

Alternate IDs , Queries, and Resolution

Technical Note

1 Scope

Most EIDR records have at least one Alternate ID, and most of those have more than one. This tech note contains a general introduction to using EIDR resolution and query services, slanted towards their use with Alternate IDs, and covers using the EIDR REST API and SDK tools to:

- Discover EIDR records that correspond to a particular Alternate ID; this is sometimes called “match by Alternate ID”
- Retrieving Alternate IDs for an existing EIDR ID

One particularly important use for the techniques described in the tech note is using Alternate Identifiers to match a catalog against the EIDR Registry. If this is done as a first pass using common identifiers¹ (e.g. IMDB, ISAN) it can substantially reduce the number of records that have to be matched based on metadata.

Please contact EIDR Operations Support (support@eidr.org) with any question related to this document.

2 Overview

2.1 Type of Alternate Identifiers

There are two kinds of Alternate IDs in the EIDR Registry. Standard IDs conform to published standards or industry practices that are so common as to be *de facto* standards. ISAN is an example of the former; IMDb is an example of the latter. Many (but not all) Standard alternate IDs are resolvable.

Proprietary IDs are specific to particular applications or particular companies. Their type is “Proprietary” and they all have a distinguishing domain attached. For example, identifiers from the DECE ecosystem (UltraViolet) have domain `decellc.com` and conform to the ID specification published by DECE. Identifiers from the UK broadcaster ITV have domain `itv.com`

EIDR will occasionally promote an ID from Proprietary to Standard, based on widespread use and industry acceptance.

2.2 Resolvable Alternate Identifiers

An identifier is *resolvable* if there is a service that can take that identifier in a standard way and return some information about it. Both Standard and Proprietary Alternate IDs can be resolvable. For example,

¹ Or indeed any type of identifier present in both the EIDR registry and the source data

the IMDb ID tt2614684 is resolvable as <http://www.imdb.com/title/tt2614684/> and all IMDB IDs follow that pattern. Identifiers from the British Film Institute, stored in in EIDR as Proprietary IDs with domain bfi.org.uk are resolvable, e.g. 150233073 resolves to <http://collections-search.bfi.org.uk/web/Details/ChoiceFilm/150233073>

Some resolvable identifiers resolve to a Web page, some resolve to machine-readable data such as XML or RDF, and some use content negotiation to select among multiple resolution types.

2.3 Using Alternate IDs

EIDR provides two mechanisms for matching Alternate IDs to EIDR records. The first of these is by Resolution and the second is by Query. Resolution also provides the means of getting the Alternate IDs for a particular EIDR record.

Resolution is the standard way in EIDR of getting metadata from a known identifier. Resolution has two orthogonal pieces. The first is the ID that is being resolved; this can be an EIDR ID or an Alternate ID. See the REST API section below for details.

The second is the data returned by the resolution. As can be seen from the following table, some of these forms will return the Alternate IDs for an EIDR record².

API Resolution Type	Schema Type Used	Notes
Full	eidr:fullObjectInfoType	This gives an object's complete metadata, filling in inherited fields and fully realizing any language additions or replacements.
Simple	eidr:simpleDataInfoType	Returns minimal information, including the name, structural type, referent type, primary language, release date, and status of an object, along with skeletal descriptions of its relationships.
SelfDefined	eidr:allSelfDefinedInfoType	Returns metadata found on the object itself, ignoring inheritance and language accumulation.
Provenance	eidr:provenanceInfoType	Returns information about an object's creation and modification history.
DOI Kernel	doi:kernelMetadata with a doi:referentCreation element	Returns metadata formatted according to the DOI Kernel metadata for a referentCreation. This includes URLs for resolvable Alternate IDs. Please see EIDR DOI Mapping Table for full details.

² Please see the *EIDR Registry User's Guide* and *EIDR and the DOI Proxy* for more information.

API Resolution Type	Schema Type Used	Notes
Alternate IDs	eidr:alternateIDsType	Returns simple list of Alternate IDs associated with the underlying EIDR record.

The EIDR Registry provides a general Query API for searching over all records' metadata. Please see Appendix B of [EIDR Registry User's Guide](#) for definition and examples. There are more examples in the Query Tool section of the [EIDR 2.0 Tools Overview](#). Most importantly for this tech note, you can make queries based on type, value, and domain of Alternate IDs.

3 Web UI

The Web UI is useful for low-volume discovery of EIDR records that match Alternate IDs, retrieving Alternate IDs from a small number of records, and so on. If you need to process many Alternate IDs or get metadata for many EIDR records, please DO NOT treat the Web UI as a programming interface (e.g. do not use Python, Perl, or cURL to manipulate the UI programmatically.) The implementation of the Web UI changes from time to time and there is no guarantee that results from reverse engineering what it does will continue to work in the future. Use the REST API, the SDK, or the command-line tools instead.

4 REST API

4.1 Resolving EIDR IDs

The EIDR REST API for Resolution uses http GET requests. To resolve an EIDR ID the call is <https://resolve.eidr.org/EIDR/object/EIDR-ID?type=TYPE>

The ?type parameter is optional; if it is not present the resolution type defaults to Full. For example, <https://resolve.eidr.org/EIDR/object/10.5240/F85A-E100-B068-5B8F-B1C8-T> returns full EIDR metadata.

<https://resolve.eidr.org/EIDR/object/10.5240/F85A-E100-B068-5B8F-B1C8-T?type=alternateIDs> returns the EIDR ID and a list of its alternate IDs

<https://resolve.eidr.org/EIDR/object/10.5240/F85A-E100-B068-5B8F-B1C8-T?type=DOIKernel> returns DOI-format metadata, including the URI forms of any actionable alternate IDs.

This works with cURL too:

```
curl -G https://resolve.eidr.org/EIDR/object/10.5240/F85A-E100-B068-5B8F-B1C8-T
```

and

```
curl -G https://resolve.eidr.org/EIDR/object/10.5240/F85A-E100-B068-5B8F-B1C8-T?type=AlternateIDs
```

4.2 Resolving Alternate IDs

Resolution of Alternate IDs uses the same API as resolving EIDR IDs, including the optional type parameter. However, the EIDR ID is replaced with ?altId=VALUE

For example, <https://resolve.eidr.org/EIDR/object/?altId=150404739> returns the full metadata for the EIDR Record that has an alternate ID with value 150404739 and

<https://resolve.eidr.org/EIDR/object/?altId=150404739&type=AlternateIDs> returns the Alternate ID list for that EIDR record.

Two errors are possible. If there is no EIDR record with the alternate ID, the result is

```
<Response xmlns="http://www.eidr.org/schema" version="2.0.9">
  <Status>
    <Code>8</Code>
    <Type>bad id error</Type>
    <Details>no match</Details>
  </Status>
</Response>
```

If there is more than one EIDR record with an Alternate ID that matches the supplied value, e.g.

<https://resolve.eidr.org/EIDR/object/?altId=50>, the reply is

```
<Response xmlns="http://www.eidr.org/schema" version="2.0.9">
  <Status>
    <Code>3</Code>
    <Type>invalid request</Type>
    <Details>
      multiple matches, including 10.5240/4637-906F-A288-EF7A-9CEC-8,
      10.5240/6056-E1F6-E8B4-6BB5-4DF5-1
    </Details>
  </Status>
</Response>
```

This is especially common with alternate IDs that are smallish integers. For example, 50, 70, and 55900 are found as both IVA Alternate IDs and multiple times as Proprietary IDs. There are two extra parameters you can use to narrow the search:

- ?altIdType=TYPE requires the type of the Alternate ID to match as well as the value. TYPE is any legal EIDR Alternate ID type. For example <https://resolve.eidr.org/EIDR/object/?altId=50&altIdType=IVA> produces a single result
- ?altIdDomain=DOMAIN requires the domain of the Alternate ID to match. <https://resolve.eidr.org/EIDR/object/?altId=50&altIdDomain=redbeemedia.com> results in a single match.

You can of course add a resolution type as a parameter and use this with cURL, e.g.

```
curl -G
https://resolve.eidr.org/EIDR/object/?altId=50&altIdDomain=redbeemedia.com&
type=alternateIDs
```

4.3 Queries

Note that this section assumes familiarity with the EIDR query language (see reference above.) The Query service uses http POST and requires EIDR credentials (user ID, party ID, and password.)

The fields that are of interest when looking for EIDR records that match Alternate IDs are:

- /FullMetadata/BaseObjectData/AlternateID
- /FullMetadata/BaseObjectData/AlternateID@domain
- /FullMetadata/BaseObjectData/AlternateID@type

The boilerplate for a REST query is

```
<?xml version="1.0" encoding="UTF-8"?>
<Request xmlns="http://www.eidr.org/schema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Operation>
<Query>
  <Expression>THEQUERY</Expression>
  <PageNumber>1</PageNumber>
  <PageSize>5000</PageSize>
</Query>
</Operation>
</Request>
```

For example, if you replace THEQUERY with (/FullMetadata/BaseObjectData/AlternateID 50) a call to the Query API will return all the records that have 50 as an Alternate ID. If you put the query into query-50.txt, you can use this cURL call.

```
curl -H "Authorization: Eidr USERID: PARTYID: MD5HASH" -H "Content-
Type: text/xml" --data @query-50.txt
https://registry1.eidr.org/EIDR/query
```

USERID and PARTYID are the standard EIDR IDs, e.g. 10.5238/your-user-name and 10.5237/your-party-id. MD5HASH is the MD5 hash of your password, described in [EIDR REST API Reference](#)³.

The default is to return SimpleMetadata for the query results. You can also add “?type=id” to the URL to get XML containing just the EIDR IDs of the results, e.g.

```
curl -H "Authorization: Eidr USERID: PARTYID: MD5HASH" -H "Content-
Type: text/xml" --data @query-50.txt
https://registry1.eidr.org/EIDR/query?type=id
```

See the REST API document for more details.

³ Sample code for encoding the password is in org.eidr.sdk.utils.PasswordEncoding in the SDK.

4.4 Performance and Practicality

Using the REST API to do resolution by Alternate ID is preferable to using the Query API – the performance is somewhat better; there is more flexibility in the return formats; and it does not require authorization credentials.

5 SDK and Command-Line Tools

The REST API exposes all of the functionality of the EIDR Registry. Using the REST API from cURL or Python can be very convenient for prototyping, experimentation, and small one-off projects. However, there is a lot of complexity, especially around error handling and to a lesser extent around authentication.

For larger projects, the EIDR SDK, available to members in Java and .NET implementations, simplifies much of this, and includes consistent error handling and centralized management of http headers. In particular, it supports all resolution types and provides utilities for building queries and iterating over the results.⁴

The SDK also includes a set of sample applications, available for Windows, Mac, and Linux platforms, which provide detailed examples of how to use the Java and .NET SDKs.

The SDK Tools work with common formats for input and output files so the output of one can be used as the input for another and processed using standard tools (spreadsheets, Perl, Python, sed, AWK, etc.) This allows partial automation of many workflows, including matching and registration, which would otherwise be either much more labor-intensive or require a larger-scale API integration.

The remainder of this section lists some of the SDK tools that can help with matching by Alternate ID and retrieving Alternate IDs from EIDR records. Please see [EIDR 2.0 Tools Overview](#) for full documentation, including installation and setup.

5.1 ResolveTool

ResolveTool takes either a single EIDR ID on the command line or a file containing a list of EIDR IDs, one per line. It supports all resolution types via the `-t` flag. The default type is Full Metadata. Other supported types are

- `f` | full
- `s` | simple
- `doi` | doikernel
- `self` | selfdefined
- `prov` | provenance
- `alternateids` | altids

Examples:

- `ResolveTool -d 10.5240/D9E0-4E92-D4AC-C6AA-63BA-5 -t AlternateIDs`

⁴ The current version does not support resolution by alternate ID.

returns the AlternateIDs resolution of the ID after the -d flag

- `ResolveTool -d 10.5240/D9E0-4E92-D4AC-C6AA-63BA-5 -t Full`

Returns the full metadata for the ID

- `ResolveTool -i file.txt -t DOIKernel`

Returns the DOIKernel metadata for all the IDs in file.txt

The -o outfile.txt redirects output to outfile.txt; otherwise results are written to stdout. By default, output that has more than one EIDR record in it is surrounded by <EIDRToolResults> and </EIDRToolResults> tags so the output is legal XML. The -raw flag suppresses the surrounding tags.

5.2 QueryTool

In its simplest form, QueryTool takes an input file containing an EIDR query, specified by the -i flag, and a resolution type specified by the -t flag. The default type is SimpleMetadata. Other supported types are

- f | full
- s | simple
- doi | doikernel
- self | selfdefined
- prov | provenance
- alternateids | altids
- id

For example, if file.txt contains the query `(/FullMetadata/BaseObjectData/AlternateID@domain bfi.org.uk)` the results will be all EIDR records that have an Alternate ID with a bfi.org.uk domain. If it contains `(/FullMetadata/BaseObjectData/AlternateID@domain bfi.org.uk) AND (/FullMetadata/BaseObjectData/AlternateID@type ISAN)` the tool will return all records that have a bfi.org.uk ID and an ISAN.

To get a list of just the IDs of records that matched the query, use “-t id”. To get a sense of how many records a query will return without running the full query, use the “-n” flag. The -o and -raw flags are the same as for ResolveTool.

Examples:

- `QueryTool -i file.txt -t id -o ids.txt`

Writes the IDs of records that match the query to the file idstxt

- `QueryTool -i file.txt -t AlternateIDs -raw -o full.txt`

Writes EIDR Full metadata to the file full.txt without enclosing tags.

5.3 AltIDToEIDR

The simplest use of AltIDToEIDR takes a file of Alternate IDs, one per line, and finds any associated EIDR records.

There are several variants, but the most convenient is `AltIDToEIDR -i file.txt -names -all`. Each line of output is tab-separated and contains the Alternate ID from the input file, an EIDR ID that has that Alternate ID, and the ResourceName in the matched EIDR record. The last two fields are blank if there is no match. If there are multiple matches, there is one line per matched record.

The tool is multi-threaded to improve performance, so the order of the output lines may differ from the order of the input file. If your application requires that the order be the same, use “-mt 1”, which sets the maximum number for threads to 1 and retains the input order in the output.⁵

The -o outfile flag works as for the other command-line tools.

AltIDToEIDR also supports two flags to help with multiply occurring alternate IDs:

- --domain DOMAIN which requires that matched records also have a matching domain on Proprietary Alternate IDs
- --type TYPE which requires that matched records also match on the type of alternate ID.

For instance, if `altstofind.txt` has these alternate IDs:

```
46145
12
18
30745
180
```

`AltIDToEIDR -i altstofind.txt -names -all` will generate this output:

30745	10.5240/80BD-2729-073D-A1B9-9EF0-H	Blinkende Lygter
30745	10.5240/75F6-F8E8-62CA-AA65-7FD9-8	The Life and Adventures of
	Ni cholas Ni kkl eby	
46145	10.5240/05EA-7715-BB44-3489-9D93-Y	Allegro
46145	10.5240/5B61-6C3E-C243-5A21-DC08-U	Lady and the Tramp
46145	10.5240/8A52-3619-E6AC-CDC4-793D-5	Samurai Cop
18	10.5240/10AF-9D4C-2BA3-264F-3E38-2	Bananerne - Skrael den for
	di n nabo!	
18	10.5240/B2C6-E369-25AD-OFFE-16C0-T	Ghosts of Mississippi
18	10.5240/DFD6-3469-1F21-3B55-6E84-V	Mindhunters
18	10.5240/D684-6E9C-58DC-OED2-446F-8	The Bible
180	10.5240/CA57-CB9E-862A-6E21-CCCA-3	Everything Is Relative
180	10.5240/FC6D-AA4A-E4EA-CEAD-D229-4	Lad isbjørnene danse
180	10.5240/AA1C-14CD-E4DF-9B17-2B89-A	The Duo
180	10.5240/5A1C-BEB2-B8FE-D55A-0561-Q	The Sunshine Boys
12	10.5240/643A-96D6-8BF9-D777-27B1-X	Babettes Gæstebud
12	10.5240/C48C-2661-4C17-9A55-CA80-7	Clean and Sober
12	10.5240/2BC0-3CF9-EFF9-41B9-61CC-T	Crash
12	10.5240/F971-9076-DC96-D9E6-ED85-U	Dolores Clai borne

However, `AltIDToEIDR -i altstofind.txt -names -all -domain dfi.dk` will produce only records with IDs from the Danish Film Institute:

⁵ See the SDK Tools Overview for a more complex way to match up input and output even with multiple threads.

46145	10. 5240/05EA- 7715- BB44- 3489- 9D93- Y	Allegro
12	10. 5240/643A- 96D6- 8BF9- D777- 27B1- X	Babettes Gæstebud
18	10. 5240/10AF- 9D4C- 2BA3- 264F- 3E38- 2	Bananerne - Skrael den for
		di nnabo!
180	10. 5240/FC6D- AA4A- E4EA- CEAD- D229- 4	Lad isbjørnene danse
30745	10. 5240/80BD- 2729- 073D- A1B9- 9EF0- H	Blinkende Lygter

6 Helpful Hints

Many databases have external identifiers, and using those IDs as a preliminary match against the EIDR registry can be very efficient. This is especially true if you have IMDb IDs or ISANs.⁶ For large-scale matching of this sort, AltIDToEIDR is probably the most efficient thing to use. You can of course do Resolution by Alternate ID through the REST API if you wish, but many users find that the pre-tabulated output simplifies further processing.

Queries are less efficient than resolutions, so if you want to find a single alternate ID it will be better to use the REST API or AltIDToEIDR. If you suspect that the ID may exist with multiple types and domains, you can use AltIDToEIDR or construct a query for use with QueryTool or the REST API.

Use of domains and types is strongly recommended, especially if the alternate IDs are unstructured strings or simple numbers.

You can extract the second column of the output from AltIDToEIDR and use it as input to ResolveTool.

If you are using the REST API and want to just get the IDs back from a query, use the form with “?type=id” as a parameter. It is then straightforward to extract a plain list of IDs, either with a proper XML tool or library or with judicious use of regular expressions and text-processing tools.

⁶ As of 31 March 2016 EIDR had over 160,000 IMDb IDs and almost 115,000 ISANs (not counting ISAN serial headers).