ATLAS SYSTEMS · ILLIAD USERS GROUP

OpenURL

PART 1

What OpenURL Is, and Why It Matters

PART 1

What Is OpenURL?

- An industry standard (NISO Z39.88) for packaging citation metadata — title, author, ISSN, genre, and more — into a URL
- It lets a link in one system (the "source," e.g. WorldCat, EDS, Alma) open a service in another system (the "target") that already knows what the patron was looking at
- ILLiad is almost always acting as the target: a patron clicks a "Request via ILL" style link in their database, and lands inside ILLiad's own web request form — already filled in
- No re-keying of citation data means fewer typos, fewer incomplete requests, and a much faster patron experience

PART 1

Why OpenURL Matters for ILL Workflows

Patron experience — One click from the database record instead of retyping an entire citation into a blank web form.

Data quality — Fields come straight from the source record, not from memory — fewer misspelled titles, wrong volumes, or missing ISSNs.

Staff time — Fewer "can you fix this citation" edits later in Borrowing Processing means less rework for your team.

Discovery — At many libraries, OpenURL links are how patrons find ILLiad in the first place — it's often their very first touchpoint.

PART 1

Where OpenURL Lives in ILLiad

- OpenURL is built directly into the ILLiad Web DLL — it is not a separate install or add-on
- Rule of thumb: if your ILLiad web pages work, ILLiad OpenURL works too
- **What you DO have to configure yourself:**
  - Your discovery/resolver system (WorldCat, EDS, Alma, etc.) needs to be told where to send its OpenURL link
  - ILLiad's OpenURLMapping table needs to know how to interpret whatever arrives
- That second point — the OpenURLMapping table — is the ILLiad-specific piece we'll spend the most time on today

PART 2

How ILLiad Handles a Request

PART 2

The Request Flow, Step by Step

1. **Patron is viewing a record** — In WorldCat, EDS, Alma, or another OpenURL-enabled database/discovery layer.
2. **Patron clicks the ILL / OpenURL link** — Configured by your library inside that source system.
3. **Citation data travels to ILLiad** — Title, author, ISSN, genre, and more ride along as tagged fields in the URL.
4. **NVTGC routes to the correct site** — Matters most on Shared Servers with multiple ILLiad sites.
5. **Patron logs on** — The familiar ILLiad OpenURL Logon page appears.
6. **The right request form opens, pre-filled** — Loan or Article page, populated from the citation data.
7. **Patron reviews and submits** — Transaction lands in Awaiting Request Processing or Awaiting Copyright Clearance, same as any other borrowing request.

PART 2

The OpenURL Server URL

Single Server

```
https://yourserver/illiad/illiad.dll/OpenURL
```

- This is the one address every source system needs — no site code required.
- It's the same base URL your ordinary ILLiad web pages already use, with /OpenURL appended.

Shared Server (example site code: ABC)

```
https://yourserver/illiad/ABC/illiad.dll/OpenURL
```

- The NVTGC site code goes right after /illiad/ — this is what routes the request to the correct site.
- If using the Atlas Authentication Portal, the link instead needs the /LOGON/?Action=10&Form=30 pattern.

PART 2

A Quick Look Inside an OpenURL Link

- An OpenURL is just the server URL plus a query string of tagged metadata — you don't need to hand-build these, but recognizing the tag names helps enormously when troubleshooting
- **Common tags you'll see referenced in the mapping table:**
  - rft.genre — what kind of item this is (article, book, conference, etc.)
  - rft.title / rft.atitle — item or article title
  - rft.au, rft.aulast, rft.aufirst — author name, whole or split
  - rft.issn / rft.isbn / rft.eissn — standard identifiers
  - rft.date — publication date
- rfr_id / sid — identifies which source system sent the link
- ILLiad doesn't care which vendor generated the link — only that the tag names match what the OpenURLMapping table is expecting

PART 3

Inside the OpenURLMapping Table

PART 3

Introducing the OpenURLMapping Table

- Location: Customization Manager → Web Interface → OpenURL → OpenURLMapping
- Pre-seeded out of the box to work correctly with most default OpenURL sources — you are rarely building this from scratch
- This table is THE place to change how an incoming OpenURL request gets translated into an ILLiad transaction
- Most real-world work here is adjusting or adding entries for a specific resolver's quirks — not redesigning the whole table
- On Shared Servers, this table can be configured differently per site via the NVTGC field

PART 3

Table Fields, Part 1 — Scope Columns

Field	What it controls
NVTGC	Applies to shared servers or sites with delivery locations only. Identifies which site this row applies to. Edits limited by the site you're logged into in Customization Manager.
URL_Ver	Which OpenURL version this row applies to. Default value "Default" rarely needs to change. OpenURL 1.0 uses the value "Z39.88-2004."
rfr_id	Which SOURCE system this row applies to. Default value "Default" matches everything. Ask your resolver vendor for their exact rfr_id string — e.g. FirstSearch uses info:sid/firstsearch.oclc.org:WorldCat. This value (minus the info:sid/ prefix) is stored in ILLiad's CitedIn field.

PART 3

ILLiadAction — Three Actions, Three Jobs

Substitute:genre

- Sets the RequestType (Article/Loan) and DocumentType for the new request
- Also picks which HTML request form is used
- ILLiadFieldName must be RequestType
- Format: RequestType:DocumentType:HTMLForm — e.g. Loan:Book:LoanRequest.html

Substitute:rft.genre

- Identical concept and configuration to Substitute:genre
- Used specifically for OpenURL version 1.0 links
- Always create BOTH a genre and an rft.genre entry for the same mapping
- Missing one means one OpenURL version silently falls through

Replace

- Pulls one piece of incoming citation data into one ILLiad field
- One Replace row per ILLiad field you want populated
- ILLiadValue is left blank for this action
- Example: ILLiadFieldName = Edition, OpenURLFieldValues = <#rft.edition>

PART 3

Table Fields, Part 2 — The Value Columns

Field	For Replace rows	For Substitute rows
ILLiadFieldName	The ILLiad field being populated (e.g. Edition, LoanAuthor)	Always literally "RequestType"
OpenURLFieldValues	The incoming OpenURL tag, formatted <#rft.edition>	The genre value being matched — article, book, default, etc.
ILLiadValue	Left blank — not used by Replace	RequestType:DocumentType:HTMLForm, e.g. Loan:Book:LoanRequest.html

Tip: OpenURLFieldValues must be formatted exactly as <#OpenURLTagName> — other surrounding text is allowed, e.g. the comma in "<#rft.aulast>, <#rft.aufirst>"

PART 3

The Pipe (|) Fallback Trick

- A single Replace row can chain multiple candidate OpenURL tags for one ILLiad field, separated by a pipe character |
- ILLiad evaluates left to right — the first segment that has a value wins
- **Default LoanAuthor mapping (a real example straight out of the table):**
  - <#rft.au> | <#rft.aulast>, <#rft.aufirst> <#rft.auinitm> | <#aulast>, <#aufirst> <#auinitm>
  - If <#rft.au> has a value, that's used. If not, ILLiad checks the aulast/aufirst/auinitm combination. If that's empty too, it falls back to the last segment.
- This lets one row gracefully handle several citation styles and sources — no need for a separate row per style

PART 3

Date Formatting — The One Field With Special Parsing

- Add DATE|yyyy to the ILLiadValue field of a Replace row (commonly on PhotoJournalYear) to pull just a 4-digit year out of a full date string
- Other date and time tokens are supported the same way (e.g. yyyymm, hh:nn) for resolvers that pass more detailed date/time data
- How to set it: Customization Manager → Web Interface → OpenURL → OpenURLMapping → open PhotoJournalYear → set ILLiadValue to DATE|yyyy → Save
- Important limitation: this DATE| parsing is currently the ONLY field type with this kind of special formatting available — there's no equivalent for stripping prefixes or patterns out of other fields

Gotcha #1: A Blank Value Still "Counts"

The problem: A library integrating Alma found that ISSN was showing up blank on incoming requests — even though the item clearly had one. The default pipe chain checks rft.isbn first, then rft.issn. When Alma sent a blank rft.isbn, ILLiad treated that blank as "a value" and stopped looking — it never got to the ISSN.

THE FIX

Combine the two candidate tags into the SAME pipe segment instead of separate ones, so ILLiad concatenates them rather than stopping at the first blank.

BEFORE (blank-prone)

```
<#rft.isbn> | <#rft.issn> | <#rft.eissn> | ...
```

AFTER (fixed)

```
<#rft.issn><#rft.isbn> | <#rft.eissn> | ...
```

Gotcha #2: Hyphenated or Non-Standard Dates

The problem: DATE|yyyy expects a normal, parseable date string. PubMed-style date ranges — like "Jan-Mar 2011" — aren't recognized by ILLiad's date parser, so the year comes through blank instead of being extracted.

THE FIX

This is a hard limitation today, not a misconfiguration you can fix in the table. ILLiad reads the rft.date tag as one whole value, not piece by piece, so there's no way to tell it "ignore the month range, just grab the year." It's logged as an enhancement request (allowing regular expressions in OpenURLMapping) but isn't available yet.

PART 3

Using rfr_id to Customize Behavior by Source

- Because rfr_id scopes a row to one specific source, you can make the same incoming genre behave differently depending on where the link came from
- Classic example: FirstSearch calls something a "Book" genre, but you want that same genre treated as an "Article" when it arrives from a different source, like Ovid
- You'd configure two separate OpenURLMapping entries for the same genre value, each scoped to a different rfr_id
- Do not guess a vendor's rfr_id string — ask their support team for the exact value; using the wrong one means your source-specific row silently never matches
- This is the exact mechanism you'll lean on when WorldCat, EDS, and Alma each need slightly different handling — which brings us to Part 4

PART 4

Setting Up Specific Resolvers (don’t hold me to this)

WorldCat/FirstSearch, EDS, Alma, and the general pattern for everything else

PART 4

Setting Up WorldCat / FirstSearch

1. **Open the FirstSearch Admin Module** — firstsearch.oclc.org/admin → Resource Sharing tab → OpenURL Links
2. **Enter your OpenURL Server URL** — Single or Shared Server format; adjust for the Atlas Authentication Portal if your library uses it
3. **Click Test** — This opens the ILLiad OpenURL logon page. Login works, but no real citation submits — that's expected, not a failure
4. **Set the Icon File URL and Link Text** — e.g. https://yourserver/images/ILLiad.gif — this controls what patrons see and click
5. **Enable the link per database** — Choose which FirstSearch databases should show the ILL link
6. **Click Save Changes** — Confirm before exiting the Admin Module

PART 4

Setting Up EDS (EBSCO Discovery Service)

1

Work in EBSCOadmin / your link resolver's target list — Configure a custom target (often labeled Interlibrary Loan or Full Text Finder) pointing at your ILLiad OpenURL Server URL

2

Confirm the rfr_id EDS actually sends — So you can scope OpenURLMapping rows to it later if EDS records ever need source-specific handling

3

Remember: the ILLiad side doesn't change — The OpenURLMapping table decides how incoming fields map, no matter which discovery layer sent the link

4

Run a full end-to-end test request — Click through from a real EDS record before rolling this out to patrons

PART 4

Setting Up Alma

1. **Configure Alma's link resolver / Discovery interface** — Add an OpenURL target pointing to your ILLiad OpenURL Server URL, same as any other resolver

2. **Watch for the ISSN/ISBN blank-field issue** — Very common with Alma-sourced links — this is Gotcha #1 from Part 3. The pipe-chain fix resolves it.

3. **Confirm Alma's rfr_id with your Ex Libris admin** — Needed only if you require Alma-specific mapping behavior

4. **Test with a real Alma record before go-live** — Confirm the correct request form opens and all fields populate as expected

PART 4

Other Common Resolvers — The General Pattern

- Any system that speaks OpenURL 0.1 or 1.0 can point to ILLiad the exact same way — SFX, 360 Link, Primo, Summon, and others all follow this pattern
- **The setup pattern never changes:**
  - 1. Give the source system your ILLiad OpenURL Server URL as its ILL / document-delivery target
  - 2. Confirm or obtain that source's rfr_id, in case you need source-specific mapping later
  - 3. Test with a real record before rolling out to patrons
- 4. Only adjust OpenURLMapping entries if something maps incorrectly — most sources work with the defaults
- Exact menu paths vary by vendor and version — check that vendor's own documentation for "OpenURL target" or "ILL link," or contact Atlas Support if the ILLiad side needs adjustment

PART 5

Shared Servers, Troubleshooting & Best Practices

PART 5

Shared Server Considerations

- The NVTGC embedded in the ILLiad base URL can be used to route to individual sites
- Atlas Auth Portal available for a single entry point across multiple SSOs/sites
- OpenURLMapping rows are scoped by NVTGC in Customization Manager — each site can have its own genre and field behavior for the very same source
- Staff limited to one site only edit your own NVTGC and delivery locations' rows; staff with an NVTGC = ALL can view and edit across sites using the site dropdown
- Because some tables (like Routing) are shared across all sites, always coordinate with other sites before changing shared-impact behavior

PART 5

Troubleshooting Checklist

1. **Confirm the ILLiad web pages work at all** — Rules out anything beyond OpenURL specifically
2. **Confirm the source system's target URL and NVTGC** — Wrong URL or site code is the most common root cause
3. **Is the issue genre-based or field-based?** — Wrong form/RequestType → look at Substitute rows. Wrong/blank data → look at Replace rows
4. **Blank fields? Check pipe-chain ordering** — Remember: a blank still "counts" as a value (Gotcha #1)
5. **Date fields blank or wrong? Check for non-standard date formats** — Hyphenated ranges like "Jan-Mar 2011" aren't parseable today (Gotcha #2)
6. **Still stuck? Contact your standard support** — support@oclc.org or support@atlas-sys.com

PART 5

Best Practices Recap

- Always pair a Substitute:genre row with a matching Substitute:rft.genre row — never add just one
- Make sure there's always an equivalent entry for EVERY NTVGC/Delivery location
- Get the real rfr_id from each vendor instead of guessing it
- Test every new resolver integration with a real, live record before announcing it to patrons
- Document any custom OpenURLMapping changes you make, with the reason why — future you (or the next staff member) will thank you
- On Shared Servers, coordinate with other sites before changing anything with shared impact
- Add in an entry for DOI – this isn't included by default
    - NVTGC = custom; URL_ver = Default; rfr_id = Default; ILLiadAction = Replace; ILLiadFieldName = DOI; OpenURLFieldValues = <#rft_id>|<#id>; ILLiadValue = *blank*
- Check for a closing tag on the <#PhotoJournalVolume entries (old bug)

WRAP-UP

What We Covered

- What OpenURL is, and why ILLiad relies on it to meet patrons where they already search
- The full request path — from a click in a database, through the NVTGC and the logon page, to a submitted transaction in your normal queue
- The OpenURLMapping table in depth: NVTGC, URL_Ver, rfr_id, ILLiadAction, ILLiadFieldName, OpenURLFieldValues, ILLiadValue, the pipe fallback trick, and date parsing
- Two real support cases — blank ISSN fields and hyphenated dates — and exactly what to do about each
- Concrete setup steps for WorldCat/FirstSearch, EDS, Alma, and the general pattern for any other OpenURL-compliant resolver
- Where to look first when something doesn't map correctly

Extras

- New Concierge Perk - flipping for unknown genres
  - (for now – can try on your own or new Atlas Feedback submission?)
- Concierge Showcase with EDS Link
  - https://training.atlas-sys.com/Course/Details/3111
- Some pertinent Atlas Feedback links
  - https://feedback.atlas-sys.com/ideas/9
  - https://feedback.atlas-sys.com/ideas/35
  - https://feedback.atlas-sys.com/ideas/109
  - https://feedback.atlas-sys.com/ideas/487 / https://feedback.atlas-sys.com/ideas/103
  - https://feedback.atlas-sys.com/ideas/786