>

<member> .dvi </member>

</attrib>

<attrib name="extract\_links\_from\_dupes" type="boolean"> yes </attrib>

<attrib name="fetch\_timeout" type="integer"> 300 </attrib>

<attrib name="force\_mimetype\_detection" type="boolean"> no </attrib>

<section name="ftp\_errors">

<attrib name="4xx" type="string"> DELETE:3 </attrib>

<attrib name="550" type="string"> DELETE:0 </attrib>

<attrib name="5xx" type="string"> DELETE:3 </attrib>

<attrib name="int" type="string"> KEEP:0 </attrib>

<attrib name="net" type="string"> DELETE:3, RETRY:1 </attrib>

<attrib name="ttl" type="string"> DELETE:3 </attrib>

</section>

<attrib name="headers" type="list-string">

<member> User-Agent: FAST Enterprise Crawler 6 used by My Company (administrator@example.com) </member>

</attrib>

<attrib name="html\_redir\_is\_redir" type="boolean"> yes </attrib>

<attrib name="html\_redir\_thresh" type="integer"> 3 </attrib>

<section name="http\_errors">

<attrib name="4xx" type="string"> DELETE:0 </attrib>

<attrib name="5xx" type="string"> DELETE:10 </attrib>

<attrib name="int" type="string"> KEEP:0 </attrib>

<attrib name="net" type="string"> DELETE:3, RETRY:1 </attrib>

<attrib name="ttl" type="string"> DELETE:3 </attrib>

</section>

<attrib name="if\_modified\_since" type="boolean"> yes </attrib>

<attrib name="javascript\_keep\_html" type="boolean"> no </attrib>

<section name="limits">

<attrib name="disk\_free" type="integer"> 0 </attrib>

<attrib name="disk\_free\_slack" type="integer"> 3 </attrib>

<attrib name="max\_doc" type="integer"> 0 </attrib>

<attrib name="max\_doc\_slack" type="integer"> 1000 </attrib>

</section>

<section name="link\_extraction">

<attrib name="a" type="boolean"> yes </attrib>

<attrib name="action" type="boolean"> yes </attrib>

<attrib name="area" type="boolean"> yes </attrib>

<attrib name="card" type="boolean"> yes </attrib>

<attrib name="comment" type="boolean"> no </attrib>

<attrib name="embed" type="boolean"> no </attrib>

<attrib name="frame" type="boolean"> yes </attrib>

<attrib name="go" type="boolean"> yes </attrib>

<attrib name="img" type="boolean"> no </attrib>

<attrib name="layer" type="boolean"> yes </attrib>

<attrib name="link" type="boolean"> yes </attrib>

<attrib name="meta" type="boolean"> yes </attrib>

<attrib name="meta\_refresh" type="boolean"> yes </attrib>

<attrib name="object" type="boolean"> yes </attrib>

<attrib name="script" type="boolean"> yes </attrib>

<attrib name="script\_java" type="boolean"> yes </attrib>

<attrib name="style" type="boolean"> yes </attrib>

</section>

<section name="log">

<attrib name="dsfeed" type="string"> text </attrib>

<attrib name="fetch" type="string"> text </attrib>

<attrib name="header" type="string"> text </attrib>

<attrib name="postprocess" type="string"> text </attrib>

<attrib name="screened" type="string"> text </attrib>

<attrib name="site" type="string"> text </attrib>

</section>

<attrib name="login\_failed\_ignore" type="boolean"> no </attrib>

<attrib name="login\_timeout" type="integer"> 300 </attrib>

<attrib name="max\_backoff\_counter" type="integer"> 50 </attrib>

<attrib name="max\_backoff\_delay" type="integer"> 600 </attrib>

<attrib name="max\_doc" type="integer"> 1000000 </attrib>

<attrib name="max\_pending" type="integer"> 2 </attrib>

<attrib name="max\_redirects" type="integer"> 10 </attrib>

<attrib name="max\_reflinks" type="integer"> 0 </attrib>

<attrib name="max\_sites" type="integer"> 128 </attrib>

<attrib name="max\_uri\_recursion" type="integer"> 5 </attrib>

<attrib name="mufilter" type="integer"> 0 </attrib>

<attrib name="near\_duplicate\_detection" type="boolean"> no </attrib>

<attrib name="obey\_robots\_delay" type="boolean"> no </attrib>

<section name="pp">

<attrib name="ds\_max\_ecl" type="integer"> 10 </attrib>

<attrib name="ds\_meta\_info" type="list-string">

<member> duplicates </member>

<member> redirects </member>

</attrib>

<attrib name="ds\_paused" type="boolean"> no </attrib>

<attrib name="ds\_send\_links" type="boolean"> no </attrib>

<attrib name="max\_dupes" type="integer"> 10 </attrib>

<attrib name="stripe" type="integer"> 1 </attrib>

</section>

<section name="ppdup">

<attrib name="compact" type="boolean"> yes </attrib>

</section>

<attrib name="proxy\_max\_pending" type="integer"> 2147483647 </attrib>

<attrib name="refresh" type="real"> 10080.0 </attrib>

<attrib name="refresh\_mode" type="string"> scratch </attrib>

<attrib name="refresh\_when\_idle" type="boolean"> no </attrib>

<attrib name="robots" type="boolean"> no </attrib>

<attrib name="robots\_auth\_ignore" type="boolean"> yes </attrib>

<attrib name="robots\_timeout" type="integer"> 300 </attrib>

<attrib name="robots\_tout\_ignore" type="boolean"> no </attrib>

<attrib name="robots\_ttl" type="integer"> 86400 </attrib>

<attrib name="start\_urifiles" type="list-string">

<member> seedlist.txt </member>

</attrib>

<attrib name="smfilter" type="integer"> 0 </attrib>

<attrib name="sort\_query\_params" type="boolean"> no </attrib>

<section name="storage">

<attrib name="clusters" type="integer"> 8 </attrib>

<attrib name="compress" type="boolean"> yes </attrib>

<attrib name="compress\_exclude\_mime" type="list-string">

<member> application/x-shockwave-flash </member>

</attrib>

<attrib name="datastore" type="string"> bstore </attrib>

<attrib name="defrag\_threshold" type="integer"> 85 </attrib>

<attrib name="remove\_docs" type="boolean"> no </attrib>

<attrib name="store\_dupes" type="boolean"> no </attrib>

<attrib name="store\_http\_header" type="boolean"> yes </attrib>

</section>

<attrib name="truncate" type="boolean"> no </attrib>

<attrib name="umlogs" type="boolean"> yes </attrib>

<attrib name="uri\_search\_mime" type="list-string">

<member> text/html </member>

<member> text/vnd.wap.wml </member>

<member> text/wml </member>

<member> text/x-wap.wml </member>

<member> x-application/wml </member>

<member> text/x-hdml </member>

</attrib>

<attrib name="use\_cookies" type="boolean"> yes </attrib>

<attrib name="use\_http\_1\_1" type="boolean"> yes </attrib>

<attrib name="use\_javascript" type="boolean"> no </attrib>

<attrib name="use\_meta\_csum" type="boolean"> no </attrib>

<section name="workqueue\_priority">

<!-- Default priority level is 2. For this specific setting it means that a URI that doesn't match the specified includes for the queues will be inserted with priority level 2 -->

<attrib name="default" type="integer"> 2 </attrib>

<!-- Define a work queue with 3 priority levels -->

<attrib name="levels" type="integer"> 3 </attrib>

<attrib name="pop\_scheme" type="string"> prr </attrib>

<attrib name="put\_scheme" type="string"> include </attrib>

<attrib name="start\_uri\_pri" type="integer"> 2 </attrib>

<section name="1">

<!-- This queues share/weight is 10 -->

<attrib name="share" type="integer"> 10 </attrib>

<!-- These include rules defines the URIs that should enter the 1st priority level -->

<section name="include\_domains">

<attrib name="suffix" type="list-string">

<member> web005.example.net </member>

<member> web006.example.net </member>

</attrib>

</section>

</section>

<!-- Settings for the second priority level queue (3) -->

<section name="2">

<attrib name="share" type="integer"> 10 </attrib>

<section name="include\_domains">

<attrib name="suffix" type="list-string">

<member> web002.example.net </member>

<member> web003.example.net </member>

</attrib>

</section>

</section>

</section>

</DomainSpecification>

</CrawlerConfig>

1. Access Details
   1. Development Application Server

|  | **App Dev** |
| --- | --- |
| **FQDN** |  |
| **IP** |  |
| **User account** | Fastcrawl |
| **Password** | Fast53crawl |
| **ESP service account** | /export/products/sunweb |
| **Password** |  |
| **Sun web server directory** |  |
| **Sun web server admin** |  |
| **Oracle** |  |

* 1. Production Application Server

|  | **App Prod** |
| --- | --- |
| **FQDN** |  |
| **IP** |  |
| **User account** |  |
| **Password** |  |
| **Sun web server directory** |  |
| **Sun web server admin** |  |
| **Oracle** |  |

* 1. Production ESP Server

|  | **ESP Prod** |
| --- | --- |
| **FQDN** |  |
| **IP** |  |
| **User account** |  |
| **Password** |  |
| **ESP service account** |  |
| **Password** |  |
| **Home directory** |  |
| **ESP directory** |  |
| **Data directory** |  |

1. The level is the number of levels to follow links in a depth first crawl of a website, beginning at the start URI. Level 0 means only crawl start URIs, level 1 means crawl start URIs and all links coming off them. By default, using levels means not following cross site links. [↑](#footnote-ref-3)
2. The feeding rate can be determined by using the Clarity tool provided with ESP. Access to this tool is discussed in the Operations guide. [↑](#footnote-ref-4)