Blacklight at Stanford: A Highly Leveraged, Reusable, Discovery Engine

Tom Cramer
Associate Director
Digital Library Systems & Services
December 2009
Blacklight at Stanford

Blacklight is an open source, "next generation" discovery application that works equally well for library catalogs as for digital repositories. Originally developed at the University of Virginia, it has since been adopted by a growing number of institutions, including Stanford University.

Within the Stanford Libraries' digital ecosystem, Blacklight is being leveraged as a central component in a broad range of applications, from a library "discovery layer" to a repository front-end, from a digital manuscript viewer to a catalogers' tool.

This is an overview of Blacklight, its capabilities and its advantages as a highly reusable discovery application. It outlines Stanford's rationale for choosing Blacklight, provides examples of the system in operation in a variety of contexts, and covers the state and direction of both the code and the community for this open source project.
Vast Information Resources

6,825,821

- 6+ million monographic and serial works
- 25,000 E-journal subscriptions
- 800,000 maps & images
- 20 Different libraries
- 800 Licensed databases
- 1 Department of Special Collections
The Features You’d Expect

- Faceted search
- Relevance ranked results
- Personalization (bookmarks, tags)
- Export citations via RSS, SMS, Email, Zotero, RefWorks, EndNote
- Sort by anything <date, author, relevance, title>
- Support for unicode / vernacular scripts
- And much more...
Plus Three Key Features

1. More than MARC: support for any kind of record or metadata

2. Object-specific behaviors
   - Books, Images, Music, Manuscripts, Finding Aids, <any>

3. Tailored views for domain or discipline-specific materials
   - UVa’s “Music Tab”
   - UWisc’s union catalog
Object Specific Behaviors & Non-MARC records

Facets are tailored to numismatics

Note the “Source” facet is the UVa Art Museum tab.

Search results data fields are customized to content type

from http://blacklightdev.lib.virginia.edu
Blacklight: Tailored Views Based on Context

Facets are tailored to Music resources & discipline

Note the “Music” tab.

Search Results are only for Music

from http://blacklightdev.lib.virginia.edu
Blacklight’s Technology Stack

Ruby on Rails Application contains both the Blacklight plugin and local code. Local code augments and over-rides (where needed) the BL plugin.

An underlying Solr index holds data from sources of interest. Indexers prepare and load data into usable form.
Ruby on Rails

- **Rapid** application development for web applications: “Convention over configuration”
  - 10x productivity
- **Supportable**: MVC (Model-View-Controller) structure separates data from logic from presentation
- **Testable**: Rspec and Cucumber give powerful, automatable test coverage tools
- **Learnable**: Stanford went from 1 to 8 Ruby savvy developers in one year (no new hires)
  - 1 week learning curve to basic proficiency
Blacklight at Stanford

- Next generation catalog
- IR discovery (ETD’s and more)
- Digital manuscript search
- Digital archival collection front end
- Repository administrative interface (coming soon)
Hydra: One Solution for Many Needs

Hydra is an effort that is developing and packaging an application framework to sit atop Fedora, and tailoring the use of this framework for specific institutional repository & digital library solutions. It is a joint development project among Stanford, University of Virginia, University of Hull and DuraSpace.

The spirit of Hydra is a common body (framework), many heads (tailored UIs). The Hydra/Fedora hybrid will produce IR applications which support the following common functions:

* Deposit
* Edit & Annotate
* Set Permissions / Access Levels
* Manage collections
* Report
* Browse
* Search
* View Object

Blacklight provides the search, browse & viewing capabilities.
ETD Application

Search & Browse powered by Blacklight

ETD-specific viewing behavior
Displaying items 1 - 10 of 559

1. 1
   Collection: Parker Medieval Manuscripts
   Manuscript Identifier: 1
   Provenance: I can discover in the book no hint of its monastic provenance. The upper R. corner of ff. 1r-2v is mutilated: probably the press-mark was there.

2. 2
   Collection: Parker Medieval Manuscripts
   Manuscript Identifier: 2
   Provenance: The book comes from Bury St Edmunds Abbey. On the upper corner of f. 2r (first leaf of text) is a mark rather smaller than in most Bury books, but of the same kind: B. 1. The press-mark shows that this was the first book in class B. The B here stands for Bibliia. Further, at f. 322r the edge of the leaf has been mended with a patch of vellum in cent. xv on which is sketched a crowned head (cut off at the neck) and a scroll inscribed hic, hic, hic. This represents St Edmund's head, which called out Here, here, to those who were searching for it after the martyrdom. It fixes the provenance in a very satisfactory way. In the old catalogue (cent. xii, xiii) of the Abbey books preserved in a MS. at Pembroke College and printed in my Essays on the Abbey of Bury, 1895, p. 23, the second item is Bibliotheca in duo volumina (!) and in the Gesta Sacristarii, Arnold. Memorials of Bury St Edmunds Abbey (Rolls Series II. p. 290) in the account of Hervey.
A solr index without the Blacklight frontend.
Multi-Institutional Project

- Originated at UVa in 2007 as a research project
  - Moved to production as “Virgo Beta” in 2008
- Stanford adopted in Jan 2009
  - Deployed SearchWorks on Blacklight in Aug ’09
- Currently dozens of installations
- ~10 committers from 6 institutions (UVa, Stanford, Nat’l Library of Agriculture, Johns Hopkins, UWisc)
Test Coverage

• Full test coverage is a core community principle
  – Unit tests with Rspec
  – Acceptance tests with Cucumber
  – Continuous integration testing with Hudson

• Tests ensure...
  – Quality
  – Compatibility
  – Clarity of code and function
  – Confidence
Testing is a Core Community Principle

See http://projectblacklight.org/?page_id=2

- “All contributed code must have full test coverage before it is committed. The current testing infrastructure is RSpec for everything but Rails views, and Cucumber for Rails views.
- “Tests must be committed at the same time code is.
- “All bugs and development tasks will be tracked in JIRA
- “All code must be documented before it’s committed.”
BL’s Current Test Coverage is 90%

<table>
<thead>
<tr>
<th>Name</th>
<th>Total lines</th>
<th>Lines of code</th>
<th>Total coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>app/controllers/application_controller.rb</td>
<td>2073</td>
<td>1357</td>
<td>90.55%</td>
</tr>
<tr>
<td>app/controllers/bookmarks_controller.rb</td>
<td>59</td>
<td>59</td>
<td>96.61%</td>
</tr>
<tr>
<td>app/controllers/catalog_controller.rb</td>
<td>59</td>
<td>8100.00%</td>
<td></td>
</tr>
<tr>
<td>app/controllers/feedback_controller.rb</td>
<td>200</td>
<td>137</td>
<td>96.00%</td>
</tr>
<tr>
<td>app/controllers/saved_searches_controller.rb</td>
<td>37</td>
<td>24</td>
<td>35.14%</td>
</tr>
<tr>
<td>app/controllers/search_history_controller.rb</td>
<td>44</td>
<td>34</td>
<td>79.55%</td>
</tr>
<tr>
<td>app/controllers/user_sessions_controller.rb</td>
<td>25</td>
<td>21</td>
<td>76.00%</td>
</tr>
<tr>
<td>app/controllers/users_controller.rb</td>
<td>43</td>
<td>20</td>
<td>100.00%</td>
</tr>
<tr>
<td>app/helpers/application_helper.rb</td>
<td>8</td>
<td>8100.00%</td>
<td></td>
</tr>
<tr>
<td>app/helpers/bookmarks_helper.rb</td>
<td>3</td>
<td>2100.00%</td>
<td></td>
</tr>
<tr>
<td>app/helpers/catalog_helper.rb</td>
<td>44</td>
<td>29</td>
<td>90.91%</td>
</tr>
<tr>
<td>app/helpers/feedback_helper.rb</td>
<td>2</td>
<td>2100.00%</td>
<td></td>
</tr>
<tr>
<td>app/helpers/saved_searches_helper.rb</td>
<td>2</td>
<td>2100.00%</td>
<td></td>
</tr>
</tbody>
</table>

http://hudson.projectblacklight.org/hudson/job/blacklight-plugin/99/rcov/
The Code Silo Problem

OSS code

Site Specific Code

Site Specific Code

Site Specific Code
The Code Silo Problem

Version 1  Version 2  Version 3

Site Specific Code

Site Specific Code

Site Specific Code
Well-Structured Code

- Blacklight 2.0 was a substantial refactoring to make the code portable
  - Core functions, common to all installations, located in a plugin
  - Local modifications made in the Ruby on Rails application container
  - Over-rides facilitate customization for local needs

- Vendor drops are straightforward
- GIT to facilitate branching and merging
Scalability: SearchWorks = known upper bound

SearchWorks Usage: April – December, 2009

- SearchWorks currently has > 6 Million records
- Peak daily load is now > 26,000 visitors
Why Blacklight?

• Reusable tool for both catalog, repository and digital library application front ends
• Ruby on Rails is an excellent platform choice for digital library development (velocity, supportability, testing)
• Code structure and engineering support distributed development and tailored uses
• UVa (et al.) are excellent community partners
Resources

http://projectblacklight.org