

# What to Do When They Say “It Doesn’t Work”

Jane Wagner

Senior Project Manager

LibLime

Prepared for KohaCon 12, June 6, 2012

# System = Support Needs

- If you build it, they will come
- And they will need support
- Bugs are inevitable
- So are users who don't know how to do something
- How to tell the difference?
- And what do you do about it?

# Information Gathering

- To solve a problem, you have to diagnose and replicate it
- To diagnose and replicate, you need details from the user
- To get details, you have to train your users to give specifics about the problem

## Who and What

- Who is reporting the error? Staff or patron?
- Is that person well trained and experienced in the system?
- What was the person doing?
- What patron record, item record, etc.?
- Can you replicate the problem?
- Does it happen for all users/transactions?

# When

- When did the problem occur?
- Any known network or server issues at that time?
- Does the action depend on a cron job, like the holds queue?
- Is it a response time issue (very large query)?
- Was the system code recently upgraded? If so, did this action work before the upgrade?

## Where/How and Why

- Was the user in staff or OPAC?
- What screen? Which module?
- Get screenshots; copy and paste any error messages
- Can you replicate?
- What was the user trying to accomplish?
- State the goal – the user might be going about it the wrong way

# Education

- Provide enough training up front for staff
- Provide good documentation
- Develop your own FAQ over time
- Give patrons help screens and tips in OPAC
- When you solve a problem, give details – why did the problem happen and how can it be avoided next time?

# Reporting Bugs

- Learn the code structure so you can backtrack a problem if necessary
- Search the bugs database at <http://bugs.koha-community.org/bugzilla3/> and report bugs if needed
- Create a patch if you can -- <http://wiki.koha-community.org/wiki/SubmittingAPatch>
- Sign off on patches if you can -- [http://wiki.koha-community.org/wiki/Sign off on patches](http://wiki.koha-community.org/wiki/Sign_off_on_patches)



# Common Problems

- Scenarios for
  - Checking out or placing holds
  - Searching
  - Notices
  - Fines
  - Reports

# Checkout/Hold Blocked

- Need exact patron and item ID
- What is CircControl syspref set to? That determines which circ rules apply
- Check applicable circ rules – is there one allowing that patron/item combination?
- Will the action work with another patron or item?
- Also check for other settings, such as on-shelf holds allowed or limits for amount owed

# Limited Circ Rule

Patron category	Item type	Current checkouts allowed	Loan period	Unit	Hard due date
Juvenile	All	10	14	days	None defined
All	All	Unlimited	14	days	None defined
All	All	<input type="text"/>	<input type="text"/>	Days	Before <input type="text"/> (MM/DD/YYYY)

Fine amount	Fine charging interval	Fine grace period (day)	Suspension in days (day)	Renewals allowed (count)	Holds allowed (count)	Rental discount (%)	
0.25	1	0	0	2	5	0.000000	Delete
0.25	1	0	0	5	50	0.000000	Delete
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Add

## Useful screen error messages (OPAC)

**Sorry**, none of these items can be placed on hold.

Title
<a href="#">Steinway &amp; Sons /</a> , by Lieberman, Richard K.
You have already requested this title.



Sorry, you cannot place holds because you owe \$100.00.

## Useful screen error messages (staff)

### Please confirm checkout

- Too many checked out. 3 checked out, only 2 are allowed.

✓ Yes, Check Out (Y)

✗ No, Don't Check Out (N)

---

### Please confirm checkout

- The patron has a debt of 100.00
- Item *Treasure box* : (39344004464761) is checked out to **PTFS STAFF** (admin\_2).  
Check in and check out?

✓ Yes, Check Out (Y)

✗ No, Don't Check Out (N)

# Misleading screen error messages

- Items are available; this actually results from a limit on number of holds in the circ rule:

**Sorry**, none of these items can be placed on hold.

Title
<a href="#">The country ahead of us, the country behind :</a> , by Guterson, David.
No available items. ←

# Searching Problems

- No searches produce results anywhere? Check to be sure zebra process is running
- Stop and restart zebra if necessary
- If staff is searchable but OPAC isn't, check OpacSuppression syspref – if on, do you have at least one suppressed item?
- Can search in general but not for recently added titles? What's the timing on the update cron job?

## Searching Problems

- Check zebraqueue table to see if the title is still waiting for indexing:

```
select zebraqueue.id, zebraqueue.biblio_auth_number,  
zebraqueue.operation, zebraqueue.server,  
zebraqueue.done, zebraqueue.time, biblio.title from  
zebraqueue join biblio on  
(zebraqueue.biblio_auth_number = biblio.biblionumber)  
where done = '0' and zebraqueue.server = 'biblioserver'
```



## Stuck in zebraqueue?

- What's the timestamp on the oldest entry?
- Try running the zebra update manually – any errors?
- Check the oldest bib record – corrupt or wrongly-encoded MARC records can block processing.  
Overlay or remove the problem record – get to it manually by constructing a URL with its biblionumber like `http://my.system.url/cgi-bin/koha/catalogue/detail.pl?biblionumber=39`

# Notices

- Patrons complain about not getting overdue, hold, advance, other notices
- Check the patron's enhanced messaging options – are those notice types selected?
- Is the patron category set to get overdue notices?
- Check the cron jobs – are overdue and advance notice crons running?
- Check the notice triggers for overdue notices

# Message Queue Queries

- Check the message queue to see what was generated

- List all, most recent first:

```
select * from message_queue order by time_queued desc
```

- Limit to one type of notice:

```
select * from message_queue where letter_code = 'HOLD' order by time_queued desc
```

# Message Queue Queries

- Limit to one patron record:

```
select * from message_queue where borrowernumber  
= '12345' order by time_queued desc
```

- Limit to a particular date:

```
select * from message_queue where time_queued like  
'2012-06-04%' order by time_queued desc
```

## Message Found?

- If the message is not in the queue, Koha did not generate it – why? Can you replicate?
- If it's in the queue, what's the status?
  - “sent” was handed off to mail server (is that running OK?)
  - “pending” is waiting for process\_message\_queue cron
  - “failed” indicates a problem with patron's email address
- If message shows as sent, have patron check their junk mail folder or their ISP for spam or block

# Fines

- Fines – check circ rules as with checkout/hold problems. Find which rule applies and see what it has for fine charge
- Is there a fine charging interval set in circ rule?
- Check your issues table for orphaned records (null itemnumber or borrowernumber) – these can choke the fines script.
- Is finesMode syspref set to production mode?

# Reports

- Reports – delivered reports and guided reports are very limited, usually need to develop saved SQL specific to needs
- Learn table structure and SQL query syntax
- Build up your own library of reports
- See

[http://wiki.koha-community.org/wiki/SQL\\_Reports\\_Library](http://wiki.koha-community.org/wiki/SQL_Reports_Library)

# Most Important Tables for Reports

- Most of the data you want will be in biblio, biblioitems, items, borrowers, issues, reserves, accountlines
- Sometimes you need others, like branchtransfers, borrower\_message\_preference, deletedbiblio, deleteditem, old\_issues, or old\_reserves
- See [http://wiki.koha-community.org/wiki/DB\\_Schema](http://wiki.koha-community.org/wiki/DB_Schema) for several versions



## Database Schema

- If your version of MySQL doesn't restrict access, you can query your database schema directly:

```
select TABLE_NAME, COLUMN_NAME from  
information_schema.columns where table_schema  
!= 'information_schema' order by TABLE_NAME asc
```

(or select \* for more information)

## Sample Reports

- Patrons who owe more than 25.00:

```
SELECT borrowers.surname, borrowers.firstname,  
borrowers.cardnumber,  
round(Sum(accountlines.amountoutstanding),2) as 'Owed'  
FROM accountlines JOIN borrowers on  
(accountlines.borrowernumber = borrowers.borrowernumber)  
group by accountlines.borrowernumber having  
sum(accountlines.amountoutstanding) > 25 order by  
borrowers.surname, borrowers.firstname
```

## Sample Reports

- Titles with patron-added reviews:  
select biblio.title, borrowers.surname,  
borrowers.firstname, borrowers.cardnumber,  
reviews.review, reviews.approved,  
reviews.dateresolved from reviews JOIN biblio on  
(reviews.bibliionumber = biblio.bibliionumber) JOIN  
borrowers on (reviews.borrowernumber =  
borrowers.borrowernumber) order by biblio.title asc

## Sample Reports

- Previous seven days' checkouts and renewals, by day of week:

```
SELECT DAYNAME(statistics.datetime) AS 'Weekday',  
statistics.type ,count(*) FROM statistics WHERE  
statistics.datetime >= (now() - interval 7 day) and  
statistics.type in ('issue', 'renew') GROUP BY  
DAYNAME(statistics.datetime), statistics.type ORDER BY  
statistics.type, DAYNAME(statistics.datetime) asc
```

## Sample Reports

- Items checked out today, by branch and patron:

```
SELECT borrowers.surname, borrowers.firstname,  
borrowers.cardnumber, biblio.title, biblio.author,  
items.itemcallnumber, items.barcode, issues.branchcode,  
issues.timestamp FROM issues JOIN borrowers ON  
(borrowers.borrowernumber=issues.borrowernumber) JOIN  
items ON (issues.itemnumber=items.itemnumber) JOIN biblio  
ON (items.biblionumber=biblio.biblionumber) WHERE  
date(issues.issuedate) = date(now()) ORDER BY  
issues.branchcode, borrowers.surname, borrowers.firstname
```

## Sample Reports

- To find patrons with particular messaging options, first

```
select * from message_attributes
```

- to identify the message\_attribute\_id number for the notice you want. Then use this to see who has e.g. Advance Notice (# 2 on this system) turned on:

## Sample Reports

```
select borrowers.cardnumber, borrowers.surname,  
borrowers.firstname from  
borrower_message_preferences JOIN borrowers on  
(borrower_message_preferences.borrowernumber =  
borrowers.borrowernumber) where  
borrower_message_preferences.message_attribute_id  
= 2 and  
borrower_message_preferences.days_in_advance != 0  
order by borrowers.surname, borrowers.firstname
```

# Questions?

Jane Wagner

Senior Project Manager

LibLime, a division of PTFS

Content Management and Library Solutions

11501 Huff Court

North Bethesda, MD 20895

(301) 654-8088 x 151

[jwagner@liblime.com](mailto:jwagner@liblime.com)