



The Kalmanovitz Library and
The Center for Knowledge Management

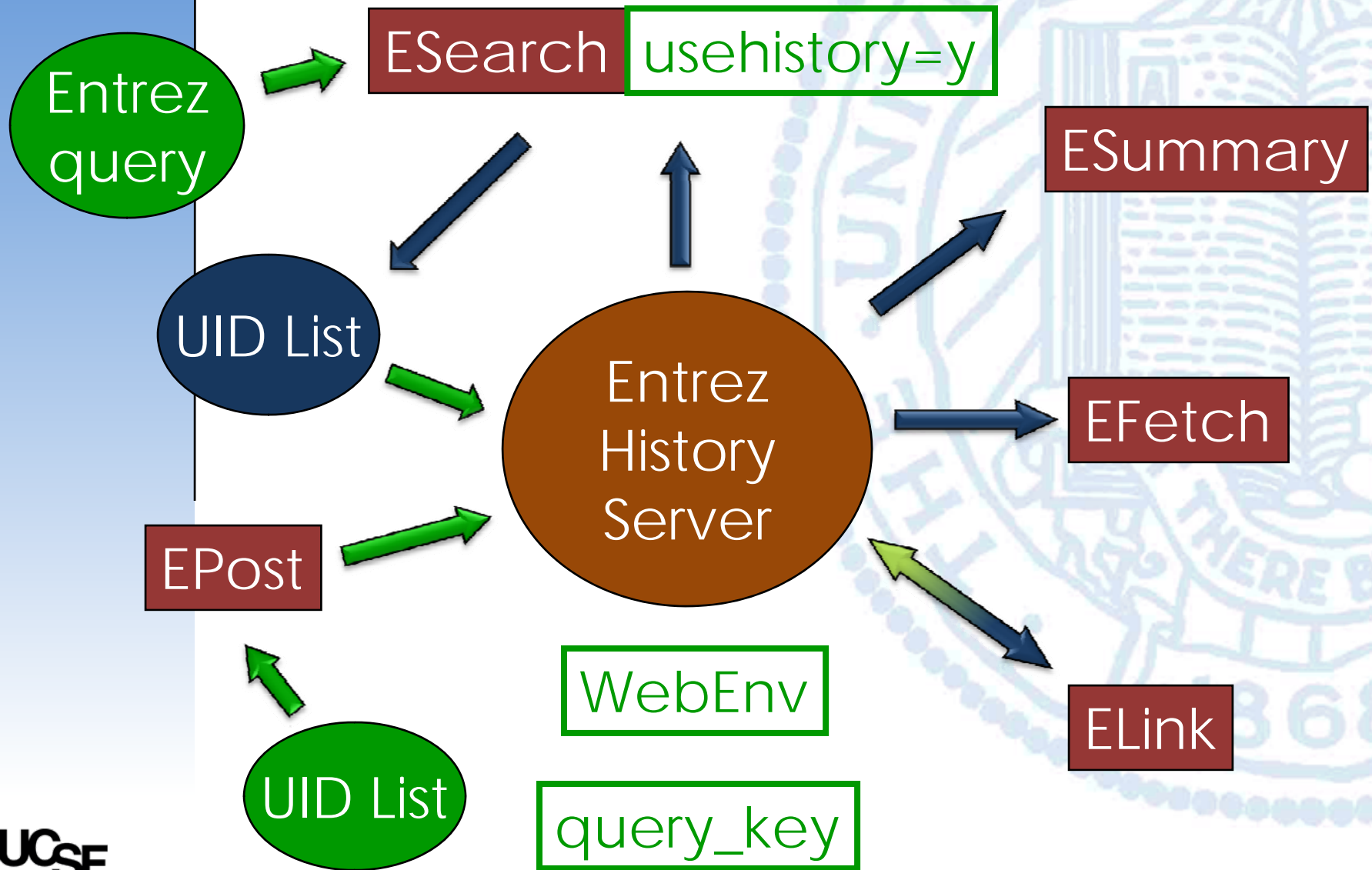
The Four Basic E-Utilities Pipelines

Gilberto da Gente
Bioinformatics Specialist
May 21th 2008

University of California, San Francisco



The Big Picture



Using the History Server

Parameter values from the XML output of eUtil 1 must be assigned to parameters in the URL of eUtil 2

post URL



parse XML

post URL



parse XML



The combination of db, query_key and WebEnv uniquely identify an entire data set as one unit

Using the History

EPost

query_key

WebEnv

&id=35



1

A3B4

WebEnv	query_key	Contents
A3B4	1	35

&id=44



1

C5D9

WebEnv	query_key	Contents
A3B4	1	35
C5D9	1	44

Using the History Cleverly

EPost

query_key

WebEnv

&id=35



1

A3B4

WebEnv

query_key

Contents

A3B4

1

35

&id=44

&WebEnv=A3B4



Use a pre-existing WebEnv!

2

C5D9

WebEnv

query_key

Contents

C5D9

1

35

C5D9

2

44

Using the History Cleverly

ESearch

`&term=rat[orgn]`

query_key

WebEnv

1

A3B4

WebEnv	query_key	Contents
A3B4	1	rat[orgn]

`&term=2005[pdat]`
`&WebEnv=A3B4`

2

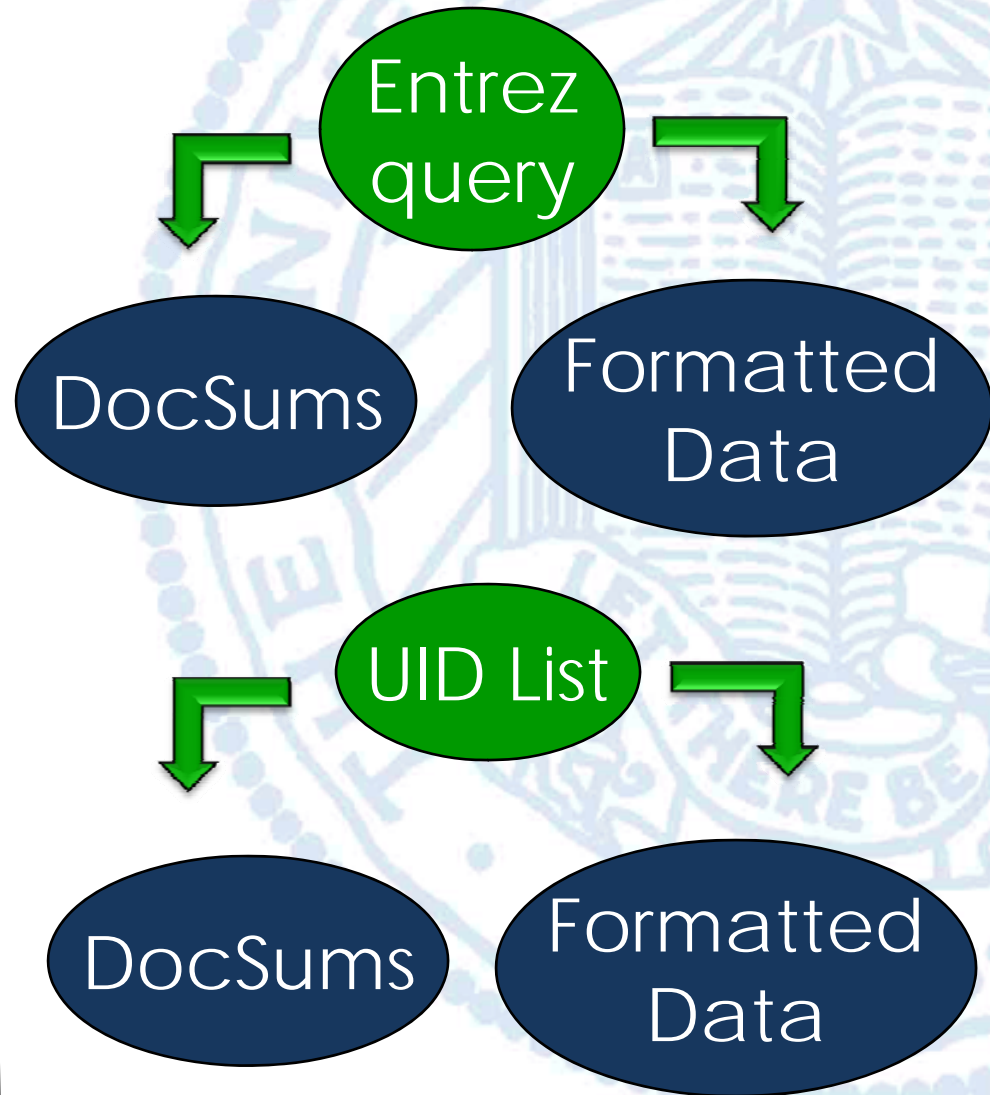
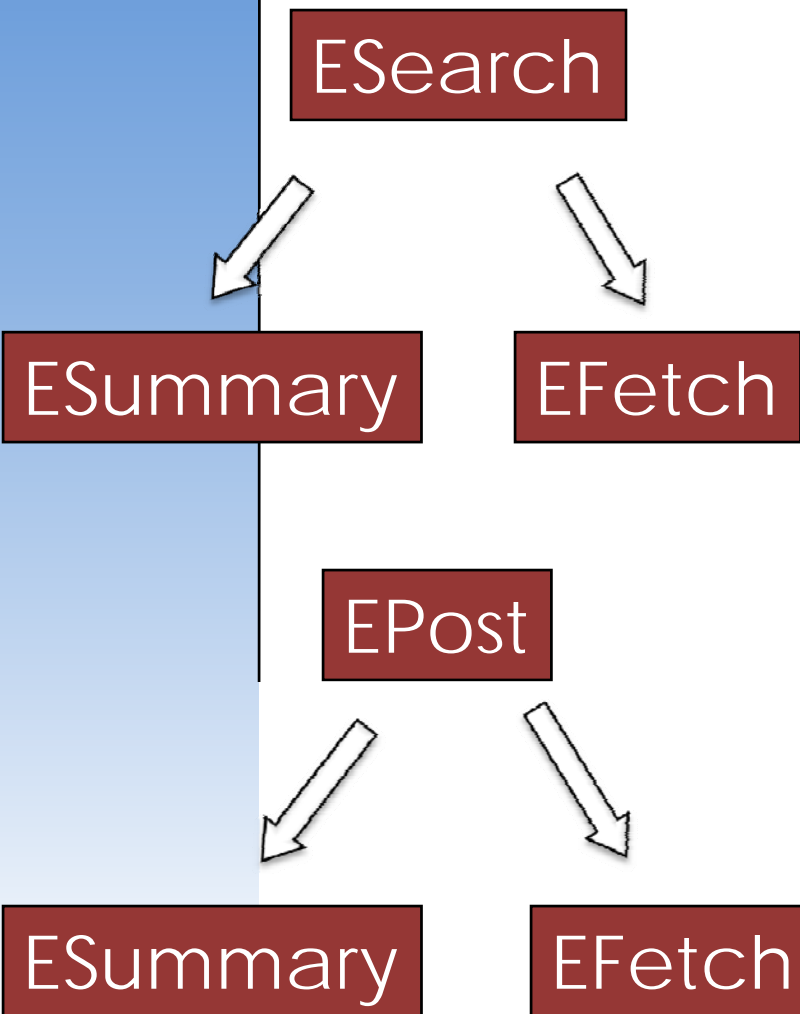
C5D9

WebEnv	query_key	Contents
C5D9	1	rat[orgn]
C5D9	2	2005[pdat]

Building a Pipeline

1. Generate a UID list on the History using
 - ESearch (Entrez query)
 - EPost (UID list)
2. Operate on the UID list on the History by
 - Limiting the list using ESearch
 - Generating a linked UID list using ELink
3. Download the UID list from the History as
 - UIDs using ESearch
 - DocSums using ESummary
 - Formatted data using EFetch

The Four Basic Pipelines



Programming Environment

- ActiveState Perl
- Komodo Edit
- Module of E-utility subroutines in Perl
 - NCBI_PowerScripting.pm
 - ✓ Data uploading subroutines
 - ✓ Data downloading subroutines
 - ✓ Linking subroutines
 - ✓ Utility subroutines
 - Designed to facilitate coding E-utility applications
- Ebot – creates an initial Perl script

GETting a URL

Sends the URL as a string of limited length

```
use LWP::Simple;

$base = 'http://eutils.ncbi.nlm.nih.gov/entrez/eutils/';
$search = 'esearch.fcgi?db=protein&term=mouse[orgn]';

$url = $base . $search;

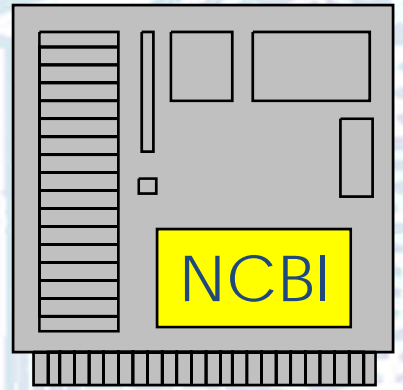
$output = get($url);

print $output;
```

POSTing a URL

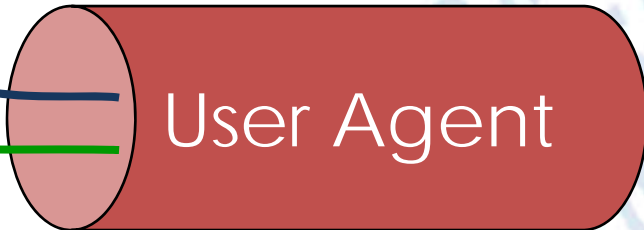


HTTP Post:
Supports URLs of
arbitrary length



HTTP
Request

HTTP
Response



```
$ua = LWP::UserAgent->new
```

```
$req = new HTTP::Request POST
```

```
$response = $ua->request($req)
```

An Example POST

```
use LWP;

my $base = 'http://eutils.ncbi.nlm.nih.gov/entrez/eutils/';
my @ids = (18484066,18483593,18483651);
my %params;
$params{'db'}='pubmed';
$params{'id'}=@ids;
$url_params = "db=$params{'db'}&id=$params{'id'}";
$url = $base . 'epost.fcgi';
#create user agent
my $ua = new LWP::UserAgent;
$ua->agent("epost_file/1.0 " . $ua->agent);
#create HTTP request object
my $req = new HTTP::Request POST => "$url";
$req->content_type('application/x-www-form-urlencoded');
$req->content("$url_params");
#post the HTTP request
$raw = $ua->request($req);
print $raw->content;
```

The E-utility Data Packet

db
query_key
WebEnv



`%params`

We will represent this data packet as a hash and use it for **input** and **output**

```
$params { db }  
$params { query_key }  
$params { WebEnv }
```

NCBI_PowerScripting routines either create **%params** as output or accept **%params** as input

Basic Subroutines

- Data uploading subroutines
 - esearch
 - epost_file
- Data downloading subroutines
 - get_uids
 - esummary
 - efetch_batch

esearch

Find all mouse RefSeq proteins

```
$params{ 'db' } = 'protein';  
$params{ 'term' } = 'mouse[orgn]+AND+srcdb+refseq[prop]';  
($params{ 'usehistory' } = 'y') #by default  
%results = esearch(%params);
```

```
$results{ 'db' } = $params{ 'db' }  
$results{ 'count' }  
$results{ 'query_key' }  
$results{ 'WebEnv' }  
@{ $results{ 'uids' } } (an array)
```

```
<eSearchResult>  
  <Count>26143</Count>  
  <RetMax>20</RetMax>  
  <RetStart>0</RetStart>  
  <QueryKey>1</QueryKey>  
  <WebEnv>0Z8q5Zy0ZaoSpa5OtLa45vo2-OVLRPA-LIbyYP9xZ8SPczp69fnTAK@QfZ9coIOFpMAAHbcWc0</WebEnv>  
  <IdList>  
    <Id>58037564</Id>  
    <Id>58037552</Id>  
    <Id>58037550</Id>
```

A Simple Search...

myscript.pl

```
#!/usr/bin/perl
use NCBI_PowerScripting;
my (%params, %results);

$params{'db'} = 'protein';
$params{'term'} = 'mouse[orgn]+AND+srcdb+refseq[prop]';

%results = esearch(%params);
print "Found $results{'count'} records.\n";
```

command

perl myscript.pl

sub epost_file

Upload a file of 1200 Gene IDs

&id is a file name containing one UID per line

```
$params{db} = 'gene';  
$params{id} = 'genes.in';  
%results = epost_file(%params);
```

input file

```
<ePostResult>  
  <QueryKey>25</QueryKey>  
  <WebEnv>0ey1GksCqOpNLoKz5VOZp_d09SRyaDM71cTGftM5vH2aqQKY1Kd0</WebEnv>  
</ePostResult>
```

```
$results{db} = params{'db'}  
$results{query_key}  
$results{WebEnv}  
$results{num} = # records in input file
```

sub get_uids

Uses ESearch to retrieve all UIDs from a history set

```
$params{db}  
$params{query_key}  
$params{WebEnv}
```

From esearch or epost_file

```
@uids = get_uids(%params);
```



Array of UIDs

sub esummary

ESummary is particularly useful for databases NOT supported by EFetch!

```
$params{db}  
$params{query_key}  
$params{WebEnv}  
$params{outfile} = 'out';  
esummary(%params);
```

From esearch or epost_file

XML DocSums
written to file 'out'

```
<eSummaryResult>  
<DocSum>  
  <Id>6492</Id>  
  <Item Name="PdbAcc" Type="String">1DAN</Item>  
  <Item Name="PdbDescr" Type="String">Complex Of Active Site Inhibited Human Blood Co  
  <Item Name="EC" Type="String">3.4.21.21</Item>  
  <Item Name="Resolution" Type="String">2</Item>  
  <Item Name="ExpMethod" Type="String">X-Ray Diffraction</Item>  
  <Item Name="PdbClass" Type="String">Complex(Serine ProteaseCOFACTORLIGAND)</Item>  
  <Item Name="PdbReleaseDate" Type="String">1997/9/4</Item>  
  <Item Name="LigCode" Type="String">CA|CAC|CL|FUC|GLC</Item>  
  <Item Name="LigCount" Type="String">5</Item>  
  <Item Name="ModProteinResCount" Type="String">13</Item>  
  <Item Name="ModDNAResCount" Type="String">0</Item>  
  <Item Name="ModRNAResCount" Type="String">0</Item>  
  <Item Name="ProteinChainCount" Type="String">5</Item>  
  <Item Name="DNAChainCount" Type="String">0</Item>  
  <Item Name="RNAChainCount" Type="String">0</Item>  
</DocSum>  
</eSummaryResult>
```

sub efetch_batch

Posts a series of EFetch URLs, each retrieving a batch of records.

`$params{batch} = -1` → All records in 1 URL
`$params{batch} = 500` → default

```
$params{db}  
$params{query_key}  
$params{WebEnv}  
$params{retmode} = 'text';  
$params{rettype} = 'fasta';  
$params{outfile} = 'prots.faa';
```

From `esearch` or `epost_file`



Protein FASTA written to file 'prots.faa'

Retrieving Batches

A single EFetch URL can download any number of records, but retrieving batches can be more reliable for large downloads.

```
&retstart=0&retmax=500
```

```
&retstart=500&retmax=500
```

```
&retstart=1000&retmax=500
```

Retrieve successive batches of 500 records until the entire dataset is downloaded

Batch retrieval is easily implemented using a for loop:

```
for ($retstart=0; $retstart < $total; $retstart+=$retmax) {  
    ... &retstart=$retstart&retmax=$retmax...  
}
```

Linking Two Calls

ESearch



ESummary

Use &WebEnv and &query_key instead of &id in esummary

1. What output does ESearch produce?

```
$results{db}
$results{count}
$results{query_key}
$results{WebEnv}
```

2. What input does ESummary require?

```
$params{db}
$params{query_key}
$params{WebEnv}
```

Use output hash from esearch as input to esummary!

Retrieving DocSums

ESearch



ESummary

```
$params{'db'} = 'protein';  
$params{'term'} = 'mouse[orgn]+AND+kinase[title]';  
  
%params = esearch(%params);
```

Now contains **db**, **WebEnv**, **query_key** for search results

```
$params{'outfile'} = 'prots.sum';
```

Tells esummary where to write its

```
esummary(%params);
```

Retrieving Data

EPost



EFetch

```
$params{ 'db' } = 'protein';  
$params{ 'id' } = 'gilist.in';  
%params = epost_file(%params);  
  
$params{ 'rettype' } = 'text';  
$params{ 'retmode' } = 'fasta';  
$params{ 'outfile' } = 'prots.faa';  
  
efetch_batch(%params);
```

The Pair Library: Part I

The Four Basic Pipelines

ESearch → ESummary

search_summary.pl

EPost → ESummary

post_summary.pl

ESearch → EFetch

search_fetch.pl

EPost → EFetch

post_fetch.pl