

# Managing Aliases

---

## Technical Note

### 1 Scope

One of the defining characteristics of the EIDR system is that EIDR IDs are unique -- one work (or version of a work) has one EIDR ID. This is a guiding principle for the required metadata fields and best practices. However, sometimes there are mistakes, and two records are registered when it later turns out that they really do represent the same work. In this case, one of the records is deprecated and aliased to the one that is retained<sup>1</sup>. Aliased records have no metadata of their own, just a connection to the retained record. The retained record is called the target of the alias.

Aliased IDs work as expected for resolution: the returned data comes from the target. At resolution time an application can tell if a record has been aliased because the <ID> element of the returned record will be the ID of the target, not the ID passed to the resolution system. You can see an example of this when you resolve an aliased ID in the EIDR UI.

The rest of the EIDR API (modify, graph traversal, use of the ID in a parent field) returns an error when it gets an aliased ID.

In general, this is fine: most external users of EIDR just resolve the IDs and get the data they expect, and this covers the vast majority of cases where aliased IDs are circulating 'in the wild.' However, systems that use EIDR IDs as input to other EIDR functions, such as creating Edits or modifying metadata, need to use the target ID. As mentioned above, this is possible if you do a Resolve first, but that complicates the application.

Most systems that use EIDR IDs for purposes other than resolution use a local database to store EIDR IDs in a way that is natural and convenient for them. Replacing aliased IDs with their targets simplifies these systems considerably because then all IDs in the system can be used for all EIDR API calls.

This document explains how to find aliased records and their targets; with that information, local systems can be updated periodically to replace aliased IDs with their targets.

This document assumes familiarity with the EIDR command-line tools.<sup>2</sup> Please see the *SDK Tools Overview* for a complete description. *Resolution, Queries, and Alternate IDs* is a quick-start guide that covers the tools mentioned in this document.

---

<sup>1</sup> Records that need aliasing are corrected whenever they are found; some are found through queries and searches, and many are found as the result of the matching process that precedes registrations.

<sup>2</sup> All of these functions can also be performed through the Java or .NET SDKs, allowing direct integration into a client's internal data management systems.

## 2 Technical Background

The important features of aliases and aliased records for this document are:

- The target of an alias is found in the VirtualField resolution of the aliased ID as /VirtualField/Alias. Only aliased records have this field
- Deleted records are actually aliased to the EIDR tombstone record, 10.5240/0000-0000-0000-0000-0000-X
- /Provenance/LastModificationDate for an aliased ID is the date the record was aliased, after which it will not change<sup>3</sup>.

Further information on aliases and queries can be found in EIDR Registry User's Guide.

## 3 Finding Aliased Records – Basic

You can use queries to find aliased records.<sup>4</sup> You will probably want to exclude deleted records, which are aliased to the EIDR tombstone record, 10.5240/0000-0000-0000-0000-0000-X.

To find all aliased records use

```
(/VirtualField/Alias EXISTS) AND NOT (/VirtualField/Alias
"10.5240/0000-0000-0000-0000-0000-X")
```

To find all deleted records use

```
(/VirtualField/Alias EXISTS) AND (/VirtualField/Alias
"10.5240/0000-0000-0000-0000-0000-X")
```

To find all aliased or deleted records use

```
(/VirtualField/Alias EXISTS)
```

The following sections assume you have stored the first query in the file query-alias.txt. Run QueryTool to find the aliased IDs:

```
QueryTool -i query-alias.txt -t id -o aliased.txt
```

aliased.txt is a list of all aliased IDs, which you can resolve to the target aliases with ResolveTool

```
ResolveTool -I aliases.txt -v -o aliases.txt
```

aliases.txt contains lines that look like

```
ID: 10.5240/7A60-C4FB-B7AA-00FF-2CAB-N alias: 10.5240/24BF-66D1-F018-FC22-D64A-7
ID: 10.5240/FD75-0652-7D3B-C390-F4CF-U alias: 10.5240/622B-58E6-4380-D142-668E-N
```

<sup>3</sup> Unless the record is deleted and resurrected, which is an extremely rare occurrence.

<sup>4</sup> A full list of all aliased and deleted records is also available from the EIDR Support page at <http://support.eidr.org/>.

```
ID: 10.5240/BA20-96E1-23C5-540B-0749-6 alias: 10.5240/974E-BBED-E59D-5472-013A-X
ID: 10.5240/8DF4-4C7A-6579-4096-7EFB-N alias: 10.5240/4820-1F64-E6D5-694D-21C6-H
ID: 10.5240/7A49-D437-3FB4-A657-9591-5 alias: 10.5240/13B1-6779-88A7-F3B5-2543-8
```

To get just the IDs, run

```
sed 's/ alias: /\t/' aliases.txt | sed 's/ID: //' >aliased-ids.txt
```

After which `aliased-ids.txt` will have the aliased ID, followed by a tab, followed by the target of the alias, e.g.

```
10.5240/7A60-C4FB-B7AA-00FF-2CAB-N      10.5240/24BF-66D1-F018-FC22-D64A-7
10.5240/FD75-0652-7D3B-C390-F4CF-U      10.5240/622B-58E6-4380-D142-668E-N
10.5240/BA20-96E1-23C5-540B-0749-6      10.5240/974E-BBED-E59D-5472-013A-X
10.5240/8DF4-4C7A-6579-4096-7EFB-N      10.5240/4820-1F64-E6D5-694D-21C6-H
10.5240/7A49-D437-3FB4-A657-9591-5      10.5240/13B1-6779-88A7-F3B5-2543-8
```

The first column has the aliased ID and the second has its target. You can look for any instances of IDs in the first column in your own data repository and replace them with the corresponding ID from the second column.

## 4 Aliased records – Useful Tricks

### *Pruning the list using provenance metadata*

The list of aliased records grows slowly, but it does grow. There is no need to process the whole list every time – all you really need to do is see aliases that have been created since last time you checked the list<sup>5</sup>. You can add a condition based on this field to your query using the Provenance metadata. For example, to find all records that have been aliased since 1 January 2016, the query would be

```
(/VirtualField/Alias EXISTS) AND NOT (/VirtualField/Alias
"10.5240/0000-0000-0000-0000-0000-X") AND
(/ProvenanceMetadata/LastModificationDate >= 2016-01-01)
```

### *Pruning the list using alternate IDs*

When EIDR Operations aliases a record, alternate IDs are transferred from the aliased record to the target. This means that if you have added your own alternate ID to everything that you match or register, you can prune the list of aliases based on your alternate ID<sup>6</sup>.

Recall that resolving an aliased record returns metadata from the target of the alias. To get the alternate IDs for all alias targets, use

```
ResolveTool -i aliased.txt -t altids -raw -o target-altids.txt
```

<sup>5</sup> On 11-Feb-2016 there were 4490 aliased records, 229 of which have a provenance date after 2016-01-01

<sup>6</sup> You may still get some extra records if you added your ID to a record that is the target of an alias rather than to a record that has been aliased.

target-altids.txt contains the alternate IDs for all aliased records, in the standard format, which you can use to search for targets that have your alternate ID. To do this, use the following set of commands, which flattens all the alternate IDs for a record into a single line.

```
sed -n '/<AlternateIDs/{:loop N;s/\n//;/AlternateIDs>$/P;d}; b
loop}' target-altids.txt >target-altidsflat.txt
```

Search for your alternate IDs in target-altids-flat.txt with

```
grep -I 'MYTYPE' target-altidsflat.txt >mytargets.txt
```

'MYTYPE' should be the domain or type you are looking for, e.g. 'itv.com' or 'IVA'

Then get the target EIDR IDs that have your alternate ID with

```
sed 's/.*<ID>/' mytargets.txt | sed 's/<\ID>.*//>' >mytarget-
ids.txt
```

mytarget-ids.txt has all EIDR IDs that have your alternate ID and are the targets of aliases. You can use these to find the aliased IDs by looking them up in aliased-ids.txt, e.g. with

```
awk 'BEGIN{FS=OFS="\t"} FNR==NR{list[$2]=$1; next} $1 in list
{print list[$1], $1}' aliased-ids.txt mytarget-ids.txt >my-
aliased.txt
```

my-aliased.txt now is the subset of aliased-ids.txt where the target has an alternate ID of 'MYTYPE'

You can use this method with a pruned-by-date list of alternate IDs (see preceding section) as well, of course.