The E-utilities In-Depth: Parameters, Syntax and More

Eric Sayers, PhD

Created: May 29, 2009; Updated: January 9, 2017.

Introduction

This chapter serves as a reference for all supported parameters for the E-utilities, along with accepted values and usage guidelines. This information is provided for each E-utility in sections below, and parameters and/or values specific to particular databases are discussed within each section. Most E-utilities have a set of parameters that are required for any call, in addition to several additional optional parameters that extend the tool’s functionality. These two sets of parameters are discussed separately in each section.

General Usage Guidelines

Please see Chapter 2 for a detailed discussion of E-utility usage policy. The following two parameters should be included in all E-utility requests.

**tool**

Name of application making the E-utility call. Value must be a string with no internal spaces.

**email**

E-mail address of the E-utility user. Value must be a string with no internal spaces, and should be a valid e-mail address.

E-utility DTDs

With the exception of EFetch, the E-utilities each generate a single XML output format that conforms to a DTD specific for that utility. Links to these DTDs are provided in the XML headers of the E-utility returns.

---

1 NCBI; Email: sayers@ncbi.nlm.nih.gov.

Corresponding author.

NLM Citation: Sayers E. The E-utilities In-Depth: Parameters, Syntax and More. 2009 May 29 [Updated 2017 Jan 9]. In: Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2010-.
ESummary version 2.0 produces unique XML DocSums for each Entrez database, and as such each Entrez database has a unique DTD for version 2.0 DocSums. Links to these DTDs are provided in the version 2.0 XML.

EFetch produces output in a variety of formats, some of which are XML. Most of these XML formats also conform to DTDs or schema specific to the relevant Entrez database. Please follow the link below for the PubMed DTD:

- PubMed DTD

**EInfo**

**Base URL**


**Functions**

- Provides a list of the names of all valid Entrez databases
- Provides statistics for a single database, including lists of indexing fields and available link names

**Required Parameters**

None. If no `db` parameter is provided, einfo will return a list of the names of all valid Entrez databases.

**Optional Parameters**

**db**

Target database about which to gather statistics. Value must be a valid Entrez database name.

**version**

Used to specify version 2.0 EInfo XML. The only supported value is ‘2.0’. When present, EInfo will return XML that includes two new fields: <IsTruncatable> and <IsRangeable>. Fields that are truncatable allow the wildcard character ‘*’ in terms. The wildcard character will expand to match any set of characters up to a limit of 600 unique expansions. Fields that are rangeable allow the range operator ‘:’ to be placed between a lower and upper limit for the desired range (e.g. 2008:2010[pdat]).

**retmode**

Retrieval type. Determines the format of the returned output. The default value is ‘xml’ for EInfo XML, but ‘json’ is also supported to return output in JSON format.
**Examples**

Return a list of all Entrez database names:


Return version 2.0 statistics for Entrez Protein:


**ESearch**

**Base URL**


**Functions**

- Provides a list of UIDs matching a text query
- Posts the results of a search on the History server
- Downloads all UIDs from a dataset stored on the History server
- Combines or limits UID datasets stored on the History server
- Sorts sets of UIDs

**Required Parameters**

`db`

Database to search. Value must be a valid Entrez database name (default = pubmed).

`term`

Entrez text query. All special characters must be URL encoded. Spaces may be replaced by `+' signs. For very long queries (more than several hundred characters long), consider using an HTTP POST call. See the PubMed or Entrez help for information about search field descriptions and tags. Search fields and tags are database specific.

esearch.fcgi?db=pubmed&term=asthma

**Optional Parameters – History Server**

`usehistory`

When `usehistory` is set to ‘y’, ESearch will post the UIDs resulting from the search operation onto the History server so that they can be used directly in a subsequent E-utility call. Also, `usehistory` must be set to ‘y’ for ESearch to interpret query key values included in `term` or to accept a WebEnv as input.
WebEnv
Web environment string returned from a previous ESearch, EPost or ELink call. When provided, ESearch will post the results of the search operation to this pre-existing WebEnv, thereby appending the results to the existing environment. In addition, providing WebEnv allows query keys to be used in term so that previous search sets can be combined or limited. As described above, if WebEnv is used, usehistory must be set to 'y'.

```
esearch.fcgi?db=pubmed&term=asthma&WebEnv=<webenv string>&usehistory=y
```

query_key
Integer query key returned by a previous ESearch, EPost or ELink call. When provided, ESearch will find the intersection of the set specified by query_key and the set retrieved by the query in term (i.e. joins the two with AND). For query_key to function, WebEnv must be assigned an existing WebEnv string and usehistory must be set to 'y'.

Values for query keys may also be provided in term if they are preceded by a '#' (%23 in the URL). While only one query_key parameter can be provided to ESearch, any number of query keys can be combined in term. Also, if query keys are provided in term, they can be combined with OR or NOT in addition to AND.

The following two URLs are functionally equivalent:

```
esearch.fcgi?db=pubmed&term=asthma&query_key=1&WebEnv=<webenv string>&usehistory=y
```

```
esearch.fcgi?db=pubmed&term=%231+AND+asthma&WebEnv=<webenv string>&usehistory=y
```

Optional Parameters – Retrieval

retstart
Sequential index of the first UID in the retrieved set to be shown in the XML output (default=0, corresponding to the first record of the entire set). This parameter can be used in conjunction with retmax to download an arbitrary subset of UIDs retrieved from a search.

retmax
Total number of UIDs from the retrieved set to be shown in the XML output (default=20). By default, ESearch only includes the first 20 UIDs retrieved in the XML output. If usehistory is set to 'y', the remainder of the retrieved set will be stored on the History server; otherwise these UIDs are lost. Increasing retmax allows more of the retrieved UIDs to be included in the XML output, up to a maximum of 100,000 records. To retrieve more than 100,000 UIDs, submit multiple esearch requests while incrementing the value of retstart (see Application 3).
**Retrieval type**

There are two allowed values for ESearch: 'uilist' (default), which displays the standard XML output, and 'count', which displays only the <Count> tag.

**Retrieval type**

Determines the format of the returned output. The default value is 'xml' for ESearch XML, but 'json' is also supported to return output in JSON format.

**Sort**

Specifies the method used to sort UIDs in the ESearch output. The available values vary by database (db) and may be found in the Display Settings menu on an Entrez search results page. If usehistory is set to 'y', the UIDs are loaded onto the History Server in the specified sort order and will be retrieved in that order by ESummary or EFetch. Example values are 'relevance' and 'name' for Gene and 'first+author' and 'pub+date' for PubMed. Users should be aware that the default value of sort varies from one database to another, and that the default value used by ESearch for a given database may differ from that used on NCBI web search pages.

**Search field.**

If used, the entire search term will be limited to the specified Entrez field. The following two URLs are equivalent:

```
esearch.fcgi?db=pubmed&term=asthma&field=title
esearch.fcgi?db=pubmed&term=asthma[title]
```

**Optional Parameters – Dates**

**Type of date used to limit a search.** The allowed values vary between Entrez databases, but common values are 'mdat' (modification date), 'pdat' (publication date) and 'edat' (Entrez date). Generally an Entrez database will have only two allowed values for datatype.

**When reldate is set to an integer n, the search returns only those items that have a date specified by datatype within the last n days.**

**Date range used to limit a search result by the date specified by datatype.** These two parameters (mindate, maxdate) must be used together to specify an arbitrary date range. The general date format is YYYY/MM/DD, and these variants are also allowed: YYYY, YYYY/MM.
Examples

Search in PubMed with the term cancer for abstracts that have an Entrez date within the last 60 days; retrieve the first 100 PMIDs and translations; post the results on the History server and return a WebEnv and query_key:

db=pubmed&term=cancer&reldate=60&datetype=edat&retmax=100&usehistory=y

Search in PubMed for the journal PNAS, Volume 97, and retrieve six PMIDs starting with the seventh PMID in the list:

db=pubmed&term=PNAS[ta]+AND+97[vi]&retstart=6&retmax=6&tool=biomed3

Search in the NLM Catalog for journals matching the term obstetrics:

db=nlmcatalog&term=obstetrics+AND+ncbijournals[filter]

Search PubMed Central for free full text articles containing the query stem cells:

+AND+free+fulltext[filter]

Search in Nucleotide for all tRNAs:

db=nucleotide&term=biomol+trna[prop]

Search in Protein for a molecular weight range:

db=protein&term=70000:90000[molecular+weight]

EPost

Base URL


Functions

- Uploads a list of UIDs to the Entrez History server
- Appends a list of UIDs to an existing set of UID lists attached to a Web Environment
Required Parameters

$db$
Database containing the UIDs in the input list. The value must be a valid Entrez database name (default = pubmed).

$id$
UID list. Either a single UID or a comma-delimited list of UIDs may be provided. All of the UIDs must be from the database specified by $db$. There is no set maximum for the number of UIDs that can be passed to epost, but if more than about 200 UIDs are to be posted, the request should be made using the HTTP POST method.

```text
epost.fcgi?db=protein&id=15718680,157427902,119703751
```

Optional Parameter

$WebEnv$
Web Environment. If provided, this parameter specifies the Web Environment that will receive the UID list sent by post. EPost will create a new query key associated with that Web Environment. Usually this WebEnv value is obtained from the output of a previous ESearch, EPost or ELink call. If no $WebEnv$ parameter is provided, EPost will create a new Web Environment and post the UID list to query_key 1.

```text
epost.fcgi?db=protein&id=15718680,157427902,119703751&WebEnv=<webenv string>
```

Example

Post records to PubMed:

```text
```

ESummary

Base URL


Functions

- Returns document summaries (DocSums) for a list of input UIDs
- Returns DocSums for a set of UIDs stored on the Entrez History server
Required Parameter

$db$

Database from which to retrieve DocSums. The value must be a valid Entrez database name (default = pubmed).

Required Parameter – Used only when input is from a UID list

$id$

UID list. Either a single UID or a comma-delimited list of UIDs may be provided. All of the UIDs must be from the database specified by $db$. There is no set maximum for the number of UIDs that can be passed to ESummary, but if more than about 200 UIDs are to be provided, the request should be made using the HTTP POST method.

```
esummary.fcgi?db=protein&id=15718680,157427902,119703751
```

Required Parameters – Used only when input is from the Entrez History server

$query_key$

Query key. This integer specifies which of the UID lists attached to the given Web Environment will be used as input to ESummary. Query keys are obtained from the output of previous ESearch, EPost or ELink calls. The $query_key$ parameter must be used in conjunction with $WebEnv$.

$WebEnv$

Web Environment. This parameter specifies the Web Environment that contains the UID list to be provided as input to ESummary. Usually this WebEnv value is obtained from the output of a previous ESearch, EPost or ELink call. The $WebEnv$ parameter must be used in conjunction with $query_key$.

```
esummary.fcgi?db=protein&query_key=<key>&WebEnv=<webenv string>
```

Optional Parameters – Retrieval

$retstart$

Sequential index of the first DocSum to be retrieved (default=1, corresponding to the first record of the entire set). This parameter can be used in conjunction with $retmax$ to download an arbitrary subset of DocSums from the input set.

$retmax$

Total number of DocSums from the input set to be retrieved, up to a maximum of 10,000. If the total set is larger than this maximum, the value of $retstart$ can be iterated while holding $retmax$ constant, thereby downloading the entire set in batches of size $retmax$. 
**Retmode**

Retrieval type. Determines the format of the returned output. The default value is ‘xml’ for ESummary XML, but ‘json’ is also supported to return output in JSON format.

**Version**

Used to specify version 2.0 ESummary XML. The only supported value is ‘2.0’. When present, ESummary will return version 2.0 DocSum XML that is unique to each Entrez database and that often contains more data than the default DocSum XML.

**Examples**

PubMed:


PubMed, version 2.0 XML:


Protein:


Nucleotide:


Structure:


Taxonomy:


UniSTS:

EFetch

Base URL


Functions

- Returns formatted data records for a list of input UIDs
- Returns formatted data records for a set of UIDs stored on the Entrez History server

Required Parameters

db

Database from which to retrieve records. The value must be a valid Entrez database name (default = pubmed). Currently EFetch does not support all Entrez databases. Please see Table 1 in Chapter 2 for a list of available databases.

Required Parameter – Used only when input is from a UID list

id

UID list. Either a single UID or a comma-delimited list of UIDs may be provided. All of the UIDs must be from the database specified by db. There is no set maximum for the number of UIDs that can be passed to EFetch, but if more than about 200 UIDs are to be provided, the request should be made using the HTTP POST method.

efetch.fcgi?db=protein&id=15718680,157427902,119703751

Required Parameters – Used only when input is from the Entrez History server

query_key

Query key. This integer specifies which of the UID lists attached to the given Web Environment will be used as input to EFetch. Query keys are obtained from the output of previous ESearch, EPost or ELInk calls. The query_key parameter must be used in conjunction with WebEnv.

WebEnv

Web Environment. This parameter specifies the Web Environment that contains the UID list to be provided as input to EFetch. Usually this WebEnv value is obtained from the output of a previous ESearch, EPost or ELInk call. The WebEnv parameter must be used in conjunction with query_key.

efetch.fcgi?db=protein&query_key=<key>&WebEnv=<webenv string>
Optional Parameters – Retrieval

retmode

Retrieval mode. This parameter specifies the data format of the records returned, such as plain text, HTML or XML. See Table 1 for a full list of allowed values for each database.

Table 1 – Valid values of &retmode and &rettype for EFetch (null = empty string)

<table>
<thead>
<tr>
<th>Record Type</th>
<th>&amp;rettype</th>
<th>&amp;retmode</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Databases</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document summary</td>
<td>docsum</td>
<td>xml, default</td>
</tr>
<tr>
<td>List of UIDs in XML</td>
<td>uilist</td>
<td>xml</td>
</tr>
<tr>
<td>List of UIDs in plain text</td>
<td>uilist</td>
<td>text</td>
</tr>
<tr>
<td>db = bioproject</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full record XML</td>
<td>xml, default</td>
<td>xml, default</td>
</tr>
<tr>
<td>db = biosample</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full record XML</td>
<td>full, default</td>
<td>xml, default</td>
</tr>
<tr>
<td>Full record text</td>
<td>full, default</td>
<td>text</td>
</tr>
<tr>
<td>db = biosystems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full record XML</td>
<td>xml, default</td>
<td>xml, default</td>
</tr>
<tr>
<td>db = gene</td>
<td></td>
<td></td>
</tr>
<tr>
<td>text ASN.1</td>
<td>null</td>
<td>asn.1, default</td>
</tr>
<tr>
<td>XML</td>
<td>null</td>
<td>xml</td>
</tr>
<tr>
<td>Gene table</td>
<td>gene_table</td>
<td>text</td>
</tr>
<tr>
<td>db = homologene</td>
<td></td>
<td></td>
</tr>
<tr>
<td>text ASN.1</td>
<td>null</td>
<td>asn.1, default</td>
</tr>
<tr>
<td>XML</td>
<td>null</td>
<td>xml</td>
</tr>
<tr>
<td>Alignment scores</td>
<td>alignmentscores</td>
<td>text</td>
</tr>
<tr>
<td>FASTA</td>
<td>fasta</td>
<td>text</td>
</tr>
<tr>
<td>HomoloGene</td>
<td>homologene</td>
<td>text</td>
</tr>
<tr>
<td>db = mesh</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full record</td>
<td>full, default</td>
<td>text, default</td>
</tr>
<tr>
<td>db = nlmcatalog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full record</td>
<td>null</td>
<td>text, default</td>
</tr>
</tbody>
</table>

Table 1 continues on next page...
Table 1 continued from previous page.

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td><code>null</code></td>
<td>xml</td>
</tr>
<tr>
<td><strong>db = nuccore, nucest, nucgss, protein or popset</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>text ASN.1</td>
<td><code>null</code></td>
<td>text, default</td>
</tr>
<tr>
<td>binary ASN.1</td>
<td><code>null</code></td>
<td>asn.1</td>
</tr>
<tr>
<td>Full record in XML</td>
<td>native</td>
<td>xml</td>
</tr>
<tr>
<td>Accession number(s)</td>
<td>acc</td>
<td>text</td>
</tr>
<tr>
<td>FASTA</td>
<td>fasta</td>
<td>text</td>
</tr>
<tr>
<td>TinySeq XML</td>
<td>fasta</td>
<td>xml</td>
</tr>
<tr>
<td>SeqID string</td>
<td>seqid</td>
<td>text</td>
</tr>
<tr>
<td><strong>Additional options for db = nuccore, nucest, nucgss or popset</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GenBank flat file</td>
<td>gb</td>
<td>text</td>
</tr>
<tr>
<td>GBSeq XML</td>
<td>gb</td>
<td>xml</td>
</tr>
<tr>
<td>INSDSeq XML</td>
<td>gbc</td>
<td>xml</td>
</tr>
<tr>
<td><strong>Additional option for db = nuccore and protein</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature table</td>
<td>ft</td>
<td>text</td>
</tr>
<tr>
<td><strong>Additional option for db = nuccore</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GenBank flat file with full sequence (contigs)</td>
<td>gbwithparts</td>
<td>text</td>
</tr>
<tr>
<td>CDS nucleotide FASTA</td>
<td>fasta_cds_na</td>
<td>text</td>
</tr>
<tr>
<td>CDS protein FASTA</td>
<td>fasta_cds_aa</td>
<td>text</td>
</tr>
<tr>
<td><strong>Additional option for db = nucest</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EST report</td>
<td>est</td>
<td>text</td>
</tr>
<tr>
<td><strong>Additional option for db = nucgss</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSS report</td>
<td>gss</td>
<td>text</td>
</tr>
<tr>
<td><strong>Additional options for db = protein</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GenPept flat file</td>
<td>gp</td>
<td>text</td>
</tr>
<tr>
<td>GBSeq XML</td>
<td>gp</td>
<td>xml</td>
</tr>
<tr>
<td>INSDSeq XML</td>
<td>gpc</td>
<td>xml</td>
</tr>
<tr>
<td>Identical Protein XML</td>
<td>ipg</td>
<td>xml</td>
</tr>
<tr>
<td><strong>db = pmc</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td><code>null</code></td>
<td>xml, default</td>
</tr>
<tr>
<td>MEDLINE</td>
<td>medline</td>
<td>text</td>
</tr>
<tr>
<td><strong>db = pubmed</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>text ASN.1</td>
<td><code>null</code></td>
<td>asn.1, default</td>
</tr>
</tbody>
</table>

Table 1 continues on next page...
Table 1 continued from previous page.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>null</td>
<td>xml</td>
</tr>
<tr>
<td>MEDLINE</td>
<td>medline</td>
<td>text</td>
</tr>
<tr>
<td>PMID list</td>
<td>uiolist</td>
<td>text</td>
</tr>
<tr>
<td>Abstract</td>
<td>abstract</td>
<td>text</td>
</tr>
<tr>
<td><strong>db = sequences</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>text ASN.1</td>
<td>null</td>
<td>text, default</td>
</tr>
<tr>
<td>Accession number(s)</td>
<td>acc</td>
<td>text</td>
</tr>
<tr>
<td>FASTA</td>
<td>fasta</td>
<td>text</td>
</tr>
<tr>
<td>SeqID string</td>
<td>seqid</td>
<td>text</td>
</tr>
<tr>
<td><strong>db = snp</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>text ASN.1</td>
<td>null</td>
<td>asn.1, default</td>
</tr>
<tr>
<td>XML</td>
<td>null</td>
<td>xml</td>
</tr>
<tr>
<td>Flat file</td>
<td>flt</td>
<td>text</td>
</tr>
<tr>
<td>FASTA</td>
<td>fasta</td>
<td>text</td>
</tr>
<tr>
<td>RS Cluster report</td>
<td>rsr</td>
<td>text</td>
</tr>
<tr>
<td>SS Exemplar list</td>
<td>ssxemplar</td>
<td>text</td>
</tr>
<tr>
<td>Chromosome report</td>
<td>chr</td>
<td>text</td>
</tr>
<tr>
<td>Summary</td>
<td>docset</td>
<td>text</td>
</tr>
<tr>
<td>UID list</td>
<td>uilist</td>
<td>text or xml</td>
</tr>
<tr>
<td><strong>db = sra</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>full, default</td>
<td>xml, default</td>
</tr>
<tr>
<td><strong>db = taxonomy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td>null</td>
<td>xml, default</td>
</tr>
<tr>
<td>TaxID list</td>
<td>uilist</td>
<td>text or xml</td>
</tr>
</tbody>
</table>

**rettype**

Retrieval type. This parameter specifies the record view returned, such as Abstract or MEDLINE from PubMed, or GenPept or FASTA from protein. Please see Table 1 for a full list of allowed values for each database.

**retstart**

Sequential index of the first record to be retrieved (default=0, corresponding to the first record of the entire set). This parameter can be used in conjunction with **retmax** to download an arbitrary subset of records from the input set.
**retmax**

Total number of records from the input set to be retrieved, up to a maximum of 10,000. Optionally, for a large set the value of `retstart` can be iterated while holding `retmax` constant, thereby downloading the entire set in batches of size `retmax`.

**Optional Parameters – Sequence Databases**

**strand**

Strand of DNA to retrieve. Available values are "1" for the plus strand and "2" for the minus strand.

**seq_start**

First sequence base to retrieve. The value should be the integer coordinate of the first desired base, with "1" representing the first base of the sequence.

**seq_stop**

Last sequence base to retrieve. The value should be the integer coordinate of the last desired base, with "1" representing the first base of the sequence.

**complexity**

Data content to return. Many sequence records are part of a larger data structure or "blob", and the `complexity` parameter determines how much of that blob to return. For example, an mRNA may be stored together with its protein product. The available values are as follows:

<table>
<thead>
<tr>
<th>Value of complexity</th>
<th>Data returned for each requested GI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>entire blob</td>
</tr>
<tr>
<td>1</td>
<td>bioseq</td>
</tr>
<tr>
<td>2</td>
<td>minimal bioseq-set</td>
</tr>
<tr>
<td>3</td>
<td>minimal nuc-prot</td>
</tr>
<tr>
<td>4</td>
<td>minimal pub-set</td>
</tr>
</tbody>
</table>

**Examples**

**PubMed**

Fetch PMIDs 17284678 and 9997 as text abstracts:

```
db=pubmed&id=17284678,9997&retmode=text&rettype=abstract
```

Fetch PMIDs in XML:
PubMed Central

Fetch XML for PubMed Central ID 212403:

Nucleotide/Nuccore

Fetch the first 100 bases of the plus strand of GI 21614549 in FASTA format:

Fetch the first 100 bases of the minus strand of GI 21614549 in FASTA format:

Fetch the nuc-prot object for GI 21614549:

Fetch the full ASN.1 record for GI 5:

Fetch FASTA for GI 5:

Fetch the GenBank flat file for GI 5:

Fetch GBSeqXML for GI 5:

Fetch TinySeqXML for GI 5:

Popset
Fetch the GenPept flat file for Popset ID 12829836:
db=popset&id=12829836&rettype=gp

Protein
Fetch the GenPept flat file for GI 8:
db=protein&id=8&rettype=gp
Fetch GBSeqXML for GI 8:
db=protein&id=8&rettype=gp&retmode=xml

Sequences
Fetch FASTA for a transcript and its protein product (GIs 312836839 and 34577063)
db=sequences&id=312836839,34577063&rettype=fasta&retmode=text

Gene
Fetch full XML record for Gene ID 2:
db=gene&id=2&retmode=xml

ELink
Base URL

Functions
- Returns UIDs linked to an input set of UIDs in either the same or a different Entrez database
- Returns UIDs linked to other UIDs in the same Entrez database that match an Entrez query
- Checks for the existence of Entrez links for a set of UIDs within the same database
- Lists the available links for a UID
- Lists LinkOut URLs and attributes for a set of UIDs
- Lists hyperlinks to primary LinkOut providers for a set of UIDs
- Creates hyperlinks to the primary LinkOut provider for a single UID
**Required Parameters**

**db**

Database from which to retrieve UIDs. The value must be a valid *Entrez database name* (default = pubmed). This is the destination database for the link operation.

**dbfrom**

Database containing the input UIDs. The value must be a valid *Entrez database name* (default = pubmed). This is the origin database of the link operation. If `db` and `dbfrom` are set to the same database value, then ELink will return computational neighbors within that database. Please see the full list of Entrez links for available computational neighbors. Computational neighbors have linknames that begin with `dbname_dbname` (examples: protein_protein, pcassay_pcassay_activityneighbor).

**cmd**

ELink command mode. The command mode specified which function ELink will perform. Some optional parameters only function for certain values of `&cmd` (see below).

**cmd=neighbor (default)**

ELink returns a set of UIDs in `db` linked to the input UIDs in `dbfrom`.

*Example: Link from protein to gene*


**cmd=neighbor_score**

ELink returns a set of UIDs within the same database as the input UIDs along with computed similarity scores.

*Example: Find related articles to PMID 20210808*


**cmd=neighbor_history**

ELink posts the output UIDs to the Entrez History server and returns a query_key and WebEnv corresponding to the location of the output set.

*Example: Link from protein to gene and post the results on the Entrez History*


**cmd=acheck**
ELink lists all links available for a set of UIDs.

Example: List all possible links from two protein GIs

dbfrom=protein&id=15718680,157427902&cmd=acheck

Example: List all possible links from two protein GIs to PubMed

dbfrom=protein&db=pubmed&id=15718680,157427902&cmd=acheck

ELink checks for the existence of links within the same database for a set of UIDs. These links are equivalent to setting db and dbfrom to the same value.

Example: Check whether two nuccore sequences have "related sequences" links.

dbfrom=nuccore&id=21614549,219152114&cmd=ncheck

Elink checks for the existence of external links (LinkOuts) for a set of UIDs.

Example: Check whether two protein sequences have any LinkOut providers.

dbfrom=protein&id=15718680,157427902&cmd=llinks

For each input UID, ELink lists the URLs and attributes for the LinkOut providers that are not libraries.

Example: List the LinkOut URLs for non-library providers for two pubmed abstracts.

dbfrom=pubmed&id=19880848,19822630&cmd=llinks

For each input UID, ELink lists the URLs and attributes for all LinkOut providers including libraries.

Example: List all LinkOut URLs for two PubMed abstracts.

dbfrom=pubmed&id=19880848,19822630&cmd=llinkslib

cmd=prlinks
ELink lists the primary LinkOut provider for each input UID, or links directly to the LinkOut provider’s web site for a single UID if `retmode` is set to `ref`.

*Example: Find links to full text providers for two PubMed abstracts.*


*Example: Link directly to the full text for a PubMed abstract at the provider’s web site.*


**Required Parameter – Used only when input is from a UID list**

**id**

UID list. Either a single UID or a comma-delimited list of UIDs may be provided. All of the UIDs must be from the database specified by `dbfrom`. There is no set maximum for the number of UIDs that can be passed to ELink, but if more than about 200 UIDs are to be provided, the request should be made using the HTTP POST method.

*Link from protein to gene.*


*Find related sequences (link from nuccore to nuccore).*


If more than one `id` parameter is provided, ELink will perform a separate link operation for the set of UIDs specified by each `id` parameter. This effectively accomplishes "one-to-one" links and preserves the connection between the input and output UIDs.

*Find one-to-one links from protein to gene.*


**Required Parameters – Used only when input is from the Entrez History server**

**query_key**

Query key. This integer specifies which of the UID lists attached to the given Web Environment will be used as input to ELink. Query keys are obtained from the output of previous ESearch, EPost or ELInk calls. The `query_key` parameter must be used in conjunction with `WebEnv`. 
**WebEnv**

Web Environment. This parameter specifies the Web Environment that contains the UID list to be provided as input to ELink. Usually this WebEnv value is obtained from the output of a previous ESearch, EPost or ELink call. The **WebEnv** parameter must be used in conjunction with **query_key**.

**Link from protein to gene:**

elink.fcgi?dbfrom=protein&db=gene&query_key=<key>&WebEnv=<webenv string>

**Find related sequences (link from protein to protein):**

elink.fcgi?dbfrom=protein&db=protein&query_key=<key>&WebEnv=<webenv string>

---

**Optional Parameters – Limiting the Output Set of Links**

**linkname**

Name of the Entrez link to retrieve. Every link in Entrez is given a name of the form `dbfrom_db_subset`.

The values of `subset` vary depending on the values of `dbfrom` and `db`. Many `dbfrom/db` combinations have no `subset` values. See the list of Entrez links for a listing of all available linknames. When **linkname** is used, only the links with that name will be retrieved.

The **linkname** parameter only functions when **cmd** is set to **neighbor** or **neighbor_history**.

**Find all links from gene to snp.**

dbfrom=gene&db=snp&id=93986

**Find snps with genotype data linked to genes.**

dbfrom=gene&db=snp&id=93986&linkname=gene_snp_genegenotype

**term**

Entrez query used to limit the output set of linked UIDs. The query in the **term** parameter will be applied after the link operation, and only those UIDs matching the query will be returned by ELink. The **term** parameter only functions when **db** and **dbfrom** are set to the same database value.

**Find all related articles for a PMID.**

dbfrom=pubmed&db=pubmed&id=19879512

**Find all related review articles published in 2008 for a PMID.**
https://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed&db=pubmed&id=19879512&term=review%5Bfilter%5D+AND+2008%5Bpdat%5Dh

holding

Name of LinkOut provider. Only URLs for the LinkOut provider specified by holding will be returned. The value provided to holding should be the abbreviation of the LinkOut provider's name found in the <NameAbbr> tag of the ELink XML output when cmd is set to llinks or llinkslib. The holding parameter only functions when cmd is set to llinks or llinkslib.

Find information for all LinkOut providers for a PMID.


Find information from clinicaltrials.gov for a PMID.


Optional Parameters – Dates

These parameters only function when cmd is set to neighbor or neighbor_history and dbfrom is pubmed.

datatype

Type of date used to limit a link operation. The allowed values vary between Entrez databases, but common values are ‘mdat’ (modification date), ‘pdat’ (publication date) and ‘edat’ (Entrez date). Generally an Entrez database will have only two allowed values for datatype.

reldate

When reldate is set to an integer n, ELink returns only those items that have a date specified by datatype within the last n days.

mindate, maxdate

Date range used to limit a link operation by the date specified by datatype. These two parameters (mindate, maxdate) must be used together to specify an arbitrary date range. The general date format is YYYY/MM/DD, and these variants are also allowed: YYYY, YYYY/MM.
**EGQuery**

**Base URL**

**Function**
Provides the number of records retrieved in all Entrez databases by a single text query.

**Required Parameter**

**term**
Entrez text query. All special characters must be URL encoded. Spaces may be replaced by `+` signs. For very long queries (more than several hundred characters long), consider using an HTTP POST call. See the PubMed or Entrez help for information about search field descriptions and tags. Search fields and tags are database specific.


**ESpell**

**Base URL**

**Function**
Provides spelling suggestions for terms within a single text query in a given database.

**Required Parameters**

**db**
Database to search. Value must be a valid Entrez database name (default = pubmed).

**term**
Entrez text query. All special characters must be URL encoded. Spaces may be replaced by `+` signs. For very long queries (more than several hundred characters long), consider using an HTTP POST call. See the PubMed or Entrez help for information about search field descriptions and tags. Search fields and tags are database specific.

**ECitMatch**

**Base URL**

**Function**
Retrieves PubMed IDs (PMIDs) that correspond to a set of input citation strings.

**Required Parameters**

**db**
Database to search. The only supported value is ‘pubmed’.

**rettype**
Retrieval type. The only supported value is ‘xml’.

**bdata**
Citation strings. Each input citation must be represented by a citation string in the following format:

journal_title|year|volume|first_page|author_name|your_key|

Multiple citation strings may be provided by separating the strings with a carriage return character (%0D). The *your_key* value is an arbitrary label provided by the user that may serve as a local identifier for the citation, and it will be included in the output. Be aware that all spaces must be replaced by ‘+’ symbols and that citation strings should end with a final vertical bar ‘|’.

$db=pubmed&retmode=xml&bdata=proc+natl+acad+sci+u+s+a|1991|88|3248|mann+bj|Art1|%0Dscience|1987|235|182|palmenberg+ac|Art2|

**Release Notes**

**ESearch &sort; JSON output format: February 14, 2014**

- ESearch now provides a supported *sort* parameter
- EInfo, ESearch and ESummary now provide output data in JSON format

**ECitMatch, EInfo Version 2.0, EFetch: August 9, 2013**

- ECitMatch is a new E-utility that serves as an API to the PubMed batch citation matcher
• EInfo has an updated XML output that includes two new fields: \(<IsTruncatable>\) and \(<IsRangeable>\).
• EFetch now supports the BioProject database.

EFetch Version 2.0. Target release date: February 15, 2012
• EFetch now supports the following databases: biosample, biosystems and sra.
• EFetch now has defined default values for \&retmode\ and \&rettype\ for all supported databases (please see Table 1 for all supported values of these parameters).
• EFetch no longer supports \&retmode=html\; requests containing \&retmode=html\ will return data using the default \&retmode\ value for the specified database (\&db).
• EFetch requests including \&rettype=docsum\ return XML data equivalent to ESummary output.

Release of new Genome database: November 9, 2011
• Entrez Genome has been completely redesigned, and database records now correspond to a species rather than an individual chromosome sequence. Please see full details of the change at https://www.ncbi.nlm.nih.gov/About/news/17Nov2011.html.
• Old Genome IDs are no longer valid. A file is available on the NCBI FTP site that maps old Genome IDs to Nucleotide GIs: ftp.ncbi.nih.gov/genomes/old_genomeID2nucGI.
• EFetch no longer supports retrievals from Genome (\&db=genome).
• The ESummary XML for Genome has been recast to reflect the new data model.
• To view the new search fields and links supported for the new Genome database, please see https://eutils.ncbi.nlm.nih.gov/entrez/eutils/einfo.fcgi?db=genome.

• ESummary now supports a new, alternative XML presentation for Entrez document summaries (DocSums). The new XML is unique to each Entrez database and generally contains more extensive data about the record than the original DocSum XML.
• There are no plans at present to discontinue the original DocSum XML, so developers can continue to use this presentation, which will remain the default.
• Version 2.0 XML is returned when \&version=2.0\ is included in the ESummary URL.

Demonstration Programs
Please see Chapter 1 for sample Perl scripts.

For More Information
Please see Chapter 1 for getting additional information about the E-utilities.