ILLiad Routing Rules

&

IDS Logic Rules
ILLiad Routing Rules
Which routing rules do we recommend?

- Rule of 5
- Direct Request Rule
- Document Type Rule
- Others?
### Anatomy of a routing rule

- **RuleNo**: Each rule waits in line. This sets the order.
- **RuleActive**: The on/off switch.
- **ProcessType**: Apply this rule to Borrowing, Lending, or Doc Del.
- **TransactionStatus**: Apply this rule to a specific queue.
- **MatchString**: Apply this rule if it matches your criteria.
- **NewProcessType**: Move the request to Borrowing, Lending, or Doc Del.
- **NewTransactionStatus**: Move the request to a specific queue.
- **RuleDescription**: A brief note explaining what this rule does.

### Biology class: We’re dissecting a routing rule!

<table>
<thead>
<tr>
<th>RuleNo</th>
<th>RuleActive</th>
<th>ProcessType</th>
<th>TransactionStatus</th>
<th>MatchString</th>
<th>NewProcessType</th>
<th>NewTransactionStatus</th>
<th>RuleDescription</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Yes</td>
<td>Borrowing</td>
<td>Awaiting Request Processing</td>
<td><code>len(t.LoanDate) = 4 and t.LoanDate = convert(varchar(4),(datepart(year,getdate())[)))</code></td>
<td>Doc Del</td>
<td>Awaiting Acquisitions Processing</td>
<td>Routes new releases to Acquisitions for automatic purchase consideration.</td>
</tr>
</tbody>
</table>
Where can I find out more about Tables & Fields?

Table names:
Users
Transactions
Tracking
History
LenderAddresses
Notes
ReasonsForCancellation
LendingReasonsForCancellation

URL for Table (older version):
https://prometheus.atlas-sys.com/display/illiad85/ILLiad+Database+Key+Table+Relationships
Tables that you will most likely use:

- Users
- Transactions

To query these tables, you need to append the first letter of the table name to the field that you want to query.

Example:

Users.FirstName would be u.FirstName
Transactions.Location would be t.Location
You can’t spell MatchString without SQL

- Boolean operators: AND, OR, NOT
- Comparison operators: =, <>, >, <, >=, <=
- Use NULL to test for empty fields
- Can incorporate advanced SQL functions (i.e. “select”)
- Never use double-quotes
- All fields have table prefixes (i.e. “u.Status”)
- Don’t exceed 255 characters
What can a MatchString do?

• MatchStrings can look at data from:
  – Transactions table
  – Users table (Borrowing & Doc Del)
  – LenderAddresses table (Lending)
• Route between Borrowing & Doc Del
• Move to any queue – including custom queues
How do you make the statement properly?

Tables: Spread, Bread

Statement:
Spread = ‘Peanut Butter’ AND Spread = ‘Jelly’ AND (Bread = ‘White’ OR Bread = ‘Wheat’)

Take note:
• Define table each time
• Use single quotes
• Use parenthesis around OR statements
• Capitalize Boolean Operators for readability
Routing rule rules to live by

- Routing rules cannot update an OCLC status
- No daisy chaining
- Order is important
- Requests can get stuck in loops
- Too many rules will slow down ILLiad
- Rube Goldbergs can break ILLiad
Now’s your turn! Let’s create a routing rule together!

Task: Create a routing rule that takes TNs from Awaiting Lending Request Processing that have a location of “BunniesField” and route them to Awaiting BunniesField Processing.

<table>
<thead>
<tr>
<th>RuleNo</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td>Yes</td>
</tr>
<tr>
<td>ProcessType</td>
<td>Borrowing</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>Awaiting Request Processing</td>
</tr>
<tr>
<td>MatchString</td>
<td><code>len(t.LoanDate) = 4 and t.LoanDate = convert(varchar(4),(datepart(year,getdate()))))</code></td>
</tr>
<tr>
<td>NewProcessType</td>
<td>Doc Del</td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td>Awaiting Acquisitions Processing</td>
</tr>
<tr>
<td>RuleDescription</td>
<td>Routes new releases to Acquisitions for automatic purchase consideration.</td>
</tr>
</tbody>
</table>
## Routing Rule Template

<table>
<thead>
<tr>
<th>Rule Number</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td></td>
</tr>
<tr>
<td>ProcessType</td>
<td></td>
</tr>
<tr>
<td>TransactionStatus</td>
<td></td>
</tr>
<tr>
<td>MatchString</td>
<td></td>
</tr>
<tr>
<td>NewProcessType</td>
<td></td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td></td>
</tr>
<tr>
<td>RuleDescription</td>
<td></td>
</tr>
</tbody>
</table>
# Routing Rule Solution

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td>Yes</td>
</tr>
<tr>
<td>ProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>Awaiting Lending Request Processing</td>
</tr>
<tr>
<td>MatchString</td>
<td>t.Location LIKE ‘%BunniesField%’</td>
</tr>
<tr>
<td>NewProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td>Awaiting BunniesField Processing</td>
</tr>
<tr>
<td>RuleDescription</td>
<td>Routes items within BunniesField collection to special processing queue</td>
</tr>
</tbody>
</table>
Advanced Routing Rules: Copyright

Borrowing - Skipping Copyright Processing

In this example, you want to send all requests that are older than 5 years straight to Awaiting Request Processing, bypassing Awaiting Copyright Clearance.

<table>
<thead>
<tr>
<th>RuleNo</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td>Yes</td>
</tr>
<tr>
<td>ProcessType</td>
<td>Borrowing</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>Awaiting Copyright Clearance</td>
</tr>
<tr>
<td>MatchString</td>
<td>t.RequestType = 'Article' and len(t.PhotoJournalYear) = 4 and t.PhotoJournalYear &lt; convert(varchar(4),(datepart(year,getdate()) - 5))</td>
</tr>
<tr>
<td>NewProcessType</td>
<td>Borrowing</td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td>Awaiting Request Processing</td>
</tr>
<tr>
<td>RuleDescription</td>
<td>Bypasses Copyright Clearance for articles older than 5 years</td>
</tr>
</tbody>
</table>

MatchString: Finding TNs that are Request Type = Article and the length of the PhotoJournalYear is equal to 4 characters and the TNs’ PhotoJournalYear value is less than the current year minus 5
(select 1 FROM GISTWeb gw WHERE gw.TransactionNumber = t.TransactionNumber AND PurchaseRecommendation='Yes') = 1 AND u.Status='Faculty' AND (u.Department='biology' OR u.Department='education' OR u.department='foreign languages')
Where’s Waldo, errr... Where’s the error?
What’s wrong with this picture?

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td>Yes</td>
</tr>
<tr>
<td>ProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>Awaiting Lending Request Processing</td>
</tr>
<tr>
<td>MatchString</td>
<td>t.Location LIKE ‘%BunniesField’</td>
</tr>
<tr>
<td>NewProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td>Awaiting BunniesField Processing</td>
</tr>
<tr>
<td>RuleDescription</td>
<td>Routes items within BunniesField collection to special processing queue</td>
</tr>
</tbody>
</table>
### What’s wrong with this picture?

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td>No</td>
</tr>
<tr>
<td>ProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>Awaiting Lending Request Processing</td>
</tr>
<tr>
<td>MatchString</td>
<td>t.Location LIKE ‘%BunniesField’</td>
</tr>
<tr>
<td>NewProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td>Awaiting BunniesField Processing</td>
</tr>
<tr>
<td>RuleDescription</td>
<td>Routes items within BunniesField collection to special processing queue</td>
</tr>
</tbody>
</table>
What’s wrong with this picture?

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td>No</td>
</tr>
<tr>
<td>ProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>Awaiting Lending Request Processing</td>
</tr>
<tr>
<td>MatchString</td>
<td>u.TransactionStatus = ‘Exhausted’ AND t.TransactionStatus = ‘Undergraduate’</td>
</tr>
<tr>
<td>NewProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td>Awaiting BunniesField Processing</td>
</tr>
<tr>
<td>RuleDescription</td>
<td>Routes items within BunniesField collection to special processing queue</td>
</tr>
</tbody>
</table>
What’s wrong with this picture?

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td>No</td>
</tr>
<tr>
<td>ProcessType</td>
<td>Lending</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>Awaiting Lending Request Processing</td>
</tr>
</tbody>
</table>

**MatchString**

- u.TransactionStatus = ‘Exhausted’ OR 
  - t.TransactionStatus = ‘Undergraduate’ OR 
  - t.RequestType = Article OR 

**NewProcessType** | Lending |

**NewTransactionStatus** | Awaiting BunniesField Processing |

**RuleDescription**
Routes items within BunniesField collection to special processing queue
Now it’s your turn!

Create a Routing Rule to share out to the group
# Routing Rule Template

<table>
<thead>
<tr>
<th>Rule Number</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td></td>
</tr>
<tr>
<td>ProcessType</td>
<td></td>
</tr>
<tr>
<td>TransactionStatus</td>
<td></td>
</tr>
<tr>
<td>MatchString</td>
<td></td>
</tr>
<tr>
<td>NewProcessType</td>
<td></td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td></td>
</tr>
<tr>
<td>RuleDescription</td>
<td></td>
</tr>
</tbody>
</table>
10 minute break
IDS Logic Rules
In Transit to Pickup Location

Skip In Transit to Pickup Location for all Loan Requests

This rule will route all loan requests directly from In DD Stacks Searching to Awaiting Doc Del Customer Contact, skipping the In Transit to Pickup Location status.

<table>
<thead>
<tr>
<th>RuleActive</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessType</td>
<td>Borrowing</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>In Transit to Pickup Location</td>
</tr>
<tr>
<td>MatchString</td>
<td>t.RequestType = 'Loan'</td>
</tr>
<tr>
<td>NewProcessType</td>
<td>Borrowing</td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td>Awaiting Customer Contact</td>
</tr>
<tr>
<td>RuleDescription</td>
<td>Moves all loan requests straight to Awaiting Customer Contact (bypassing In Transit to Pickup Location) for immediate notification.</td>
</tr>
</tbody>
</table>

OR...do this through IDS Logic, with a timer!
LogicRule: In Transit to Pickup Location

---

# BorrowingLoanDelayReceived: InTransitToPickupLocationDelayer #

LRBorrowingLoanDelayReceived:
  NVTGCList:
    - ILL

UpdatedMinutes: 2
How to Build Your Own Logic Rule

LogicRule_1:
   Enabled: "0"
   RuleName: "BorrowingUserAccountSuspended"
   PreventReruns: "1"
   NVTGCList:
      - "ILL"
   ProcessTypeList:
      - "Borrowing"
   RequestTypeList:
      - "Loan"
   TransactionStatusList:
      - "Checked Out to Customer"
   FlagList: ""
   Query: ""
# This is anything that was updated within X minutes
   UpdatedMinutes: ""
# This is anything created within X minutes
   CreatedMinutes: ""
# This is anything that was marked as due within X minutes
   DueMinutes: ""
# This is anything overdue
   OverDueMinutes: "30240"
   RouteTo: ""
   RouteToBorrowing: ""
   RouteToDocDel: ""
   AddFlag: "Overdue: 14+ Days"
   RemoveFlag: ""
   EmailTemplate: "Borrowing User Account Suspended"
   EmailSubject: "Borrowing User Account Suspended"
   AddNote: ""
   SetFields:
      - ItemInfo1: ""
      - ItemInfo2: ""
      - ItemInfo3: ""
      - ItemInfo4: ""
      - ItemInfo5: ""
# The valid actions for below are "Cancel", "ILLAnswer – Unfilled", "Cond", "ILLAnswer – Conditional", "Shipped"
   OCLCUpdate: ""
LogicRule: The Search Part

LogicRule_1:
   Enabled: "0"
   RuleName: "BorrowingUserAccountSuspended"
   PreventReruns: "1"
   NVTGCList:
      - "ILL"
   ProcessTypeList:
      - "Borrowing"
   RequestTypeList:
      - "Loan"
   TransactionStatusList:
      - "Checked Out to Customer"
   FlagList: ""
   Query: ""

# This is anything that was updated within X minutes
   UpdatedMinutes: ""
# This is anything created within X minutes
   CreatedMinutes: ""
# This is anything that was marked as due within X minutes
   DueMinutes: ""
# This is anything overdue
   OverDueMinutes: "30240"

*Anything with “List” can have multiple values

(example: RequestTypeList could have both Borrowing and Lending)
LogicRule: The Action Part

RouteTo: ""
RouteToBorrowing: ""
RouteToDocDel: ""
AddFlag: "Overdue: 14+ Days"
RemoveFlag: ""
EmailTemplate: "Borrowing User Account Suspended"
EmailSubject: "Borrowing User Account Suspended"
AddNote: ""
SetFields:
  - ItemInfo1: ""
  - ItemInfo2: ""
  - ItemInfo3: ""
  - ItemInfo4: ""
  - ItemInfo5: ""

*Queues do not need to exist in advance
*Flags and email templates needs to exist in advance
*SetFields can be any field within the Transactions table

#The valid actions for below are “Cancel”, “ILLAnswer – Unfilled”, “Cond”, “ILLAnswer – Conditional”, “Shipped”
OCLCUpdate: ""
How to start: Ask yourself these questions:

- What is the process type?
- What is the request type?
- What is the transaction status?
- Does the TN have any flags attached?
- Is there a specific query you want to run to identify the TN?
- Is this relevant: Created minutes, Updated minutes, Due minutes, Overdue minutes?

Then...

What do you want to do to the TN?

- Do you want to route it somewhere?
- Do you want to change the process type?
- Do you want to add a flag?
- Do you want to remove a flag?
- Do you want to leave a note?
- Do you want to send an email?
- Do you want to add additional information to the TN?
Things to consider:

• It is recommended that these run on a test queue first!

• LogicRules automatically leave notes behind

• You can daisy chain Logic Rules

• You can prevent them from re-running on a TN

• Logic Rules run in order

• You will still need to contact me to set these up at your site
Now it’s your turn!

Create a LogicRule to share out to the group
Bonus!

How to Create a Test Queue
**Routing**

**Edit Row**

<table>
<thead>
<tr>
<th>RuleNo</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>RuleActive</td>
<td>Yes</td>
</tr>
<tr>
<td>ProcessType</td>
<td>Borrowing</td>
</tr>
<tr>
<td>TransactionStatus</td>
<td>Acq Request Processing</td>
</tr>
<tr>
<td>NewProcessType</td>
<td>Doc Def</td>
</tr>
<tr>
<td>NewTransactionStatus</td>
<td>Needs Faculty Rep Review</td>
</tr>
<tr>
<td>RuleDescription</td>
<td>Routing rule sending purchase requests from specific departments to queue for rep review (overflow)</td>
</tr>
</tbody>
</table>
Soon there will be a form online where you can submit your LogicRule ideas.

For now, email me (jonesw@geneseo.edu) or call me (585-245-5172) and let me know about your ideas!
Questions?
Thank you!

http://rulekit.idsproject.org

jonesw@geneseo.edu