Search for: Search

Change Log

Below is a list of all the XTF releases, in reverse chronological order. Click on a release to see new features, bug fixes, and changes for each release.

Version 3.0-beta

Release date: January 31, 2011

Highlights

- Scanned book display support in default UI
- Stability improvements to index rotation support
- Globalization and RSS support
- More Unicode improvements
- Many bug fixes

Changes

- Added an end-to-end test of index rotation.
- Unicode is now stored in normalized form C (NFC) in the index, and queries are also normalized internally. This way, a query for “a” followed by a combining diacritic “umlaut” (which is not normalized) will still match the combined “a-umlaut” character (which is normalized). Thanks to Marcos Fragomeni for finding the problem and suggesting a fix.
- New attribute can be specified on a query: returnMetaFields=”field1,field2,…” which can cut down on processing and/or transmission time for a query that has lots of hits on docs with lots of fields in them.
- indexDump utility now supports an optional “-xml” mode, and an “-allFields” flag to obviate the need to list every field explicitly on the command line.
- Index version bumped to “3.0b” but should still work with XTF 2.2 indexes. Still, it’s generally a good idea to re-index after upgrading.
- Switched to nicer XTF logo in the default UI.
- Added default interface support for scanned books using the Open Library BookReader.
- Centralized some XSL utility functions in xtfCommon stylesheet directory.
- Added globalization feature: can now supply strings to present the UI in various languages, and allow the user to pick a language. Note: the individual translations are all incomplete and meant as a starting point only. We love contributions!
- Added a link to get an RSS feed from a search (along with recognizable RSS icon).
- Changed old way of matching element names with any namespace, “matches(local-name(), ‘foo’)” to more elegant syntax, “*:foo”. This was especially prevalent in the TEI stylesheets.

Bug Fixes

- Downgraded iText library (used for PDF combining only) to ver 2.1.7, since in a blatantly commercial move the iText people changed (unbeknownst to me when I upgraded to the 5.x line) from a friendly

LGPL license to an unfriendly Affero GPL, making it incompatible with XTF's license stable.
- Index rotation now uses rsync instead of a giant Perl command, which should eliminate crashing problems when large numbers of documents are changed in an index.
- Fixed bug when warming indexes in background: if an index failed to open once (e.g. if it's still under construction) the warmer thread would crash and never warm again.
- Fixed handling of empty facet values.
- Fixed inconsistent tokenization of "sectionType" values — now they are always uniformly tokenized.
- Fixed crash bug if no pluralization map is used.
- Fixed a bug: tokenizer was incorrectly and bizarrely lower-casing "X" between dashes.
- Accent maps and plural maps now play better together. Thanks to Marcos Fragomeni for finding the bug and posting a fix.
- Fixed URL handling for document IDs containing special characters.
- Fixed prefilter bug for any item with multiple dates.

Version 2.2

Release date: June 8, 2010

This is a fairly major release containing several features as well as bug fixes and stability improvements.

Highlights

- Index-related improvements: solid incremental indexing, index rotation, validation and background warming.
- Popular new "sub-document" feature for fine-grained searching of structured texts.
- Many smaller changes and bug fixes.
- As always the core of XTF 2.2 is compatible with all prior XTF stylesheets and config files.

Features

- Sub-documents: A single document can now be broken into logical "sub-documents" that come up individually in crossQuery but all together in dynaXML. Useful for documents with many unrelated parts (e.g. a book of poetry.) Sub-documents inherit their parent document's meta-data, and can add their own (or override.)
- Stylesheet profiling now enabled for crossQuery (not just dynaXML)
- New cacheControl configuration option for crossQuery and dynaXML. Default out-of-the-box is to disable browser caching of all requests because they're dynamic. Previous behavior maintained if flag is omitted to maintain backward compatibility.
- textIndexer and servlets support index rotation, so that a partially completed index won't be displayed. This advanced feature is disabled by default as it's best used in final deployment only. Note that index rotation depends on Perl and on hard links; thus it probably won't work on Windows.
- textIndexer can now perform simple validation on the index. Validation tests include number of hits in crossQuery and/or dynaXML. If an index fails validation, it won't rotate (servlets will continue to use the old but good index.)
- With validation enabled, servlets can use it to "warm up" an index in the background while serving up the prior version; once validation is done, they switch instantly to the new index. This way frequent incremental indexing runs do not cause lag for users.
- New sorting option for crossQuery results: "totalHits" which sorts the documents by the number of hits

within them. Note that sorting this way is somewhat slower but it shouldn't be noticeable for small to medium collections.
- New servlet parameter is available to all servlet stylesheets: "http.rawURL" which contains the original URL with all its un-escaped percent codes intact.
- New indexer option ("-force") to force indexing regardless of whether objects have changed. This way one does not have to "touch" a bunch of files to redo them.
- Code is now in Mercurial (hg) easing development as well as making it possible for XTF users to maintain a unified code base with the main XTF code in the default branch and their changes in another, allowing for more seamless upgrades.
- New extension functions to:
    - Create a temporary file that will be automatically deleted after the current request: FileUtils:createTempFile(prefixStr, suffixStr)
    - Apply an accent map (as used by textIndexer) during stylesheet processing: CharUtils:applyAccentMap(pathToMapFile, stringToMap)
    - Stylesheets can now optionally produce a PDF file instead of an HTML page, using XSL FO (Formatting Objects) with the Apache FOP library. Additionally they can glue on an existing PDF file before or after the FOP-generated PDF.
    - Run an external command which is expected to produce a PNG or JPEG image on stdout. That image is in turn streamed directly to the client via the servlet's output stream.
    - Pipe the contents of a file directly to the client via the servlet's output stream.
    - Create an HTTP error (e.g. 500) directly from a stylesheet.
    - Generate an HTTP redirect (either temporary or permanent) from a stylesheet.
    - Convert the size of a file to a human-readable string, e.g. "36 Kb", "1.2 Mb", etc.
    - Calculate the MD5 hash of a string
    - Compute the number of days, hours or minutes that have elapsed between a given time and now.

Changes

- textIndexer now defaults to traversing all input directories instead of only those with no files above them. This is enabled by a setting in the config file, and in the default distribution it is set. Old behavior can still be obtained by removing the flag (maintaining backward compatibility.)
- Index tokenization now recognizes the paragraph symbol "§" as a queryable word.
- Speed optimizations in the default crossQuery resultFormatter.xsl
- EAD stylesheets now support "chunking" like the TEI stylesheets, allowing them to handle very large EAD files. Hits now have linking arrows to take the user to the next or previous hit even if it's in a different chunk.
- Changed EAD docFormatter to be able to handle top-level C02 elements.
- Optimized key creation to skip dynamic keys during indexing.
- Socket exceptions are no longer treated as severe errors since clients often disconnect prematurely from the server.
- Changed error reporting to avoid a Java stack trace for Saxon errors. Still reports the XSLT file and line number of course.
- URL parameters are now passed to the stylesheet in the same order they were specified in the URL (rather than alphabetical).
- iText (used for MS Word text extraction) upgraded to version 5.0.1
- PDFBox (used for PDF text extraction) upgraded to version 1.1.0

Bug fixes

- Fixed default sorting of author and title browse pages (stylesheet fixes only.)

- Minor queryParser and resultFormatter fixes for OAI interface validation
- Fixed stylesheet error causing max # of result pages to be one short of the correct number.
- Fixed stylesheets to deal properly with documents containing multiple dates.
- Many fixes in the stylesheets and within the Java code to support special symbols in metadata values (including facet values). These characters include “<”, “>”, “&”, “=”, non-latin symbols, etc.
- If a document fails during indexing due to an XSLT exception, its chunks are now rolled back (previously they were tacked onto the start of the next document producing bizarre search results.)
- Accent maps and plural maps weren’t playing together properly. Now plural words will be depluralized whether or not they started out accented. Thanks to Marcos Fragomeni for the fix suggestion.
- Made some fixes to CJK (Chinese/Japanese/Korean) tokenization.
- A few fixes for hit and note links in TEI display.
- More robust in the facet of minor indexing errors.
- Fixed to roll back chunks of a document if it suffers an XSLT failure part-way through being indexed.
- More descriptive message if xtf:tokenize is inconsistently applied for a given field by the various index preFilter stylesheets.
- Fixed latency cutoff reporting to be more accurate.
- Added descriptive error message if one tries a “more-like-this” query on full text (which we don’t support… yet anyway.)
- Fixed crash in query request parsing logic when query consists of nothing but stop words.
- Lazy files are no longer created for non-XML files during indexing.
- Fixed bug where stored keys in lazy files would occasionally be corrupted.
- Fixed NullPointerException? when reading XML stubs that have certain kinds of DTD declarations.

Experimental additions

- Support for Dublin Core Kernel/ERC metadata output from crossQuery
- Support for OpenURL resolving in crossQuery stylesheets.

Known issues

- Many new functions from XTF 2.1 and 2.2 are not yet documented. For now, a message to the xtf-user list, or examining the code, are the only ways to learn how to use these new features.

Version 2.1.1

Release date: June 18, 2008

This is a minor release containing bug fixes and stability improvements.

Changes

- By default, dynaXML will no longer build lazy trees by itself, instead depending upon textIndexer to build them. This is generally better as the lazy files are kept in sync with the index, but the old behavior can be enabled with the <lazyTrees> configuration option in dynaXML.conf.
- Small stylesheet tweaks for EAD viewing, and for author and title browsing.
- Search robot handling has been revised to be friendlier to robots, presenting a much flatter (and thus more easily crawled) view of the document set.

Bug fixes

- When running the indexer in `-clean` mode, previously it could fail mysteriously (or possibly create a corrupt index) if it was unable to delete any part of the old index. It will now fail consistently with a comprehensible error message.
- In previous versions, the indexer attached a list of tokenized fields to the first document indexed. Unfortunately, if that document were later updated or deleted, the list was lost. XTF now stores the list in a separate file to avoid this problem (it still reads lists in existing indexes for backward compatibility.)
- There were two bugs in the IP authentication code for dynaXML. If overlapping ranges were specified, or a single range with a range in the second or third component, dynaXML would incorrectly place addresses as in/out of the list.
- HTTP headers were inconsistently named; on Resin they'd be mixed case (e.g. "User-Agent") while on Tomcat they'd be lower-case (e.g. user-agent). Hence, it was difficult for stylesheets to reliably switch on them. Now they're always available with the lower-case name (with the mixed-case name present as well for backward compatibility).
- Footnote links within TEI documents were not properly linking to the associated notes, due to namespace matching problems.

Version 2.1

Release date: June 12, 2008

While the last version focused on getting the documentation in order, this is a major new feature release with special focus on the default stylesheets and user interface.

Highlights

- Extensive UI improvements, including new search forms, built-in faceted browsing, and a crisp new look.
- Built-in XHTML and OAI/PMH output, NLM article in/out, and Microsoft Word in.
- For those starting fresh with XTF the stylesheets are now easier to understand and adapt. For existing implementations, 2.1 is compatible with all prior XTF stylesheets and config files.
- Experimental "freeform" boolean query language.
- Many bug fixes and minor changes/features.

Features

- Revised default stylesheets:
  - Faceted browsing and query drill-down
  - Improved look and feel
  - Simple and Advanced search
  - Browse by Author or Title
  - Improved bookbag
  - Support for searching and navigating NLM-formatted XML articles
  - Support for searching most Microsoft Word documents
  - docSelector.xsl uses ReadXMLStub extension (below) to decide how to index XML docs, rather than relying on file/directory name.

    - Now uses web-standard YUI library for AJAX instead of custom Javascript code.
    - Basic OAI/PMH support now included.
    - Support for efficiently handling robot crawling (e.g. Googlebot)
    - Stylesheet code has been streamlined.
- New extension functions to:
    - Send email from a stylesheet.
    - Quickly read in the first part of an XML file.
    - Check if cookies are enabled.
    - Read and "tidy" HTML documents into XHTML documents (helpful for screen scraping and similar activities)
- A new query type is supported: `<allDocs>`. This is mainly for convenience in browsing the entire collection (usually used with facets)
- Ability to index most Microsoft Word documents
- New "field:emptyFirst" and "field:emptyLast" doc sort modifiers; emptyLast is default instead of inscrutable behavior.
- Also support "field:ascending" and "field:descending" sort modifiers, as a more verbose form of "+field" and "-field".
- New facet sorting option: reverseValue (useful for date field).
- New facet selection operator: singleton (useful for auto-expanding single selections in a hierarchical facet).

Changes

- Expanded set of sample documents.
- Session tracking is now enabled by default. Cookie-less mode is deprecated.
- Improved error reporting includes file name and line number for all XSLT errors.
- Changed to avoid loading DTD associated with documents. This can greatly boost speed, and also makes the system more reliable as it no longer depends on external web servers.
- Upgraded to new version of PDFBox which is faster and less buggy.
- There was previously no way to select facet values containing parentheses, brackets, asterisks, and other special characters. You can now use quotes to disambiguate these in a facet select expression.
- You can now explicitly specify 'score' (or synonym 'relevance') in sortDocsBy on meta-data fields. Score sorting has always been the default, but this enables one to make it obvious.

Bug fixes

- Fixed occasional NullPointer exception in crossQuery when 'raw' mode was used.
- Fixed other exception handling problems. Thanks to Jakob Saternus for finding these and submitting patches.
- Fixed rare NullPointerException when trying to scan attributes of a lazy document.
- Fixed bug in facet sorting: would sometimes drop single subgroup and all its descendants.
- Fixed HTML text extraction to remove or map illegal characters.
- Fixed bug in path processing: XTF would often erroneously remove "../.." from a path name.
- Fixed bug that caused an exception if non-XTF attributes were placed on an untokenized meta-data element. Thanks to Richard Padley for reporting this.
- Fixed potential out-of-memory situation if multiple threads simultaneously load facet data.
- Fixed NullPointer exception when using wildcards in a <near>…<not> query. Thanks to Seth Cherney for reporting this.
- XTF would fail mysteriously if certain names were used for meta-data fields, such as "text" and "key". A descriptive error is now reported instead.

- In high-volume situations when a new or changed index was loaded, it was possible for multiple threads to load the same facet data at the same time, wasting time and possibly causing an out-of-memory situation. These are now synchronized to be in strict sequence instead of parallel.
- A stack trace was being reported for TermLimit exceptions, but isn't needed.
- Fixed bug: sometimes Saxon on Windows would report include "%20″ in a system-id() path instead of space, which caused FIleUtils.exists() to fail. FileUtils now handles this case.
- Fixed handling of multiple values for the same parameter name in the URL. XTF servlets now pass these correctly to stylesheets (which can deal correctly with them, or not, as they wish.)
- Fixed assertion failure when a single stop word is specified as a keyword query.
- Fixed bug: single-term, single-field keyword queries were failing mysteriously.
- Fixed handling of drive-letter paths on Windows under Java 6 that caused the textIndexer to crash on start-up.
- Corrected handling of <xsl:result-document> in servlets, so that it can change the output format during the transformation. Needed for XHTML/Frameset output.
- Took out dynaXML caching of docSelector output. This cache didn't actually enhance performance, and caused tricky multi-threading bugs to occur.
- Removed obsolete DirectSearch and PreviewXML servlets (these were never used in XTF)
- Fixed crossQuery problems relating to query terms containing ampersands. Thanks to Richard Padley for submitting these.

Experimental additions

- "Freeform" query language which allows users to type in fielded boolean queries similar to those supported by Google's advanced search.
- RawQuery servlet, which receives a single XML query in the URL, runs it, and returns XML results.
- Extension to read a PNG file, extract a piece of it, add yellow highlights, and send image over HTTP. Eventually may be used to implement image page flipping with hits in context (e.g. PDFs, scanned texts, etc.)
- Support for alternate URL parameter tokenizers in crossQuery and dynaXML.

Version 2.0

Release date: December 14, 2007

The main changes in XTF 2.0 are under the hood (mainly upgrades to the underlying Lucene and Saxon engines) but a major new feature has been added: multi-word spelling correction. Additionally, the XTF code has been reformatted to make it easier for the main developers and outside parties to work on it and collaborate. Finally, several bugs have been fixed and the documentation has been updated.

As always, every effort has been made to maintain compatibility with existing stylesheets and configuration files. No changes should be necessary (though note that queries with upper case letters are handled slightly differently now).

Following are details on all the changes in this version.

- A robust and efficient spelling correction engine is now available and enabled by default for all queries in crossQuery. Programming, reference, and under the hood documentation is provided.
- XTF now uses the Java 1.5 / 5.0 platform, which provides better XML parsing, enhanced performance, and support for handy programming constructs. If you are using Java 1.4.x, you'll need to upgrade

before using this version of XTF.
- The latest versions of Saxon (8.9) and Lucene (2.1) are now included with XTF. These provide better performance and many bug fixes, as well as opportunities for to integrate new features into XTF.
- Servlets in XTF now detect the output method specified in the <xsl:output> stylesheet tag and automatically set the HTTP mime type appropriately. Previously the mime type was always set to "text/html" which is incorrect for XML documents.
- Documented the xtf:sectionTypeAdd prefilter attribute, which has existed for quite some time and comes in handy storing hierarchical section types.
- XTF now inserts version numbers into index and lazy tree files, and checks them to verify compatibility. This should prevent problems encountered by users trying to use indexes generated by previous versions of XTF.
- Multi-character diacritics as well as non-spacing marks are now removed by default during index tokenization and query processing. Previously only single base/diacritic characters were being handled. Thanks to Joao Lima for pointing this out and testing the fix.
- The redirect:send, Saxon extension now properly handles attribute-value templates.
- There were several situations where errors during indexing could result in empty or corrupt lazy tree files. These have been fixed. Thanks to Jakob Saternus for pointing this out and sending fixes. Also, lazy files that are generated by dynaXML will be automatically re-generated if the source file changes (this would not work for lazy files generated by the textIndexer since they would then be out of sync with the index.)
- Certain tokens (such as identifiers) containing underscores were tokenized differently by the textIndexer as opposed to the query parser, making it impossible to query for them. Now they are tokenized consistently.
- Fixed a bug in the "book bag" Javascript code that prevented removing books from the bag.
- In the default stylesheets, most instances of "&" are now replaced with a simple semicolon. Semicolons work just as well and make the code and URLs more compact.
- The input and output tags for the Query Router stylesheet (only needed by advanced users) are now documented.
- Broken links and missing images in the documentation have been fixed.
- When lazy files are generated by the textIndexer, the printed count of stored XSL keys is now correct; previously it was always one higher than the actual count.
- The servlets and indexer now properly handle ampersand and other special characters within filenames. However, the default stylesheets don't produce correct dynaXML links yet. Fixing this would take a lot of work, and it's not clear that it would be worth it. So for now, these files work in crossQuery only.
- Added a new section on Optimizing Performance to XTF Tips & Tricks. This covers stylesheet tips, profiling, and adjusting Java's virtual memory.
- Stylesheets can use a new extension function to get the current date and time.
- URL parameter tokens are no longer converted to lower case before passing them to the stylesheets, which enables one to echo a user's query in the same case they typed it. Case mapping is now handled internally in the text engine, after the query has been parsed.

Version 1.9

Release date: April 25, 2007

The main goal of XTF 1.9 was to update the documentation, attempting to cover all features and enhancements made in the past year. Here's a summary of the new documentation:

- An extensive Tips & Tricks guide has been added, covering topics such as various handy ways to debug stylesheets, setting up XTF to work in external tools such as <oXygen/> and Eclipse, and other helpful

tips.
- Multi-field <and> queries have been added. This is a powerful way to provide a simple "keyword" search that automatically searches for *all* of the terms in *any* of the specified fields. For instance, this would give good results for a query spanning title and author such as against all enemies clarke.
- A hybrid <orNear> query operator has been added, that operates like an <or> query but uses proximity for better scoring.
- New useProximity attribute is available on <and> queries.
- Completely rewrote the Hit Scoring section in XTF Under the Hood to reflect how XTF actually scores and ranks document hits (which has changed dramatically from the old versions that were previously documented.)
- Documented the explainScores attribute that enables comprehensive (if sometimes obscure) output of the score factors used by XTF to score a particular document hit.
- Wrote up the experimental <moreLike> query that searches for documents similar to a given one.
- Faceted browsing has graduated from experimental status into the permanent documents. The main description can now be found in the Faceted Browsing section of the Programming Guide.
- Documentation has been restructured, simplifying filenames and adding this main index page.
- The experimental Spelling Correction feature is now documented.
- XTF's very experimental support for dynamic creation of FRBR "work sets" has been documented, at least in basic form.
- Documented XTF extensions to call command-line programs from stylesheets, and to check a file's existence, length, and modification time.
- An Saxon extension instruction, redirect:send, has been added to allow a stylesheet to force an immediate HTTP redirect.
- For clarity, the documentation now refers to "bi-grams" instead of "n-grams", since XTF exclusively uses two-word n-grams.
- The new xtf:store attribute is available for meta-data fields at index time. Additionally, xtf:index can now be specified for meta-data fields as well (it used to only apply to text.)
- An option for numeric range searching is now available, which allows efficient processing of very granular numeric data, as long as it is in a fixed, consistent format.
- Some of the error tags that dynaXML could send to the **Error Generator Stylesheet** were not previously documented: <InvalidDocument> and <NoPermission>.
- Documented how to specify multiple pre-filters in the <file> tag produced by the **Document Selector Stylesheet**. The pre-filters are chained together in the order listed.
- Experimental support for MARC record parsing is now documented.
- Pass-through configuration parameters have been supported for a long time, but are now properly documented.

Version 1.8

Release date: September 8, 2006

- The default dynaXML docFormatterCommon.xsl stylesheet was not properly computing the path for figure references if the doc source is external (e.g. specified by `source=http://xxx` in the URL).
- The sample "book bag" and "more like this" features were broken, due to missing script files in the distribution. Also, a few source files needed to rebuild the internal JavaCC parsers in XTF were missing.

Version 1.8-beta

- Many users have requested EAD support in XTF out-of-the-box. While XTF has always been capable of handling these, the default stylesheets were very TEI-centric. This release contains brand new stylesheets that support TEI, EAD, PDF, HTML, and Text. Flexible meta-data handling will use *.dc files if present. If not present, will look inside TEI and EAD documents. Also, the confusing reliance on *.mets files has been completely removed.
- Disabled non-standard whitespace stripping while building lazy tree files. Previously, XTF stripped whitespace between elements, which caused differing results from the same stylesheets run through Saxon from the command-line. If absolutely necessary, there is an undocumented index config flag to turn stripping back on: <whitespace strip="yes"/>
- Upgraded PDFBox to most recent version (0.7.2) which offers greater speed and stability, and better results.
- Fixed FileUtils.exists() function called by some stylesheets to automatically handle a "file:" prefix if present.
- Fixed PDF filter in indexer to automatically escape XML characters such as '<', and to strip out invalid characters.
- Same fix for text files.
- Certain unusual queries caused an assertion in FieldSpanSource: "kept span was cancelled". Fixed.
- Fixed problem that kept JavaDocs from building.
- XTF now avoids loading external DTDs for documents pulled in through the Saxon document() function. This helps speed the processing, and reduces external dependencies.
- Fixed bug that caused indexer to crash if resulting index is empty (e.g. if no docs found).
- Fixed bug: indexDump would only output first of multiple un-tokenized values for a field.
- Experimental support for spelling correction has been added. Documentation to follow.
- Experimental new query operator added: <orNear>, which is like a standard OR query except that it will take proximity into account when multiple terms are present in one document.
- Improvements to the experimental "more like this" query. It may be getting close to prime-time.
- The XTF icon has been changed to be more descriptive, less confusing, and arguably less fun. "XTF Man" is gone.

Version 1.7

Release date: June 1, 2006

- Change log now contains item numbers from the SourceForge trackers ("Feature Requests" or "Bugs") which can be referenced for more detailed information.
- Added new front end to crossQuery servlet. The new "query router" stylesheet allows the use of multiple query parsers. Those just starting out, or who only need one parser, can use the default queryRouter.xsl without change.
- textIndexer now allows "deep" section type indexing. A new attribute "sectionTypeAdd" can be inserted by the prefilter stylesheet. This causes the text in that section to inherit its parent's sectionType and add the specified text. This allows simple processing of hierarchical sections without complex prefilter code.
- Many users have expressed confusion over the way document IDs were handled in dynaXML, and observed that much CDL-centric code is present in the default stylesheets. These have been refactored, and document IDs are now simply the path from the data directory to each document, instead of a strict 10-character code.
- XTF now allows stylesheets to track data on a per-user-session basis. A simple API is provided to get and set state data. The session identifier is tracked using cookies, or if the user has cookies disabled, though URL rewriting.
- Default stylesheets now expose "Book Bag" and "More Like This" functionality. The former is based

on the session state API, the latter on the new <moreLikeThis> query operator. These also demonstrate an AJAX style of programming, updating pages on the fly.

- New “exact” query operator added. To match, the field must contain exactly the query phrase; no more, no less.
- Added new “moreLike” query operator which uses a simple index-based algorithm to locate additional documents that resemble a specified document. This feature is considered experimental and subject to improvement/change. [Feature Req 1470968]
- Made minor changes to the experimental “boost set” facility.
- Fixed bug in phrase query if stop-words appeared at start or end of a meta-data field.
- Fixed bug with where apostrophe and other combined words at start or end of a meta-data field would cause queries to not match.
- Fixed bug causing boost values to have no effect on an <or> query.
- Refactored Lucene integration. The result is more modular, which will help in upgrading to Lucene 1.9 and 2.0. Back-ported selected classes to improve span processing on indexes with millions of records.
- Config file parameters are now case insensitive. Also, boolean parameters all uniformly accept “true”, “yes”, “1″ as synonyms, and “false”, “no”, and “0″ as synonyms.
- Added ability to display non-normalized scores (or raw) scores in crossQuery.
- Added optional “score explanation” in crossQuery, to give a very detailed description of how each document’s score was computed.
- Made several changes and fixes to the experimental ‘facet’ feature.
- Multiple index prefilters may now be specified for one document by the docSelector stylesheet. The prefilters will be run in a chain.
- Added support for parsing MARC21 data files. The indexer will break them into records, convert them to MARCXML format, and pass each converted record to the prefilter(s). Very large files are supported, and the indexer will try to skip bad records and recover.
- Fixed null pointer exception in dynaXML when an empty query was specified.
- Servlets now allow “;” to separate URL parameters. This can be quite handy as opposed to “&”, since the latter requires special escaping in stylesheets. Both are now supported interchangeably.
- All references to “ngrams” have been changed to the more specific term “bigrams”.
- Improved efficiency of span collection in the Text Engine.
- Vastly reduced memory usage of cached sorting arrays for indexes that contain only meta-data.
- All servlets now pass a “servlet.dir” parameter to stylesheets. This is the home directory of the XTF installation, and can be used by stylesheets to locate data files or for other purposes.
- crossQueryResult input to resultFormatter stylesheet now contains the original parsed URL parameters, and the query that resulted from the queryParser stylesheet. Both of these can be quite useful in result formatting.
- Queries output from queryParser stylesheet may now optionally contain <resultData> elements. These are ignored by the Text Engine, but passed on to the result formatter stylesheet. They’re a handy way for the query parser to pass data directly to the result formatter.
- Meta-data fields can now be marked in index prefilter as xtf:store=”no”, which prevents them from showing up in query results. The field is still indexed, just not stored or displayed.
- Similarly, the index prefilter can mark a field with xtf:index=”no”, causing it to not be indexed (and this not searchable) but still be stored and displayed.
- Improved efficiency of textIndexer’s culling phase. In particular, it no longer runs out of memory and crashes on indexes with millions of documents.
- ‘indexStats’ tool is now much faster, and attempts to provide as much information early in the process as possible. Also, doesn’t crash on large indexes.
- Added new ‘indexDump’ tool, which can dump selected meta-data fields from all documents in an index.
- Fixed bug where indexer would occasionally crash when trying to create a lazy tree file without creating its directory first.

- Fixed bug that caused XML namespace declarations to be dropped from the beginning of in lazy tree files.
- textIndexer now tracks and displays the elapsed time of each indexing run.
- crossQuery wasn't paying attention to the MIME type specified by Result Formatter stylesheet output specification. Now the default is (text/html) is only used if none specified.
- Fixed assertion failure when a <not> clause appeared within a <near> query.
- Fixed a bug in the internal simplification of boolean queries that caused an assertion failure when searching for "the".
- Fixed bug in dynaXML that gave an unenlightening error message if the source file specified by the docReqParser is actually a directory.

Version 1.6.1

- New "debug step" mode added, which can be very handy both to understand crossQuery and to debug stylesheet problems. This is enabled by adding "&debugStep=1″ to the crossQuery URL. This also works in the experimental SRU servlet.
- Added optional ability to turn on a "runaway" timer, that will report and optionally kill off requests that exceed specified time limits. This can help in tracking down intermittent server slowdowns. This is configured in conf/crossQuery.conf and conf/dynaXML.conf.
- Added optional cutoff size for latency reporting. After a request has exceeded this amount of data, the servlet will report the latency immediately. When the request finally finishes, the final latency is also reported. This is configured in conf/crossQuery.conf and conf/dynaXML.conf.
- Minor improvements to paging behavior in crossQuery resultFormatterCommon.xsl.
- Fixed "Modify Query" link in crossQuery default/resultFormatter.xsl.
- Fixed bug that caused the Content-Type of "raw" mode output from dynaXML to be "text/html" instead of the proper "text/xml".
- Fixed a potential thread synchronization issue in lazy file access.
- Changed timestamp output to be more compatible with Resin and Tomcat.
- Fixed thread contention issue with query rewriting.
- Fixed memory leak with performing searches in dynaXML.
- Fixed handling of '"' and '&' characters in meta-data fields during indexing (was throwing an exception instead of passing these through.)
- Switched indexer to using Lucene's "compound files" mode. This results in indexes that have many fewer files, and thus avoids problems with running out of filesystem handles. The indexes are compatible, and the indexer will silently upgrade older indexes to the new compound files.
- TextIndexer now outputs the XTF version number (1.6.1) instead of the perpetual "1.0″.
- Reduced memory usage of accent and plural mapping facilities.
- Clarified error message when an exception is encountered during Saxon processing.
- Minor documentation updates to reflect new features above, but a full documentation revision will have to wait until the 1.7 release.

Version 1.6.0

- Fixed caching problem that caused sort/group data to be reloaded on each query, rather than cached between queries as was intended.
- Fixed static variable problem that was causing the SRU and crossQuery servlets to conflict with each other.
- Fixed multi-threading bug: when many simultaneous crossQuery threads tried to access the same index,

they would sometimes corrupt each others' span results.
- Fixed bug: for apps that use the QueryProcessor Java API, the hit count and score normalization were not being reset from one use to the next. This bug did not affect crossQuery or dynaXML, which make a new QueryProcessor for every request.
- Fixed to avoid marking terms specified in a <not> query.
- Fixed a bug causing the indexer to crash when tokenizing certain fields ending in "."
- Fixed 'textIndexer' and 'indexStats' scripts to work properly under Microsoft Windows.
- Fixed a bug in handling of '&', '<', and '>' in source documents: they were being double-escaped. For instance, '&' would become '&amp' instead of '&'.
- Fixed a bug in handling of the XSLT 'previous::*' axis. The axis would operate incorrectly on lazy trees, essentially acting just like 'previous-sibling::*'.
- The SRU servlet was completely broken, but is now working again.
- Sample stylesheets now provide an option to reverse the order of sort-by-year.
- Added a new feature (as yet undocumented) that allows stylesheets to call out to external command-line tools. Robustly handles XML input and output, and allows a timeout specification. See regress/CrossQuery/K-External for examples of how to use this facility.
- Distribution now available as either a full distribution as before, or split into "core" and "example" pieces. The "core" piece is especially useful for existing users to upgrade the core while leaving all their stylesheets and configuration files intact.
- Minor documentation corrections.

Version 1.5.1

- Now works again under Java 1.4 (was using Integer.valueOf(int), which is only present in Java 1.5)
- Corrected problem where dynaXML wouldn't run unless an index was present (tried to create lazy file in a directory that didn't exist yet.)
- Fixed textIndexer and indexStats scripts to allow spaces in the XTF_HOME path, and to properly switch between ":" classpath separation on Unix and ";" on Windows.
- crossQuery servlet now passes the time (in seconds) it took to parse and process the query to the resultFormatter stylesheet. Documentation reflects this.
- Installation procedures corrected and simplified. Please see new documentation for more details.

Edit this entry.

Latest XTF News

- XTF 3.0 beta
- XTF Website Launched
- XTF Community Preview
- XTF 2.2 released

Subscribe to XTF News

- RSS