# Slide 1 – Welcome

Good afternoon and thank you for joining us for “Welcome to E-utilities for PubMed”, which is the first in our “Insider’s Guide to Accessing NLM Data” series of classes.

My name is Mike Davidson, and I’m a librarian at the National Library of Medicine in Bethesda, Maryland. I play a couple of different roles here at NLM:

As a member of the NLM Training Team, I help develop and present training on NLM products in a variety of formats. A lot of our training is on PubMed, but we cover other areas as well.

I’m also on the PubMed Customer Service Team, so if you’ve ever written in to the Help Desk, there’s a chance I was the one who answered your question.

# Slide 2 – Agenda (ANIMATED)

Before we get into today’s agenda, let me start off by saying that we know that our audience has lots of different levels of experience.

We expect you all have some background searching and working in PubMed, but some may have more technical expertise than others.

We’re trying to offer a little something today for everyone:

If you’re more technically experienced, and the stuff we’re covering is old news to you, hang in there, there’s some more advanced stuff coming.

If you’re a little less technically experienced, stick with us, there’s some more introductory stuff for you too.

The Agenda:

* **[CLICK]** We’re going to start off today by taking a look at some particularly challenging PubMed questions
* **[CLICK]** We’re then going to introduce you to a solution that may help you answer those questions
* **[CLICK]** We’ll provide a little background on how that solution works, to give you a good solid foundation
* **[CLICK]** We’re going to explore some practical implementations of the solution, including some tools you can use in the real world, which will hopefully help us
* **[CLICK]**Answer our questions!

As I said at the beginning, one of my jobs is on the PubMed Customer Service Team, and we’ve started to see an increase in a certain type of questions. These questions all follow the same pattern:

# Slide 3 – How can I get “X” out of PubMed in “Y” format?

How can I get a particular set of data out of PubMed in a certain format?

Here’s an example:

# Slide 4 – An Example (ANIMATED)

We want to figure out which authors have a lot of recent publications on a subject? Who’s “hot” right now in a particular field?

We don’t just want authors with a lot of publications overall, but authors with a lot of recent publications.

We also want to know which organizations are funding research on this topic.

And I want this data in a format where I can analyze it or visualize it, like in Microsoft Excel.

**[CLICK]** This is a challenge in PubMed, because, while you can download results in a CSV format, the format groups all the authors together in the same field. You’d have to pull them out and separate them to analyze.

Also, the CSV format doesn’t include funding information, so you’d have to get that data a different way.

Here’s another example…

# Slide 5 – Another Example (ANIMATED)

Let’s say we’re trying to visualize trends in research on opioid abuse, broken down by specific types of opioids.

Here’s a screenshot of a results page for a very basic search for “fentanyl abuse”.

Look over on the right side and you’ll see what we call the Discovery Bar, which contains a lot of useful information to help expand and provide context for your search results. One of the features you’ll sometimes see is this Results by Year histogram.

**[CLICK]** I’ve blown it up here so you can see it a little better.

The chart displays how many of your search results were published in each year.

If this were a live demo of PubMed, we could click this Download CSV link and pull the data behind this graphic to use however we want.

Now let’s say I want to compare this trend to a different search, **[CLICK]** for “oxycodone abuse”.

We can get the histogram over here again **[CLICK]**, but if we want to compare that oxycodone graph with the fentanyl graph **[CLICK]**, we’d have to do a little bit of work. While this results by year graph is very useful, **[CLICK]** it doesn’t give us any options for customization or advanced analysis.

Now, these are kind of advanced questions: the overwhelming majority of PubMed users can go to [www.pubmed.gov](http://www.pubmed.gov/) and find what they need. It is an incredibly useful tool that I’m sure a lot of you use all the time, with great success.

But as we’ve seen, for certain jobs, advanced users like you sometimes need different tools. It’s all about having the right tool for the right job.

# Slide 6 – What do you need your tools to do? (ANIMATED)

So advanced users, what do you need your tools to do? **[CLICK]**

You need them to get **exactly** the data you need **[CLICK]** and only the data you need **[CLICK]** in the specific output format you need it in.

And so, we want to introduce you to…

# Slide 7 – E-utilities

E-utilities!

E-utilities is an API for accessing PubMed and other NCBI databases.

Great! But…

# Slide 8 – What does that ***mean***?

What does that **mean**?

# Slide 9 – What is an API?

Let’s start off with, what’s an API? It’s a term that gets thrown around a lot, but is only rarely explained. So if you’ve never used an API before (or you have used one, but still don’t know what API is), let’s start with a definition:

API stands for Application Programming Interface. It’s a set of tools, routines and protocols for building software applications.

Okay, that’s a good start, but, as a definition, it’s still pretty generic. So let’s get a little more specific about…

# Slide 10 – The E-utilities API

A set of tools, routines, and protocols that allow you to interact directly with the data in over two dozen NCBI databases, including PubMed, the MeSH database, and PMC.

# Slide 11 – Okay, but what does ***that*** mean?

Okay, but what does **that**mean?

# Slide 12 – The E-utilities API is just a series of rules for querying a database.

Really, it’s simpler than it sounds.

The E-utilities API is just a series of rules for querying a database.

For today, the database we’re going to be talking about is PubMed, but, as we mentioned, these rules work for querying a bunch of different NCBI databases.

For those of you who aren’t familiar, NCBI is one of the divisions of the National Library of Medicine. They do a lot of different stuff, but one of their jobs is developing and maintaining a variety of databases of biomedical information, including PubMed, and they have developed the E-utilities API as a way of querying those databases.

# Slide 13 – URLs as Database Queries (ANIMATED)

The way you construct these queries is in the form of structured URLs. The URL asks the NCBI server for information, returns it in a certain format. What information you get, what format it’s in, depends on how you construct the URL.

Now, in practice, you yourself probably won’t be constructing these URLs on a regular basis. **[CLICK]** There are some concrete, practical tools that can do that for you!

In order to understand how E-utilities works, and what your tools are doing, **[CLICK]** we’re going to go over the basics of creating an E-utilities URL.

Now, it might seem a little daunting at first: because each E-utilities URL is designed to ask a question to the server, and there’s basically an infinite number of questions you can ask, there are just as many URLs you can create!

However, all of these URLS follow the same set of rules and the same basic pattern.

Let’s take a look.

# Slide 14 – The three parts of an E-utilities URL

We’re going to go into a bunch more detail soon, but, in short, the three parts of an E-utilities URL are:

* The Base URL, which is the address of the E-utilities server
* A Utility Name, which is the name of the specific tool you are using
* And some Parameters, which are the details of your query.

In practice, those parts look something like this:

# Slide 15 – The three parts of an E-utilities URL

The Base URL is this chunk of URL here [POINT], and that’s always the same.

The Utility name will be one of nine options that look something like this. [POINT] They all have this .fcgi extension on the end. We’ll talk about those options more in just a second.

And the Parameters, which define the details of your query.

Now, this part looks a little scary. Once you learn the syntax, though, you’ll start to see some familiar search terms and tags here. It starts getting pretty easy for someone who has spent any time using PubMed.

For example, we can see a “[Journal]” tag here [POINT], and this is [POINT] is a Boolean AND. Don’t get thrown off by all of the pluses: we just use them in place of spaces, since we can’t have spaces in our URLs.

Okay, let’s go through these three parts one step at a time.

# Slide 16 – The Base URL

First is the Base URL.

The Base URL is always the same. This is the beginning of every E-utilities URL, and identifies the address of the E-utilities server.

# Slide 17 – E-utilities Utilities

The second part is choosing a specific utility.

Each of the nine utilities asks a different kind of question.

For example, ESearch searches a database and returns a list of unique identifiers (or UIDs) for the records that meet your search criteria.

So if you’re searching PubMed, this would return a list of PMIDs.

It’s important to note that ESearch doesn’t retrieve full records, just PMIDs. For that you’d use a different Utility:

EFetch takes a series of unique identifiers (like PMIDs; like, for example, the list of PMIDs you’d get from an ESearch), and retrieves the full record (including all of the data in all of the fields) for each of those UIDs.

Note that the records you’re retrieving are the exact same data that’s in the web version of PubMed at www.pubmed.gov, so there are no differences in the content.

# Slide 18 – E-utilities Utilities, cont’d

There’s also seven more utilities that each do different things, but we’re not going to get into all of them now.

If you want to see the full list, and what they all do, check out our Insider’s Guide website at <https://dataguide.nlm.nih.gov>

# Slide 19 – Parameters

The parameters vary by utility. Not all parameters apply to all utilities.

Some are required, some are optional.

One parameter you’ll see a lot is the “db” parameter, which specifies which of the NCBI databases you’ll be querying. In this case, it’s PubMed, as we’ll be working in PubMed for all of our examples today.

The “term” parameter would be used with ESearch to specify a search string you’d like to search for.

# Slide 20 – Parameters, cont’d

Some other parameters you might see include the “id” parameter, which specifies a list of UIDs (in this case, PMIDs). EFetch uses this to specify which records we want to fetch.

You might also see the “retmode” and “rettype” parameters, which specify what format you’d like your results returned in.

There are also many, many more parameters. You can get details on all of them at <https://dataguide.nlm.nih.gov>.

# Slide 21 – The Three Parts…Together! (ANIMATED)

Now let’s put the three parts together.

**[CLICK]** We start with the Base URL, which is always the same.

**[CLICK]** We then add our utility, which in this case is ESearch (with that .fcgi extension). This means we’re going to be searching a database.

**[CLICK]** Finally, we have our parameters, which specify the details of our query. We have a “db” parameter, letting us know we’ll be searching PubMed, and a “term” parameter to define our search string. Again, the pluses here are in place of spaces.

**[CLICK]** And that’s it! We have an E-utilities URL!

# Slide 22 – Using an E-utilities URL

And we can just put that URL right in our browser and see what happens.

[LIVE DEMO]

We paste the URL into a browser and hit Enter, and we get our results right in our browser window.

First off, note that we have the “Count” at the top that shows us how many results we have.

Then we have a list of PMIDs here. Note again that since we used ESearch, we have only PMIDs, not full records.

This stuff at the bottom is NOT the full records. It’s actually the query translation, just like you would find in Search Details in the web version of PubMed.

All of the query translation and Automatic Term Mapping that happens in the web version of PubMed also happens with ESearch.

Again, the exact query is determined by our parameters. In this case, we’re searching for articles on breast cancer published in the journal Science in 2008. If we wanted to change that to a different journal, we can just tweak that parameter.

Try changing “science” in that URL to “nature”, and see what happens.

We’re now getting a different set of PMIDs, because we’re looking for records from a different journal.

So that’s two “ESearch” examples. Let’s take a look at an “EFetch” example:

# Slide 23 – Another E-utilities URL

Looking for the three parts, we have the Base URL, we have an EFetch instead of an ESearch for our utility, and we have some different parameters, including an “id” parameter, to tell us which PMIDs we will be fetching, and “retmode” and “rettype” parameters, which determine the output format.

[LIVE DEMO]

Popping this URL into our browser, we see that we have retrieved two full records, in a text format, using the abstract view. This is similar to what you would see if you used the Abstract (text) option in PubMed.

We can change the way our results are formatted by changing our parameters.

We can delete the current “retmode” and “rettype”, and just use “retmode=xml”.

This gets us the same two records, because we haven’t changed our “id” parameter, but now we see them in full XML.

Now we’ve spent about 10-15 minutes to accomplish something we could do in PubMed in about 5 seconds.

Remember what we said before: It’s all about the right tool for the right job. For some questions, going to [www.pubmed.gov](https://www.pubmed.gov/) is absolutely the right choice.

But simple queries like this is not really the point of E-utilities.

Where it’s strongest is when you combine multiple URLs together, using the results of one URL to create the next.

Let’s take a look at an example:

This is our “fentanyl abuse” search from earlier. It gives us XXX PMIDs. Ideally, we then want to retrieve the full records for these PMIDs, so we can look at the publication date and create the histogram, to compare to the “oxycodone abuse” histogram.

To do this, we need an EFetch.

I’m going to type this in to the URL bar, which you may or may not be able to see, but I’ll try and narrate as I go.

We start with the base URL again: https://eutils.ncbi.nlm.nih.gov/entrez/eutils/

Then we add in our EFetch utility: efetch with our .fcgi? extension.

Then we add in our “db=pubmed” parameter.

And then we add “&id=”, so we can specify which PMIDs we want to fetch. And I’ll copy and paste this first one, then put a comma, then copy and paste this second one, and put a comma…

This is ridiculous. There is no way this is practical. And even if it wasn’t a terrible idea to hand copy XXX PMIDs, remember that we would also have to do this for the “oxycodone abuse” search, as well.

This is why we want to use E-Utilities in a programming environment.

# Slide 24 – The ***real*** power of E-utilities

Using E-utilities in a programming environment gives us a number advantages. We don’t have to create the URLs by hand; we can write the rules for creating the E-utilities URLs right into the program.

We can automatically take the results of one query, and use them to generate the next query. For example, we can create an EFetch URL using the PMIDs that were the result of our ESearch.

We can also use programming tools to manipulate our output. For example, we can pull out just the publication date from the records we fetched.

# Slide 25 – Using E-utilities in a programming environment (ANIMATED)

Ordinarily, in order to use E-utilities in a programming environment, you would need to complete a series **[CLICK]** of complicated **[CLICK]** time-consuming **[CLICK]** steps **[CLICK]** just to get started, before you even get close to actually answering your question.

**[CLICK]**

**But wait…**

We’ve already done a lot of that for you!

# Slide 26 – EDirect

Let me introduce you to EDirect. EDirect is software developed by NCBI. It’s basically a set of tools with the E-utilities URL creation rules built in.

You’ll execute EDirect commands, and EDirect will create the URLs, query the server, and give you your results.

EDirect also has tools to extract the exact elements you need from the full PubMed XML records you download.

And because EDirect works in a Unix environment, it doesn’t take that much to get started, but it can work with other tools and programming languages, so advanced users can build more elaborate and powerful solutions.

# Slide 27 – What do I need to get started with EDirect?

In order to use EDirect, there are a couple of things you’ll need to do first.

To start, you’ll need access to a Unix command line environment.

If you’re using a Mac or a Linux machine, you already have a Unix terminal ready to go.

If you’re using a Windows computer, you’ll need to install a Unix terminal emulator. I use a free application called Cygwin, which you can download and install, and use as your Unix command line environment.

In addition to a Unix environment, you’ll also need the EDirect installation package. You can download and install this from within your Unix environment. We have detailed, easy-to-follow instructions for how to install EDirect (which also includes a link to download Cygwin), up on the Insider’s Guide website.

# Slide 28 – What do I need to know to use EDirect?

EDirect doesn’t take too much prior knowledge to learn, but it helps to have some knowledge of Unix…

…or, a knowledgeable friend you can pester with questions.

Even if you don’t have someone like this, there’s lots of help online and it’s pretty easy to find. Just go to your favorite search engine and search for “Unix” plus whatever you’re trying to do. Someone’s probably asked the same question, and hopefully got an answer you can use.

One of the big advantages of EDirect is its ability to extract specific data elements from the PubMed XML format. It will help to have a familiarity with the basic structure and syntax of XML. If you’re already used to looking at PubMed data in XML format, you’re good to go. If not, feel free to start taking a look, which will help you get a feel for the structure of the data.

It can also be helpful to have some programming knowledge, particularly Perl or Python. It’s absolutely not required, but the more you know about programming, and about Unix, the more you’ll be able to do with EDirect.

That said, even without much prior knowledge, you should still be able to get EDirect working for you.

# Slide 29 – EDirect in Action (ANIMATED)

So let me show you what EDirect looks like in action.

As I said, this works in a Unix environment. What that means is that we’re going to use this “terminal” (in this case Cygwin) to interact with the computer.

This is our command prompt. We just type in the commands that we want to run, and execute them directly from here.

So, for example, if I wanted to search PubMed, I’d type in an esearch command **[CLICK]** for “fentanyl abuse”, hit enter **[CLICK]** and get my results back in XML.

Now, as we said at the top, we know we have people attending today with a wide range of experiences.

I want to give those of you with less programming experience fair warning: as you’ve already seen a little bit, in the next couple of minutes, we’re going to start showing you some things and throwing around some terms that might be a little scary, outside your comfort zone.

# Slide 30 – Don’t Panic! (ANIMATED)

Please don’t panic! **[CLICK]**

Even if you’re less technically proficient, if you’re willing to put in a little effort, I **[CLICK]** promise you, you will be able to pick this stuff up and put it to use.

Over the next couple of slides, we’re going to try and show some examples of E-utilities in action, and what it can do for you.

If you’ve got a little more of a technical background, some of this stuff will be old news to you, and you’ll start to see right away how you can get started with E-utilities.

If this is all a little newer to you, don’t worry too much about the details of how we did all of this stuff. For now, just take a broader view of the kind of stuff we can do with E-utilities. If it seems like something that might be useful, you can take one of our future classes, read some of the material on the Insider’s Guide website, or find the techie at your organization (every organization has at least one) and have them help you through the early stages.

With all that said, let’s take a look at the way EDirect can quickly and seamlessly put E-utilities to work for us.

# Slide 31 – Where did the URLs go? (ANIMATED)

So EDirect commands look a bit different than the URLs we constructed before, but it’s really just a faster, simpler way of creating those URLs.

Behind the scenes, this command is actually creating this URL **[CLICK].**

And if you look closely, you can see some of the similarities.

**[CLICK]** This “esearch” here is saying which utility we want to use, just like we do in the URL over here.

**[CLICK]** This –db argument says to use “pubmed” as the “db” parameter for our URL.

**[CLICK]** And here’s our search query “fentanyl abuse”, which is going into the “term” parameter.

# Slide 32 – Where did the URLs go? (Cont’d) (ANIMATED)

And it works just the same for efetch. Here’s the command…

**[CLICK]** …and here’s the URL. And again, we can see that the…

**[CLICK]** ...“efetch” command tells us to use the “Efetch” utility.

**[CLICK]** The –db argument tells us to use pubmed.

**[CLICK]** We have an –id argument that tells us which PMIDs we want to fetch ([POINT] it’s all of these right here).

**[CLICK]** And we also have a –format argument, which creates a URL that uses the “retmode” parameter XML to get us XML records.

Now, you won’t actually see these URLs. EDirect creates them in the background, queries the server, and returns your results.

But again, this doesn’t get us any farther than we were before, when we had to create a URL, then take the results of that manually and feed them into the next URL.

# Slide 33 – Putting EDirect commands together

Where EDirect ***really*** gets good is when you can put multiple commands together. **[CLICK]**

These two little characters here tell us to take the result of this esearch command and pipe them into our next command, which we can find on the next line.

You’ll notice that this efetch is missing a few things

It doesn’t have a –db argument, because it already knows what database we’re in: PubMed. We’ve already told it that in the esearch.

It also doesn’t have an –id argument, because it knows to take the PMIDs that were the output of the esearch command on the previous line.

These two lines together allow us to search PubMed, then pull the complete XML records of all of our results. And now we’re ready to start answering some questions.

# Slide 34 – Answering questions with E-utilities

We’re going to look at a few of the examples we talked about at the beginning of the session.

Again: Don’t worry about following along every detail now. This is just to give you an idea of what E-utilities can do.

Don’t even pay too much attention to the details of the code: All of this sample code is available online, so you can review, tweak, reuse or repurpose.

# Slide 35 - Active authors on a topic (ANIMATED)

Remembering back to our earlier examples, we were trying to find authors that are especially prolific in the last few years on our chosen topic.

We start again with a Unix Terminal, then we’d put in our series of commands.

You could type them in by hand, but I usually write them out in a text editor and then copy and paste the whole script into the terminal. **[CLICK]**

Again, we won’t go into details, but you’ll see some familiar elements:

We have the esearch command here, our –db is pubmed, and here’s our search query: It’s composed of a topic string, ANDed together with a date range.

Then we pipe the results of this esearch into our efetch, and then we pipe the fetched records into our next line.

We haven’t talked about “xtract” yet, and we’re not going to go into a lot of detail on it right now. It’s an incredibly useful command, built into EDirect, which extracts specific elements from XML output, and rearranges them in a table. It’s very customizable: you choose the output, the arrangement, the separators, etc.

This is what makes EDirect really useful. It’s a little outside the scope of today’s class, but in our EDirect for PubMed Basics class, we’ll spend a LOT of time talking about xtract.

We have a few more lines of post-processing here, which makes sure we’re getting the data we want in the format we want.

Once we have all our commands in there, we hit enter **[CLICK]**, and we see our results.

Normally, this would take a little longer, but I prepared these screenshots ahead of time.

We can see a list of the top 10 authors who have the most publications on this topic in the last few years.

# Slide 36 - Agencies funding research on a topic (ANIMATED)

Our next example, pulling out the list of the agencies that have funded research on this topic, is very similar actually.

Starting with our blank command prompt, we put in our pre-written code **[CLICK]**, and we can see that the esearch and efetch lines are exactly the same.

The only difference is which elements I’m extracting.

When I run this code, instead of getting a list of the most prolific authors **[CLICK]** I get a list of the top ten agencies that are most frequently listed as funding research on these records.

# Slide 37 – Results by Year (LIVE DEMO)

Now, for our last big demo, we’re going to try and do a version of that Results by Year histogram, but comparing our two opioid abuse searches. This is just an example of the kind of thing you can do with E-utilities, to give you a little more customizability.

For this, one, we’re actually going to demo it live, so who knows what’s going to happen!

[LIVE DEMO]

So again, this is our command prompt.

I’ve already written my code in my text editor, and here’s what it looks like.

[SHOW CODE IN TEXT EDITOR.]

esearch -db pubmed -query "fentanyl abuse" -datetype PDAT -mindate 1988 -maxdate 2017 | \

efetch -format xml | \

xtract -pattern PubmedArticle -block PubDate -element Year MedlineDate | \

cut -c -4 | \

sort-uniq-count-rank | \

sort -n -t $'\t' -k 2 > fentanyl\_abuse.txt

esearch -db pubmed -query "oxycodone abuse" -datetype PDAT -mindate 1988 -maxdate 2017 | \

efetch -format xml | \

xtract -pattern PubmedArticle -block PubDate -element Year MedlineDate | \

cut -c -4 | \

sort-uniq-count-rank | \

sort -n -t $'\t' -k 2 > oxycodone\_abuse.txt

join -j 2 -o 0,1.1,2.1 -a1 -a2 -e0 -t $'\t' <(cat fentanyl\_abuse.txt) <(cat oxycodone\_abuse.txt) > abuse\_compare.txt

You’ll notice it’s quite a bit longer, and I’ll talk you through it in just a moment, but I’m going to copy and paste it into my command prompt first to get it running. **[CLICK]**

It’s going to take a few moments to complete.

Just like before, we’re running a search, retrieving full XML records, and extracting data from the XML, but rather than outputting the results right on our screen, we’re saving the output to a file.

For this example, I’m limiting the results to a date range starting in 1988, which is the time frame I’m interested in looking at.

Also, there’s also a few more lines of code for each search to get the data into the correct format before we save it. Then we have to do that whole thing over again for the second search.

Finally, we have this last line, which will merge the two files together.

[ONCE COMPLETED] We can then open up that text file in a text editor, and see our results.

We can then open up that text file in a text editor, and see our results.

We can also open the file up in software that can analyze or visualize data, like Microsoft Excel, and we can use that data as the basis for a chart that lets us compare the two histograms.

I’ve made this chart ahead of time, using that same method to get the data.

As I said before, this is just an example, and it’s pretty scaled down so it would run fast enough to complete. You could use a different search, or extract different data elements. You could also style or visualize this data differently.

For example, we could look at a chart is a comparison of PubMed results for “heroin addiction” vs “opioid addiction”, which also might tell an interesting story.

Because we’re using E-utilities, we’ve got a more flexible set of tools at our disposal.

Okay, before we finish up today, let’s go back to the slides for just a minute. I just want to zoom back out a little bit to the big picture.

# Slide 38 –E-utilities: the bigger picture. (ANIMATED)

The whole point of all of the stuff we’ve been talking about today is figuring out different ways to access PubMed Data **[CLICK].**

As we’ve seen, **[CLICK]** the E-utilities API is a great channel that you can use to get at that data, and **[CLICK]** EDirect is a tool that can make using E-utilities easier, if you’re working in **[CLICK]** a Unix Terminal environment.

However, if you’re working in a different environment, you will probably want to use different tools.

# Slide 39 – E-utilities: the bigger picture (ANIMATED)

For example, let’s saying you’re working with the R programming language. **[CLICK]**

Some of you may be familiar with R: it’s really good for doing statistical analysis and data visualization.

Let’s say you or someone in your organization wants to do some analysis or visualization of PubMed data, and you want to use R to do it.

In order to get at that PubMed data quickly and easily, you’ll want to use the same E-utilities API, **[CLICK]** but because you’re in a different environment, you’ll want to use a different tool.

In R, there are a number of “packages” of commands, kind of like EDirect, but designed to work in the R environment. **[CLICK]** These packages, **[POINT]** like RISmed, rentrez, or reutils, help make it easier to use the **[POINT]** E-utilities API to access **[POINT]** PubMed data from the **[POINT]** R environment.

# Slide 40 – E-utilities: the bigger picture. (ANIMATED)

Now, if you’re working in a different environment (not Unix or R **[CLICK]**), you might have different tools altogether **[CLICK]** (maybe you’d even have to build your own tools), but you’d still be using the same **[CLICK]** E-utilities API.

# Slide 41 – E-utilities: the bigger picture. (ANIMATED)

And in some environments, like a web browser **[CLICK]**, you might not use special tools at all. We can just use the URL creation rules to use E-utilities **[CLICK]** directly.

This is what makes E-utilities so useful and flexible: you can use E-utilities to access PubMed data, regardless of what environment you’re working in. You just have to find the right tool for the job.

# Slide 42 – Where do we go from here?

Hopefully, today’s presentation introduced you to the basic idea of E-utilities, and to EDirect, but we know there’s still a lot more learn. Fortunately, this is just the beginning of the Insider’s Guide.

If you’re interested in something a little more interactive, we’d encourage you to sign up for one of the other Insider’s Guide classes.

EDirect for PubMed is more of a hands-on workshop that introduces you to the basics of using EDirect to get exactly the PubMed data you want, in the format you want it in. It’s a five-part series, with an optional “office hours” session, and you can register at dataguide.nlm.nih.gov.

# Slide 43 – Where can I get help?

If you are interested in continuing to learn about these tools, I would encourage you to check out the Insider’s Guide website at dataguide.nlm.nih.gov.

[LINK IN CHATBOX: <https://dataguide.nlm.nih.gov>]

We have documentation, sample code, recordings and slides for previous classes, as well as detailed installation instructions for EDirect

You can also check out NCBI’s documentation for E-utilities and EDirect. While we focus mostly on the use of these tools to access PubMed data, NCBI’s documentation is much broader, as it covers all of the different database that E-utilities works with.

# Slide 44 – One more example, to get you thinking…

Before we finish up, I just have one more example I want you to think about.

Say we want to know what topics our organization is publishing on. Our organization has many authors and research components with a variety of names, acronyms, abbreviations, etc.

How can we efficiently search for our organization and analyze the topics of published articles?

This is the kind of question we get at Customer Service at lot, and it is absolutely doable with E-utilities.

I’m not going to demo it right now, but we have some sample code on the website that you can check out.

If this is the kind of question you have, or folks in your institution have, maybe take a look at our sample code and see if it’s something you could adapt.

And even if it isn’t, consider taking a look at the code anyway, just as a practice exercise. Start thinking about how you could accomplish this, or how you could adapt it to something that would be relevant to you.

This might be similar to a problem that you’ve had trouble solving in the past because you didn’t have the right tools for the job. And I’m sure many of you have some jobs that E-utilities is the right tool for.

So get out there and put E-utilities to work!

# Slide 45 – Questions?