



Technology Overview



MARCH 2025

Bob Foreman
Principal Software Engineer
LexisNexis Risk Solutions



Welcome to the 2025 KSU Data for Social Good Hackathon!

- ✓ Our challenge uses the HPCC Systems platform and ECL (Enterprise Control Language). This overview gives a detailed look at both!
- ✓ HPCC defined is a *distributed data parallel processing* platform.

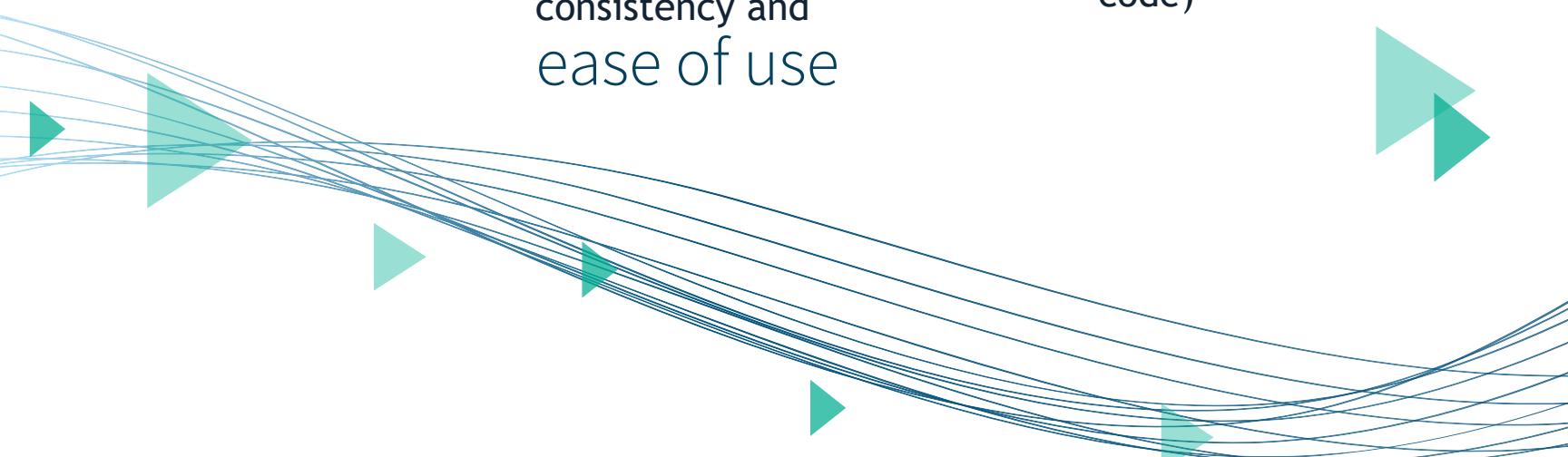
High Performance Computing Cluster

- ✓ Contains a THOR cluster where the majority of your coding will be done, and a ROXIE cluster to deliver your results.
- ✓ A proven platform for LexisNexis for over 20 years, and open source since 2011!

HPCC Systems: End to End Data Lake Management



Completely free
open source data
lake solution



Out of the box
capabilities for
consistency and
ease of use



Less coding
and more using (even
though we love to
code)



We are your
one stop
shop for all
your data
integration,
querying and
analytical
needs

HPCC Systems Evolution

2001



Original version of HPCC Systems released

2011



Open source Apache license and code release to GitHub
Exceeded market-leading performance benchmark achieved

2012 – 16



Continuous **QUALITY-FOCUSED** improvements
Better support and training with improved integration – faster and easier to use

2017-2025

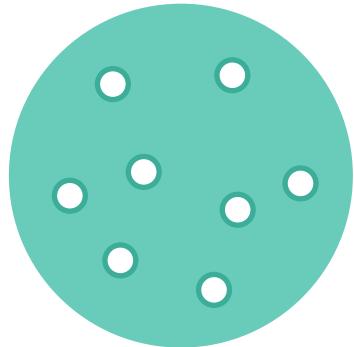


Improved processing architecture
IoT enabled
ML Expansion!
Cloud Native!

The Data Centric Approach

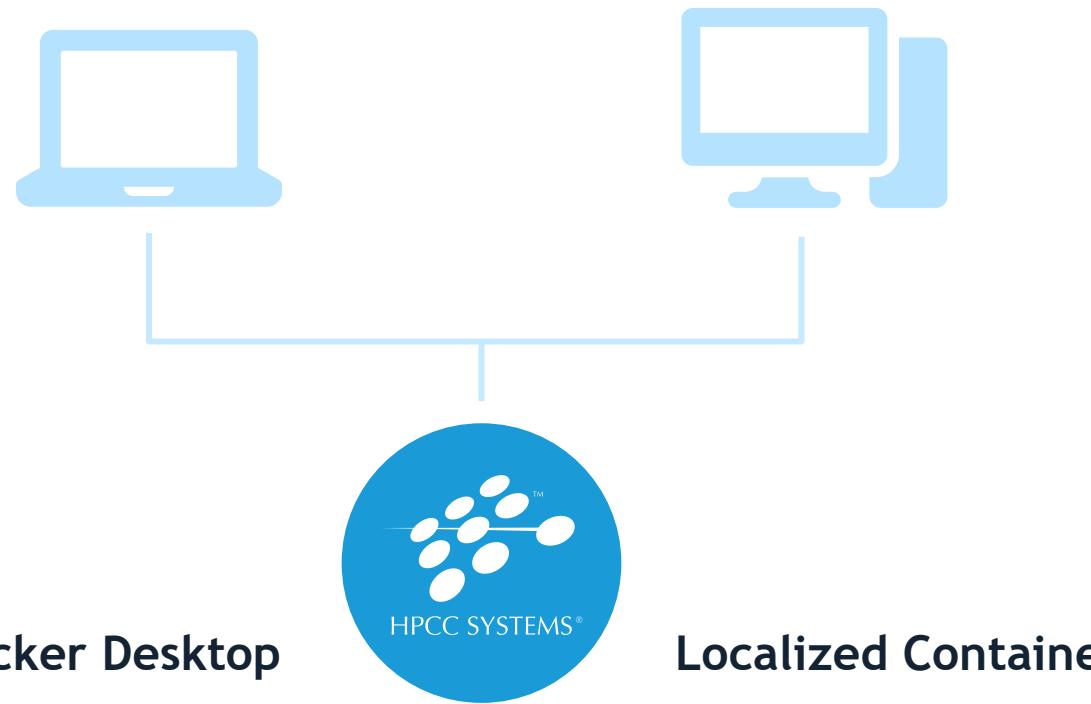
A single source of data is insufficient to overcome inaccuracies

Our platform is built on the premise of absorbing data from many data sources and transforming them to **actionable smart data**

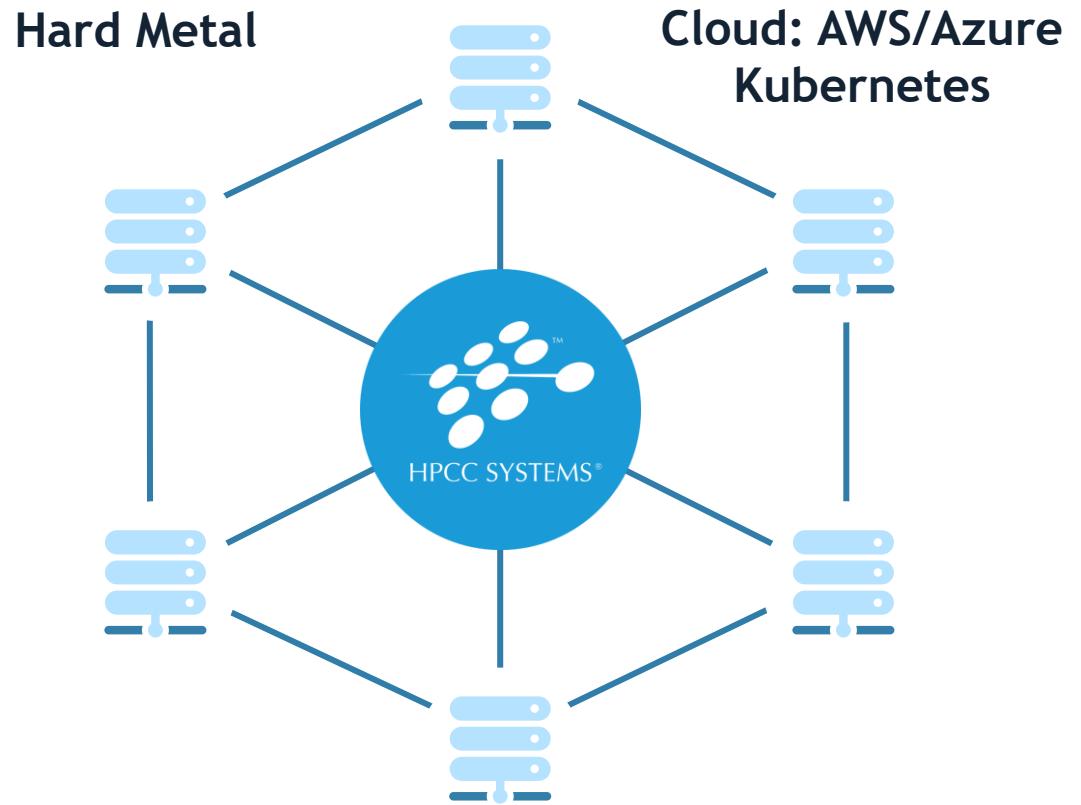


Scale from Small to Big

The stack can run on a single laptop or desktop.



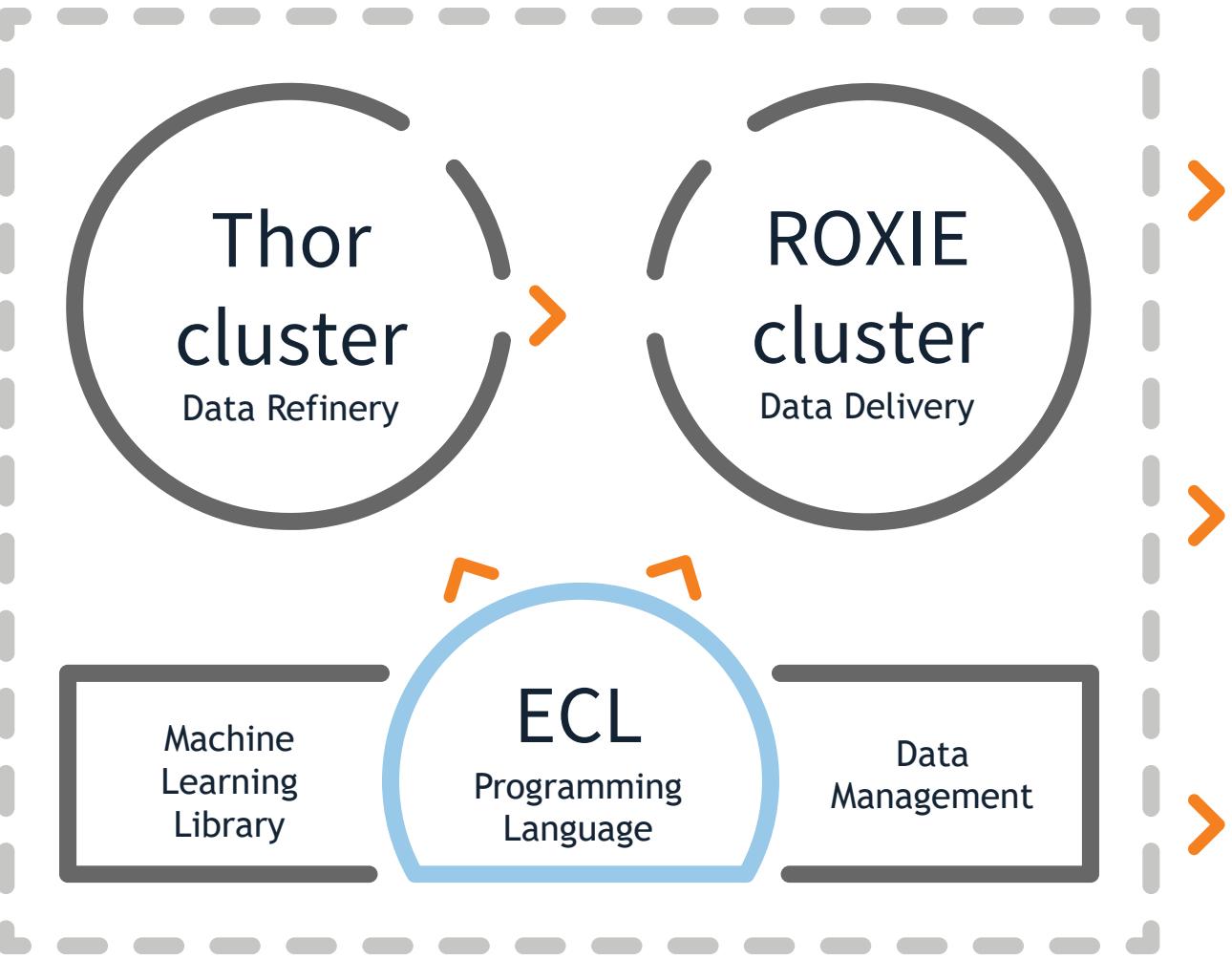
In more sophisticated cases, HPCC Systems run *clusters*, hundreds of servers working as a single processing entity, to transform and deliver big data.



The HPCC Systems Components



Developer
ECL IDE
VS-Code
Cloud IDE
CLI



- Query
- Visualize
- Connect

Technology – The Open Source Stack



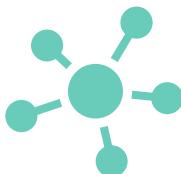
Thor: Data Refinery Cluster

Extraction, loading, cleansing, transforming, linking and indexing



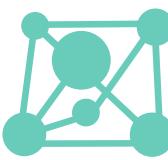
ROXIE: Data Delivery Engine

Rapid data delivery cluster with high-performance online query delivery for big data



Data Management Tools

Data profiling, cleansing, snapshot data updates, consolidation, job scheduling and automation



Machine Learning Library

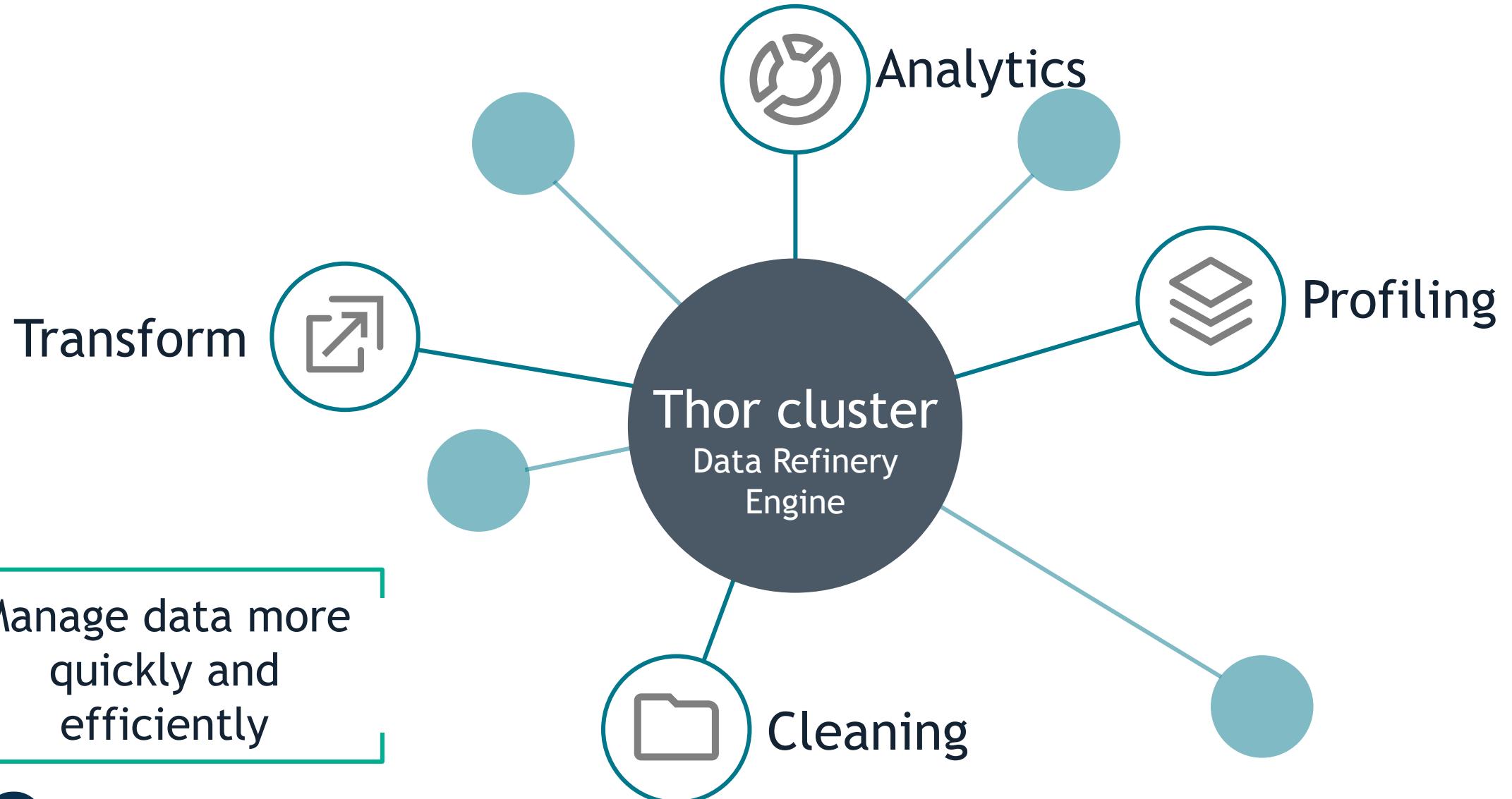
Linear regression, logistic regression, decision trees and random forests



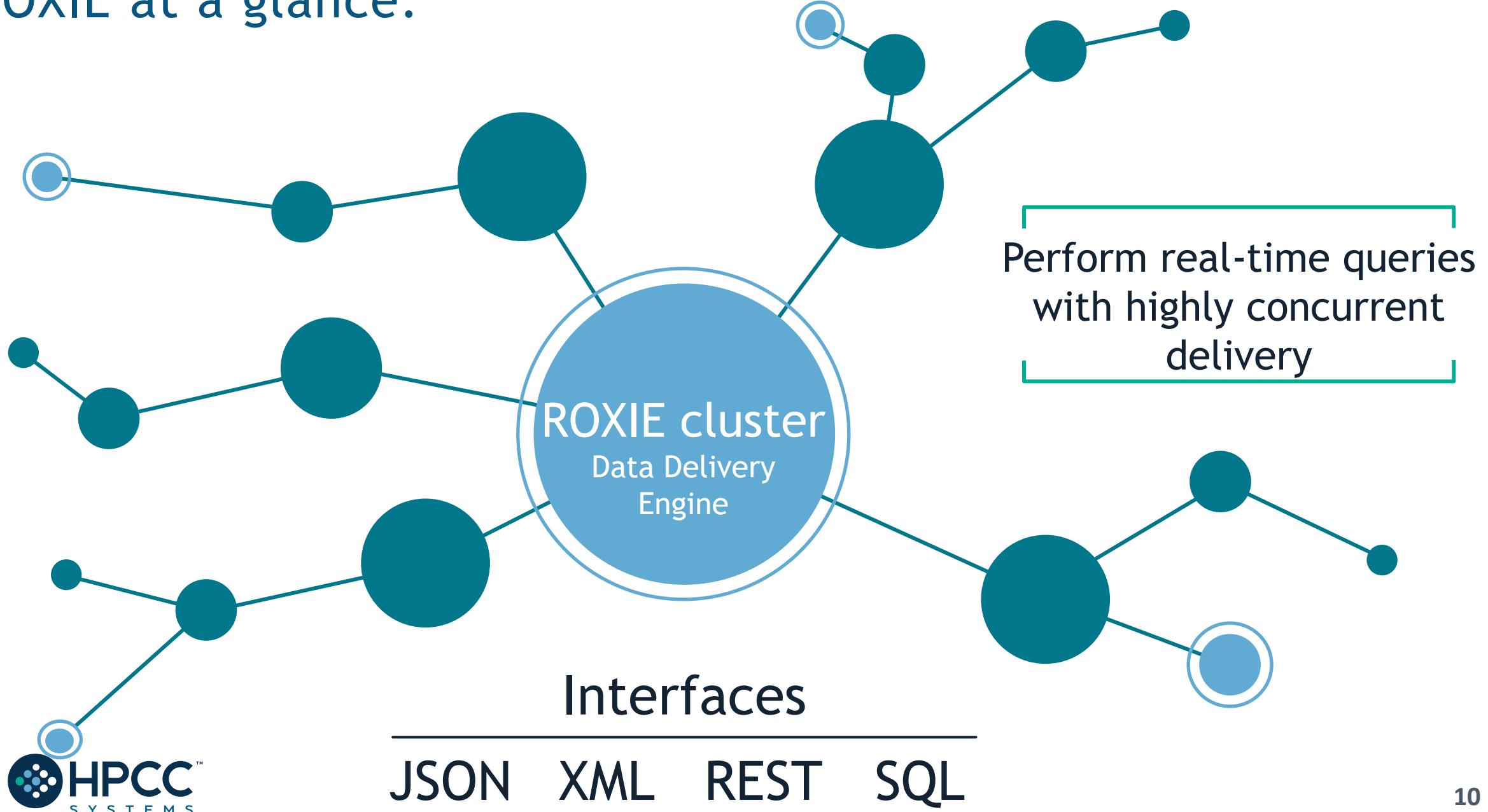
Connectivity & Third-Party Tools

New plugins to help integrate third party tools with the HPCC Systems platform

THOR at a glance:



ROXIE at a glance:



An Introduction to ECL

ECL
Enterprise Control
Language



```
IMPORT $, STD, ML;
EXPORT Func(UNSIGNED C, UNSIGNED2 Dist, UNSIGNED size, STRING Fld, REAL Parm1=0, REAL Parm2=0, REAL Parm3=0) := MODULE
  SHARED Node := STD.system.Thorlib.Node() + 1;
  SHARED PersistPrefix := $.Parms.PersistPrefix;
  SHARED TotalRecs := $.Parms.RecCnt * CLUSTERSIZE;
  SHARED UIDval := IF(C=1, node, node + ((C-1)*CLUSTERSIZE));
  SHARED BOOLEAN IsRandFile := $.Parms.Randomness = $.ut.RandomSrc.file;
  SHARED Normal := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Normal(Parm1, Parm2),
      ML.Distribution.Normal(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'NormalDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED Normal2 := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Normal2(Parm1, Parm2),
      ML.Distribution.Normal2(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'Normal2DistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED Uniform := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Uniform(Parm1, Parm2),
      ML.Distribution.Uniform(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'UniformDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED StudentT := FUNCTION
    Thisdist := ML.Distribution.StudentT(Parm1, Parm2);
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'StudentTDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
```

- Transparent and implicitly parallel programming language
- Both powerful and flexible



How to do
it



vs.



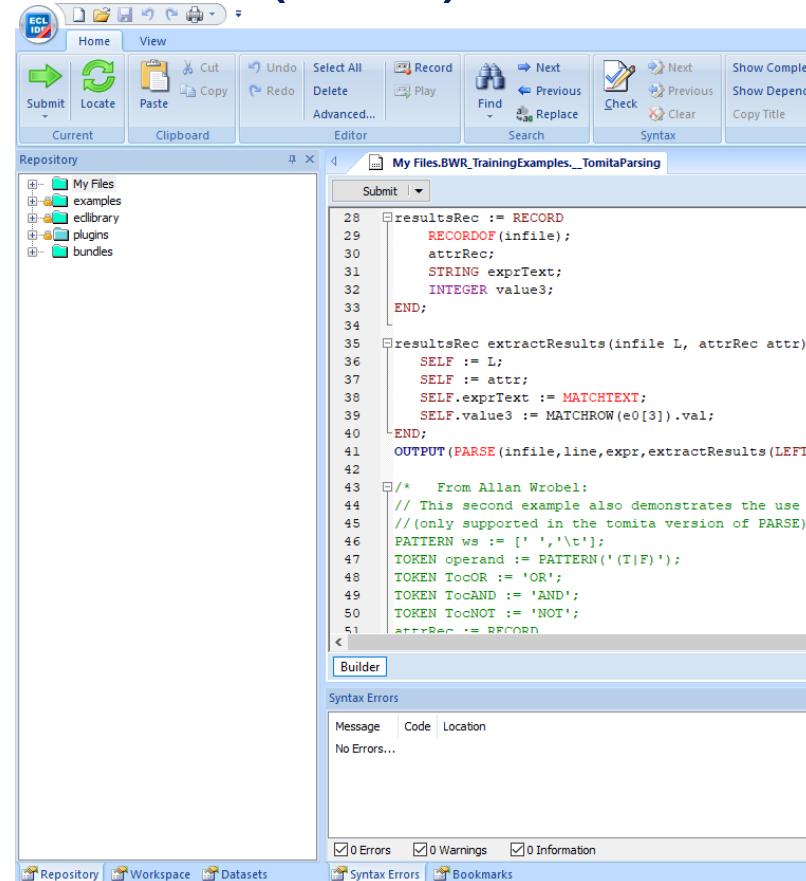
What to
do

- Optimized for data-intensive operations, declarative, non-procedural and dataflow oriented
- Uses intuitive syntax which is modular, reusable, extensible and highly productive

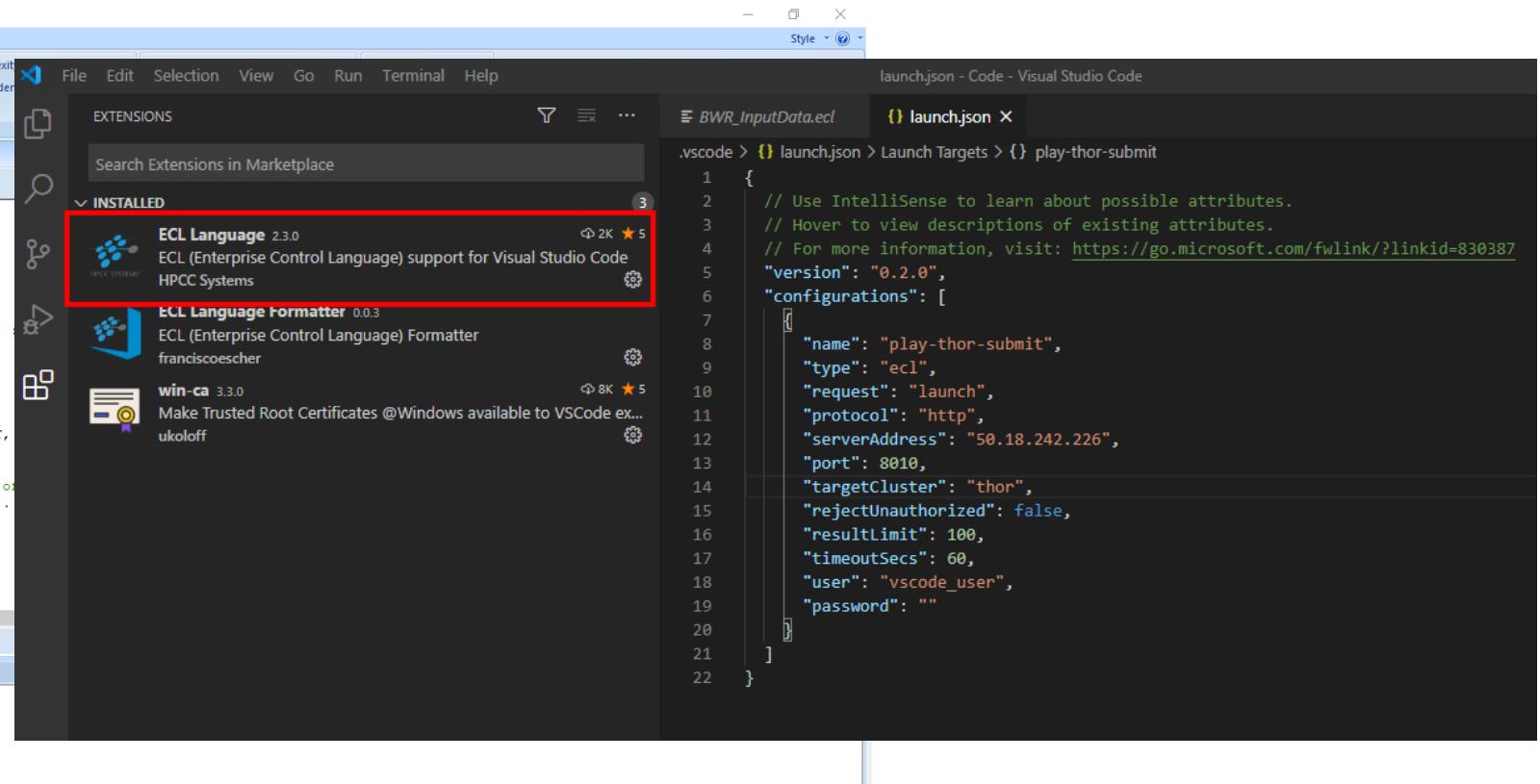


Integrated Development Environments

ECL IDE (Win)



Visual Studio Code (Ux/MacOS)

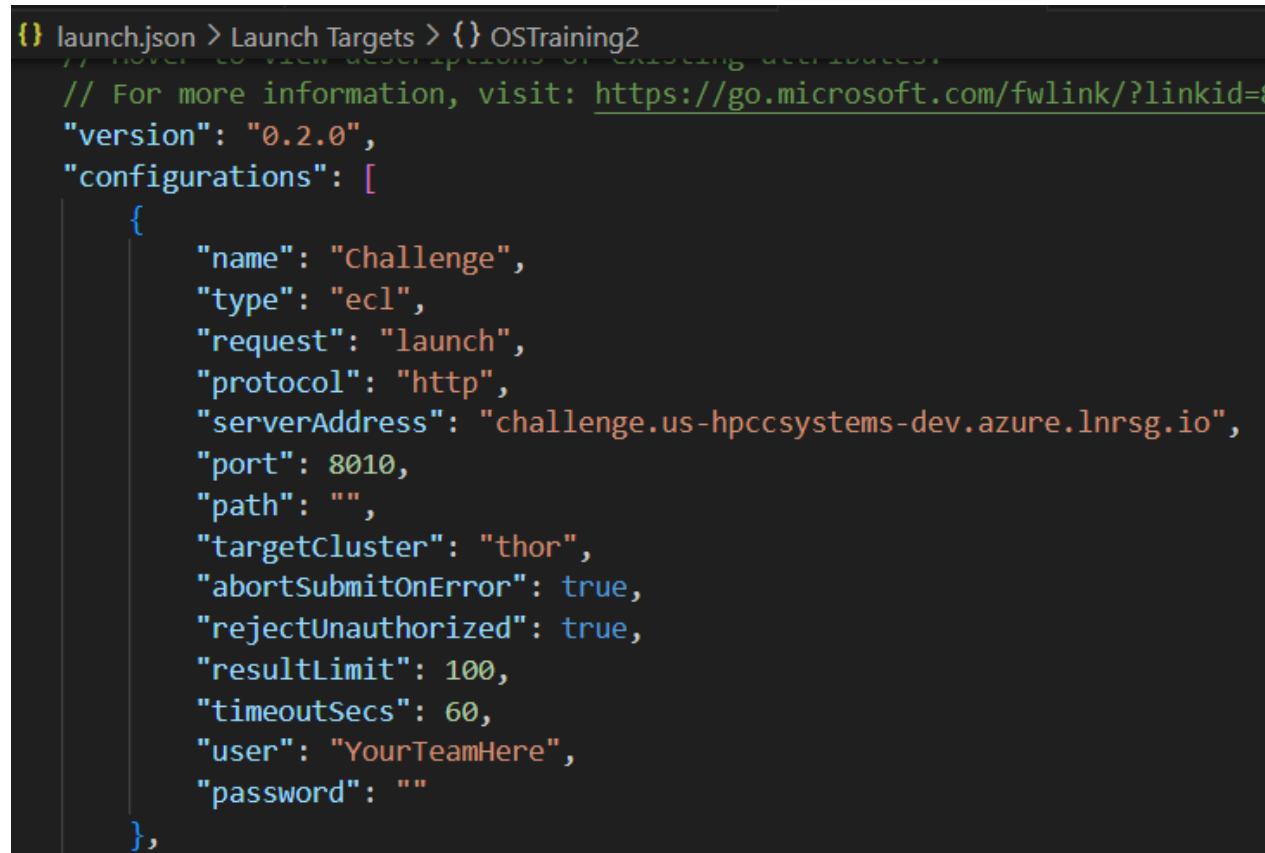


And CLI too! ECL.EXE

The Playing Field!

HPCC Cluster ECL Watch:

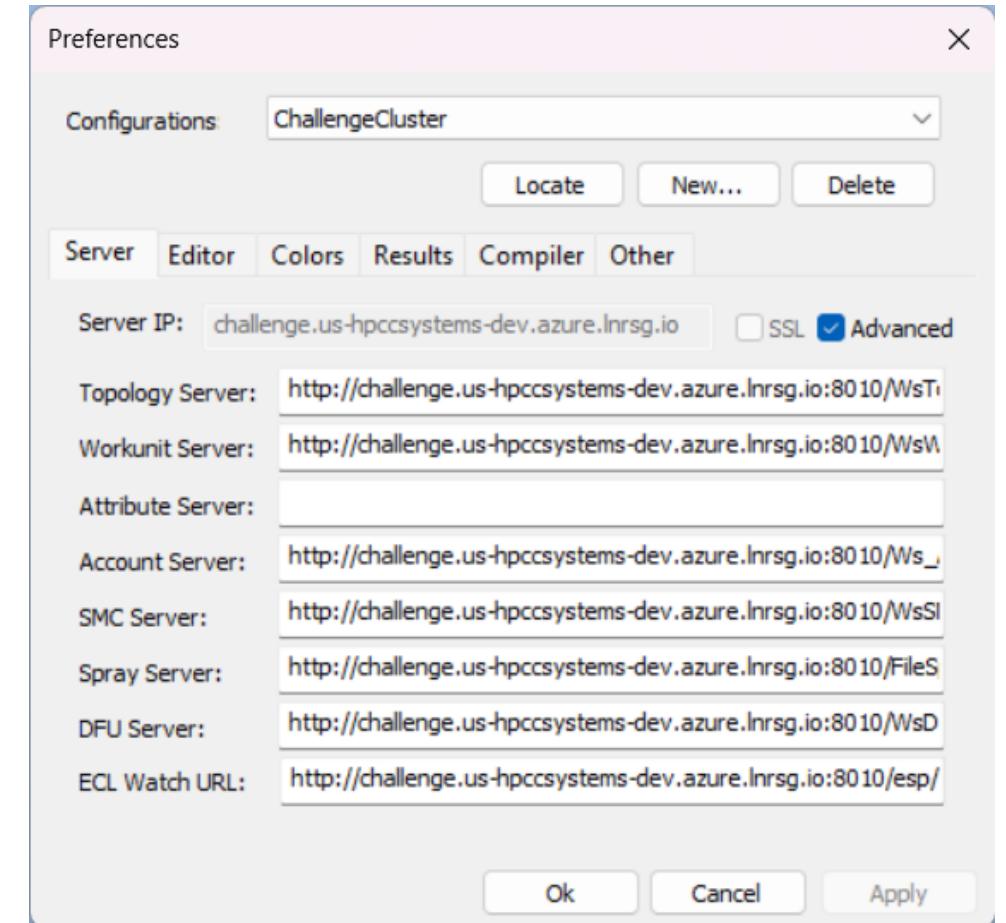
<http://challenge.us-hpccsystems-dev.azure.lnrsg.io:8010/>



```
{} launch.json > Launch Targets > {} OTraining2
// For more information, visit: https://go.microsoft.com/fwlink/?LinkId=8
"version": "0.2.0",
"configurations": [
    {
        "name": "Challenge",
        "type": "ecl",
        "request": "launch",
        "protocol": "http",
        "serverAddress": "challenge.us-hpccsystems-dev.azure.lnrsg.io",
        "port": 8010,
        "path": "",
        "targetCluster": "thor",
        "abortSubmitOnError": true,
        "rejectUnauthorized": true,
        "resultLimit": 100,
        "timeoutSecs": 60,
        "user": "YourTeamHere",
        "password": ""
    },
}
```



{} launch.json X



The Repo!

<https://github.com/hpccsystems-solutions-lab/HelpMissingKids25>

The screenshot shows the GitHub repository page for 'HelpMissingKids25'. The repository is public and has 1 branch and 0 tags. The main commit is by 'bobf2000' titled 'Update README.md' made 2 minutes ago. Below the commit list, there's a table of files with their descriptions and last update times. At the bottom, there are links to 'README' and 'Apache-2.0 license'. In the top right, there are buttons for 'Edit Pins', 'Watch', 'Code' (which is highlighted with a red circle), and 'Clone'. The 'Clone' section includes options for HTTPS, SSH, GitHub CLI, and a URL field. An arrow points from the 'Code' button to the 'Download ZIP' button in the Clone section, which is also highlighted with a red circle.

File	Description	Last Updated
.vscode	Code Update	last year
Code	Create BWR_CleanChurches.ecl	last year
Docs	Create quickstart.mod	last year
Images	Update repo folders and ReadMe	last year
LICENSE	Initial commit	2 years ago
README.md	Update README.md	2 minutes ago
edlcc.log	Update Docs	last year
quickstart.mod	Create quickstart.mod	last year

IDE Features:



A full-featured GUI for ECL development providing access to the ECL repository and many of the ECL Watch capabilities.

Uses various ESP services via SOAP.

Provides the easiest way to create:

1. Queries into your data, instant results!
2. ECL Definitions to build your queries which:
 - Are created by coding an expression that defines how some calculation or record set derivation is to be done.
 - Once defined, can be used in succeeding ECL definitions.

The ECL Watch

ECL Watch v9

Wuid, User, (ecl:*, file:*, dfu:*, query:*)...

Tech Preview On

Workunits Playground

Refresh Open Delete Set To Failed Abort Protect Unprotect Filter Mine

WUID	Owner	Job Name	Cluster	Roxie Cluster	State	Total Cluster Time	Execution Cost	File Access Cost
W20220907-165435	siuray01	XTAB_Persons_Gender	thor		completed	5.296	0.00 (USD)	0.00 (USD)
W20220901-153139	Jimi	testme-263372	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-075140	vscode_user	Step01b	thor		compiled	0.000	0.00 (USD)	0.00 (USD)
W20220906-074606	vscode_user	Step01-215848	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-074327	vscode_user	Step01	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-075102	vscode_user	Step01	thor		completed	1:06.111	0.00 (USD)	0.01 (USD)
W20220906-074437	vscode_user	Step01	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-074723	vscode_user	Step01	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-074449	vscode_user	Step01	hthor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220912-070601	aaa	PREDICT1	thor		completed	44.480	0.00 (USD)	0.00 (USD)
W20220912-064736	aaa	PREDICT1	thor		completed	21.994	0.00 (USD)	0.00 (USD)
W20220912-132624	aaa	PREDICT1	thor		completed	39.644	0.00 (USD)	0.00 (USD)
W20220906-170731	siuray01	Persons	thor		completed	1.103	0.00 (USD)	0.00 (USD)
W20220902-181153	siuray01	Persons	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220902-175740	siuray01	Persons	thor		completed	0.000	0.00 (USD)	0.00 (USD)
W20220902-181411	siuray01	Persons	thor		failed	0.000	0.00 (USD)	0.00 (USD)

1 - 25 of 632 Rows

1 2 3 4 5 25

ECL Watch Features:

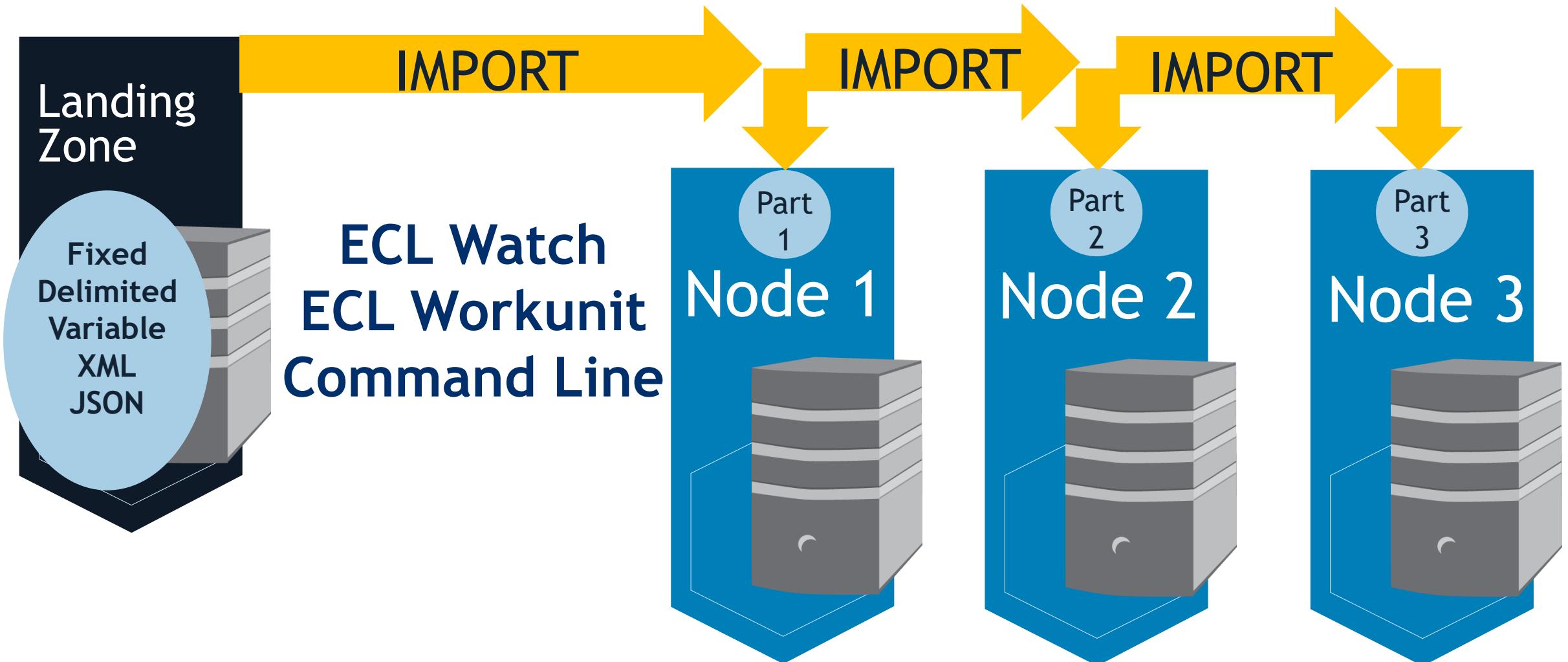


A web-based query execution, monitoring and file management interface. It can be accessed via ECL IDE or a web browser.

ECL Watch allows you to:

1. See information about active workunits.
2. Monitor cluster activity.
3. Browse through previously submitted Workunits.
4. See a visual representation of the data flow within the WU, complete with statistics which are updated as the job progresses.
5. Search through files and see information including:
 - Record counts and layouts.
 - Sample records.
 - The status of all system servers whether they are in clusters or not.
6. View log files.
7. Start and stop processes.

Getting the data to the cluster!



Treated as a *single logical file*...



ECL Overview



MARCH 2025

Bob Foreman
Principal Software Engineer
LexisNexis Risk Solutions

ECL (Enterprise Control Language)

ECL is a language design to query/manipulate massive data and is used for ETL (Extract, Transform, Load) and data visualization.

Extract

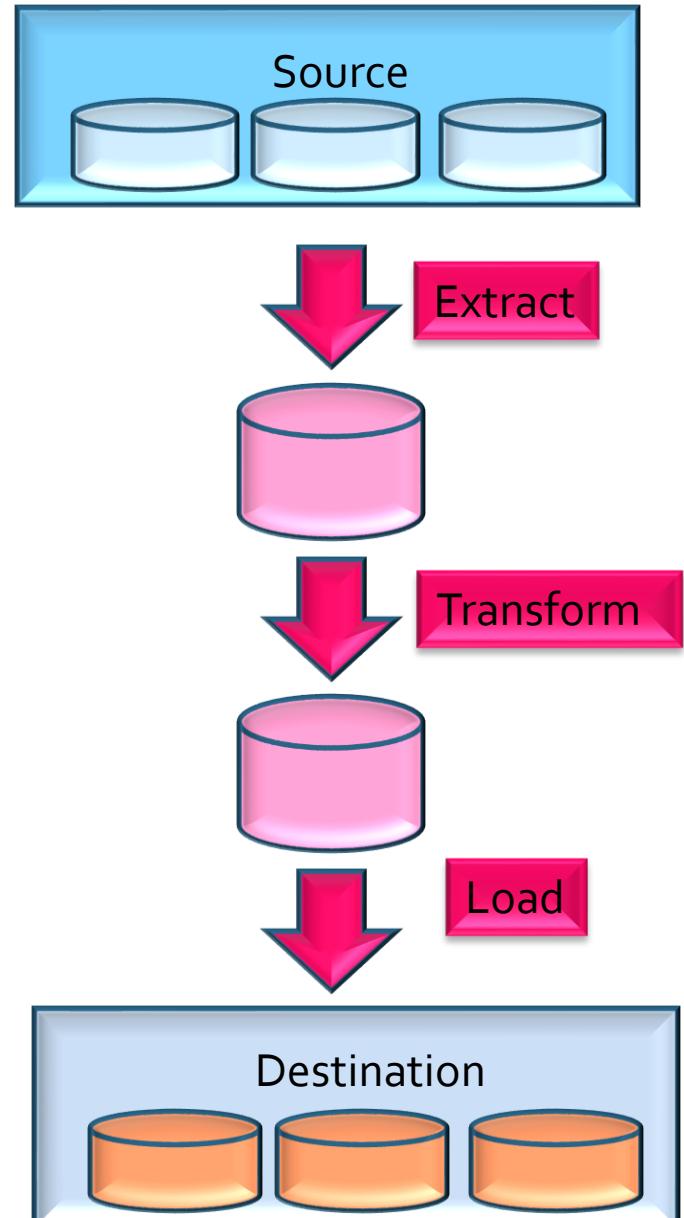
Reading data from different type of datasets

Transform

Formatting/converting data to needed shape

Load

Writing (Delivering) dataset to its target location



Fundamentals of ECL

- ✓ Declarative Language
- ✓ **Not** case-sensitive
- ✓ White space is ignored (Makes your code more readable)

```
// This is a single line comment  
/* A block comment */
```

- ✓ *Object.Property* syntax is used to qualify definition scope and disambiguate field references within datasets:
- ✓ *FolderName.Definition* //reference a definition from another module/folder
- ✓ *Dataset.Field* //reference a field in a dataset or record set

Fundamentals of ECL (Continued)

- ✓ Definition operator is **`:=`** “is defined as”
- ✓ Semicolon is line terminator: `num := 12 ;`
- ✓ Equality test is `=` `valOne = valTwo`
- ✓ Not equal: Use `<>` or `!=`
- ✓ Definitions can be defined only once.
- ✓ *Only those definitions that contribute to a result are compiled and used.*
- ✓ There are no loops as we know them! TRANSFORM and PROJECT is used instead.

Common Data Types

Character

- STRING[n]
- UTF8
- UNICODE[_locale][n]

Numeric

- INTEGER[n]
- UNSIGNED[n]
- REAL[n]
- DECIMAL<n>[_y]
- UDECIMAL<n>[_y]

Other

- BOOLEAN
- SET OF <type>
- RECORD
- DATASET

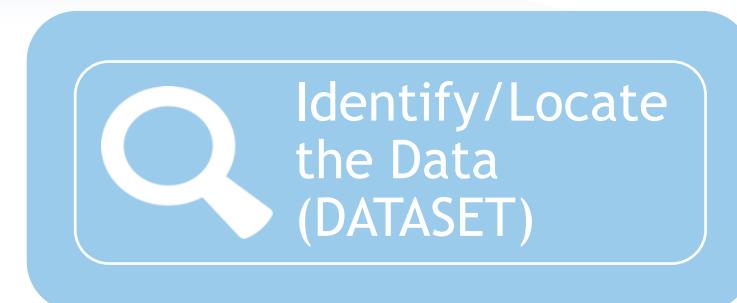
Usage:

```
Type      Name      := default value
UNSIGNED1 MyNumber := 0;
```

Name must start with a letter and can contain letters, numbers and the underscore character.

Three ECL Data Rules

Before you begin to work on any data in the HPCC cluster,
you must always do three things:



RECORD Structure

Defines the layout of fields in the dataset, order of the fields should be the same as the dataset.

DATASET

A physical data file. It can be defined in code (inline) or can be read from disk.

Job	Catergory	City	State	Avg_Salary
Manager	IT	Atlanta	GA	87000
Director	Art	Atlanta	GA	100000
CIO	IT	Tampa	FL	112000
Sales	General	Chicago	IL	55000

RECORD Structure Example:

```
EXPORT Layout_Company := RECORD
    UNSIGNED sic_code;
    STRING1 source;
    STRING120 company_name;
    STRING10 prim_range;
    STRING2 predir;
    STRING28 prim_name;
    STRING4 addr_suffix;
    STRING2 postdir;
    STRING5 unit_desig;
    STRING8 sec_range;
    STRING25 city;
    STRING2 state;
    STRING5 zip;
    STRING4 zip4;
    STRING10 phone;
END;
```

DATASET

```
name := DATASET( file, recorddef, THOR [options]);  
name := DATASET( file, recorddef, CSV [ ( options ) ] );  
name := DATASET( file, recorddef, XML( path,[options] ) );  
name := DATASET( file, recorddef, JSON( path,[options] ) );
```

- ✓ *name* - The definition name by which the file is subsequently referenced.
- ✓ *file* - A string constant containing the logical filename.
- ✓ *recorddef* - The RECORD structure of the dataset.
- ✓ *options* - options specific to the dataset type.
- ✓ *path* - A string constant containing the full XPATH to the tag that delimits the records in the *file*
- ✓ *command* - third-party program that creates the dataset.

DATASET introduces a new data file into the system with the specified *recorddef* layout.

RECORDOF

RECORDOF(*recordset*)

- *recordset* – The set of data records whose RECORD structure to use. This may be a DATASET or any derived recordset.

The **RECORDOF** declaration specifies inheriting just the record layout (without default values) of the specified *recordset*.

```
t := TABLE(People,{LastName,FirstName});
```

```
r := RECORD
  RECORDOF(t);
  UNSIGNED1 NewByte;
END;
```

RECORD and DATASET example

Layout_Company := RECORD

```
UNSIGNED    sic_code;
STRING120   company_name;
STRING10    prim_range;
STRING2     predir;
STRING28   prim_name;
STRING4    addr_suffix;
STRING2    postdir;
STRING5    unit_desig;
STRING8    sec_range;
STRING25   city;
STRING2    state;
STRING5    zip;
STRING4    zip4;
END;
```

EXPORT File_Company_List := DATASET('~CLASS::Company_List', Layout_Company, THOR);

Inline Dataset

```
SalaryAvg_Layout := RECORD
    STRING Job;
    STRING Category;
    STRING City;
    STRING2 State;
    INTEGER Avg_Salary;
END;

// Inline Dataset
SalaryAvg_DS := DATASET([
    {'Manager', 'IT', 'Atlanta', 'GA', 87000},
    {'Director', 'Art', 'Atlanta', 'GA', 100000},
    {'CIO', 'IT', 'Tampa', 'FL', 112000},
    {'Sales', 'General', 'Chicago', 'IL', 55000}
], SalaryAvg_Layout //Layout definition
);
```

OUTPUT

Let's display the result.

Job	Catergory	City	State	Avg_Salary
Manager	IT	Atlanta	GA	87000
Director	Art	Atlanta	GA	100000
CIO	IT	Tampa	FL	112000
Sales	General	Chicago	IL	55000

CHOOSE

Returns the first n number of records.

```
// A simple output  
OUTPUT(SalaryAvg_DS, NAMED('SalaryAvg_DS'));  
  
//CHOOSE  
OUTPUT(CHOOSEN(SalaryAvg_DS, 2), NAMED('SalaryAvg_Chosen'));
```

##	job	category	city	state	avg_salary
1	Manager	IT	Atlanta	GA	87000
2	Director	Art	Atlanta	GA	100000
3	CIO	IT	Tampa	FL	112000
4	Sales	General	Chicago	IL	55000

##	job	category	city	state	avg_salary
1	Manager	IT	Atlanta	GA	87000
2	Director	Art	Atlanta	GA	100000

SORT

Ascending or descending sort

Job	Category	City	State	Avg_Salary
Manager	IT	Atlanta	GA	87000
Director	Art	Atlanta	GA	100000
CIO	IT	Tampa	FL	112000
Sales	General	Chicago	IL	55000

Filter

Choosing a smaller part of dataset. A BOOLEAN expression following any recordset or dataset.

```
//Filter  
OUTPUT(SalaryAvg_DS(City = 'Tampa'), NAMED('Tampa_Filter'));
```

##	job	category	city	state	avg_salary
1	CIO	IT	Tampa	FL	112000

```
//Sort  
SortJobs := SORT(SalaryAvg_DS, Job);  
OUTPUT(SortJobs, NAMED('SortJobs'));
```

##	job	category	city	state	avg_salary
1	CIO	IT	Tampa	FL	112000
2	Director	Art	Atlanta	GA	100000
3	Manager	IT	Atlanta	GA	87000
4	Sales	General	Chicago	IL	55000

More on Filtering

All records within *dataset* will be evaluated

If *boolean_expression* evaluates to TRUE for a particular record, it will be included in the result

Logical Operators

AND

OR

NOT or ~

Comparison Operators

=

<> or !=

<

>

<=

>=

<=>

```
youngeOrLowIncome := allPeople(age < 20 OR  
                                avgHouseIncome <= 10000);
```

Math Functions

```
MathLayout := RECORD  
  INTEGER Num1;  
  INTEGER Num2;  
  INTEGER Num3;  
END;  
  
DS := DATASET([{20,45,34},  
               {909,56,45},  
               {30,-1,90}],  
              MathLayout);
```

Num1	Num2	Num3
20	45	34
909	56	45
30	-1	90

```
COUNT(DS);          //Counts the number records in a dataset -- Returns 3  
MAX(DS, Num1);    //Returns the MAX value on a field in a dataset -- Returns 909  
MIN(DS, Num2);    //Returns the MIN value on a field in a dataset -- Returns -1  
AVE(DS, Num1);    //Returns the AVERAGE value on a field in a dataset -- Returns 319.6666666666667  
SUM(DS, Num1 + Num3); //Returns the result of adding numbers together -- Returns 1128  
TRUNCATE(AVE(DS, Num1)); //Returns the integer portion of the real_value. -- Returns 319  
ROUND(3.45);       //Returns the rounded value -- Return 3  
ROUND(3.76);       //Returns the rounded value -- Return 4
```

CORRELATION

NumOne	NumTwo
1	1
2	2
3	3
4	4
5	5
6	6

NumObe	NumTwo
1938960000.00	2044820000.00
1779710000.00	854858000.00
2961810000.00	1248480000.00
2774400000.00	1263570000.00
1144160000.00	434290000.00
3387280000.00	1302380000.00
3195380000.00	1711770000.00

`CORRELATION(ds1, NumOne, NumTwo)`

`CORRELATION(ds2, NumOne, NumTwo)`

Returns 1.0

Returns 0.4978702535543908

FUNCTION (ECL Definitions with parameters)

```
EXPORT myfunc (STRING val) := FUNCTION
| Result := 'Hello ' + val + ' , welcome to this function';
| RETURN Result;
END;

//Using myfunc
res := myfunc('Jonny');
OUTPUT(res, NAMED('res'));

OUTPUT(myfunc('Sunny'), NAMED('Sunny'));
```

One Line Function

Sunny	Hello Sunny , welcome to this function
res	Hello Jonny , welcome to this function

```
INTEGER checkMax (SET OF INTEGER numList) := MAX(numList);
OUTPUT(checkMax([2,5,8,10,45,11]), NAMED('checkMath'));
```

MODULE

Is a container that allows you to group related definitions.
The *parameters* passed to the module are shared by all the related *members* definitions.

Variable Scope

- Local definitions are visible only up to an EXPORT or SHARED
- SHARED definitions are visible within module.
- EXPORT definitions are visible within and outside of a module .

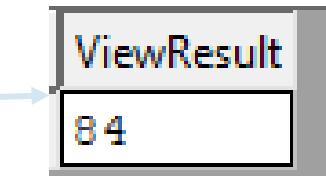
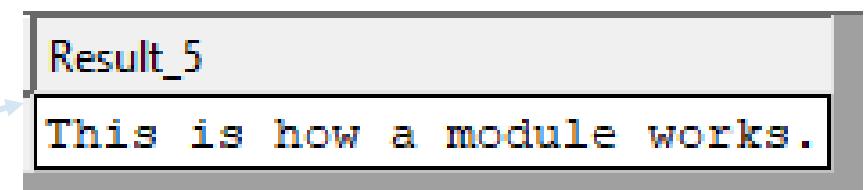
```
MyMod := MODULE

    // Visible only by MyMod
    SHARED x := 88;
    SHARED y := 42;

    // Visible by MyMod and outsiders
    EXPORT See := 'This is how a module works.';
    EXPORT res := Y * 2;
END;

OUTPUT(MyMod.See);

OUTPUT(MyMod.Res, Named('ViewResult'));
```



TRANSFORM

Specifies exactly how each field in the output record set is to receive its value.

- It should include the result type.
- Should contain name
- Contains parameter list
- SELF: refers to fields in result type.

PROJECT

Processes through all the records in the dataset performing the TRANSFORM.

- LEFT: refers to dataset getting passed to PROJECT.
- COUNTER: Optional counter that counts calls to TRANSFORM

Standalone TRANSFORM

```
Person_Layout := RECORD
  STRING FirstName;
  STRING LastName;
END;

NameDS := DATASET([{'Sun','Shine'},
                   {'Blue','Sun'},
                   {'Silver','Rose'}],
                  Person_Layout);

NameOutRec := RECORD
  STRING FirstName;
  STRING LastName;
  STRING CatValues;
  INTEGER RecCount
END;

NameOutRec CatThem(Person_Layout L, INTEGER C) := TRANSFORM
  SELF.CatValues := L.FirstName + ' ' + L.LastName; //Defines value for new field
  SELF.RecCount := C; // Adding Counter
  SELF := L; // Assign everything with same field name from NameDS
END;

CatRecs := PROJECT(NameDS, // Dataset to loop through
                    CatThem //Transform name
                    (LEFT, //Left dataset which is NameDS
                     COUNTER //Simpler Counter
                    ));

OUTPUT(CatRecs, NAMED('CatRecs'));
```

FirstName	LastName
Sun	Shine
Blue	Moon
Silver	Rose

firstname	lastname	catvalues	reccount
Sun	Shine	Sun Shine	1
Blue	Moon	Blue Moon	2
Silver	Rose	Silver Rose	3

NameOutRec: Result Layout

CatThem: Transform Name

Person_Layout: Input Dataset Layout

L : Reference to Person_Layout fields

SELF: Refers to fields in result dataset

C: Will do the Counting

```

Person_Layout := RECORD
    INTEGER PersonalID;
    STRING FirstName;
    STRING LastName;
END;

NameDS := DATASET([{100, 'Jo', 'Smith'},
                    {203, 'Dan', 'Carpenter'},
                    {498, 'Sally', 'Fryman'},
                    {302, 'Silver', 'Rose'}],
                  Person_Layout);

NameOutRec := RECORD
    INTEGER RecCount;
    INTEGER PersonalID;
    STRING PersonName;
    STRING FutureAddress;
END;

CatRecs := PROJECT(NameDS,
                    TRANSFORM(NameOutRec,
                               SELF.PersonName := LEFT.FirstName + ' ' + LEFT.LastName;
                               SELF.RecCount := COUNTER;
                               SELF := LEFT;
                               SELF := []);
                  );
OUTPUT(CatRecs, NAMED('Inline_CatRecs'));

```

PersonalID	FirstName	LastName
100	Jo	Smith
203	Dan	Carpenter
498	Sally	Fryman
302	Silver	Rose

Inline TRANSFORM

CatRecs: Project Name

NameDS: Input Dataset to loop through

NameOutRec: Result layout

SELF: Refers to fields in result dataset

SELF := LEFT: Assign everything with same field name from NameDS

SELF := []: All un-assigned fields will be set to default values

reccount	personalid	personname	futureaddress
1	100	Jo Smith	
2	203	Dan Carpenter	
3	498	Sally Fryman	
4	302	Silver Rose	

TABLE (recordsets in memory, cross-tab tool)

```
Pickup_Layout := RECORD
    STRING10  pickup_date;
    DECIMAL8_2 fare;
    DECIMAL8_2 distance;
END;
Pickup_DS  :=  DATASET([{'2015-01-01', 25.10, 5},
                        {'2015-01-01', 40.15, 8},
                        {'2015-01-02', 30.10, 6},
                        {'2015-01-02', 25.15, 4}],
                        Pickup_Layout);

crossTabLayout := RECORD
    Pickup_DS.pickup_date;
    avgFare   := AVE(GROUP, Pickup_DS.fare);
    totalFare := SUM(GROUP, Pickup_DS.fare);
END;

crossTabDs := TABLE(Pickup_DS, // Input Dataset
                    crossTabLayout,
                    pickup_date);

OUTPUT(crossTabDs, NAMED('crossTabDs'));
```

pickup_date	fare	distance
2015-01-01	25.1	5
2015-01-01	40.15	8
2015-01-02	30.1	6
2015-01-02	25.15	4

pickup_date	avgfare	totalfare
2015-01-01	32.625	65.25
2015-01-02	27.625	55.25

JOIN

The JOIN function produces a result set based on the intersection of two or more datasets or indexes.

INNER: Only those records that exist in both datasets.

LEFT OUTER: At least one record for every record in the left.

RIGHT OUTER: At least one record for every record in the right.

LEFT ONLY: One record for each left record with no match in the left.

RIGHT ONLY: One record for each left record with no match in the right.

FULL ONLY: One record for each left and right record with no match in the opposite.

EmpDS

EmpID	Name	HireYear
1000	Jack	2014
2000	Blue	2016
3000	Mary	2016
5000	Mart	2000
8000	Cat	2002

JobCatDS

EmpID	Department	Title
1000	IT	developer
2000	Biz	Manager
4000	Fin	accountant
8000	IT	analyst

```
InnerJoin := JOIN(EmpDS, JobCatDS,
    LEFT.EmpID = RIGHT.EmpID,
    TRANSFORM(EmpResult_Layout,
        SELF := LEFT,
        SELF := RIGHT));
```

empid	name	title	department
1000	Jack	developer	IT
2000	Blue	Manager	Biz
8000	Cat	analyst	IT

```
LeftOuterJoin := JOIN(EmpDS, JobCatDS,
    LEFT.EmpID = RIGHT.EmpID,
    TRANSFORM(EmpResult_Layout,
        SELF := LEFT,
        SELF := RIGHT),
    LEFT OUTER);
```

empid	name	title	department
3000	Mary		
5000	Mart		

```
FullOuterJoin := JOIN(EmpDS, JobCatDS,
    LEFT.EmpID = RIGHT.EmpID,
    TRANSFORM(EmpResult_Layout,
        SELF := LEFT,
        SELF := RIGHT),
    FULL OUTER);
```

empid	name	title	department
1000	Jack	developer	IT
2000	Blue	Manager	Biz
3000	Mary		
0		accountant	Fin
5000	Mart		
8000	Cat	analyst	IT

VISUALIZATION (built-ins and an ECL Bundle)

Methods include

- Two-Dimensional
- Multi-Dimensional Methods
- Geospatial
- General

A basic visualization typically requires the following steps:

1. Creation of a suitable dataset.
2. Output the dataset with a suitable name, so that visualization can locate the data.
3. Create (and output) the visualization, referencing the named output from step 2

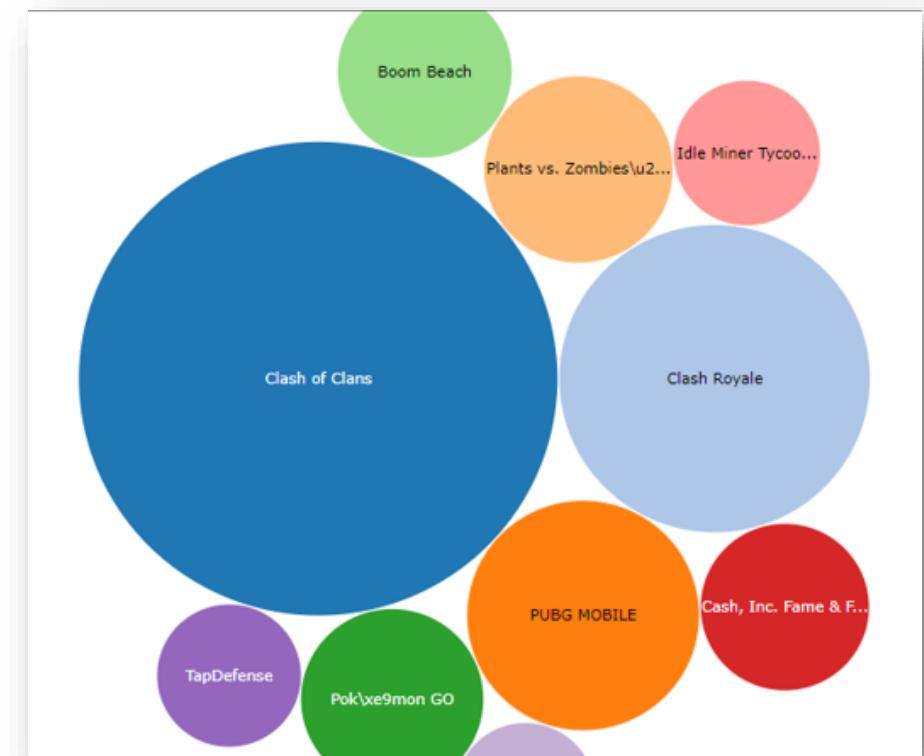
```

top_user_rating_count := TOPN(
    TABLE(clean_mod.games_ds,
        {name,
         user_rating_count}) ,
    10,
    -user_rating_count);

OUTPUT(analysis_mod.top_user_rating_count, NAMED('user_rating_count'));
Visualizer.TwoD.Bubble(['user_rating_count',
    /*datasource*/,
    'user_rating_count']);

```

Bubble
Pie
Bar
Scatter
Line
WorldCloud
Area





Help Missing Kids! Challenge



MARCH 2025

Bob Foreman
Principal Software Engineer
LexisNexis Risk Solutions

The KSU Data for Social Good Challenge!

Help Missing Kids!

On average, 1000 children go missing every day in the United States. According to the National Center for Missing and Exploited Children® (NCMEC), 1 in 6 missing children who run away become victims of human trafficking. HPCC Systems has partnered with NCMEC to help notify first responders, law enforcement, and the community when a child is reported missing.

What can we do as developers to help understand this problem, and what can we do to help?

This year's challenge will analyze different social factors such as poverty, unemployment, and other areas to develop possible insights as to *why* so many are missing *and* to provide additional first responders and other resource information to the requestor.

The goal of the challenge is to answer two questions:

1. Is there a correlation between the locations that kids are reported missing and other social factors in that area (unemployment, education, poverty, and population).
2. Can we provide additional information to the NCMEC feed to help find the missing children? (fire and police stations, hospitals, churches, food banks, etc.)

Join in the effort to help find missing children

- The ADAM Program was launched in November 2000 and donated by LexisNexis® Risk Solutions to the **National Center for Missing & Exploited Children**, (NCMEC) to assist law enforcement in the recovery efforts of missing children.
- **Time is of the essence** when a child is missing, and photos play a critical role in the recovery efforts. NCMEC uses the ADAM Program to **quickly distribute a poster** of a missing child, targeting a specific search area.
- NCMEC can leverage the ADAM Program to distribute alerts on all missing child cases **including endangered runaways** and other missing child cases that do not meet AMBER Alert criteria.
- **The community can help** in the recovery of missing children by signing up for free to receive missing child alerts in their area at adamprogram.com
- Help **spread awareness** about The ADAM Program and encourage your network of contacts to also signup to receive missing child alerts.
- **Thank you for your interest and support** in this important cause.
- **Questions?**

Trish McCall
The ADAM Program, Co-Founder
Sr Director, Program Management
LexisNexis Risk Solutions
Trish.McCall@lexisnexisrisk.com
Twitter: [@cybertrish](https://twitter.com/cybertrish)

Patti Willingham
Executive Director,
Case Management Services
National Center for Missing &
Exploited Children (NCMEC)
PAWillingham@ncmec.org



For more information

▶ Watch the ADAM Program overview video

✉ Download the postcard

❓ Learn more at adamprogram.com



The Data!

NCMEC Data has been collected from all 50 states and organized into a simple dataset to use as your central source. In addition, many public datasets have also been gathered and cleaned to help get you started.

These datasets include:

Education

Unemployment

Poverty

Population

Police

Fire Stations

Hospitals

Places of Worship

Food Banks

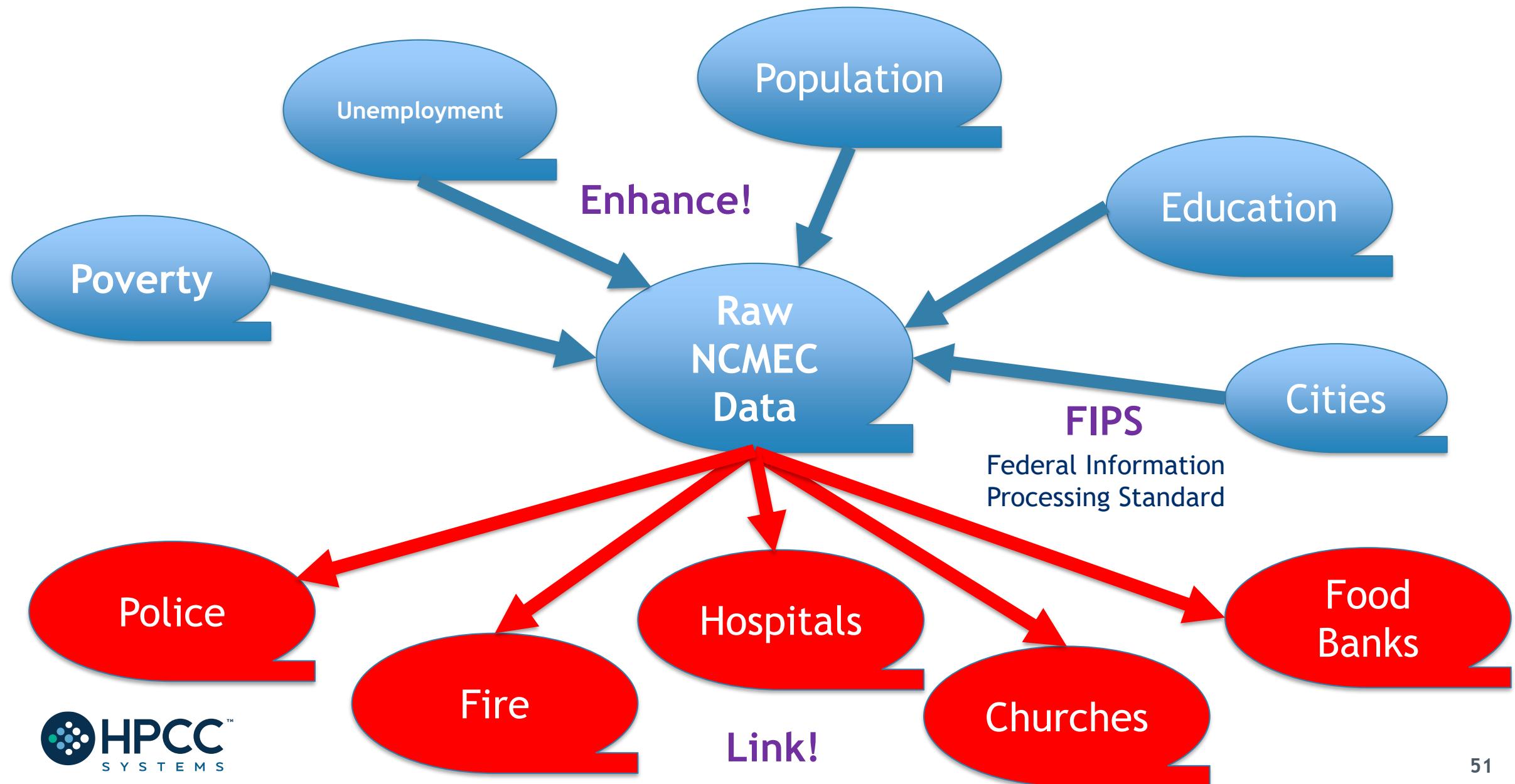
Auxiliary Datasets:

A **Cities** dataset with related FIPS and Zip Codes (used for linking the above datasets to the NCMEC data)

Unemployment Rates (Not really used in this challenge but interesting data!)

You are not limited to using these datasets! Extra credit will be rewarded by linking in other pertinent data!

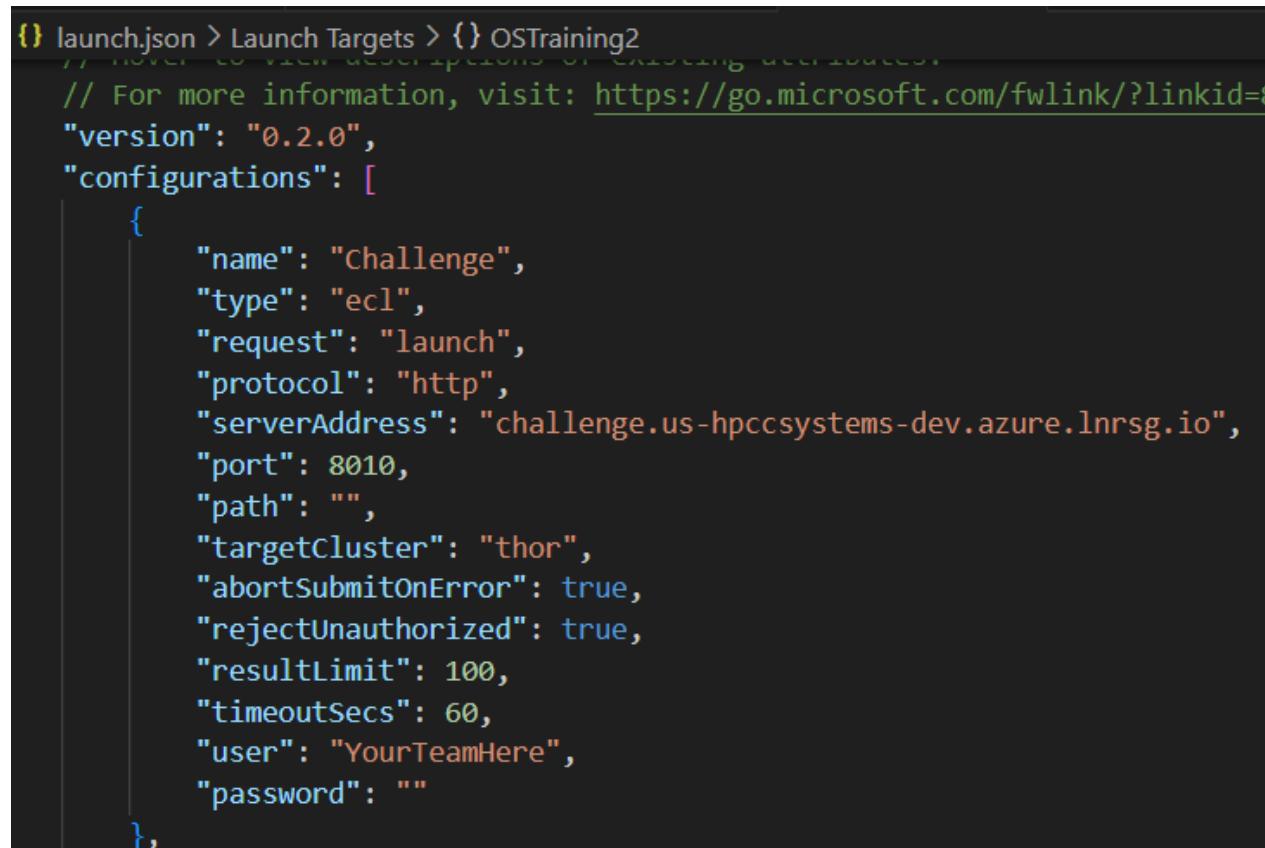
Data Flow



The Playing Field!

HPCC Cluster ECL Watch:

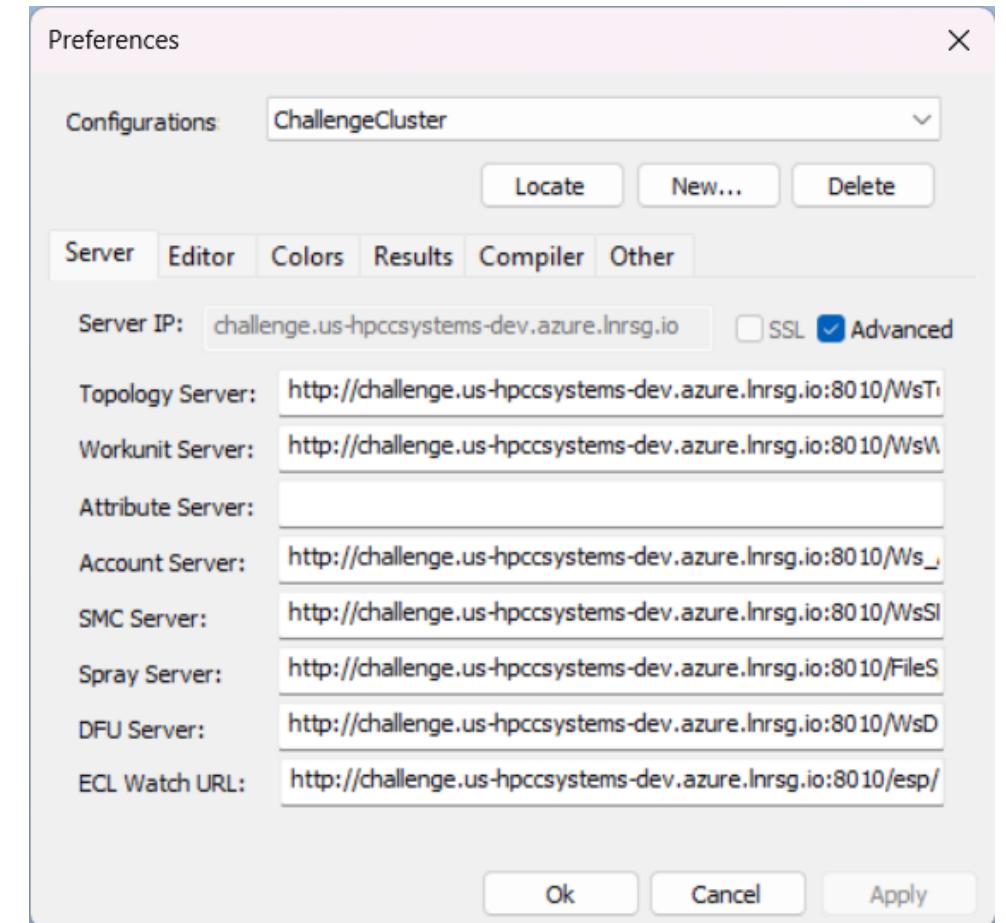
<http://challenge.us-hpccsystems-dev.azure.lnrsg.io:8010/>



```
{} launch.json > Launch Targets > {} OTraining2
// For more information, visit: https://go.microsoft.com/fwlink/?LinkId=8
"version": "0.2.0",
"configurations": [
    {
        "name": "Challenge",
        "type": "ecl",
        "request": "launch",
        "protocol": "http",
        "serverAddress": "challenge.us-hpccsystems-dev.azure.lnrsg.io",
        "port": 8010,
        "path": "",
        "targetCluster": "thor",
        "abortSubmitOnError": true,
        "rejectUnauthorized": true,
        "resultLimit": 100,
        "timeoutSecs": 60,
        "user": "YourTeamHere",
        "password": ""
    },
}
```



{} launch.json X



The Repo!

<https://github.com/hpccsystems-solutions-lab/HelpMissingKids25>

The screenshot shows the GitHub repository page for 'HelpMissingKids25'. The repository is public and has 1 branch and 0 tags. The main commit by 'bobf2000' is 'Update README.md' (commit ceb6130, 2 minutes ago). Below the commit list, there's a table of files with their descriptions and last update times. At the bottom, there are links to 'README' and 'Apache-2.0 license'. On the right side, there's a 'Clone' section with options for HTTPS, SSH, and GitHub CLI, and a link to 'Clone using the web URL'. A red circle highlights the 'Code' button in the top right corner of the main repository area. Another red circle highlights the 'Download ZIP' button in the 'Clone' section.

File	Description	Last Updated
.vscode	Code Update	last year
Code	Create BWR_CleanChurches.ecl	last year
Docs	Create quickstart.mod	last year
Images	Update repo folders and ReadMe	last year
LICENSE	Initial commit	2 years ago
README.md	Update README.md	2 minutes ago
edlcc.log	Update Docs	last year
quickstart.mod	Create quickstart.mod	last year



Examples and Tips



NCMEC Data

<https://www.missingkids.org/gethelpnow/search/rss>



RSS NEWS FEEDS

You can receive missing child alerts in RSS format. To subscribe to any of the RSS feeds below, right click on the link and copy the URL. Then paste the URL into your reader.

These are not AMBER Alerts. For information about the AMBER Alert Program or to see active AMBER Alerts visit the [AMBER Alert](#) page.

Missing Children Alert Cases

Add this RSS feed to receive alerts when a new missing child is added to our group of missing-children-alert cases.

[Missing Children Alerts](#)

Regional Missing Children Alerts

Add these RSS feeds to receive alerts whenever a missing child from the chosen area is posted to NCMEC's Web database

Alabama	Missouri
Alaska	Montana
Arizona	Nebraska
Arkansas	Nevada
California	New Hampshire
Colorado	New Jersey

NCMEC Data

CSV format:

#	recid	dateposted	firstname	lastname	currentage	datemissing	missingcity	missingstate	contact	photolink
1	1311686	12/11/2018	Melvin	Horst	99	19281227	ORRVILLE	OH	Orrville Police Department (Ohio) 1-330-684-5025	http://www.missingkids.org/poster/NCMC/
2	1176415	8/10/2012	Marjorie	West	90	19380508	HAMILTON TOWNSHIP	PA	Pennsylvania State Police (Pennsylvania) 1-814-938-0510	http://www.missingkids.org/poster/NCMC/
3	1148707	12/20/2021	Beverly	Sharpman	93	19470911	PHILADELPHIA	PA	Philadelphia Police Department (Pennsylvania) 1-215-685-3252	http://www.missingkids.org/poster/NCMC/
4	1007456	8/23/2021	Ricky	Bryant	78	19491219	MAUSTON	WI	Juneau County Sheriff's Department (Wisconsin) 1-608-847-5649	http://www.missingkids.org/poster/NCMC/
5	1154349	7/26/2018	Connie	Smith	81	19520716	LAKEVILLE	CT	Connecticut State Police (Connecticut) 1-860-626-7975	http://www.missingkids.org/poster/NCMC/

Step 1: Enhance this data by adding data points from Unemployment, Poverty, Population, Education, or any other datasets you can think of!

NCMEC Data

Cleaned and Enhanced NCMEC data:

recid	dateposted	firstname	lastname	currentage	datemissing	missingcity	primaryfips	missingstate	ump_rate	pov_pct	popest	edu_high	contact	photolink
1475911	20230823	AXEL	GOMEZ HERNANDEZ	9	20180227		0		5.87	16.9	332276874	25.17	NCMEC MISSING CHILDREN'S DIVISION - 333 JOH...	http://www.missingkids.org
601773	20161219	TYLER	INMAN	44	19821221	ABERDEEN	53027	WA	9.14	20.8	76560	24.84	ABERDEEN POLICE DEPARTMENT (WASHINGTON) 1-3...	http://www.missingkids.org
603596	20200514	DEAN	PYLE PETERS	57	19810205	ADA	0	MI	5.87	16.9	332276874	25.17	KENT COUNTY SHERIFF'S OFFICE (MICHIGAN) - M...	http://www.missingkids.org
1481918	20230615	JACINTO	BRITO-RAMIREZ	17	20230530	ADDISON	17043	IL	5.38	7.5	926198	13.48	ADDISON POLICE DEPARTMENT (ILLINOIS) 1-63...	http://www.missingkids.org
830487	20210714	SUSAN	SMALLEY	54	19880319	ADDISON	48113	TX	5.77	20.3	2599587	27.07	CARROLLTON POLICE DEPARTMENT (TEXAS) - MISS...	http://www.missingkids.org
707842	20210714	STACIE	MADISON	53	19880319	ADDISON	48113	TX	5.77	20.3	2599587	27.07	CARROLLTON POLICE DEPARTMENT (TEXAS) 1-972-...	http://www.missingkids.org
1330863	20180618	EVA	BARRIOS-TORNEZ	22	20180528	ADELANTO	6071	CA	7.3	17.1	2189926	27.26	SAN BERNARDINO COUNTY SHERIFF'S OFFICE (CAL...	http://www.missingkids.org
1456392	20220727	NAYELI	RODRIGUEZ GONZALEZ	18	20220724	ADELPHI	24033	MD	5.34	15.7	956172	19.1	PRINCE GEORGE'S COUNTY POLICE DEPARTMENT (...)	http://www.missingkids.org
941306	20220831	ANGEL	TORRES-IRIZARRY	49	19760706	ADJUNTAS	72001	PR	15.75	0	17980	53.39	DEPARTMENT OF JUSTICIA (PUERTO RICO) 1-787...	http://www.missingkids.org
803713	20160421	OMAR	VARGAS	32	19950422	AGUA PRIETA	0		5.87	16.9	332276874	25.17	COCHISE COUNTY SHERIFF'S OFFICE (ARIZONA)	http://www.missingkids.org

Step 2: Clean and Standardize the Dataset! This will allow a more accurate search result when we build the delivery system!

NCMEC Data ECL Example:

```
1 IMPORT STD,$;
2
3 NCMEC_Rec := $.File_AllData.mc_byState;
4 NCMEC_DS  := $.File_AllData.mc_byStateDS;
5 Cities     := $.File_AllData.City_DS;
6 UNEMP      := $.File_AllData.unemp_byCountyDS;
7 EDU        := $.File_AllData.EducationDS;
8 POVTY      := $.File_AllData.pov_estimatesDS;
9 POP         := $.File_AllData.pop_estimatesDS;
```

```
6 EXPORT NCMECPlusLayout := RECORD
7     UNSIGNED3 recid;
8     UNSIGNED4 dateposted;
9     STRING18 FirstName;
10    STRING24 LastName;
11    UNSIGNED1 currentage;
12    UNSIGNED4 datemissing;
13    STRING23 missingcity;
14    UNSIGNED3 PrimaryFIPS;
15    STRING2 missingstate;
16    DECIMAL5_2 ump_rate; //New field
17    DECIMAL5_2 pov_pct; //New Poverty percent for children 0-17
18    UNSIGNED4 PopEst; //Population Estimate from 2020-2022
19    DECIMAL5_2 edu_High; //less than a high school diploma (percent)
20    STRING131 contact;
21    STRING96 photolink;
22 END;
```

```
19  NewNCMECLayout CleanNCMEC(NCMEC_DS Le,UNSIGNED2 CNT) := TRANSFORM
20      // SELF.RecID   := CNT; //Now uses Case Number
21      SELF.DatePosted := STD.Date.FromStringToDate(Le.DatePosted, '%m/%d/%Y');
22      SELF.FirstName  := STD.Str.ToUpperCase(Le.FirstName);
23      SELF.LastName   := STD.Str.ToUpperCase(Le.LastName);
24      // SELF.DateMissing := STD.Date.FromStringToDate(Le.DateMissing, '%m/%d/%Y'); //Processed earlier
25      SELF.Contact    := STD.Str.ToUpperCase(Le.Contact);
26      SELF.PrimaryFIPS := 0;
27      SELF.ump_rate   := 0;
28      SELF.pov_pct    := 0;
29      SELF.PopEst     := 0;
30      SELF.edu_High  := 0;
31      SELF            := Le;
32  END;
33  //Step 1: Make room for new metrics, standardize dates, names, contact and sequence records
34  Clean_NCMEC_DS := PROJECT(NCMEC_DS,CleanNCMEC(LEFT,COUNTER));
```

NCMEC Data ECL Example:

Adding the FIPS code:

```
37  □NewNCMECLayout GetFIPS(Clean_NCMEC_DS Le,Cities Ri) := TRANSFORM
38      SELF.PrimaryFIPS := (UNSIGNED3)Ri.county_fips;
39      SELF              := Le;
40  END;
41
42  AddFIPS := JOIN(Clean_NCMEC_DS,Cities,
43                    LEFT.missingcity = STD.STR.ToUpperCase(RIGHT.city) AND
44                    LEFT.missingstate = RIGHT.state_id,
45                    GetFIPS(LEFT,RIGHT),LEFT OUTER);
```

NCMEC Data ECL Example: Analyzing and Visualizing!

```
51 //Cross-Tab by City:  
52  
53 CT_City := TABLE(AddFIPS,{missingcity,missingstate,cnt := COUNT(GROUP)},missingstate,missingcity);  
54 Out_CT_City := OUTPUT(SORT(CT_City,-cnt),NAMED('MissByCity'));  
55  
56 //Cross-Tab by State:  
57  
58 CT_ST := TABLE(AddFIPS,{missingstate,cnt := COUNT(GROUP)},missingstate);  
59 Out_CT_ST := OUTPUT(SORT(CT_ST,-cnt),NAMED('MissByState'));  
60 Visualizer.Choropleth.USStates('MissingByState', , 'MissByState', , , DATASET([{'paletteID': 'PuBuGn'}]), Visualizer.KeyValueDef));  
61  
62 //Cross-Tab by Date Missing:  
63  
64 CT_date := TABLE(AddFIPS,{DateMissing,cnt := COUNT(GROUP)},DateMissing);  
65 Out_CTDates := OUTPUT(SORT(CT_date,-cnt),NAMED('MissByDate'));  
66  
67 //Cross-Tab by Primary FIPS:  
68  
69 CT_FIPS := TABLE(AddFIPS,{PrimaryFIPS,cnt := COUNT(GROUP)},PrimaryFIPS);  
70 Out_CT_FIPS := OUTPUT(SORT(CT_FIPS,-cnt),NAMED('MissByFIPS'));
```

NCMEC Data

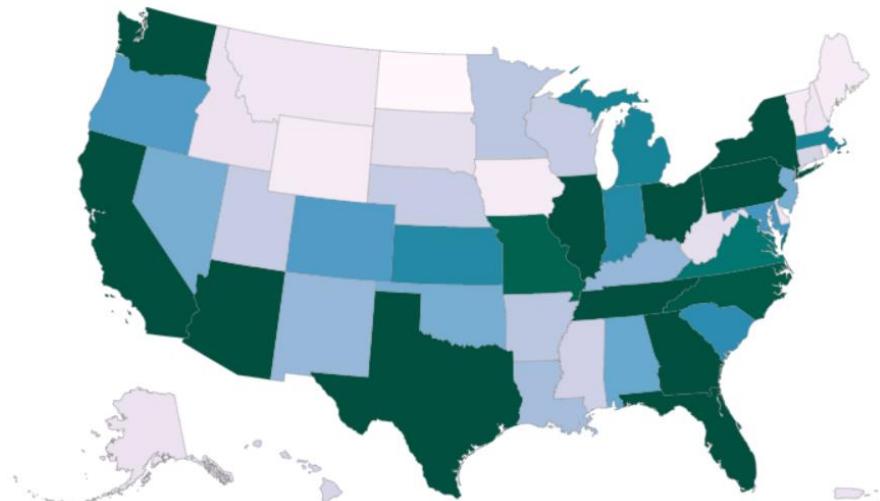
Analyzing and Visualizing!

##	missingcity	missingstate	cnt
1	LOS ANGELES	CA	52
2	PHOENIX	AZ	48
3	CHICAGO	IL	33
4	PHILADELPHIA	PA	33
5	HOUSTON	TX	26
6	MEMPHIS	TN	21
7	PORTRLAND	OR	19
8	TUCSON	AZ	19
9	LAS VEGAS	NV	19
10	MIAMI	FL	19

##	primaryfips	cnt
1	0	870
2	6037	135
3	4013	91
4	48201	42
5	17031	39
6	12011	37
7	6065	35
8	42101	33
9	6071	30
10	6059	29

##	datemissing	cnt
1	20240105	16
2	20240108	13
3	20231223	13
4	20231206	12
5	20230911	11
6	20231106	11
7	20240109	11
8	20240112	11
9	20231231	11
10	20240111	11

##	missingstate	cnt
1	CA	541
2	FL	278
3	TX	250
4	AZ	173
5	OH	127
6	NY	124
7	PA	118
8	GA	106
9	TN	105
10	WA	100



NCMEC Data ECL Example:

Adding additional data points:

```
72 //Add Unemployment Rate for area:  
73 CT_UNEMP := TABLE(UNEMP((STD.Str.Find(attribute, 'Unemployment_rate',1) <> 0)),  
74     {Fips_Code,cnt := ROUND(AVE(GROUP,value),2)},Fips_Code);  
75 // OUTPUT(SORT(CT_UNEMP,-cnt),NAMED('UNEMP_Rate'));  
76  
77 ADDUMP := JOIN(AddFIPS,CT_UNEMP,LEFT.PrimaryFIPS=RIGHT.Fips_Code,  
78                 TRANSFORM(NewNCMECLayout,  
79                     SELF.ump_rate := RIGHT.cnt,  
80                     SELF := LEFT),LEFT OUTER,LOOKUP);  
81  
82  
84 //Add Poverty Percentage ages 0-17 for FIPS area:  
85 POVTBL := TABLE(POVTY((STD.Str.Find(attribute, 'PCTPOV017_2021',1) <> 0)),  
86     {Fips_Code,attribute,value});  
87 // OUTPUT(SORT(POVTBL,-value),NAMED('PovertyPct0to17'));  
88  
89 ADDPOV := JOIN(AddUMP,POVTBL,LEFT.PrimaryFIPS=RIGHT.Fips_Code,  
90                 TRANSFORM(NewNCMECLayout,  
91                     SELF.pov_pct := RIGHT.value,  
92                     SELF := LEFT),LEFT OUTER,LOOKUP);  
93
```



Data Delivery (Roxie and Visualization)



Delivering your Results!

Step 3 - Clean, Declare and Build your Indexes (Fire):

```
11  SHARED CleanFireRec := RECORD
12    STRING100 name;
13    STRING60 addressbuildingname;
14    STRING65 address;
15    UNSIGNED3 PrimaryFIPS := 0; //New - Added from Cities DS
16    STRING35 city;
17    STRING2 state;
18    STRING10 zipcode;
19  END;
20  EXPORT CleanFire := PROJECT(Fire,TRANSFORM(CleanFireRec,
21                                SELF.name          := STD.STR.ToUpperCase(LEFT.name),
22                                SELF.addressbuildingname := STD.STR.ToUpperCase(LEFT.addressbuildingname),
23                                SELF.address        := STD.STR.ToUpperCase(LEFT.address),
24                                SELF.city           := STD.STR.ToUpperCase(LEFT.city),
25                                SELF.State          := STD.STR.ToUpperCase(LEFT.state),
26                                SELF.zipcode        := STD.STR.ToUpperCase(LEFT.zipcode)));
27  EXPORT CleanFireFIPS := JOIN(CleanFire,Cities,
28                                LEFT.city = STD.STR.ToUpperCase(RIGHT.city) AND
29                                LEFT.state = RIGHT.state_id,
30                                TRANSFORM(CleanFireRec,
31                                SELF.PrimaryFIPS := (UNSIGNED3)RIGHT.county_fips,
32                                SELF            := LEFT),LEFT OUTER,LOOKUP);
33
34  EXPORT CleanFireDS   := DATASET('~/HMK::OUT::Fire',CleanFireRec,FLAT);
35  EXPORT CleanFireIDX  := INDEX(CleanFireDS,{city,state},{CleanFireDS}, '~HMK::IDX::Fire::CityPay');
36  EXPORT CleanFireFIPSIDX := INDEX(CleanFireDS,{PrimaryFIPS},{CleanFireDS}, '~HMK::IDX::Fire::FIPSPay');
37  EXPORT BuildFireIDX  := BUILD(CleanFireIDX,OVERWRITE);
38  EXPORT BuildFireFIPSIDX := BUILD(CleanFireFIPSIDX,OVERWRITE);
```



Delivering your Results!

Step 4 - Clean, Declare and Build your Indexes (Police):

```
40 //Police
41 SHARED CleanPoliceRec := RECORD
42     STRING135 name;
43     STRING80 address;
44     STRING30 city;
45     STRING2 state;
46     STRING5 zip;
47     STRING15 zip4;
48     STRING15 telephone;
49     STRING25 type;
50     STRING15 status;
51     INTEGER3 population;
52     STRING25 county;
53     UNSIGNED3 countyfips;
54     STRING3 country;
55     REAL8 latitude;
56     REAL8 longitude;
57 END;
58
59 EXPORT CleanPolice      := PROJECT(Police,TRANSFORM(CleanPoliceRec,SELF.countyfips := (UNSIGNED3)LEFT.countyfips,SELF := LEFT));
60 // EXPORT CleanPoliceFIPS := JOIN(CleanPolice,Cities,
61 //                                 // LEFT.city = STD.STR.ToUpperCase(RIGHT.city) AND
62 //                                 // LEFT.state = RIGHT.state_id,
63 //                                 // TRANSFORM(CleanPoliceRec,
64 //                                         // SELF.countyFIPS := (UNSIGNED3)RIGHT.county_fips,
65 //                                         // SELF          := LEFT),LEFT OUTER,LOOKUP);
66 EXPORT CleanPoliceDS   := DATASET('~HMK::OUT::Police',CleanPoliceRec,FLAT);
67 EXPORT CleanPoliceIDX := INDEX(CleanPoliceDS,{countyfips,City,State},{CleanPoliceDS},'~HMK::IDX::Police::CityPay');
68 EXPORT BuildPoliceIDX := BUILD(CleanPoliceIDX,OVERWRITE);
```

Delivering your Results!

Step 5 - Clean, Declare and Build your Indexes (Hospitals and NCMEC):

```
23 //Dataset Generated in BWR_STD_NCMEC:  
24 EXPORT NCMECPlusDS      := DATASET('~HMK::OUT::NECMCPlus',NCMECPlusLayout,FLAT);  
25 EXPORT NCMECPlusIDXPay  := INDEX(NCMECPlusDS,{PrimaryFIPS,missingstate,missingcity},{NCMECPlusDS},'~HMK::IDX::NECMC::FIPSStCity');  
26 EXPORT BuildNewNCMECIDX := BUILD(NCMECPlusIDXPay,OVERWRITE);  
  
70 //Hospital  
71 SHARED CleanHospitalRec := RECORD  
72     STRING95 name;  
73     STRING80 address;  
74     STRING35 city;  
75     STRING2 state;  
76     STRING5 zip;  
77     STRING15 zip4;  
78     STRING15 telephone;  
79     STRING20 type;  
80     STRING6 status;  
81     INTEGER2 population;  
82     STRING20 county;  
83     UNSIGNED3 countyfips;  
84     STRING3 country;  
85     REAL8 latitude;  
86     REAL8 longitude;  
87 END;  
88  
89 EXPORT CleanHospital      := PROJECT(Hospital,TRANSFORM(CleanHospitalRec,SELF.countyfips := (UNSIGNED3)LEFT.countyfips,SELF := LEFT));  
90 EXPORT CleanHospitalDS    := DATASET('~HMK::OUT::Hospital',CleanHospitalRec,FLAT);  
91 EXPORT CleanHospitalIDX   := INDEX(CleanHospitalDS,{countyfips,City,state},{CleanHospitalDS},'~HMK::IDX::Hospital::CityPay');  
92 EXPORT BuildHospitalIDX   := BUILD(CleanHospitalIDX,OVERWRITE);
```

Delivering your Results!

Step 6 - The Build Action:

```
1 IMPORT $;
2 //Indexes used for ROXIE demo
3
4 $.File_CleanResponders.BuildFireIDX;
5 $.File_CleanResponders.BuildFireFIPSIDX;
6 $.File_CleanResponders.BuildPoliceIDX;
7 $.File_CleanResponders.BuildHospitalIDX;
8
9 $.File_EnhanceNCMEC.BuildNewNCMECIDX;
```

36	ABBEVILLE	SC	ABBEVILLE COUNTY FIRE DEPARTMENT LONG CANE			5 DURHAM ROAD	45001	29620
37	ABBEVILLE	SC	ABBEVILLE COUNTY FIRE DEPARTMENT MONTEREY			1970 MONTEREY ROAD	45001	29620
38	ABBOTSFORD	WI	ABBOTSFORD FIRE DEPARTMENT AND AMBULANCE SERVICE			203 BIRCH STREET	55073	54405
39	ABBOTTSTOWN	PA	UNITED HOOK AND LADDER ABBOTTSTOWN STATION 33			38 EAST KING STREET	42001	17301
40	ABBYVILLE	KS	RENO COUNTY FIRE DISTRICT 4 STATION 2			100 EAST AVENUE F STREET	20155	67510
41	ABERCROMBIE	ND	ABERCROMBIE FIRE DEPARTMENT			606 BROADWAY	38077	58001
42	ABERDEEN	ID	ABERDEEN - SPRINGFIELD FIRE DEPARTMENT			51 NORTH MAIN STREET	16011	83210
43	ABERDEEN	MD	ABERDEEN FIRE DEPARTMENT HOUSE 1			21 NORTH ROGERS STREET	24025	210...

6	1003	BAY MINETTE	AL	BAY MINETTE POLICE DEPARTMENT	300 NORTH HOYLE AVENUE	36507	4525	(251) 580-1682	LOCAL POLICE DEPARTMENT	OPEN	28
7	1003	BAY MINETTE	AL	JAMES H FAULKNER STATE COMMUNITY COLLEGE CAMPUS P...	1900 SOUTH UNITED STATES HIGHWAY 31	36507	NOT AVAILABLE	(251) 580-2222	SPECIAL JURISDICTION	OPEN	4
8	1003	DAPHNE	AL	DAPHNE POLICE DEPARTMENT / DAPHNE CITY JAIL	1502 UNITED STATES HIGHWAY 98	36526	NOT AVAILABLE	(251) 621-2834	LOCAL POLICE DEPARTMENT	OPEN	75
9	1003	ELBERTA	AL	ELBERTA POLICE DEPARTMENT	1362 MAIN STREET	36530	NOT AVAILABLE	(251) 986-5300	LOCAL POLICE DEPARTMENT	OPEN	6
10	1003	FAIRHOPE	AL	FAIRHOPE POLICE DEPARTMENT	107 NORTH SECTION STREET	36532	2430	(251) 928-2385	LOCAL POLICE DEPARTMENT	OPEN	54
11	1003	FOLEY	AL	FOLEY POLICE DEPARTMENT	200 EAST SECTION AVENUE	36535	2703	(251) 952-4010	LOCAL POLICE DEPARTMENT	OPEN	86
12	1003	GULF SHORES	AL	GULF SHORES POLICE DEPARTMENT	220 CLUBHOUSE DRIVE	36542	NOT AVAILABLE	(251) 968-2431	LOCAL POLICE DEPARTMENT	OPEN	54
13	1003	GULF SHORES	AL	GULF STATE PARK RANGER STATION	20115 STATE HIGHWAY 135	36542	4501	(251) 948-7275	LOCAL POLICE DEPARTMENT	OPEN	-999
14	1003	LOXLEY	AL	LOXLEY POLICE DEPARTMENT	2131 EAST RELHAM DRIVE	36551	2401	(251) 964-6000	LOCAL POLICE DEPARTMENT	OPEN	17

1151	12011	FL	FORT LAUDERDALE	2010176	20240118	JUAN	CASTRO-ARIAS	14	20240108	5.3	16.5	1942007	22.97	FORT LAUDERDALE POLI
1152	12011	FL	HALLANDALE BEACH	1185285	20111209	BRIANA	CONKLIN	15	20111203	5.3	16.5	1942007	22.97	FAIRFAX COUNTY POLIC
1153	12011	FL	HALLANDALE BEACH	1185285	20111209	SEBAS...	CONKLIN	17	20111203	5.3	16.5	1942007	22.97	FAIRFAX COUNTY POLIC
1154	12011	FL	HOLLYWOOD	1297203	20191015	YESENIA	DUVAL DEL ROSARIO	33	20101014	5.3	16.5	1942007	22.97	HOLLYWOOD POLICE DEP
1155	12011	FL	HOLLYWOOD	1321877	20210427	YUSUF	SHIKDER	13	20150315	5.3	16.5	1942007	22.97	CONTACT THE NEAREST
1156	12011	FL	HOLLYWOOD	1321877	20210427	ZAHRA	SHIKDER	9	20150315	5.3	16.5	1942007	22.97	CONTACT THE NEAREST
1157	12011	FL	LAUDERHILL	601922	20110112	JASON	TOWNSEND	47	19800520	5.3	16.5	1942007	22.97	BROWARD COUNTY SHERI
1158	12011	FL	PEMBROKE PINES	961542	20220624	ANAST...	ARGENTOVA-STEVENS	26	20030423	5.3	16.5	1942007	22.97	PEMBROKE PINES POLIC
1159	12011	FL	PLANTATION	915648	20120809	LEAH	VAN SCHOICK	58	19820815	5.3	16.5	1942007	22.97	BROWARD COUNTY SHERI

Delivering your Results!

Step 7 - Function to search by City,State or FIPS:

```
1  IMPORT $,STD;
2  UpperIt(STRING txt) := Std.Str.ToUpperCase(txt);
3
4  EXPORT fn_FindKids := MODULE
5      SHARED NCMECKey      := $.File_EnhanceNCMEC.NCMECPlusIDXPay;
6
7  //PrimaryFIPS,missingstate
8  EXPORT By_CityST(STRING23 city_in,STRING2 st_in)      := FUNCTION
9      RETURN IF(st_in = '',
10             NCMECKey(KEYED(missingcity=UpperIt(city_in)),WILD(PrimaryFIPS),WILD(missingstate)),
11             NCMECKey(KEYED(missingcity=UpperIt(city_in)),KEYED(missingstate=UpperIt(st_in)),WILD(PrimaryFIPS)));
12
13
14  EXPORT By_FIPS(UNSIGNED3 fips_in) := FUNCTION
15      RETURN NCMECKey(primaryFIPS=fips_in);
16
17 END;
```

Delivering your Results!

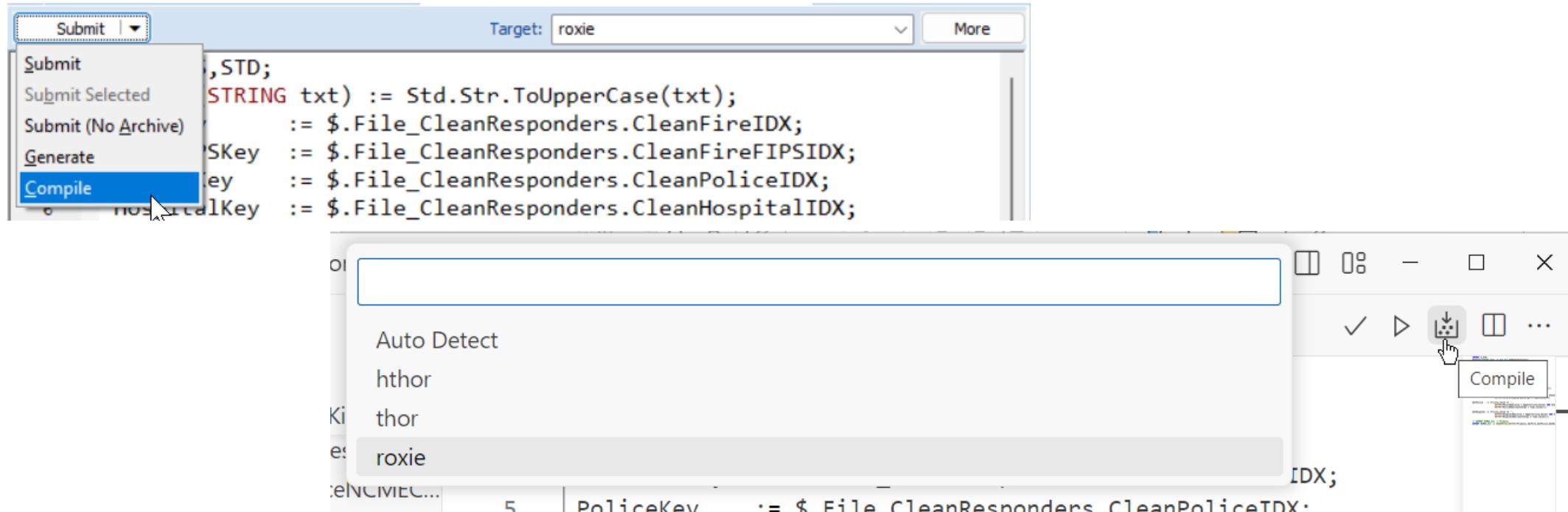
Step 8 - Design and Write Your Query:

```
1  IMPORT $,STD;
2  UpperIt(STRING txt) := Std.Str.ToUpperCase(txt);
3  FireKey      := $.File_CleanResponders.CleanFireIDX;
4  FireFIPSKey  := $.File_CleanResponders.CleanFireFIPSIDX;
5  PoliceKey    := $.File_CleanResponders.CleanPoliceIDX;
6  HospitalKey  := $.File_CleanResponders.CleanHospitalIDX;
7  UNSIGNED3 fips_value  := 0 : STORED('FIPS');|
8  STRING23 city_value  := '' : STORED('City');
9  STRING2 state_value  := '' : STORED('State');

10
11
12 Primary     := IF(fips_value <> 0,
13                     $.FN_FindKids.By_FIPS(fips_value),
14                     $.FN_FindKids.By_CityST(city_value,state_value));
15
16 GetFire      := IF(city_value <> '',
17                     OUTPUT(FireKey(City = UpperIt(city_Value) AND State = UpperIt(state_Value))),
18                     OUTPUT(FireFIPSKey(primaryfips = fips_value)));
19
20 GetPolice    := IF(city_value <> '',
21                     OUTPUT(PoliceKey(City = UpperIt(city_Value) AND State = UpperIt(state_Value) AND WILD(countyfips))),
22                     OUTPUT(PoliceKey(countyfips = fips_value)));
23
24 GetHospital  := IF(city_value <> '',
25                     OUTPUT(HospitalKey(City = UpperIt(city_Value) AND State = UpperIt(state_Value) AND WILD(countyfips))),
26                     OUTPUT(HospitalKey(countyfips = fips_value)));
27
28 // EXPORT NCMEC_Svc := Primary;
29 EXPORT NCMEC_Svc := SEQUENTIAL(OUTPUT(Primary),GetFire,GetPolice,GetHospital);
```

Delivering your Results!

Step 9 - Deploy(Publish) and then Test Your Query:



The screenshot shows the HPCC Systems IDE interface. At the top, there is a toolbar with a 'Submit' button, a 'Target' dropdown set to 'roxie', and a 'More' button. A context menu is open over some code, with the 'Compile' option highlighted in blue. Below the toolbar, there is a code editor window containing C++ code related to file cleaning. In the bottom right corner of the code editor, there is a toolbar with several icons, one of which is a downward arrow labeled 'Compile'. The cursor is hovering over this 'Compile' icon.

```
    ,STD;  
    STRING txt) := Std.Str.ToUpperCase(txt);  
    FireKey := $.File_CleanResponders.CleanFireIDX;  
    SKey := $.File_CleanResponders.CleanFireFIPSIDX;  
    PoliceKey := $.File_CleanResponders.CleanPoliceIDX;  
    HospitalKey := $.File_CleanResponders.CleanHospitalIDX;
```

OX

Auto Detect

hthor

thor

roxie

PoliceKey := \$.File_CleanResponders.CleanPoliceIDX;

Delivering your Results!

Step 9 - Deploy(Publish) and then Test Your Query:

The screenshot shows the HPCC Systems Job Management interface. At the top, there's a navigation bar with tabs for Variables (14), Outputs (7), Inputs, and Timers (14). Below the navigation bar is a toolbar with various actions: Refresh, Save, Delete, Restore, Reschedule, Deschedule, Set To Failed, Abort, Recover, Resubmit, Clone, Publish, Z.A.P, and Worker Logs. A file icon with a plus sign and a save icon are also present.

The main area displays a job entry for "W20240205-141019". The job details are as follows:

- Action:** compile
- State:** compiled
- Owner:** Rob

A modal dialog is open, titled "W20240205-141019". It contains the following configuration fields:

- Job Name:** Bob.NCMEC_Svc
- Remote Dali:** (empty field)
- Source Process:** (empty field)
- Comment:** (empty field)
- Priority:** None
- Allow Foreign Files:**
- Update Super Files:**

At the bottom right of the dialog is a "Submit" button. At the bottom left of the dialog are buttons for Clear, Copy, and Download. Below the dialog, there are tabs for Severity and Source, with Severity being the active tab.

Delivering your Results!

Step 9 - Deploy(Publish) and then Test Your Query:

Queries bob.ncmc_svc.1

Summary Errors/Status (1) Logical Files (5) Super Files Libraries Used (0) Summary Statistics Graphs (7) Resources Test Pages W20240205-141019

SOAP JSON WSDL Request Schema Response Schema Sample Request Sample Response Parameter XML Legacy Form Links

Reset |

bob.ncmc_svc.1 Response

Dataset: Result 1

primarytpe	missingdate	missingcity	recid	dateposted	firstname	lastname	currentage	datemissing	imp_rate	pov_pct	pop_pct	edu_high	contact	photolink	internal_tpos	
1	12021	FL	NAPLES	857694	20200819	WENDY	HUDAKOC	39	19981115	5.39	17.3	387509	22.51	COLLIER COUNTY SHERIFF'S OFFICE (FLORIDA) 1-239-793-9300	http://www.missingkids.org/poster/NCMC/857694/1	0
2	12021	FL	NAPLES	1216222	20140108	BRENDA	OVALLE	37	2004203	5.39	17.3	387509	22.51	COLLIER COUNTY SHERIFF'S OFFICE (FLORIDA) 1-239-252-9300	http://www.missingkids.org/poster/NCMC/1216222/1	0
3	12021	FL	NAPLES	1232511	20200824	GABRIELA	MEDINA	34	20060811	5.39	17.3	387509	22.51	COLLIER COUNTY SHERIFF'S OFFICE (FLORIDA) 1-239-774-4434	http://www.missingkids.org/poster/NCMC/1232511/1	0
4	12021	FL	NAPLES	2005754	20240103	BRIDGET	ALVARADO	15	20231110	5.39	17.3	387509	22.51	COLLIER COUNTY SHERIFF'S OFFICE (FLORIDA) 1-239-252-9300	http://www.missingkids.org/poster/NCMC/2005754/1	0

Dataset: Result 2

	city	state	name	addressbuildingname	address	primarytpe	zipcode	internal_tpos
1	NAPLES	FL	BIG CORKSCREW ISLAND FIRE CONTROL AND RESCUE DISTRICT STATION 10		13240 IMMOKALEE ROAD	12021	34120	0
2	NAPLES	FL	BIG CORKSCREW ISLAND FIRE CONTROL AND RESCUE DISTRICT STATION 12		21520 IMMOKALEE ROAD	12021	34120	0
3	NAPLES	FL	CITY OF NAPLES FIRE DEPARTMENT STATION 2		977 26TH AVENUE NORTH	12021	34103	0
4	NAPLES	FL	CITY OF NAPLES FIRE DEPARTMENT STATION 3		300 CITATION POINT	12021	34104	0
5	NAPLES	FL	EAST NAPLES FIRE CONTROL AND RESCUE DISTRICT STATION 21		11121 EAST TAMiami TRAIL	12021	34113	0
6	NAPLES	FL	EAST NAPLES FIRE CONTROL AND RESCUE DISTRICT STATION 23		6055 COLLIER BOULEVARD	12021	34114	0
7	NAPLES	FL	EAST NAPLES FIRE DEPARTMENT CONTROL AND RESCUE DISTRICT STATION 20 HEADQUARTERS		4798 DAVIS BOULEVARD	12021	34100	0
8	NAPLES	FL	EAST NAPLES FIRE DEPARTMENT CONTROL AND RESCUE DISTRICT STATION 22		4375 BAYSHORE DRIVE	12021	34112	0
9	NAPLES	FL	GOLDEN GATE FIRE CONTROL AND RESCUE DISTRICT STATION 70		4741 GOLDEN GATE PARKWAY	12021	34116	0
10	NAPLES	FL	GOLDEN GATE FIRE CONTROL AND RESCUE DISTRICT STATION 72		3820 BECK BOULEVARD	12021	34114	0
11	NAPLES	FL	GOLDEN GATE FIRE DEPARTMENT CONTROL AND RESCUE DISTRICT STATION 71		100 13TH STREET SOUTH/WEST	12021	34117	0
12	NAPLES	FL	GOLDEN GATE FIRE DEPARTMENT CONTROL AND RESCUE DISTRICT STATION 73		14575 COLLIER BOULEVARD	12021	34119	0
13	NAPLES	FL	GREATER NAPLES FIRE AND RESCUE STATION 50		175 CAPRI BOULEVARD	12021	34113	0
14	NAPLES	FL	GREATER NAPLES FIRE RESCUE STATION 24		2795 AIRPORT PULLING ROAD NORTH	12021	34105	0
15	NAPLES	FL	GREATER NAPLES FIRE RESCUE STATION 61		525 NEWPORT DRIVE	12021	34114	0
16	NAPLES	FL	NAPLES FIRE DEPARTMENT STATION 1		835 8TH AVENUE SOUTH	12021	34102	0
17	NAPLES	FL	NORTH COLLIER FIRE CONTROL AND RESCUE DISTRICT STATION 47		2795 AIRPORT PULLING ROAD NORTH	12021	34105	0
18	NAPLES	FL	NORTH COLLIER FIRE DEPARTMENT CONTROL AND RESCUE DISTRICT STATION 45		1865 VETERANS PARK DRIVE	12021	34109	0
19	NAPLES	FL	NORTH COLLIER FIRE RESCUE DISTRICT - STATION NUMBER 48		16280 LIVINGSTON ROAD	12021	34110	0
20	NAPLES	FL	NORTH NAPLES FIRE CONTROL AND RESCUE DISTRICT STATION 40		1441 PINE RIDGE ROAD	12021	34109	0
21	NAPLES	FL	NORTH NAPLES FIRE CONTROL AND RESCUE DISTRICT STATION 42		7010 IMMOKALEE ROAD	12021	34119	0
22	NAPLES	FL	NORTH NAPLES FIRE CONTROL AND RESCUE DISTRICT STATION 46		3410 PINE RIDGE ROAD	12021	34109	0
23	NAPLES	FL	NORTH NAPLES FIRE CONTROL DISTRICT STATION 44		8970 HAMMOCK OAK DRIVE	12021	34108	0

Dataset: Result 4

countytpe	city	state	name	address	zip	zip4	telephone	type	status	population	county	country	latitude	longitude	internal_tpos	
1	12021	NAPLES	FL	COLLIER COUNTY SHERIFF / COLLIER COUNTY JAIL	3301 EAST TAMiami TRAIL, BUILDING J	34112	4902	(239) 793-9352	SHERIFF'S OFFICE	OPEN	1265	COLLIER	USA	26.1270613	-81.7656812	0
2	12021	NAPLES	FL	COLLIER COUNTY SHERIFFS OFFICE - DISTRICT 1 - NORTH NAPLES SUBSTATION	776 VANDERBILT BEACH PARKWAY	34108	8707	(239) 597-1607	SHERIFF'S OFFICE	OPEN	999	COLLIER	USA	26.2506766000001	-81.8056981999999	0
3	12021	NAPLES	FL	COLLIER COUNTY SHERIFFS OFFICE - DISTRICT 2 - GOLDEN GATE SUBSTATION	4707 GOLDEN GATE PARKWAY	34116	6901	(239) 455-3121	SHERIFF'S OFFICE	OPEN	999	COLLIER	USA	26.1831239000001	-81.7035825999999	0
4	12021	NAPLES	FL	COLLIER COUNTY SHERIFFS OFFICE - DISTRICT 3 - EAST NAPLES SUBSTATION	11121 TAMiami TRAIL EAST	34113	7753	(239) 793-9328	SHERIFF'S OFFICE	OPEN	999	COLLIER	USA	26.0907545000001	-81.7321468	0
5	12021	NAPLES	FL	COLLIER COUNTY SHERIFFS OFFICE - DISTRICT 4 - GOLDEN GATE ESTATES SUBSTATION	1195 NAPLES ROAD 858	34120	4482	(239) 304-3520	SHERIFF'S OFFICE	OPEN	999	COLLIER	USA	26.29030783	-81.5838641999999	0
6	12021	NAPLES	FL	COLLIER COUNTY SHERIFFS OFFICE - SPECIAL OPERATIONS CENTER - AVIATION	250 PATRIOT WAY	34104	3526	(239) 252-0301	SHERIFF'S OFFICE	OPEN	999	COLLIER	USA	26.1581269420001	-81.7783393	0
7	12021	NAPLES	FL	FLORIDA HIGHWAY PATROL TROOP F - NAPLES	3205 BECK BOULEVARD	34114	1201	(239) 354-2377	PRIMARY STATE AGENCY	OPEN	999	COLLIER	USA	26.1552853150001	-81.671261548	0
8	12021	NAPLES	FL	NAPLES POLICE DEPARTMENT	355 RIVERSIDE CIRCLE	34102	1404	(239) 213-4844	LOCAL POLICE DEPARTMENT	OPEN	176	COLLIER	USA	26.1518262000001	-81.7901421	0
9	12021	NAPLES	FL	SHERIFFS OFFICE-CRIME PREVENTION	2373 NORTH HORSESHOE DRIVE	34104	NOT AVAILABLE	(239) 793-9091	SHERIFF'S OFFICE	OPEN	999	COLLIER	USA	26.1634116000001	-81.7751189999999	0

Dataset: Result 6

countytpe	city	state	name	address	zip	zip4	telephone	type	status	population	county	country	latitude	longitude	internal_tpos	
1	12021	NAPLES	FL	LANDMARK HOSPITAL OF SOUTHWEST FLORIDA	1265 CREEKSIDE BLVD E	34109	0590	(239) 529-1800	GENERAL ACUTE CARE	OPEN	50	COLLIER	USA	26.270174	-81.788584	0
2	12021	NAPLES	FL	NAPLES COMMUNITY HOSPITAL	350 7TH ST N	34102	5754	(239) 624-4000	GENERAL ACUTE CARE	OPEN	391	COLLIER	USA	26.15064469	-81.7987429700001	0
3	12021	NAPLES	FL	NCH HEALTHCARE SYSTEM NORTH NAPLES HOSPITAL CAMPUS	11190 HEALTH/PARK BLVD	34110	NOT AVAILABLE	(239) 624-5000	GENERAL ACUTE CARE	OPEN	322	COLLIER	USA	26.2743069	-81.78874501	0
4	12021	NAPLES	FL	PHYSICIANS REGIONAL MEDICAL CENTER - COLLIER	6300 COLLIER BLVD	34114	NOT AVAILABLE	(239) 354-6001	GENERAL ACUTE CARE	OPEN	100	COLLIER	USA	26.10553542	-81.68488477	0
5	12021	NAPLES	FL	PHYSICIANS REGIONAL MEDICAL CENTER - PINE RIDGE	6101 PINE RIDGE RD	34119	NOT AVAILABLE	(239) 304-5145	GENERAL ACUTE CARE	OPEN	109	COLLIER	USA	26.21395287	-81.73268379	0
6	12021	NAPLES	FL	THE WILLOUGH AT NAPLES	9001 TAMiami TRL E	34113	NOT AVAILABLE	(239) 775-4500	PSYCHIATRIC	OPEN	67	COLLIER	USA	26.09602651	-81.73630976	0

Alternate Delivery: Visualization

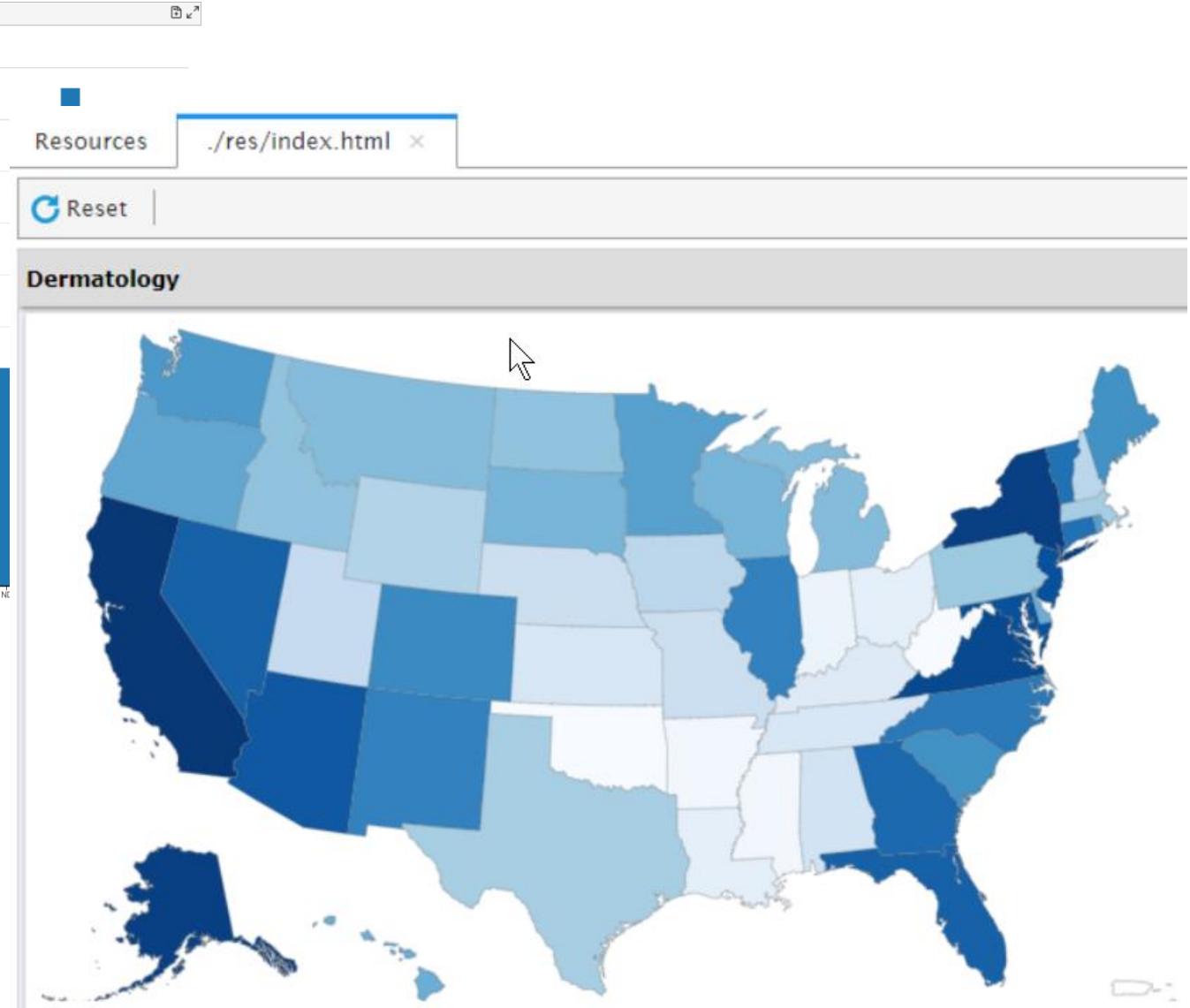
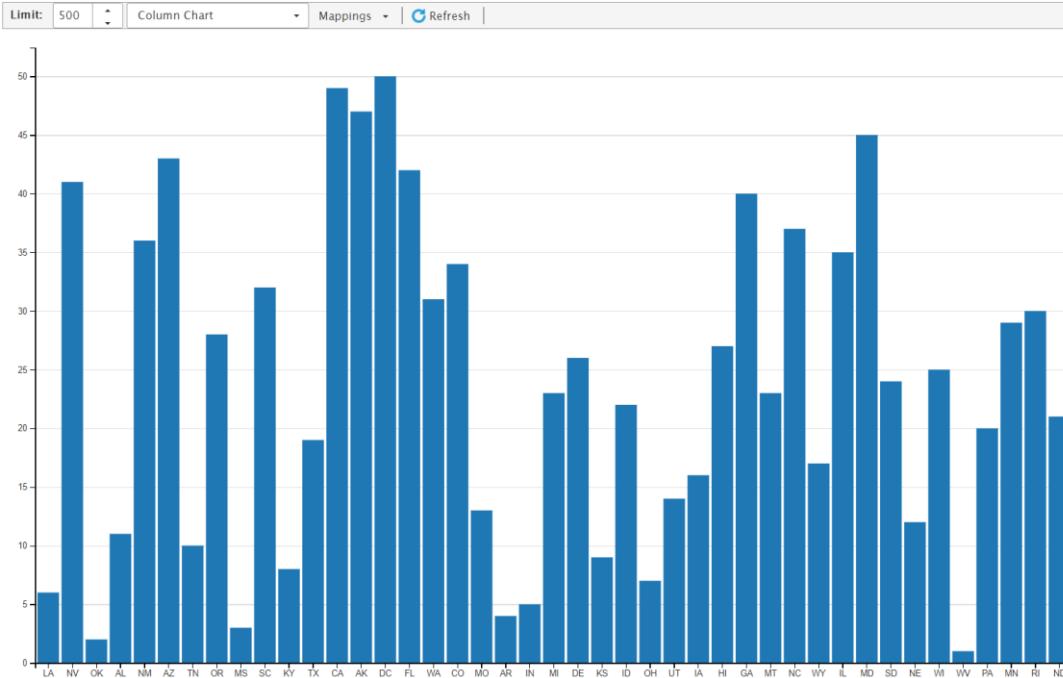
HPCC Systems provides built-in Visualization of your output data in a variety of charts and graphs. You can visualize your data in three ways:

- Using the Chart Tool in the ECL Playground.
- Accessing the Visualize tab in all ECL workunits
- Using the Resources tab in conjunction with the ECL Visualizer bundle.

Installing:

```
ecl bundle install https://github.com/hpcc-systems/Visualizer.git
```

Visualization Examples:



Final Thoughts

- ✓ Since your solution is the key part to this challenge you can use `#OPTION('obfuscateOutput', TRUE)`; at the start of your code to hide it from being viewed on ECL Watchpage. If you decide to use `#OPTION` make sure to remove it from the WUID that you shared with the judges. When `obfuscateOutput` set to true, details are removed from the generated workunit, including ECL code, estimates of record size, and number of records.
- ✓ If you want to write the result to a file, make sure the file name starts with your team's name for uniqueness purpose.
- ✓ Make sure the query names are unique and easy to identify. Do not use generic names like test, mentors, or roxie. We suggest adding your team's name as well. General names will result in other teams overwriting your files, queries, and results
- ✓ We encourage team play so teams that help answer questions in Slack will be considered favorably in judging.
- ✓ Direct emails and direct messages to judges asking for support will be *ignored* and it won't work in team's favor
- ✓ We also encourage students to leverage our community forum and/or StackOverflow for ECL coding related questions. Please make sure to tag your questions with `hpcc-ecl`.

The Resources!

KSU Challenge Wiki Page:

<https://wiki.hpccsystems.com/display/hpcc/Kennesaw+State+University+Hackathon+2024>

Learn ECL Academy

<https://hpccsystems-solutions-lab.github.io>

ECL Training containing six short videos

https://www.youtube.com/watch?time_continue=192&v=Lk78BCCtM-0

ECL Documentation

http://cdn.hpccsystems.com/releases/CE-Candidate-9.10.14/docs/EN_US/ECLLanguageReference_EN_US-9.10.14-2.pdf

Visualization Document

https://cdn.hpccsystems.com/releases/CE-Candidate-9.10.14/docs/EN_US/VisualizingECL_EN_US-9.10.14-2.pdf

Standard Library

https://cdn.hpccsystems.com/releases/CE-Candidate-9.10.14/docs/EN_US/ECLStandardLibraryReference_EN_US-9.10.14-2.pdf

Machine Learning

<https://hpccsystems.com/download/free-modules/machine-learning-library>

Get in Touch

Teams Channel

hackathons@hpccsystems.com

Robert.Foreman@lexisnexisrisk.com





HPCC
SYSTEMS