Project Overview

Bibliographic merging, Endeca, and Web application
Three Processes

Merging of bibliographic records
- Pre-processing stage
- 8M contractual record limit

Endeca Forge, Dgidx, and MDEX

Web application
- Presentation platform
- Can be used to present more than data from the MDEX Engine
Merging of bibliographic records

BIB and HOL data extracted from Aleph Oracle (z00) x 11

Merge routine

- 'Endeca Field Mapping and Pipeline' shows the action that is taken during the merge routine for each MARC field
- Deduplication based on OCLC no
- HOL data written into the Union MARC
- The Aleph service p_print_03 is run for all merged records to apply UTF8 encoding and Material Types for the BIB records.
  - Material Types for all SUL BIB records are applied using a single instance of Aleph’s tab_type_config.
  - The Aleph file tab_type_config translates the information encoded in the LDR/008/007/006 of the BIB record into a two-letter material type code that is placed in the BIB record, which is the source of the format facet in Endeca.
- The merge routine happens before the field mapping.

Other data sources in the future (digital libraries)

- Via DLU01
- Direct
Endeca Forge Dgidx and MDEX

Forge is a data processing program
Dgidx is an indexing program
MDEX is the search engine/API that serves data in response to a query, includes all of the information needed to build an entire page

- ‘Endeca Field Mapping and Pipeline’ shows how MARC fields are mapped to Endeca record fields.
- ‘Endeca Dimensions (Facet) Mappings’ shows how MARC fields are mapped to Endeca dimensions (facets).
- ‘Endeca Search Configuration’ shows the search 'interfaces' and the Endeca record fields that are searched
Forge is a data processing program

Endeca provided a custom MARCadapter to transform MARC records into records that are readable by Forge

- FCLA has modified the “MARCadapter” files to help define the “Online” format for bibliographic records and assist with other features of the WebApp where there needs to be a custom field in the Endeca record.

- E.g. we apply the Online format code to the record based on the presence of http in $u of the 856 but excluding "table of contents," "publisher," "sample text," or "contributor."

Transforms your source data into standardized, tagged Endeca records

Each record has a list of dimension (text) values tagged to it.
Dgidx

dexing program that reads the tagged Endeca records that were prepared by Forge
Creates the proprietary indices for the Endeca MDEX Engine
  – Dgraph: An Index for every N-value
  – Entire Endeca Database stored in memory
  – Output stored in directories the file system of the Endeca box.

Indexing Configuration in Endeca (pipeline) includes:
  – Stop words
  – Character normalization and ‘internationalization’
  – Thesaurus and Stemming (automatic)
  – Taxonomies (‘hierarchies’) e.g. LCC/NLM
  – Search Configuration (‘interfaces’)
  – Relevance ranking
  – DYM and Spell Correction
  – Truncation
MDEX Engine

Serves data in response to a query via the API, includes all of the information needed to build an entire page.

Queries include Search and Navigation.

An entire page (object) is returned in response to a query, constructed from a subset of the Dgraph.

Subsequent navigation is applied to this object, not the entire Dgraph (Follow-up queries are faster).
“WebApp”

Apache Tomcat and JSP

- Maintains a connection state with Endeca Nav Engine (Dgraph/MDEX)
- Similarly, maintains a connection with ORACLE via JDBC
- Restarted every morning (with refreshed configuration files).
"WebApp"

Other than the Forge, Dgidx programs, and some control scripts, this is what Endeca provides:

Endeca API includes:
- Http Connections into the MDEX
- Method to query via a URL
- An result object that can be parsed and manipulated for display
- A method to get the dimensions, dimension values, and corresponding IDs for a particular Navigation state.
- Other Classes and Methods
- Boolean query mode and other query match modes
  - Interaction with other features (no stop words, stemming, spelling, thesaurus, ranking)
  - Proximity searching (NEAR/n, ONEAR/n)

Statistical Report (See [http://www.fcla.edu/FCLAinfo/stats/endeca_stat/endeca_stat.html](http://www.fcla.edu/FCLAinfo/stats/endeca_stat/endeca_stat.html) and linked document that explains the reporting categories)

Everything else is a customization of the Web application by FCLA.
“WebApp” Features

Hooks into Aleph
– Display of item information
  • Sublibrary/Collection (Aleph tabs)
  • According to item status (Aleph tabs)
  • Availability (Circ) status (SQL)
  • Detailed holdings (SQL)
– Patron Empowerment
  • Loans list (SQL)
  • Renewals (API)
  • Requests and Holds (API)
– SFX Contextual links for Full Text (Query SFX server)
– List Functions and Session ID (SQL)
“WebApp” Features

Custom Features

– Advanced Search and limits
– RSS for Result Set
– Email list of records
– Hooks to RefWorks
– Permalink
– Marc Views via SQL
– Debug View (raw Endeca record)
– Browse lists (in development)
– Book covers (in development)
– …What’s next?
Development and System Environment

The Endeca MDEX Server
- 125 gig mem
- Uses 317 gig of file space (out of 392 gig).
- Each dgraph uses about 16% of the memory
  - 1-5 dgraphs are running at a given time.
  - Each dgraph uses about 20 gig memory.

The WebApp Server
- 7 gig mem.
- File space is not an issue. No dgraphs.
- Handles 19 meg http requests.

Forge box
- 32 gig mem.
- 197 gig file space.
- Runs dgidx which uses 17 gigs, and forge which uses 2.4 gigs.

Subversion
- svnlog (http://catalog.fcla.edu/svnlog.xml)