

# ECL Playground

Boca Raton Documentation Team



## ECL Playground

Boca Raton Documentation Team

Copyright © 2026 HPCC Systems®. All rights reserved

We welcome your comments and feedback about this document via email to <docfeedback@hpccsystems.com>

Please include **Documentation Feedback** in the subject line and reference the document name, page numbers, and current Version Number in the text of the message.

LexisNexis and the Knowledge Burst logo are registered trademarks of Reed Elsevier Properties Inc., used under license.

HPCC Systems® is a registered trademark of LexisNexis Risk Data Management Inc.

Other products, logos, and services may be trademarks or registered trademarks of their respective companies.

All names and example data used in this manual are fictitious. Any similarity to actual persons, living or dead, is purely coincidental.

2026 Version 10.2.10-1

|                                |   |
|--------------------------------|---|
| ECL Playground .....           | 4 |
| Using the ECL Playground ..... | 5 |

# ECL Playground

This manual covers the ECL Playground which is a component found in ECL Watch.

Figure 1. The ECL Playground in ECL Watch

The screenshot displays the ECL Watch ECL Playground interface. At the top, the ECL Watch logo and navigation icons are visible. The main area is divided into three sections:

- Code Editor:** Contains ECL code for defining a record, a dataset, and sorting it in various ways. The code is as follows:

```
1 MyRec := RECORD
2   STRING1 Value1;
3   STRING1 Value2;
4 END;
5
6 SomeFile := DATASET({{
7   { 'C', 'G' },
8   { 'C', 'C' },
9   { 'A', 'X' },
10  { 'B', 'G' },
11  { 'A', 'B' }
12 }, MyRec);
13 SortedRecs1 := SORT(SomeFile, Value1, Value2);
14 SortedRecs2 := SORT(SomeFile, -Value1, Value2);
15 SortedRecs3 := SORT(SomeFile, Value1, -Value2);
16 SortedRecs4 := SORT(SomeFile, -Value1, -Value2);
17 SortedRecs5 := SORT(SomeFile, Value2, Value1);
18 SortedRecs6 := SORT(SomeFile, -Value2, Value1);
19 SortedRecs7 := SORT(SomeFile, Value2, -Value1);
20 SortedRecs8 := SORT(SomeFile, -Value2, -Value1);
21 OUTPUT(SortedRecs1);
22 OUTPUT(SortedRecs2);
```
- Workflow Diagram:** Shows a visual representation of the execution process. It includes two 'Inline Dataset' nodes, two 'Sort' nodes (with parameters +300, -100), and two 'Output Result' nodes (Output Result #1 and Output Result #2). Arrows indicate the flow of data between these components.
- Data Table:** Displays the results of the execution. The table has columns for row number (##), value1, and value2. The data is as follows:

| ## | value1 | value2 |
|----|--------|--------|
| 1  | C      | C      |
| 2  | C      | G      |
| 3  | B      | G      |
| 4  | A      | B      |
| 5  | A      | X      |

Below the table, there are navigation controls including 'Refresh', 'Download' options (Zip, GZip, XLS, CSV), and a 'Filter' dropdown. The status bar at the bottom indicates '1 - 5 of 5 results' and shows tabs for 'Result 1' through 'Result 8', with 'Result 2' currently selected.

# Using the ECL Playground

ECL Playground is a tool hosted on an ESP server. A page runs in your browser, allowing you to access and execute self-contained ECL code on your HPC Systems platform without the use of any other tools. The ECL Playground then shows you the results and the graph in your browser. The view is very similar to what the ECL IDE displays.

## Accessing ECL Playground

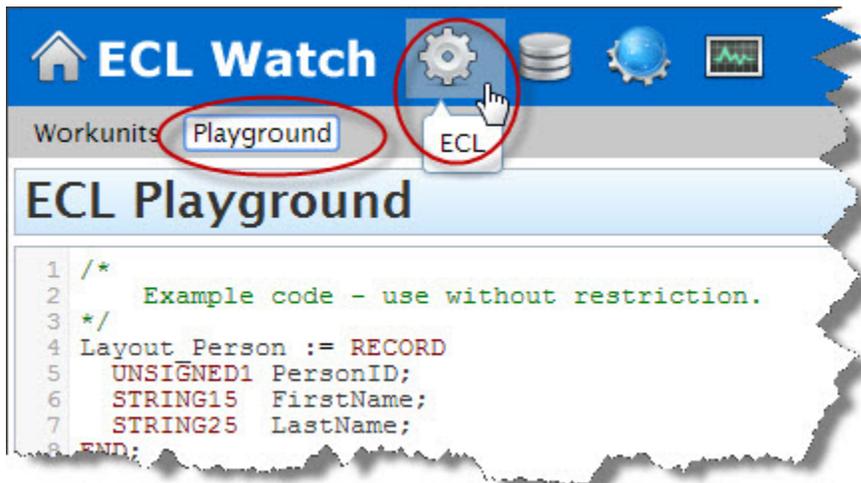
ECL Playground is installed with the HPC Systems platform. You can access it through the ECL Watch page.

1. In your browser, go to the **ECL Watch** URL. For example, `http://nnn.nnn.nnn.nnn:8010`, where `nnn.nnn.nnn.nnn` is your ESP server node's IP address.

|   |   |
|---|---|
|  | Your IP address could be different from the ones provided in the example images. Please use the IP address of <b>your</b> node. |
|---|---|

2. From ECL Watch, click on the **ECL** icon, then click the **Playground** link from the navigation sub-menu.

**Figure 2. ECL Playground link**

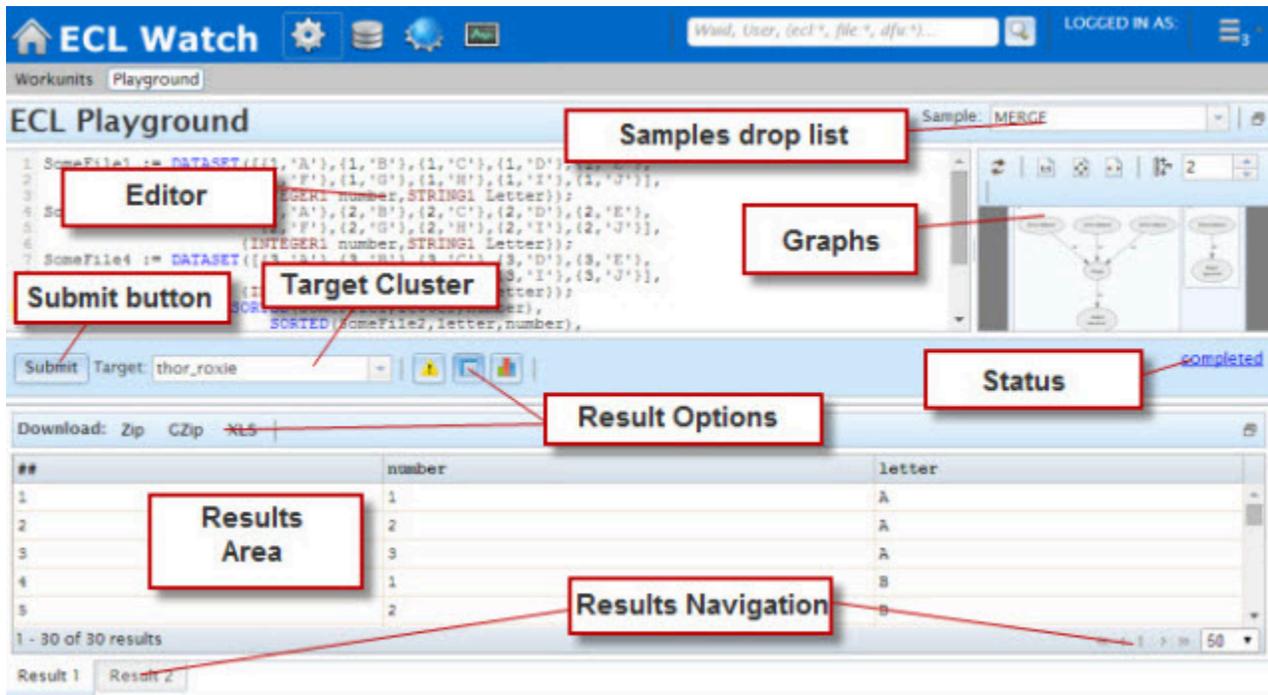


The ECL Playground displays.

## Introducing the ECL Playground

The ECL Playground page is a work area where you can see and run self-contained ECL code. You can see the code, submit it, and see the results. You can even change the code and resubmit it to instantly see the new results right in your browser. This is an ideal tool for the user who is not an ECL programming expert who wants to change some of the ECL code and see the results.

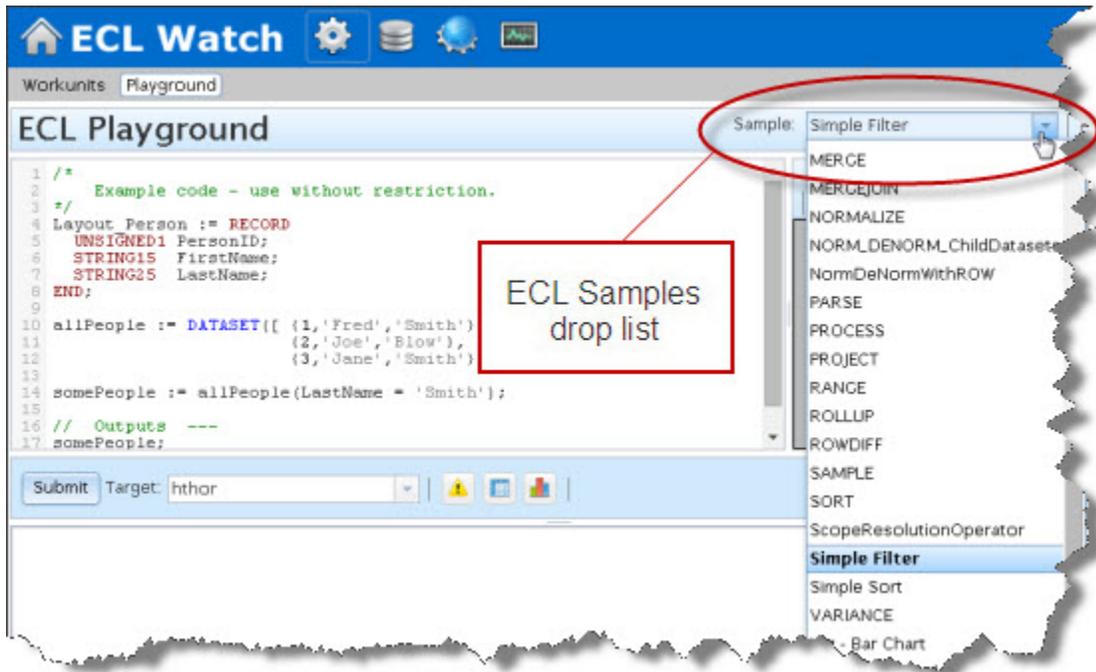
**Figure 3. The ECL Playground**



The ECL Playground page is divided into areas. The top portion contains the *Editor* area and the *Graph Viewer*. The Sample code drop list is at the top right. The bottom portion of the page displays the results.

The ECL Playground comes with a set of ready to run sample ECL code. The drop list contains code samples. Select any one of these samples and it loads in the editor.

Figure 4. Sample drop list



The selected code displays in the *Editor* area. You can then submit it as-is, or modify and submit. The results display at the bottom portion of the page.

## Running ECL Code

To run the selected sample code, choose a target cluster from the drop list, then press the **Submit** button.

A successful run displays the word **completed** as the status and the results display in the results viewer. You can also view the graph in the upper right.

Figure 5. Success

The screenshot shows the ECL Watch Playground interface. At the top, there's a blue header with the ECL Watch logo and navigation icons. Below that, the 'Workunits' tab is set to 'Playground'. The main area is titled 'ECL Playground' and shows a sample named 'JOIN\_dupes'. The code editor contains the following ECL code:

```
1 set1 := [1,2,3,4,5,6,7,8,9,10];
2 set2 := [10,20,30,40,50,60,70,80,90,100];
3
4 r1 := {integer1 fred};
5 r2 := {integer1 fred,integer1 sue};
6 ds1 := dataset(set1,r1);
7
8 ds2 := dataset(set2,r1);
9
10 r2 XF(ds1 L, ds2 R) := transform
11   self.fred := L.fred;
12   self.sue := R.fred;
13 end;
```

To the right of the code editor is a graph visualization showing a flow from 'ds1' and 'ds2' through a 'Join' node to a 'Result' node. A red box labeled 'Graph' points to this visualization. Below the code editor, there's a 'Submit' button (circled in red) and a 'completed' status indicator (circled in red). The interface also shows download options (Zip, CZip, XLS) and a table of results.

| ## | fred | sue |
|----|------|-----|
| 1  | 1    | 10  |
| 2  | 1    | 20  |
| 3  | 1    | 30  |
| 4  | 1    | 40  |
| 5  | 1    | 50  |

The table shows 5 results out of 100. The 'Result 1' tab is selected at the bottom.

A completed job generates a graph. You can examine the graph in greater detail by double-clicking the graph to zoom in. You can also zoom in with the mouse wheel. A double-click on a blank area of the graph will zoom out. You can use the scroll bars on the border of the graph to navigate or you can drag the graph with your mouse.

Selecting a node in the graph highlights the relevant section of the code in the Editor. This is helpful in troubleshooting or modifying code since it shows you the code that corresponds to a node in the graph.

Figure 6. Error

The screenshot shows the ECL Playground interface. At the top, the title is "ECL Playground" and the sample name is "JOIN\_dupes". The code editor contains the following ECL code:

```
r2 := {integer1 fred, integer1 sue};  
ds1 := dataset(set1, r1);  
ds2 := dataset(set2, r14);  
XF(ds1 L, ds2 R) := transform  
  self.fred := L.fred;  
  self.sue := R.fred;  
end;  
:= JOIN(ds1, ds2, right.fred % 2 = 0, XF(left  
output(j)
```

The code editor has several lines highlighted in red, indicating errors. A box labeled "Error Indicators" points to these lines. Below the code editor is a "Submit" button and a "Target" dropdown set to "hthor". To the right of the submit button is a "failed" status indicator. Below the submit button is a table of error messages:

| Severity | Source | Code | Message                      | Col | Line | File Name |
|----------|--------|------|------------------------------|-----|------|-----------|
| Error    | ecfcc  | 2167 | Unknown identifier "r14"     | 21  | 8    | stdin:    |
| Error    | ecfcc  | 2167 | Unknown identifier "R"       | 18  | 10   | stdin:    |
| Error    | ecfcc  | 2025 | SELF not legal here          | 3   | 12   | stdin:    |
| Error    | ecfcc  | 2167 | Unknown identifier "sue"     | 8   | 12   | stdin:    |
| Error    | ecfcc  | 3002 | syntax error near "end"      | 1   | 13   | stdin:    |
| Error    | ecfcc  | 3002 | syntax error near "ds2": ... | 15  | 15   | stdin:    |

At the bottom of the interface, there are checkboxes for "Error(s)", "Warning(s)", and "Info", all of which are checked.

The status area displays the job status. If a job fails, errors display in the result viewer and the code is highlighted in red in the *Editor*. If there are warnings they are displayed in yellow.

## Analyze the results

When running ECL Code that has multiple results, each result is on a separate tab. Select a tab to see that set of results. You can also change number of results displayed or page through the results with the links at the bottom.

Figure 7. Multiple results

The screenshot shows the ECL Playground interface. At the top, the title is "ECL Playground" and the sample is "ENTH". The code editor contains the following ECL code:

```
1 SomeFile := DATASET({'A'},{'B'},{'C'},{'D'},{'E'},  
2                 {'F'},{'G'},{'H'},{'I'},{'J'},  
3                 {'K'},{'L'},{'M'},{'N'},{'O'},  
4                 {'P'},{'Q'},{'R'},{'S'},{'T'},  
5                 {'U'},{'V'},{'W'},{'X'},{'Y'},  
6                 (STRING1 Letter));  
7  
8 Set1 := ENTH(SomeFile,2,10,1);  
9 Set2 := ENTH(SomeFile,2,10,2);  
10 Set3 := ENTH(SomeFile,2,10,3);  
11 Set4 := ENTH(SomeFile,2,10,4);  
12 Set5 := ENTH(SomeFile,2,10,5);  
13
```

Below the code editor is a "Submit" button and a "Target" dropdown set to "hthor". The status is "completed".

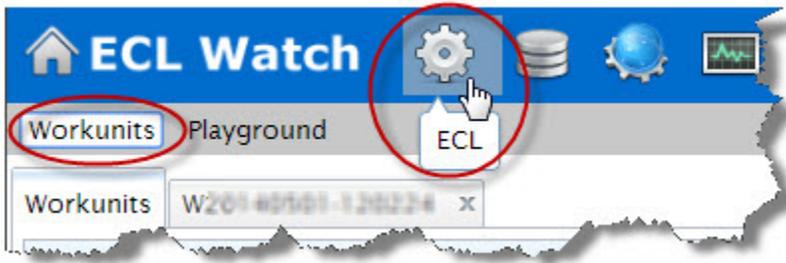
The results table is displayed below, with columns "##" and "letter". The table is empty, and the status is "1 - 5 of 5 results". A red box labeled "Results Navigation" points to the navigation controls at the bottom of the table, which include buttons for "Result 2", "Result 3", "Result 4", and "Result 5", and a pagination control showing "50" and navigation arrows.

## ECL from a Workunit

You can access ECL code from inside a Workunit Details page in ECL Watch.

1. Select **Workunits** from the ECL Watch  menu.

**Figure 8. Browse Workunits**



2. Click on a workunit hyperlink to open the Workunit Details page.
3. Click on the **ECL** tab to view the workunit's ECL code.

**Figure 9. ECL link**

