

ECL Playground

Boca Raton Documentation Team



ECL Playground

Boca Raton Documentation Team

Copyright © 2025 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.

2025 Version 9.12.16-1

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. The top navigation bar includes the ECL Watch logo, settings, database, and user information. The main area is titled "ECL Playground" and shows a sample of the "SORT" operation. The code editor on the left contains the following ECL script:

```
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'}], MyRec);
12 SortedRecs1 := SORT(SomeFile, Value1, Value2);
13 SortedRecs2 := SORT(SomeFile, -Value1, Value2);
14 SortedRecs3 := SORT(SomeFile, Value1, -Value2);
15 SortedRecs4 := SORT(SomeFile, -Value1, -Value2);
16 SortedRecs5 := SORT(SomeFile, Value2, Value1);
17 SortedRecs6 := SORT(SomeFile, -Value2, Value1);
18 SortedRecs7 := SORT(SomeFile, Value2, -Value1);
19 SortedRecs8 := SORT(SomeFile, -Value2, -Value1);
20
21 OUTPUT(SortedRecs1);
22 OUTPUT(SortedRecs2);
```

The workflow diagram on the right shows two parallel paths. Each path starts with an "Inline Dataset" block, followed by a "Sort" block with parameters "+300, -100", and ends with an "Output Result" block. The first path outputs to "Output Result #1" and the second to "Output Result #2".

Below the workflow diagram, there is a "Submit" button and a "Target" dropdown menu set to "thor". The status "completed" is displayed on the right.

The results table at the bottom shows the output of the first sort operation. It has columns for "##", "value1", and "value2". The table contains 5 rows of data:

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

The table indicates "1 - 5 of 5 results" and shows a pagination control set to 50.

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 HPCC 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 HPCC Systems platform. You can access it through the ECL Watch page.

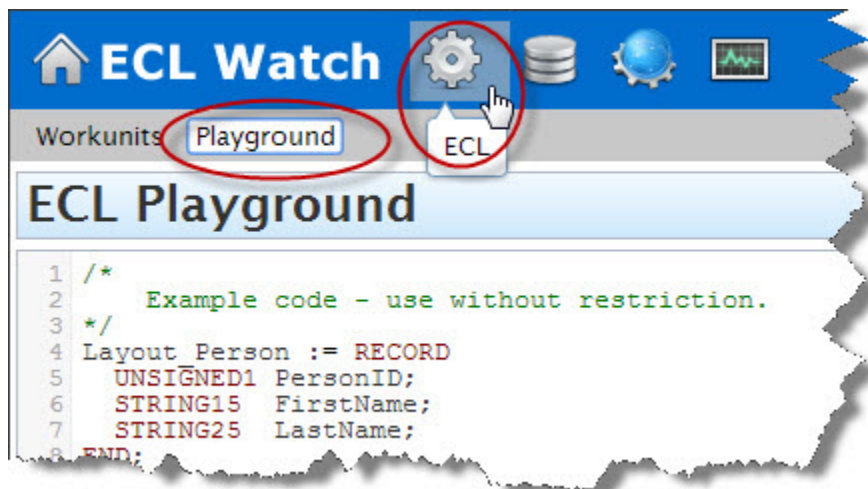
1. In your browser, go to the **ECL Watch** URL. For example, `http://nnn.nnn.nnn.nnn:8010`, where `nnn.nnn.nnn.nnn` is your ESP server node's IP address.



Your IP address could be different from the ones provided in the example images. Please use the IP address of **your** 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 it, a grey bar shows 'Workunits' and 'Playground' tabs. The main area is titled 'ECL Playground' and includes a 'Sample: JOIN_dupes' dropdown. On the left, a code editor displays ECL code for creating datasets and performing a transform. On the right, a graph visualization shows the execution flow. A red box labeled 'Graph' points to this area. Below the code editor, a 'Submit' button is circled in red, and a 'completed' status is circled in red on the right. At the bottom, a table shows the results of the job, with columns for row number, 'fred', and 'sue'. The table displays 5 rows of data, and a pagination bar indicates '1 - 50 of 100 results'.

```
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;
```

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

1 - 50 of 100 results

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 is "JOIN_dupes". The code editor contains the following ECL code:

```
r2 := {integer1 fred, integer1 sue};  
ds1 := dataset(set1, r1);  
ds2 := dataset(set2, r14);  
r2 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 a red box labeled "Error Indicators" pointing to the errors. The status bar at the bottom shows "Submit" and "Target: hthor". The result viewer on the right shows a "failed" status. Below the code editor is an error table:

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 error table, there are checkboxes for "Error(s)", "Warning(s)", and "Info".

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, there's a header with the title "ECL Playground" and a "Sample: ENTH" dropdown. Below the header is a code editor with the following 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: hthor" dropdown. To the right of the code editor is a sidebar with a tree view showing the execution plan. Below the submit button is a "Download: Zip GZip XLS" section and a "Filter" dropdown. The main area displays a table with the following columns: "##" and "letter". The table shows 5 rows of results. Below the table, there's a pagination control showing "1 - 5 of 5 results" and a set of buttons: "Result 2", "Result 3", "Result 4", and "Result 5". A red box labeled "Results Navigation" points to the pagination controls and the result tabs. The pagination controls include a set of arrows and a page number "50".

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

