Data Feeds and the Library

What we've learned
• Jon Gorman, Library IT
  – jtgorman@illinois.edu
Introduction – Goals

• Better understanding of our data sources and what information we need to function.
• How we use that information.
• Share mistakes we’ve made and hopes for the future.
• Start discussion on ways we can improve
Introduction – The Library

• Part of CARLI – Consortium of Academic and Research Libraries in Illinois
• CARLI hosts our instance of Voyager – the Integrated Library System (ILS).
• Voyager keeps track of circulation and inventory.
• Shared Voyager service allows easy borrowing and lending between CARLI members.
The Data Feeds

- Billing
- Voucher
- Patron
Billing

- charges & credits for students, staff, and faculty for fines and libprint charges
Billing

Voyager SIF feed → Internal DB → AITS → Reconcile Document Direct Files → Produce Reports for Billing Office
Invoices/Vouchers

- purchases from vendors stored in Voyager.
Vouchers

1. Voyager DB Query
2. Internal Database
3. AITS
4. Reports printed out next day
Patron Feed

• Gathering information on patrons for services such as checking out materials, off-campus electronic resource use, and inter-library loan.
Patron Feed - Inputs

I-Card → LDAP → Patron DB → Voyager
Patron Feed – Sources

- I-Card feed
- LDAP Query (for email address)
- Voyager Library System – feedback loop for manual entries since last load
Patron Feed - Destinations

Patron DB

Voyager

ILLIAD

Voyager
Patron Feed - Destinations

- Voyager
- Off-campus proxy (access to electronic resources)
- InterLibrary Loan system – borrow books from libraries outside of the consortium
Patron Feed – Information Needs

- Address in banner – billing, sending material to on-campus offices
- Affiliation
- Employee class
- Enrollment status

Last three needed for: Mapping to Voyager Patron Groups & License requirements for electronic materials
Lessons We’ve Learned
Brief History

• Different people, different technologies, and partial but incomplete ports lead to wide diversity within even one feed:
  – Cron, Scheduled task, sql server jobs
  – perl scripts, VB executables, stored procedures, DTS packages, and other application calls.
Lesson: Backup

- Backup. Soon or later, something really, really bad will happen.
- Similar to backup, try to make processes reversible.
Lesson: Version Control

- Get history of development, better tracking of “why did they do x?” for group of changes
- Reduces number of files (foo.1.pl, oldfoo.pl, etc)
- Easy to revert if needed
Lesson: Standardize/Simplify

• Choose Perl, made up majority of processes and was known by most of Library IT
Lesson: Config Files

- Servers change.
- Changing hardcoded values dangerous.
- Can make it easy to change between production and test environment.
- Using Config::General for perl.
Lesson: Test

- Have a test environment that mirrors production
- Test cases to test specific behaviors – using Test::More
Lesson: Log

- Use a standard library like Log4Perl.
- Makes it easier to change log settings on all parts of a feed.
- Easy to adjust verboseness.
- Also de-couples alerts – can change config file and libraries to add/modify alerting mechanisms.
Lesson: Error check

- Whenever you make a connection, parse a file, or generally interact with another system
- TEST the error checking. Break the system and see if it responds properly.
Lesson: Automate

- Humans make mistakes, some more than others
- Consistency – a computer will fail the same way with the same inputs.
- Be careful of automating without error checking, can be catastrophic
What was, what is, and what may be

- **Was:** ad-hock, intertwined, no error checks, sequential without feedback
- **Is:** Version control, one platform, some test cases, testing environments, mostly automated, better error checking, logging
- **May be:** Full regression testing suite, highly automated, modular code, better separation of systems
Perl Tips

- Binary files? Use binmode.
- Wrapper script used “do” execute:
  - do downloadfiles.pl
  - Sets $! And $@
- Die callback - $SIG{__DIE__}
- Don’t copy and paste, modules can be your friend.
- Perl Best Practices by Conway