Finding All Alternate IDs of a Particular Type

Technical Note

1 Scope

Many EIDR registrants add their own identifiers to EIDR records as Alternate IDs. For public identifiers, it is also possible that others may add them as well. For example, many registrants will add IMDB IDs, ISANs, or British Film Institute IDs. This means that if you are keeping your own database or mapping of EIDR to other IDs, there may be mappings of which you are unaware. As another example, sometimes an EIDR record is aliased to another and in this case alternate IDs are moved from the deprecated ID to the retained record; although the aliased ID will still resolve, it can simplify some applications if the external database updates its records to use the retained ID.

Fortunately, the EIDR command line tools make it easy to find all the Alternate IDs of a given type and the EIDR IDs to which they are attached. This document is a tutorial on how to do that.

2 Types 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.

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.

3 How To Do It

First, you have to find all EIDR records that have an Alternate ID of the type you’re interested in. This is done with QueryTool.

3.1 Getting All Alternate IDs

If the Alternate ID you are looking for is a standard alternate ID, the query will be something like

```
(/FullMetadata/BaseObjectData/AlternateID@type ISAN)
```

If the Alternate ID you are looking for is Proprietary, the query will be something like

```
(/FullMetadata/BaseObjectData/AlternateID@domain “bfi.org.uk”)
```

Put the query in a file (e.g. query.txt) and run QueryTool:

```
QueryTool -i query.txt -t id -o my-eidr-ids.txt
```

my-eidr-ids.txt now has all EIDR IDs that have the alternate ID type you are looking for. To get the Alternate IDs, run

```
ResolveTool -i my-eidr-ids.txt -t altids -o xml-altids.txt
```

xml-altids.txt now has EIDR IDs and their Alternate IDs in the `eidr:alternateIDsType` format.[1] Most of these will not be the Alternate IDs you're looking for. You can write your own perl, awk, or XSLT scripts to extract the EIDR ID/Alternate ID pairs you want, or follow the instructions in the next sections.

3.2 Getting What You Want, Method 1

This method gives you titles as well as EIDR IDs and Alternate IDs, but takes longer to run.

Extracting your Alternate IDs

Now you have to extract the alternate IDs of your preferred type from the larger list. Here is a template for doing that with the Linux sed utility:[2] the 'uniq | sort' idiom removes any duplicates from the list; duplicates will occur if he same alternate ID is on more than one EIDR record.

```
sed -n '/TYPESTRING/p' xml-altids.txt | sed 's:</.*>::' | sed
's:<.*>::' | sort | uniq >altids-only.txt
```

For a standard Alternate ID, TYPESTRING is the name of the type, e.g for ISAN you'd start the line with

```
sed -n '/ISAN/p' xml-altids.txt
```

For a Proprietary Alternate ID, TYPESTRING is the full domain in quotation marks. For the domain “bfi.org.uk” you'd start the line with

```
sed -n '/\”bfi\.org\.uk\”/p' xml-altids.txt
```

After you run the full chain of commands, altids-only.txt has all the alternate IDs of the required type.

[1] If you don't need a separate file of EIDR IDs you can run `QueryTool -I query.txt -t altids -o xml-altids.txt`

[2] The first sed extracts all lines with the desired type; the second removes the closing XML tag; the third removes the opening XML tag. You have to do it in that order because * in a sed regular expression is greedy; sed 's:<.*>::/g' will delete everything between the first < on the line and the last > (the whole line in this case.)

Final Results

Now you need to find the EIDR records that match the Alternate IDs. The standard way to do this is with AltIDToEIDR. For standard Alternate IDs:

```
AltIDToEIDR -all -names -i altids-only.txt -o altid-results.txt
```

Or for Proprietary Alternate IDs (replace DOMAIN with the domain you want, bfi.org.uk in the case of the example above

```
AltIDToEIDR -all -names -dom DOMAIN -i altids-only.txt -o altid-results.txt
```

altid-results.txt now has lines that look like this. The fields are tab-separated and the file can be processed further, read into a spreadsheet, etc.

```
150789754     10.5240/F6C9-2712-5E43-E932-1BFE-Y      $64,000 Question
150221520     10.5240/D486-DFEF-3A58-6232-DF4B-5      "Major" The Red Cross Dog
154107132     10.5240/4D7B-8937-7885-85E5-6E36-G      'arry On The Steamboat
…
```

If you sort the results on column 1, records that have the same Alternate ID will be in adjacent rows[3], e.g.

```
150211939    10.5240/7132-A751-B823-02B4-7C5D-O    The Little Door Into The World
150211944    10.5240/1862-12F5-7BE7-0253-A2E5-7    Little Dorrit
150211944    10.5240/C308-094A-CD5C-FCA8-6F8D-J    Little Dorrit, Part 1: Nobody's Fault
150211961    10.5240/0931-A1E2-5DB0-D0A9-8196-Z    Little Emily
…
150776764    10.5240/54E3-2DFE-A418-5D2A-D53B-5    Conviction
150776911    10.5240/33E0-7DC4-F391-6CF9-9BEB-H    Shame
150776911    10.5240/2F1C-02FD-7D82-52E1-58B4-C    Tyrannosaur
150777234    10.5240/E17E-5379-8BB1-49B2-0CC4-X    Stormhouse
```

If you sort the results on column 2, all the alternate IDs of your type for a single EIDR record will be in adjacent rows[4], e.g.

```
150044176    10.5240/0690-08AD-B32F-FEAF-A94A-E    Trog
150232168    10.5240/0690-6637-37C5-B42A-B799-L    Men of Steel
150232169    10.5240/0690-6637-37C5-B42A-B799-L    Men of Steel
150187096    10.5240/0691-3084-7368-AAF3-CD86-4    Get Cracking
…
150018481    10.5240/185C-7349-E124-A8B7-CC26-P    Double Exposures
150211944    10.5240/1862-12F5-7BE7-0253-A2E5-7    Little Dorrit
```

[3] Or you can find the multiply-occurring Alternate IDs with `cut -f 1 altid-results.txt | sort | uniq -d >multiply-occurring-altID.txt`

[4] Or you can find EIDR IDs with multiple Alternate IDs of your type with `cut -f 2 altid-results.txt | sort | uniq -d >multiple-alt-per-EIDR-ID.txt`

```
150434091      10.5240/1862-12F5-7BE7-0253-A2E5-7     Little Dorrit
150015767      10.5240/1863-95B8-8C09-4E05-A488-0     Debt Of Honour
```

Gettting What You Want, Method 2

This method does not give you titles but is faster to run because it doesn't require another set of registry operations.

Run this command[5]. TYPESTRING is described above in *Extracting Your Alternate IDs*.

```
sed -n '/10\.5240|TYPESTRING/p' xml-altids.txt | sed 's:</.*>::' |
sed 's: *<.*>::' | awk '/10\.5240/{if (buf!="") print buf; buf=$1}
!/10\.5240/{$0=$0;buf=buf"\t"$1} END {if (buf!="") print buf}'
>simple-results.txt
```

Each line of the results contains an EIDR ID followed by the Alternate IDs of your type, all tab-separated, e.g.

```
10.5240/AA3F-1A9E-F033-6509-6EA0-V    154109922
10.5240/1862-12F5-7BE7-0253-A2E5-7    150434091      150211944
10.5240/EB5B-AAFC-B5B5-D9A6-8C71-5    150034446
```

4 Helpful Hints

Some EIDR records will have more than one Alternate ID of a particular type; be careful of this if you are writing your own processing scripts.

If you are looking for proprietary Alternate IDs, use the whole domain name in the query file and in the first call to `sed`. Many Alternate IDs use subdomains, e.g. bfi.org.uk/Gifford and bfi.org.uk, or spe.sony.com/AlphaID and spe.sony.com/MPM

If you are parsing the Alternate ID resolutions file yourself, or using line counts to see if things make sense, some things to keep in mind are:

- Some EIDR records might have more than one Alternate ID of the same type or domain, either intentionally or in error. Once you have the
- The same Alternate ID can appear on more than one EIDR record, either intentionally or in error

[5] The first sed extracts all lines with an EIDR ID or the desired type; the second removes the closing XML tag; the third removes the opening XML tag and any leading spaces. You can replace the awk clause with this slightly faster and slightly inscrutable sed command:
sed '/10\.5240/ {:loop N;/\n10\.5240/{P;D; b loop}; s/\n/\t/; b loop}' >simple-results.txt