Open Repository, Open Source

Where we were,
What we've learnt,
And what it means to you

The Hardware

- 2x DL360 G2
- 1xMSA 1500cs
- 1xMSA 1000
- 1xMSA 20
  - 2 TBs capacity

The Software

- Windows Advanced Server 2000
- Microsoft Cluster Server
- Oracle 10 (on Sun/Solaris)
- Sun JDK 5
- Tomcat 5.5
- DSpace 1.3.2 + customisations

Problem #1: Handle Servers

- Ran as console applications
- Would not start automatically at boot
- Required a user login to start the servers
- Servers would stop if remote login disconnected
- Keeping open remote logins for extended periods cause maintenance problems (limited connections)

Solution #1: Services

- Use open source Java Service Wrapper to create Windows services
  - http://wrapper.tanukisoftware.org/
  - Uses single, simple configuration file
- Services started automatically on boot

Problem #2: Lucene

- Could not run filter-media, etc. whilst Tomcat was running and had an open searcher
- Existing solution was to stop Tomcat for entire duration of indexing, across all repositories
- Sometimes Tomcat would not be restarted correctly after indexing

Solution #2: DSIndexer Patch

- Added configuration option to index in an 'offline' directory
- Indexer attempts to swap in the new index on completion
- If it can't, sets a flag in the filesystem, and the DSQuery will swap it in on next search
- SourceForge patch #1655583 - Avoid index lock with filter-media

Problem #3: Oracle

- Database had been installed as ISO-LATIN
- To store Unicode characters, key columns used the national character set (NCLOB / NVARCHAR)
- N... columns can not be identified correctly with java.sql types

Solution #3: Separate Instance

- New instance configured as UTF-8
- All workarounds for N... column handling could be removed from the code
- All Oracle fixes that were still required (ie. CLOB support) could be contributed back
  - Patch #1665400 - Oracle schema changes for CLOB columns
  - Patch #1660752 - Workaround Numeric/Integer handling in Oracle

And on the 7th Day

Began building for the future

Until...

Problem #4: PDFBox

- PDFBox likes to create temporary files
- Lots of temporary files (well, one per PDF)
- Doesn't do too well at cleaning them up either (Windows thing?)
- And whilst not exactly the fault of PDFBox:

  on restarting the servers, the Cluster FS no longer worked

Opportunity: Linux Migration

- Debian 4
  - all required hardware and software support in Debian repositories
- Oracle Cluster Filesystem (OCFS2)
  - already used for Oracle installation
  - easy to configure

```
SERVER1:/# cat /etc/apt/sources.list

deb http://ftp.uk.debian.org/debian/ etch main contrib non-free
deb-src http://ftp.uk.debian.org/debian/ etch main contrib non-free

deb http://security.debian.org/ etch/updates main contrib
deb-src http://security.debian.org/ etch/updates main contrib



SERVER1:/# apt-get install apache2 libapache2-mod-jk tomcat5.5 sun-java5-jdk

SERVER1:/# apt-get install ocfs2-tool ocfs2console



SERVER1:# cat /etc/ocfs2/cluster.conf

node:
     ip_port = 7777
     ip_address = 192.168.2.101
     number = 0
     name = SERVER1
     cluster = ocfs2

node:
     ip_port = 7777
     ip_address = 192.168.2.102
     number = 0
     name = SERVER2
     cluster = ocfs2

cluster:
     node_count = 2
     name = ocfs2



SERVER1:/#  mkfs.ocfs2 -b 4K -C 32K -N 4 -L oracle_home /dev/sdb5
```

Handle Servers (Again)

- How do you:
  - Run multiple handle servers on a single machine?
  - Retain ability to stop / start them individually?
- Use Java Service Wrapper!
  - Same simple configuration file – some paths, etc. changed for Linux
  - Provided script stores PID in file, so each instance can be managed separately

The Future: DSpace 1.4.x

- New OR code based on DSpace 1.4.1 with
  - bug fixes (from 1.4.2)
  - researcher pages (Nathan Sarr / Tim Donohue)
  - configurable item submission (Tim Donohue)
  - configurable browse (Richard Jones) / browse ordering
    - added Oracle support
    - submitted and integrated ordering code

Browse: Richard's Marketing Blurb

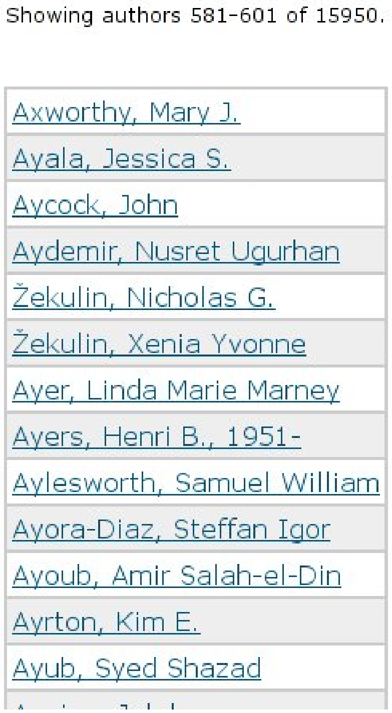
- Switching browse order on configured sort fields
- Configurable (by end user) results per page
- Long author list truncation (end user configurable)
- Significant performance improvements
- Paging on all result sets in all contexts
- Configurable cross linking to other browse contexts (e.g. authors in author list to items by that author)
- Access to browse tables via DAOs (easily allows different databases to be supported – ie. Postgres, Oracle)

Configurable Browse

- Can be configured to use any metadata fields
- Configurable sort fields
- No hard coded limits for
  - number of browse lists
  - number of sort fields

```
###### Browse Configuration ######
#
# Use this to configure the browse indices.  The form is:
#
# webui.browse.index.<n> = <index name> : \
#                                        <schema prefix>.<element>[.<qualifier>|.*] : \
#                                        (date | title | text) : \
#                                        (full | single) \
#
# (date | title | text | <other>) refers to the datatype of the field.
#                 date: the index type will be treated as a date object
#                 title: the index type will be treated like a title, which will include
#                            a link to the item page
#                 text: the index type will be treated as plain text.  If single mode is
#                            specified then this will link to the full mode list
#           <other>: any other datatype will be treated the same as 'text', although
#                   it will apply any custom ordering normalisation configured below
# (full | single) refers to the way that the index will be displayed in the
#                            browse listing.  "Full" will be the full item list as specified
#                            by webui.itemlist.columns; "single" will be a single list of
#                            only the indexed term
#
# NOTE: the text to render the index will use the <index name> parameter to select
# the message key from Messages.properties using a key of the form:
#
# browse.type.<index name>
#
# Note: the index numbers <n> must start from 1 and increment continuously by 1
# thereafter.  Deviation from this will cause an error during install or
# configuration update
#
# For compatibility with previous versions:
#
webui.browse.index.1 = dateissued:dc.date.issued:date:full
webui.browse.index.2 = author:dc.contributor.*:text:single
webui.browse.index.3 = title:dc.title:title:full
webui.browse.index.4 = subject:dc.subject.*:text:single
webui.browse.index.5 = dateaccessioned:dc.date.accessioned:date:full

# Set the options for what can be sorted by
#
# Sort options will be available when browsing a list of items (i.e. only in
# "full" mode, not "single" mode).  You can define an arbitrary number of fields
# to sort on, irrespective of which fields you display using webui.itemlist.columns
#
# the format is:
#
# webui.browse.sort-option.<n> = <option name> : \
#                                                    <schema prefix>.<element>[.<qualifier>|.*] : \
#                                                    (date, text)
#
# This is defined much the same as above.  The only difference is that the final
# parameter just lets the sorter know whether to expect to sort by plain text (using
# the "text" option), or by a proper date (using the "date" option).  If no options
# are specified, you will not be able to sort any results by anthing other than the
# key values.
#
webui.browse.sort-option.1 = title:dc.title:text
webui.browse.sort-option.2 = date:dc.date.issued:date
```

Existing Browse Ordering



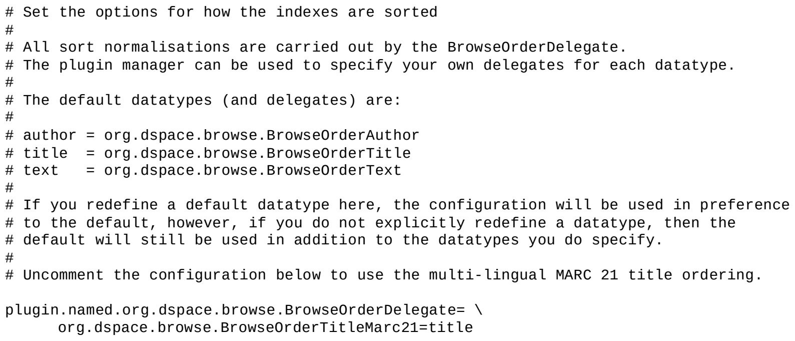
Showing authors 581-601 of 15950.

Axworthy, Mary J.
Ayala, Jessica S.
Aycock, John
Aydemir, Nusret Ugurhan
Žekulin, Nicholas G.
Žekulin, Xenia Yvonne
Ayer, Linda Marie Marney
Ayers, Henri B., 1951-
Aylesworth, Samuel William
Ayora-Diaz, Steffan Igor
Ayoub, Amir Salah-el-Din
Ayrton, Kim E.
Ayub, Syed Shazad

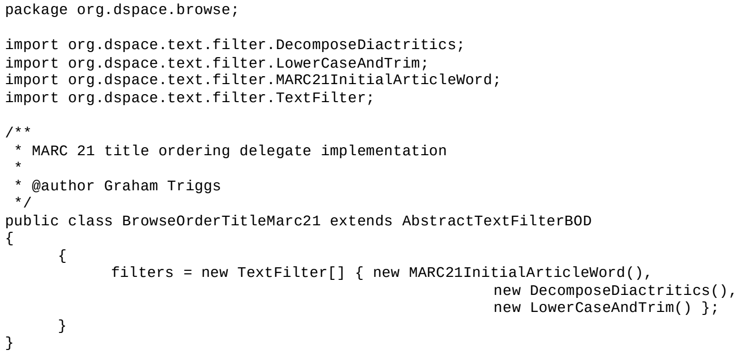
- Relies on database ordering of sort columns
- Doesn't deal with Unicode characters well

Configurable Ordering

- Order 'type' defined by browse datatype
- Each type can be configured to have a delegate
- Simple delegates defined as combinations of TextFilters



```
# Set the options for how the indexes are sorted
#
# All sort normalisations are carried out by the BrowseOrderDelegate.
# The plugin manager can be used to specify your own delegates for each datatype.
#
# The default datatypes (and delegates) are:
#
# author = org.dspace.browse.BrowseOrderAuthor
# title  = org.dspace.browse.BrowseOrderTitle
# text   = org.dspace.browse.BrowseOrderText
#
# If you redefine a default datatype here, the configuration will be used in preference
# to the default, however, if you do not explicitly redefine a datatype, then the
# default will still be used in addition to the datatypes you do specify.
#
# Uncomment the configuration below to use the multi-lingual MARC 21 title ordering.

plugin.named.org.dspace.browse.BrowseOrderDelegate= \
      org.dspace.browse.BrowseOrderTitleMarc21=title
```



```
package org.dspace.browse;

import org.dspace.text.filter.DecomposeDiactritics;
import org.dspace.text.filter.LowerCaseAndTrim;
import org.dspace.text.filter.MARC21InitialArticleWord;
import org.dspace.text.filter.TextFilter;

/**
 * MARC 21 title ordering delegate implementation
 *
 * @author Graham Triggs
 */
public class BrowseOrderTitleMarc21 extends AbstractTextFilterBOD
{
      {
            filters = new TextFilter[] { new MARC21InitialArticleWord(),
                                                           new DecomposeDiactritics(),
                                                           new LowerCaseAndTrim() };
      }
}
```

Unicode Ordering

- IBM's ICU4J used to decompose Unicode characters
- ICU4J is the basis of the JDK unicode handling
- Decomposition is not a public API until JDK 6

```
package org.dspace.text.filter;

/**
 * Define an interface for all browse ordering filters.
 * @author Graham Triggs
 */
public interface TextFilter
{
      public String filter(String str);

      public String filter(String str, String lang);
}
```

```
package org.dspace.text.filter;

import com.ibm.icu.text.Normalizer;

/**
 * Decompose diacritic characters to character + diacritic
 *
 * @author Graham Triggs
 */
public class DecomposeDiactritics implements TextFilter
{
    public String filter(String str)
    {
        return Normalizer.normalize(str, Normalizer.NFD);
    }

    public String filter(String str, String lang)
    {
        return Normalizer.normalize(str, Normalizer.NFD);
    }
}
```

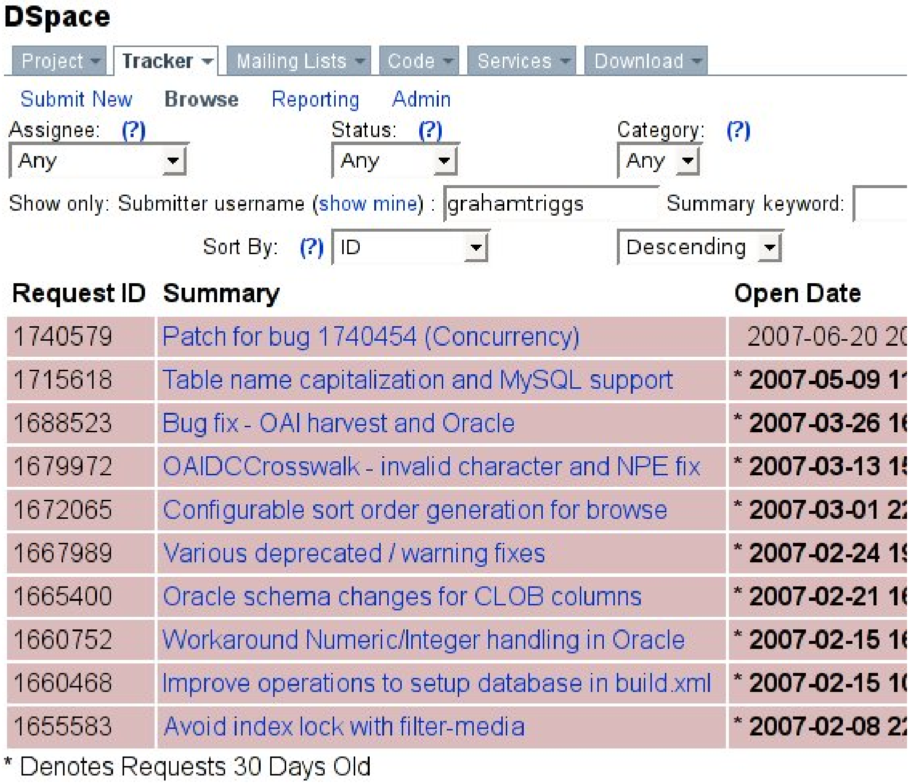
Advanced Ordering

- Sort strings do not necessarily have to be human readable
- Example filter for doing Locale correct ordering (based on a fixed configured Locale, not per-visitor)

```
/**
 * Makes a sort string that is Locale dependent.
 * Uses the same Locale for all items, regardless of source language.
 *
 * You can set the Locale to use by setting 'webui.browse.sort.locale'
 * in the dspace.cfg to an ISO code.
 *
 * If you do not specify a Locale, then it defaults to Locale.ENGLISH.
 *
 * IMPORTANT: The strings that this generates are NOT human readable.
 * Also, you will not be able to meaningfully apply any filters *after* this,
 * however, you can apply other filters before.
 *
 * @author Graham Triggs
 */
public class LocaleOrderingFilter implements TextFilter
{
    private static Logger log = Logger.getLogger(LocaleOrderingFilter.class);

    /**
     * Uses a Locale dependent Collator to generate a sort string
     * @param str The string to parse
     * @return String the sort ordering text
     */
    public String filter(String str)
    {
        RuleBasedCollator collator = getCollator();

        // Have we got a collator?
        if (collator != null)
        {
            int element;
            StringBuffer buf = new StringBuffer();

            // Iterate throught the elements of the collator
            CollationElementIterator iter = collator.getCollationElementIterator(str);

            while ((element = iter.next()) != CollationElementIterator.NULLORDER)
            {
                // Generate a hexadecimal string representaion of the Collation
element
                // This can then be compared in a text sort ;-)
                String test = Integer.toString(element, 16);
                buf.append(test);
            }

            return buf.toString();
        }
...
```

OR Contributions

- Oracle support
- Code cleanup
- Browse ordering
- Code review
  - concurrency issues
  - configurable browse
- Preliminary MySQL support



Assignee: (?) Any

Status: (?) Any

Category: (?) Any

Show only: Submitter username (show mine) : grahamtriggs

Summary keyword:

Sort By: (?) ID Descending

Request ID	Summary	Open Date
1740579	Patch for bug 1740454 (Concurrency)	2007-06-20 2
1715618	Table name capitalization and MySQL support	* **2007-05-09 1**
1688523	Bug fix - OAI harvest and Oracle	* **2007-03-26 1**
1679972	OAIDCCrosswalk - invalid character and NPE fix	* **2007-03-13 1**
1672065	Configurable sort order generation for browse	* **2007-03-01 2**
1667989	Various deprecated / warning fixes	* **2007-02-24 1**
1665400	Oracle schema changes for CLOB columns	* **2007-02-21 1**
1660752	Workaround Numeric/Integer handling in Oracle	* **2007-02-15 1**
1660468	Improve operations to setup database in build.xml	* **2007-02-15 1**
1655583	Avoid index lock with filter-media	* **2007-02-08 2**

* Denotes Requests 30 Days Old

Conclusion

- Open source and community are good things
- Being more closely aligned with the DSpace core is allowing OR to
  - collaborate with other developers
  - support other institutions running DSpace
  - contribute code