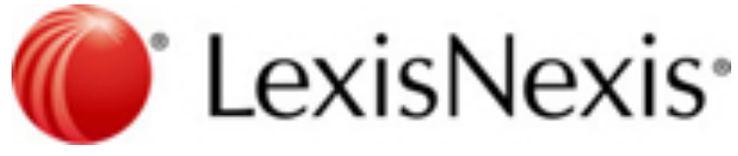
HPCC Systems - Big Data

Roxie -

In-Memory Data & Index ,

Sub-Second Query Cluster

By Fujio Turner



@FujioTurner

Who is LexisNexis®?

LexisNexis is a provider of legal, tax, regulatory, news, business information, and analysis to legal, corporate, government, accounting and academic markets.

LexisNexis has been in business since *1977* with over 30,000 employees worldwide.

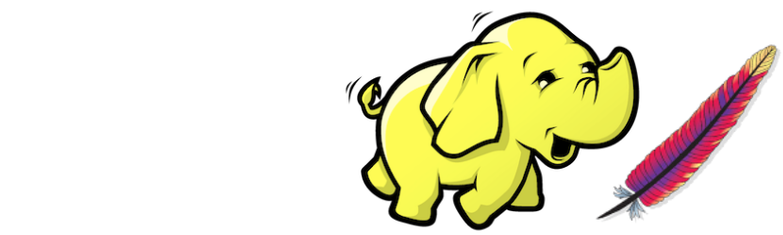
What is HPCC Systems?

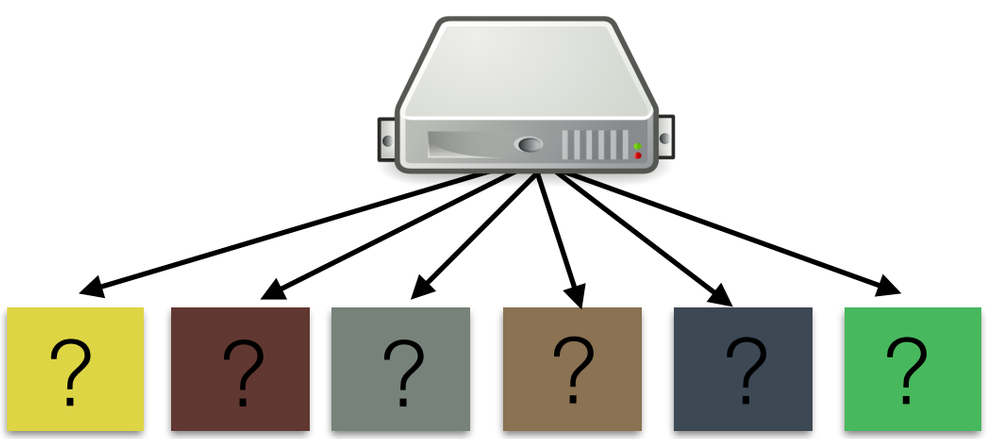
HPCC SYSTEMS®

LexisNexis Risk is the division of the LexisNexis which focuses on data, Big Data processing, linking and vertical expertise and supports HPCC Systems as an open source project under Apache 2.0 License.

Comparison

Block Based





JAVA

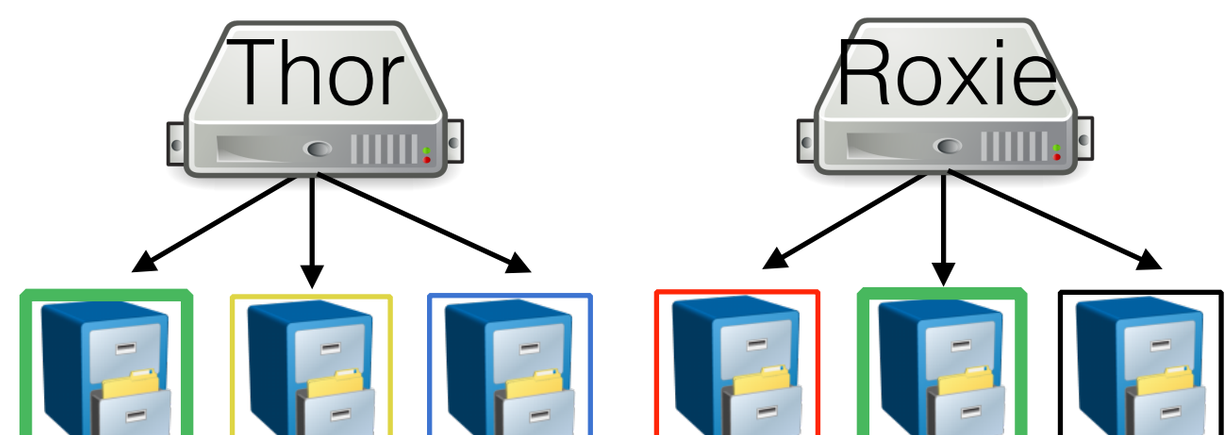
Petabytes

1-80,000 Jobs/day

Since 2005

File Based

HPCC Systems



C++

Exabytes

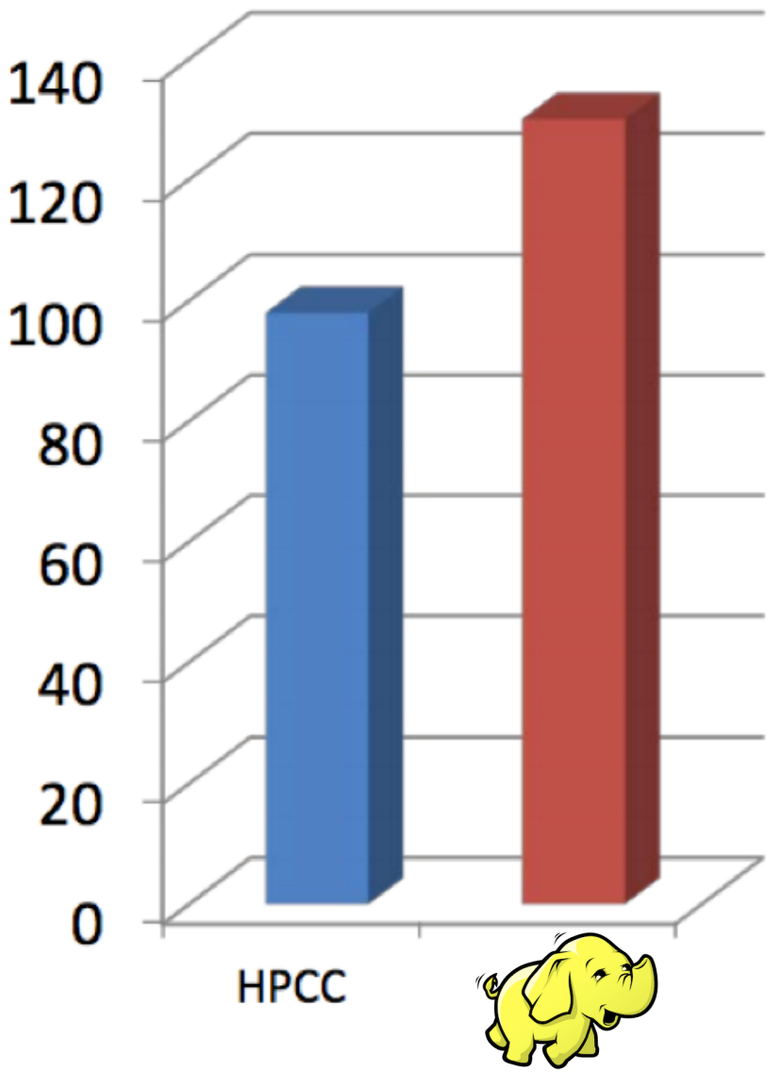
Non-Indexed 4X-13X

Indexed: 2K-3K Jobs/sec

Since 2000

Non-Indexed Full Data Set

Execution Time (seconds)



Productivity

```
// Perform global terasort
rec := record
      string10  key;
      string10  seq;
      string80  fill;
       end;
in := DATASET('nhtest::terasort1',rec,FLAT);
OUTPUT(SORT(in,key,UNSTABLE),,'nhtest::terasort1out',overwrite);
//End
```

3 ECL statements

```
  }
  abstract int findPartition(Text key);
  abstract void print(PrintStream strm) throws IOException;
  int getLevel() {
    return level;
  }
}

/**
 * An inner trie node that contains 256 children based on the next
 * character.
 */
static class InnerTrieNode extends TrieNode {
  private TrieNode[] child = new TrieNode[256];

  InnerTrieNode(int level) {
    super(level);
  }
  int findPartition(Text key) {
    int level = getLevel();
    if (key.getLength() <= level) {
      return child[0].findPartition(key);
    }
    return child[key.getBytes()[level] & 0xff].findPartition(key);
```

700+ Lines of Java MapReduce Code

Space/Cost

HPCC

1 | 20

Customers | Development | Business

http://hpccsystems.com/why-hpcc/benchmarks

How do I Query HPCC Systems?

ECL (Enterprise Control Language) is a C++ based query language for use with HPCC Systems Big Data platform. ECLs syntax and format is very simple and easy to learn.

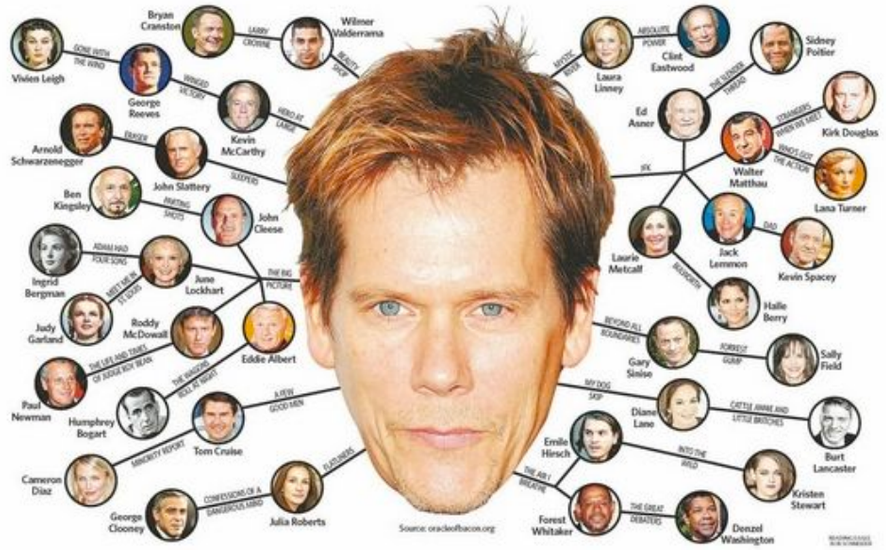
```
/*
    Example code - use without restriction.
*/
Layout_Person := RECORD
  UNSIGNED1 PersonID;
  STRING15  FirstName;
  STRING25  LastName;
END;

allPeople := DATASET([ {1,'Fred','Smith'},
                       {2,'Joe','Blow'},
                       {3,'Jane','Smith'}],Layout_Person);

somePeople := allPeople(LastName = 'Smith');

//  Outputs  ---
somePeople;

```

Note - ECL is very similar to Hadoop's pig ,but more expressive and feature rich.

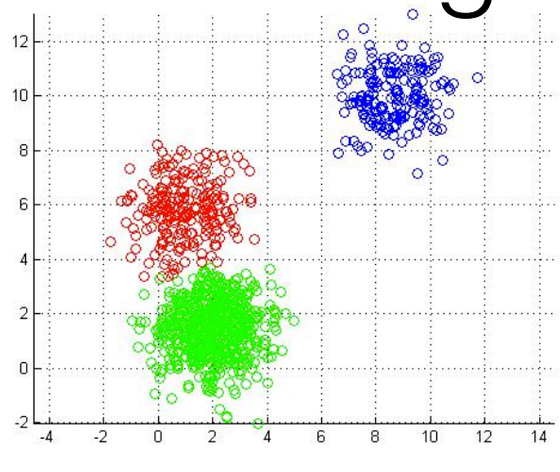
ECL (Enterprise Control Language)

C++ based query language

GraphDB

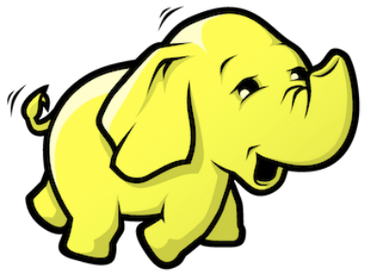


Machine Learning



SQL w/ JOINS

Map/Reduce



Simple to Complex Queries

HPCC Systems
Cluster Architecture

"I can query all or part of your data."

"I'm sub-second fast."

Thor

Hard Disk
Index(optional)

Either/Both

Roxie

Hard Disk
Index(optional)
In-memory Index
SSD

Rapid **O**nline **X**ML **I**nquiry **E**ngine

	HPCC Systems *Roxie*
Date	2004
Query Languages	**ECL**     
In-Memory	Index Only or Index w/ Part or All Data
Data Type	Normalized or DeNormalized or Unstructured
Query Methods	REST Direct TCP SOAP

Example 1

File → **Load Into Thor** → **Query**

VS

File → **Load Into Roxie** → **Index** → **Publish** → **Query**

HPCC Systems Sample Data for Examples 1

hpccsystems.com/download/docs/learning-ecl

Download

Home › Download › Docs

- Getting Started
- Free Community Edition
  - Server Platform
  - ECL IDE
  - Client Tools
  - Graph Control
  - Monitoring
  - View All
- HPCC VM Image
- **Documentation**
  - Installation & Administration
  - **Learning ECL**
  - ECL IDE & Client Tools
  - Machine Learning
  - Tutorials
  - View All
- Source Code

Learning ECL

Community Wiki

The HPCC Systems Community Wiki includes information provided by the community covering best practices, tips, sample code and examples. The HPCC Systems Red Book also contains useful information to help users manage the transition between releases.

HTML

HPCC Data Tutorial: with an Introduction to Thor and Roxie

This tutorial provides a walk-through of the development process from beginning to end and is designed to be an introduction to working with data on HPCC, as well as give an introduction to **Thor** and **Roxie**. The Sample Data Files are required references for the tutorial.

PDF

Sample Data File (one needed)	
In ZIP format	OriginalPerson.zip
In tar.gz format	OriginalPerson.tar.gz
Uncompressed	OriginalPerson

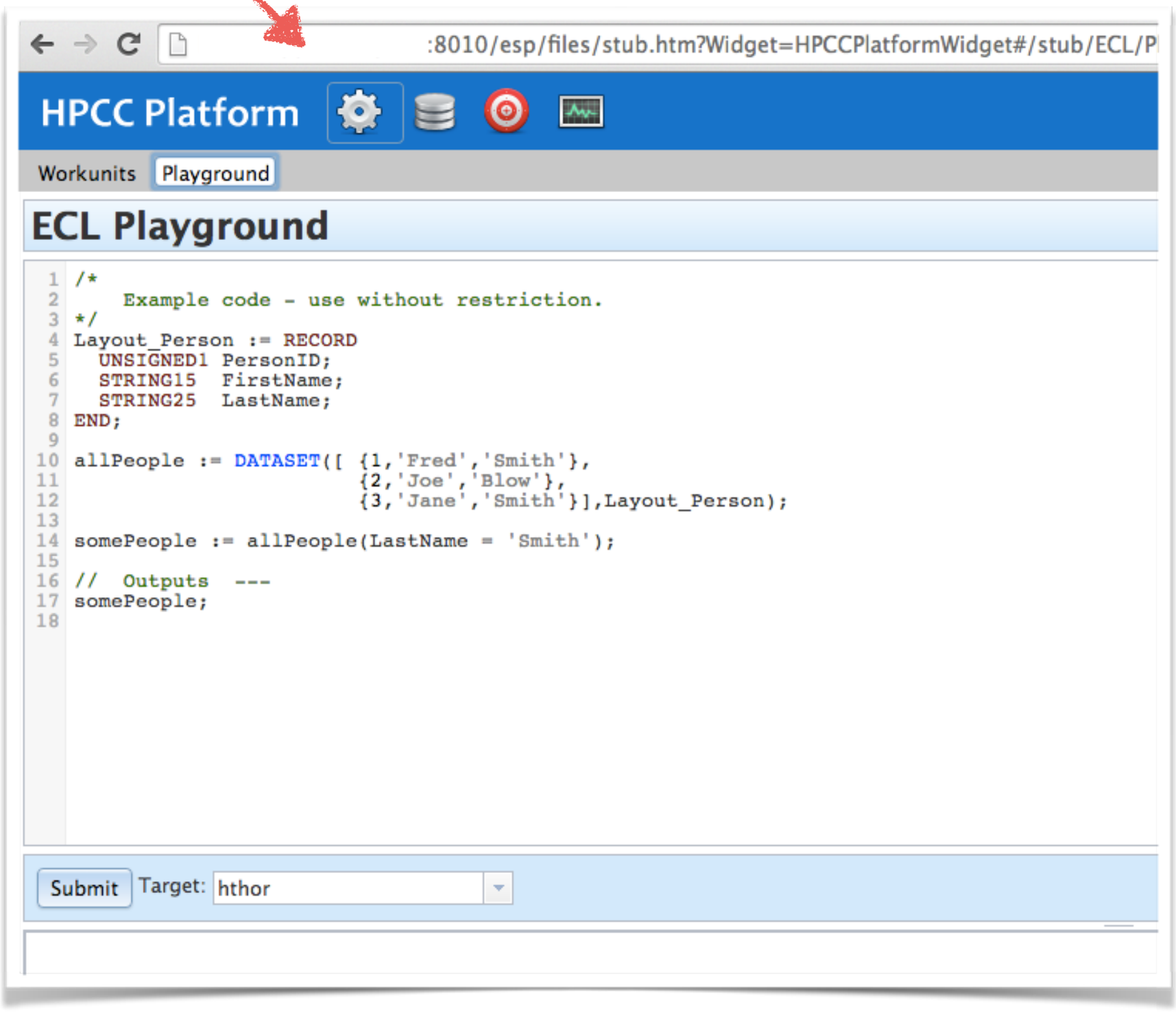
Sample Data

http://hpccsystems.com/download/docs/learning-ecl

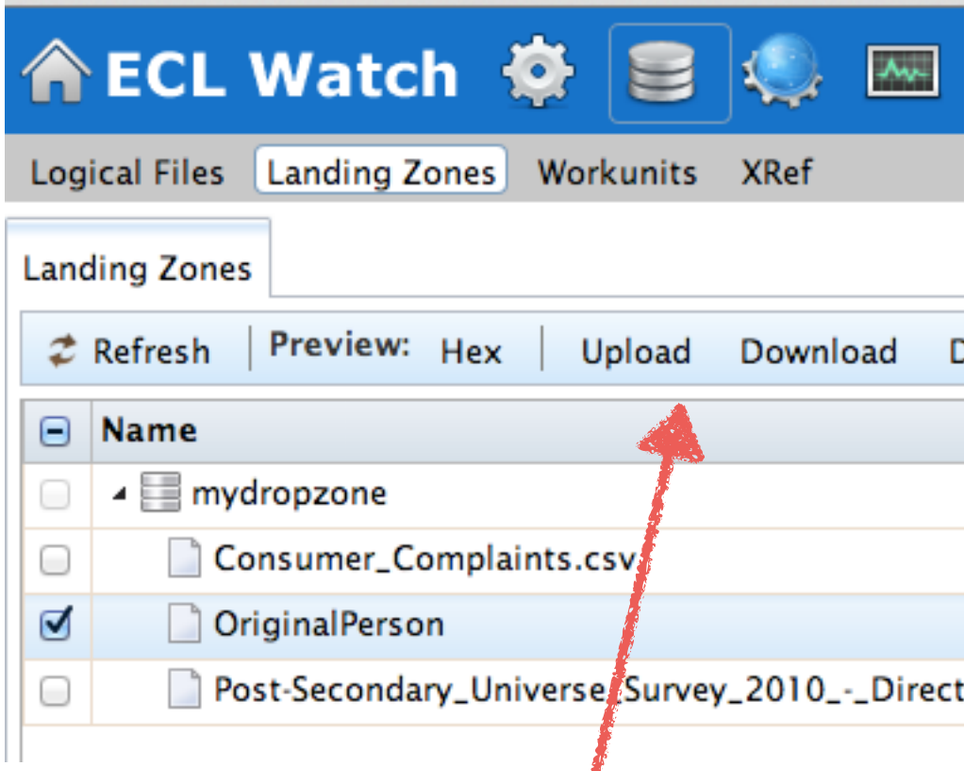
Administrator Web GUI

IP / Url of HPCC install

on
Port 8010

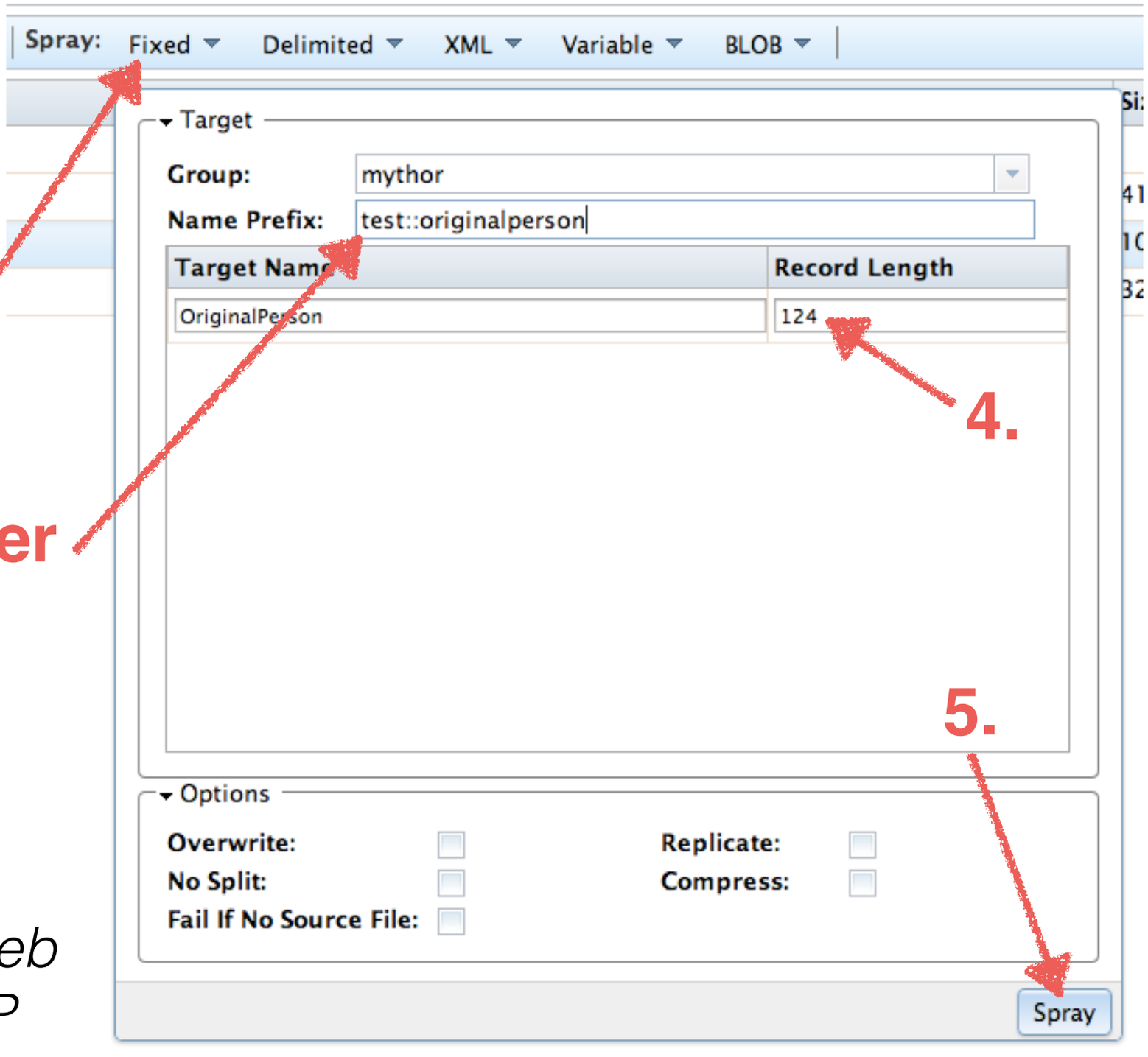


Load Data



1. Upload file*
2. Distribute to cluster
3. Name of file in cluster
4. Size of each row
5. Push to cluster

*2GB file size limit through web
No limit if uploaded via SOAP



Loaded

ECL Watch

Logical Files | Landing Zones | Workunits | XRef

Logical Files

Refresh | Open | Delete | Remote Copy | Copy

	Logical Name
	hthor::test::key_consumer_complaints
	test::consumer_complaints
	test::originalperson
	test::secondary_school
	test::superfile_consumer

In Thor Cluster

Logical Files | test::originalperson

Summary | Contents | ECL | DEF | XML | File Parts | Queries | Workunit

Download: Zip GZip XLS | Filter

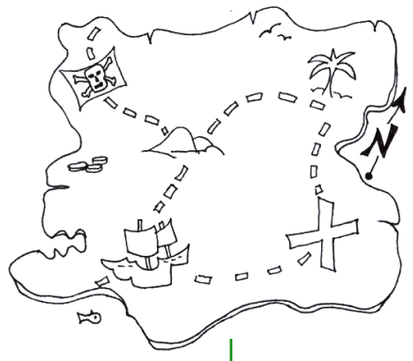
##	contents
1	Cherianne Khatchatourian N 5453069 BOULDER RIDGE RD # 25A HAWKINS WI
2	Muyesser Raplee X 2074755 SWAMP RD DISTRICT HEIGHT MD
3	Roselin Viceconte 97828107 HILL TER ENTERPRISE OR
4	Inda Provines 72941290 W MOUNT PLEASANT AVE LAVACA AR
5	Inderdeep Laurence D 3233044 PROSPECT PL GREENSBORO FL
6	Chrystine Mangiapane 800071806 1ST AVE APT 8F ARVADA CO
7	Adelene Stock R 199011117 FARM RD DOVER DE
8	Mendy Rufenblanchette 296973 W 83RD ST APT 4C WILLIAMSTON SC
9	Lannie Amerantes I 25312200 W 20TH ST APT 909 CHARLESTON WV
10	Tare Gonyeau T 799246 CANDLE CT EL PASO TX
11	Finney Aristilde P 31220222 1ST AVE APT 2B MACON GA
12	Oreoluwa Marthaler 04210176 CLAREMONT GDNS AUBURN ME
13	Surge Abbottkrepp D 4408722 LE PARC CT TWINSBURG OH

Query

Example 1

```
Layout_People := RECORD
  STRING15 FirstName;
  STRING25 LastName;
  STRING15 MiddleName;
  STRING5 Zip;
  STRING42 Street;
  STRING20 City;
  STRING2 State;
END;
```

Schema



Data

"USE DATABASE;"

File Location, "FROM Table"

```
allPeople := DATASET('~test::originalperson',Layout_Person,THOR);
```

WHERE `LastName` = 'Smith'

Query

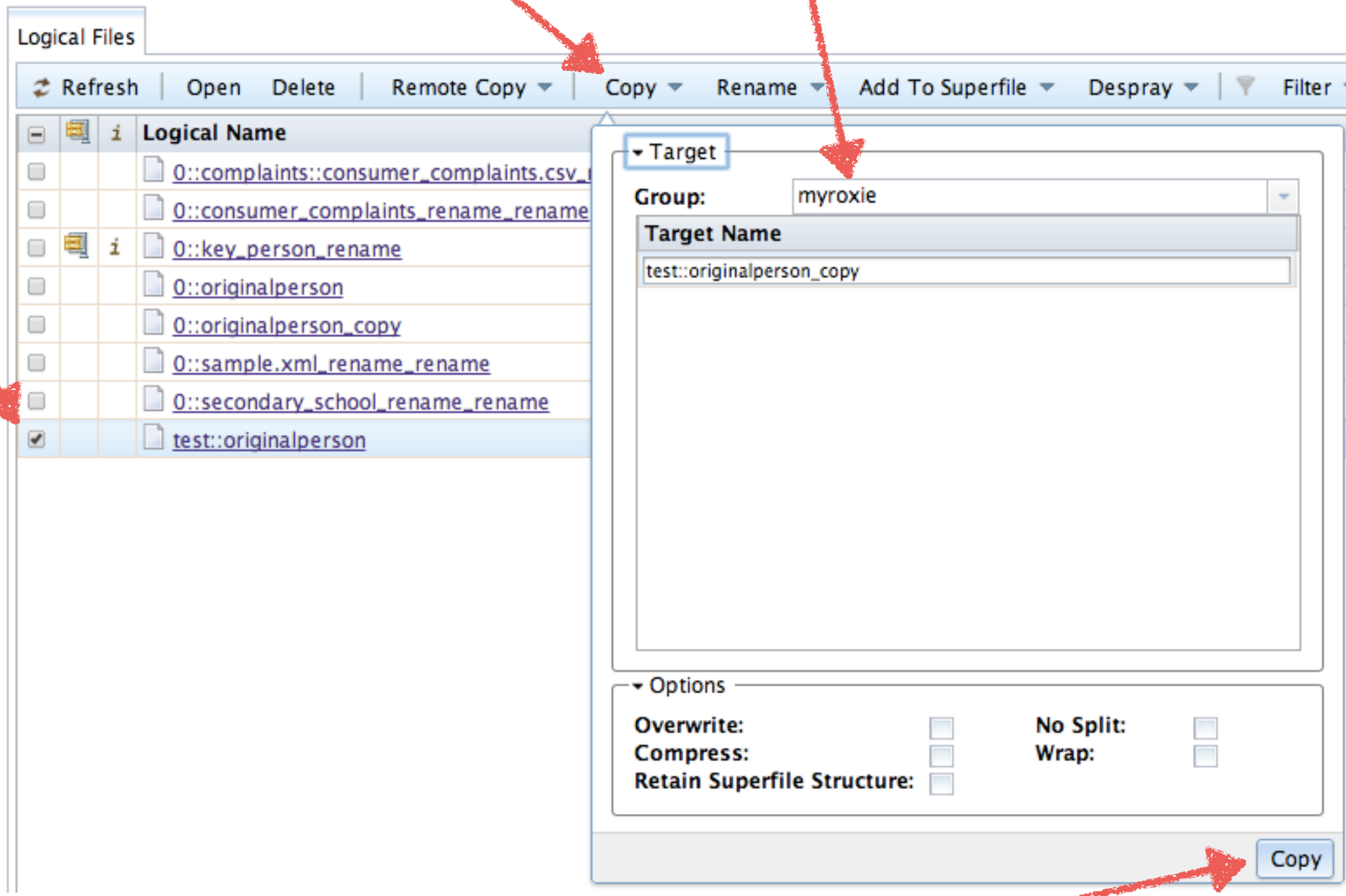
```
smith := allPeople(LastName= 'Smith');
```

*"SELECT * ...."*

```
smith;  //Output
```

Copy Data
From Thor to Roxie

1.Select Thor File(s) 2.Copy Tab 3. Pick Roxie cluster



4. Click Copy

Indexing

In-Memory

Ex. Creating an index by ***"LastName"***

```
allPeople := DATASET('~test::originalperson_copy',
    {Layout_People, UNSIGNED8 RecPtr {virtual(fileposition)}},
                                                    FLAT);
```

File Position Number
pseudo recordID
"Alter Table"(new column)

Index Filename

```
rx := INDEX(allPeople,{LastName,RecPtr},'~test::key_person_copy',PRELOAD);
```

In-Memory

Make Index

```
BUILDINDEX(rx);
```

Indexing

In-Memory with Luggage

Index Only

rx := INDEX(allPeople,{LastName,RecPtr},'~test::key_person_copy',PRELOAD);

Index + Part or All Data

rx := INDEX(allPeople,**{FirstName,MiddleName},**{LastName,RecPtr},'~test::key_person_copy',PRELOAD);

```
Layout_People := RECORD
  STRING15 FirstName;
  STRING25 LastName;
  STRING15 MiddleName;
  STRING5 Zip;
  STRING42 Street;
  STRING20 City;
  STRING2 State;
END;
```

Store Data In-Memory with Index

+

Query w/ Index

Data

```
allPeople := DATASET('~test::originalperson_copy', {Layout_People, UNSIGNED8 RecPtr {virtual(fileposition)}},FLAT);

datax:= INDEX(allPeople,{LastName,RecPtr},'~test::key_person_copy');
```

WHERE `LastName` = 'Smith' **from Index**

Query

```
filterdata:= FETCH(allPeople,datax(LastName='Smith'),RIGHT. RecPtr);

filterdata; //Output
```

Publish Your Code

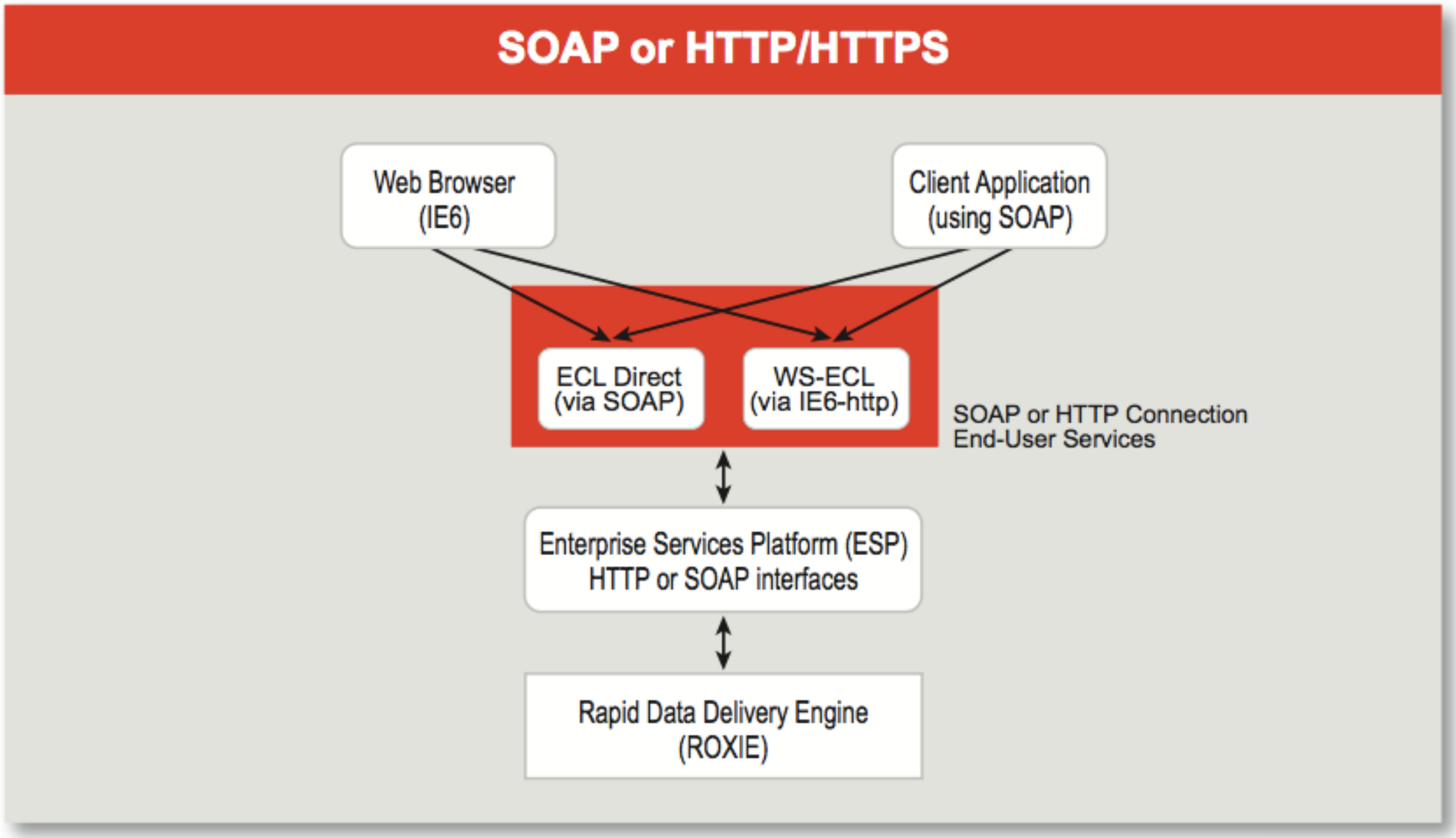
What Is publishing your code?

ECL is built from C++. So your ECL needs to be compiled before it runs.When you publish a query, you pre-compile your ECL and send it to the ESP server where it will be stored. ESP, on port 8002 , will listen to any requests and execute the published query.

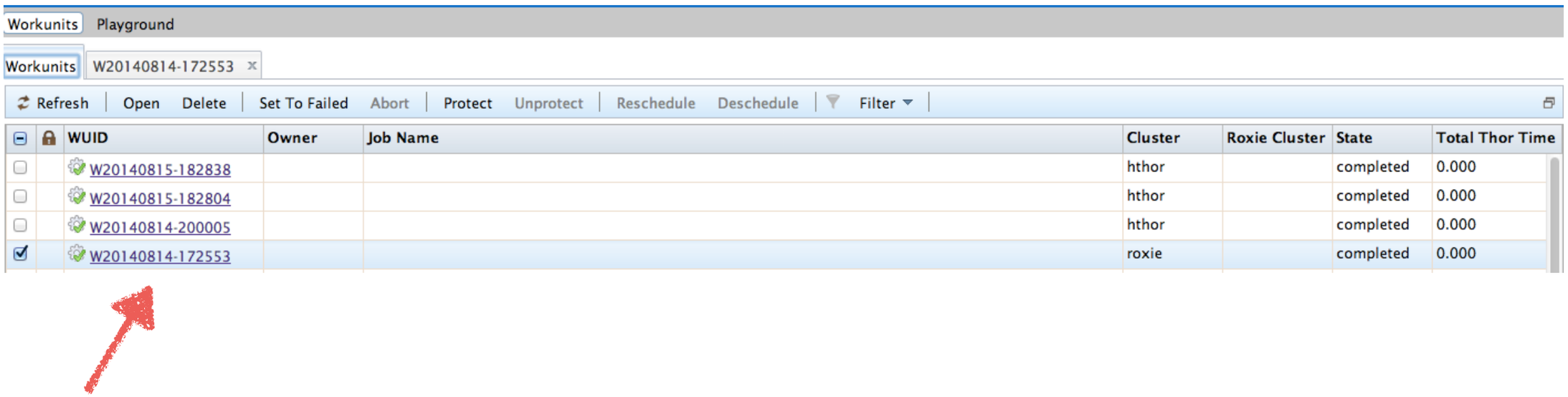
Can I do ad-hoc queries without publishing?

Yes, but it will not be sub-second fast as it is not pre-compiled.

How Querying ESP Works

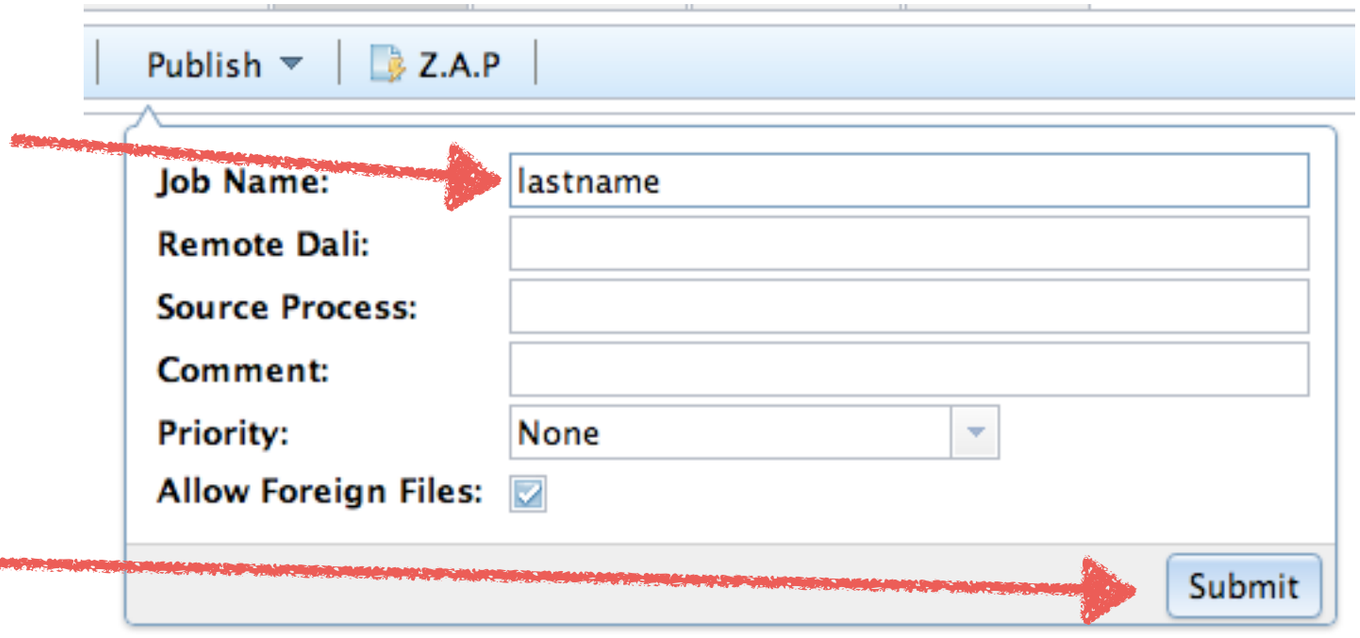


How to Publish Your ECL



1.Select your ad-hoc “lastname” query from before

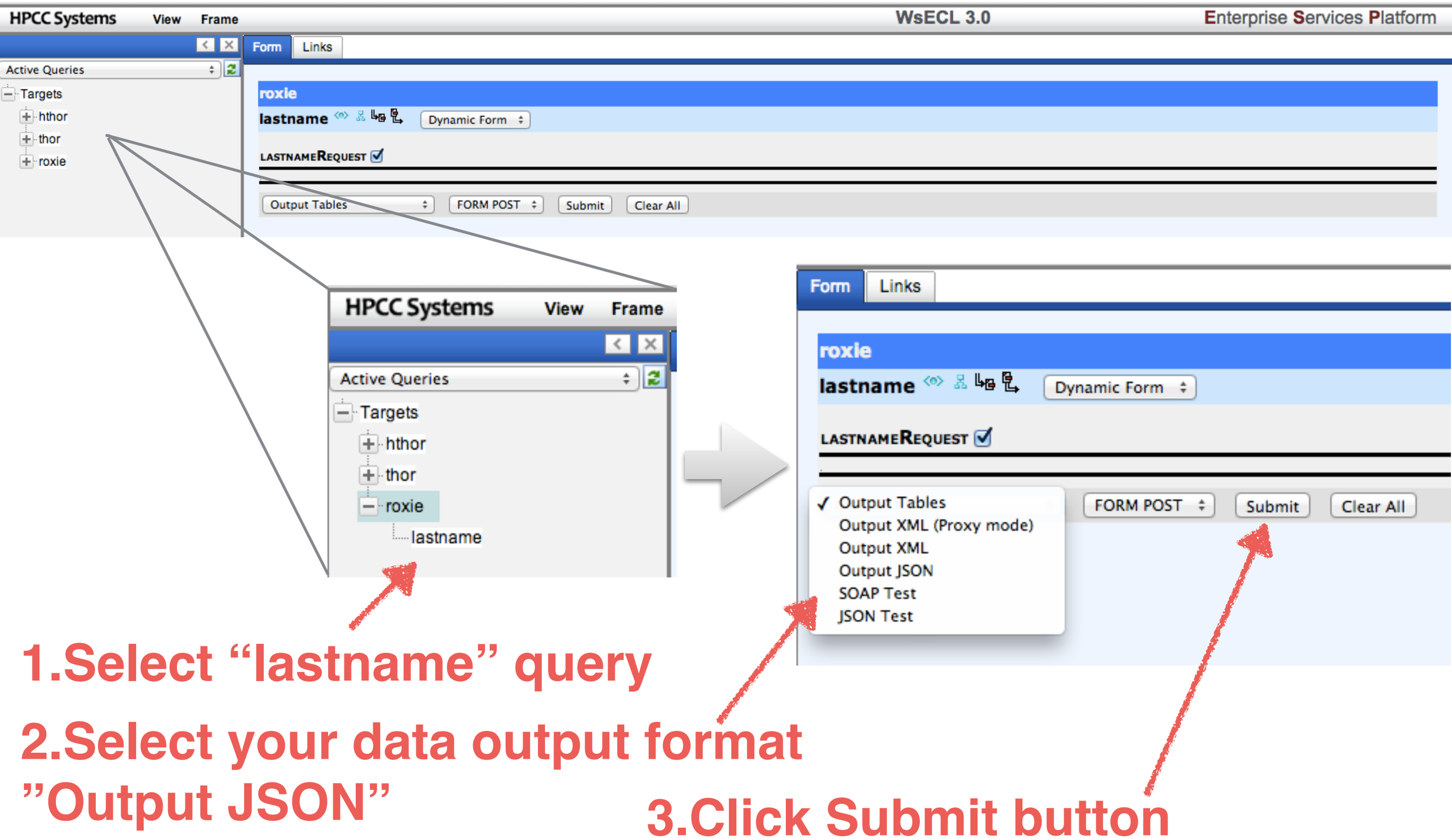
2.Name you query “lastname”



3. Publish

ESP on port 8002

Enterprise Service Platform



1.Select “lastname” query

2.Select your data output format ”Output JSON”

3.Click Submit button

JSON Format

JSON Test | **roxie / lastname**

Destination: /WsEcl/json/query/roxie/lastname?ver_=0

Request:

Headers:

```
Content-Type: application/json; charset=UTF-8
```

Response:

Headers:

```
{
 "lastname": {

 }
}
```

1.Send Request = Query or hit this Url

Send Request | Check well-formness before send

JSON Format

JSON Test | **roxie / lastname**

Destination: /WsEcl/json/query/roxie/lastname?ver_=0

Results - in less then a second

Request:

Headers:

```
Content-Type: application/json; charset=UTF-8
```

```
{
 "lastname": {

 }
}
```

Response:

Headers:

```
Content-Type: application/json
```


```
{
 "lastnameResponse":  {
  "sequence": 0,
   "Results":  {
    "result_1": {
     "Row": [
      {
       "firstname": "Ganija",
        "lastname": "Smith",
        "middlename": "Z",
        "zip": "07444",
        "street": "2090 POTTS HILL RD",
        "city": "POMPTON PLAINS      NJNaftali       S",
        "state": "te",
        "recptr": 5565740
      },
       {
       "firstname": "Montakarn",
        "lastname": "Smith",
        "middlename": "Q",
        "zip": "16635",
        "street": "45 MALLARD RD",
        "city": "DUNCANSVILLE       PAAmgelo        W",
        "state": "ha",
        "recptr": 34427484
```


Send Request ☑ Check well-formness before send

SuperFile

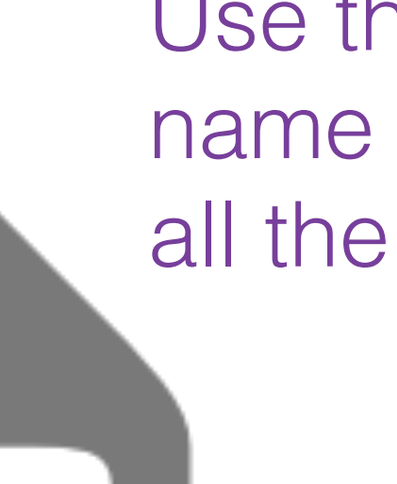
Organizing Your Files

Use this file name to get all the real files

2013-06
Twitter

2013-06-06
…….. -07
…….. -08

Logical File



Real File

2013-06-08
Twitter

2013-06-07
Twitter

2013-06-06
Twitter

✚ Append new real files on the fly

SuperKeys

Organizing Your Indexes

No Sub-Super Files or Keys in Roxie

2013
Twitter

2013-06-08
Twitter

2013-06-07
Twitter

2013-06-06
Twitter

HPCC Systems

Load, Index & Query Big Data the EZ way

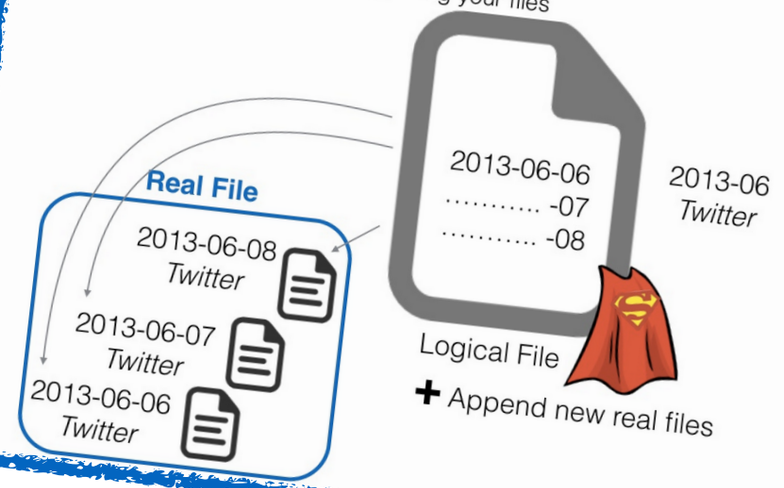
LexisNexis

By Fujio Turner

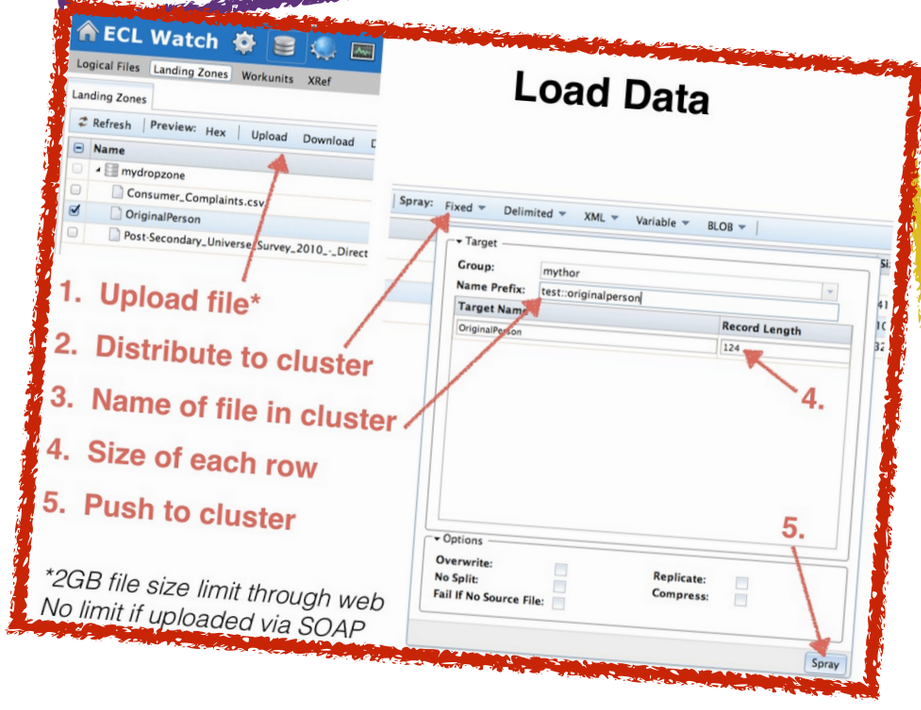
@myhousehippo

SuperFile

organizing your files



Load Data



Why Index ?

~~Full Table or Data Scan~~

MySQL + + and from date to date

slideshare

For More HPCC "How To's" Go to SlideShare

http://www.slideshare.net/FujioTurner/

Download HPCC Systems Open Source Community Edition

CentOS *or* ubuntu

http://hpccsystems.com/download/

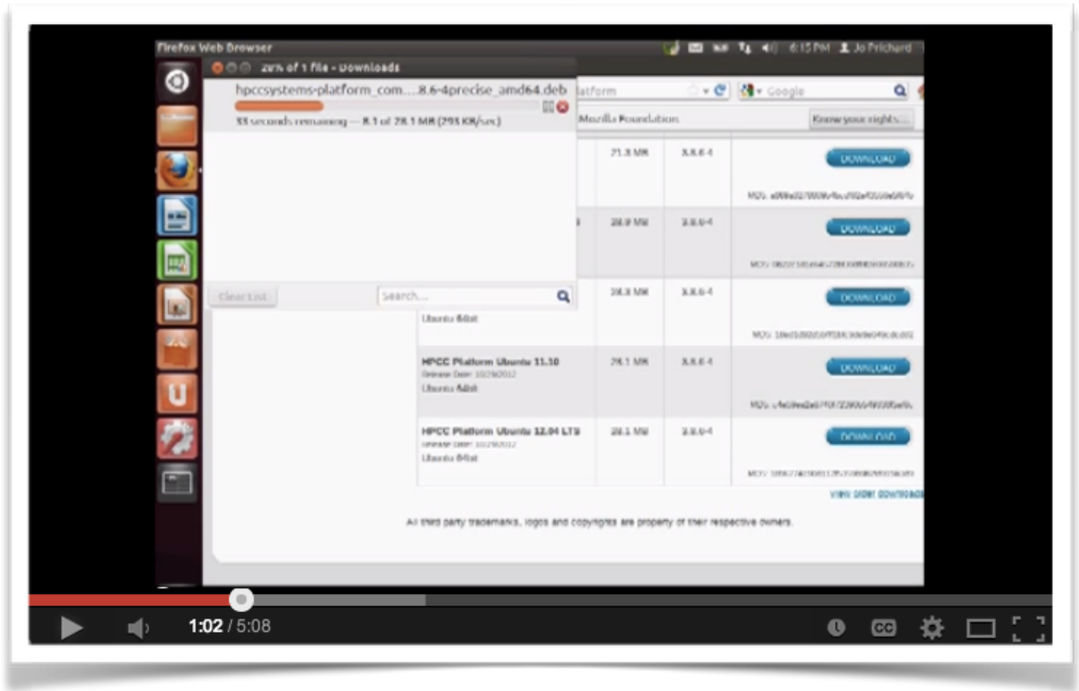
Source Code

https://github.com/hpcc-systems

GitHub

Watch how to install HPCC Systems in *5* Minutes

YouTube



http://www.youtube.com/watch?v=8SV43DCUqJg