

# Altair Breeze getting started

Version 2.14.0



---

## Contents

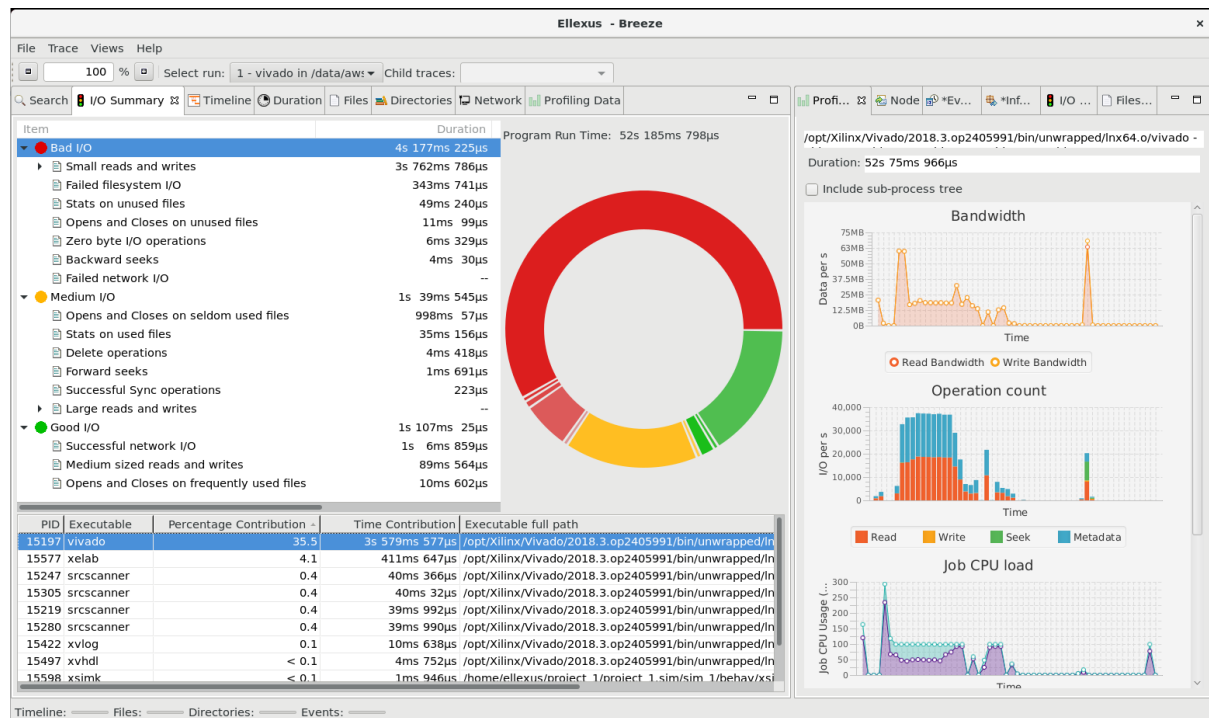
<b>1 Introduction and product overview</b>	<b>3</b>
1.1 Breeze: Detailed I/O profiling and dependency analysis . . . . .	3
1.2 Why profile I/O? . . . . .	3
1.3 Track application dependencies for good data hygiene . . . . .	3
1.4 Use Breeze to become fast, agile and cloud ready . . . . .	3
1.4.1 System administrators . . . . .	3
1.4.2 HPC user education . . . . .	4
1.4.3 Tool and system optimisation . . . . .	4
1.5 Breeze tells you: . . . . .	4
<b>2 Installation and getting started</b>	<b>5</b>
2.1 Download . . . . .	5
2.2 Generating a Breeze trace . . . . .	5
2.3 Licensing . . . . .	5
2.4 Opening the GUI . . . . .	5
2.5 Understanding the Breeze data . . . . .	5
2.6 Working with very big trace files . . . . .	6
2.7 Exporting Breeze data with the Breeze Automation Platform . . . . .	6
2.8 Exporting Breeze data with the Breeze Healthcheck Report . . . . .	6
2.9 MPI and jobs that run on multiple machines . . . . .	6
2.10 MPI I/O and parallel I/O libraries . . . . .	6
2.11 Benchmarking storage performance and troubleshooting . . . . .	6

# 1 Introduction and product overview

This guide will show you how to get started with Altair Breeze, I/O profiling tool. For more information about Breeze please refer to the demo videos and user manuals. Breeze was originally developed by Ellexus, now an Altair company.

## 1.1 Breeze: Detailed I/O profiling and dependency analysis

Breeze lets you solve software deployment problems and resolve file and network dependencies quickly by profiling applications in detail. With data for storage experts and summary reports for sharing, Breeze makes every engineer an I/O expert, identifying good and bad I/O patterns for easy wins.



## 1.2 Why profile I/O?

Even highly-optimised applications can have bad I/O patterns in deployment so it is a good idea to profile all applications, even when you can't change the source code. Problems such as storing files in the wrong place or trawling the PATH variable happen all the time and are easy to fix. By using Breeze and following our best practices, you will fix short-term problems, understand storage requirements and plan for the future. Breeze profiles application file I/O so you can ensure you are storing files in the right place.

## 1.3 Track application dependencies for good data hygiene

Breeze automatically detects which files, programs and libraries have been used by an application to create a bill of materials for each run. This can be used to solve deployment issues or to prove the correctness of a deployment.

## 1.4 Use Breeze to become fast, agile and cloud ready

Breeze gives you the information you need to solve deployment and performance problems quickly and tune your applications for your IT setup whether on-prem or in the cloud. It forms a common language for IT managers, users and software vendors to work together, taking the guesswork out of issue resolution.

### 1.4.1 System administrators

“Why does this application work for you and not for me?”

Deep dive into application arguments environment, libraries and dependencies to debug deployment and performance issues. Breeze gives you everything you always wanted from strace.

#### **1.4.2 HPC user education**

Make every engineer an I/O expert with an I/O healthcheck report. The Breeze Healthcheck reports give a straightforward I/O overview and performance recommendations based on common I/O issues.

#### **1.4.3 Tool and system optimisation**

"Why is my application slow today?"

Automate performance and regression testing with detailed I/O profiling. Export file and dependencies for application correctness and migration.

#### **1.5 Breeze tells you:**

For each file or process in your application - Which files and network locations your program has accessed - How much time has been spent in good and bad I/O - How much data has been read or written - How many meta-data operations such as move() or create() your program has performed - The full process tree with arguments and environment - ... and much more

## 2 Installation and getting started

This guide assumes you have access to the Breeze user manuals and documentation.

### 2.1 Download

Go to the [Ellexus Website Downloads](#) page and fill out the form. The site will send you an e-mail which you can use to download tar files of the products. To install, just untar the product on any filesystem which is accessible from the hosts running workloads which you wish to trace or profile.

### 2.2 Generating a Breeze trace

To trace or profile an application you use the `trace-program.sh` script in the download. This will save the Breeze trace to disk. You need to specify the output directory with the `-f` option and that directory must not already exist. You then just give the script the command and arguments you would normally run. If you want to run the application through a batch scheduler then the whole `trace-program` command should be given to the scheduler.

```
bsub /path/to/breeze/trace-program.sh -f /tmp/trace.out \
```

The trace can get quite large so we recommend that users put the trace on local storage if possible. We do have an option to relocate the trace at the end of the application.

### 2.3 Licensing

You do not need a license key to trace an application, but the trace is not useful without the licensed front end GUI and command-line applications.

This makes Breeze useful in a customer services context. It means that users can ask customers to generate and then send a Breeze trace for analysis. Cloud vendors can use it in customer service engagements when migrating workloads (this is not the most common use case but it is useful). It also means you can trace applications or machines that are on a completely different network from your license server.

**You do not need to worry about accessing the license server when tracing an application.**

Customers who just want to trace an application, and send the trace to someone else for analysis, can choose the smaller trace-only download from the [website](#) that does not include the GUI.

- See the guide for field application engineers for more information.

### 2.4 Opening the GUI

Once the application has finished you can open the trace in the Breeze GUI with `breeze . sh`. You will need to set the license location. Then you can import the trace from the files menu. The trace is a top level directory with many files and directories below. You want to choose the top level directory for import.

The UI runs on the Java 8 runtime. It's actually an OpenJDK build from Azure systems (Azul Zulu 8 with FX).

### 2.5 Understanding the Breeze data

The I/O summary shows when Breeze opens. This outlines time spent in good, medium and bad I/O. All of the views on the left of the GUI show data for the whole application. All of the views on the right show data for specific programs or files that have been selected on the left.

To find out more information about where an application is spending its time, the files view is a great resource. This lists every file that has been accessed, the I/O operations for that file and how much time was spent. For example, if your application is spending a lot of time opening files, you can see in the files view whether that time was spent opening different files or instead the same file was opened many times. Do this by expanding "other operations" and sorting the open latency column to see the time spent opening files.

The Profiling Data view shows I/O patterns over time so is another good way to understand how the I/O patterns of the application change over time.

- For more information take a look at the Breeze demo videos and user manuals.

## 2.6 Working with very big trace files

For large trace files Breeze will automatically switch to a faster import mode that drops some of the I/O data. You may also have to increase the amount of memory available to the Java virtual machine with the `--mem` option when Breeze GUI is started.

If the trace is too big to import into the GUI you can still export data via the Breeze Automation Platform command-line API and you may be able to generate the Breeze Healthcheck Report. The Breeze Automation Platform has several lightweight modes including the faster dependencies mode, which doesn't include any I/O data other than file dependencies.

If the application is too big to profile with Breeze then we recommend that users switch to profiling with Mistral instead.

## 2.7 Exporting Breeze data with the Breeze Automation Platform

The Breeze Automation Platform is the command-line API to Breeze. There are many options to export Breeze profiling data. The most useful mode is the dependencies export, which includes a list of every file and mount point used by the application.

- See the Breeze user manual for more information.
- See the [Ellexus HPCPros whitepaper presented at DAC](#) for details of how Qualcomm have built a bill-of-materials workflow using Breeze to automate their cloud migration projects.

## 2.8 Exporting Breeze data with the Breeze Healthcheck Report

The Breeze Healthcheck report is a static HTML report that is easy to share with colleagues to communicate information about an application. It contains a number of checks for bad I/O patterns in addition to the information found in Breeze, so is a great source of information for users who don't know where to start with the Breeze data. The report is exported via a command-line API.

- See the Breeze user manual for information on how to export the Breeze Healthcheck report.

## 2.9 MPI and jobs that run on multiple machines

Many jobs run across multiple machines using MPI libraries or using in-house scripts or micro-schedulers to launch child jobs on other machines using ssh or the job scheduler. Breeze will attempt to follow the application and trace the child job. Breeze will create a separate trace for every child job. The launch point for the child job will be shown in the timeline view and a list of all child jobs is visible in the drop-down at the top of the UI.

Currently it is not possible to view aggregated data in the Breeze GUI for jobs that span multiple machines and generate multiple traces. It is possible to generate some overview reports using Breeze AP.

## 2.10 MPI I/O and parallel I/O libraries

In Breeze, multiple ranks on a single machine will be aggregated into one trace. Each rank is displayed as a separate process. MPI ranks that run on other machines will have their own traces, grouped by machine.

Breeze tracks some MPI I/O calls to the file system directly and those can be seen in the MPI I/O view. Parallel I/O is tracked alongside the normal POSIX MPI calls.

MPI communications I/O between ranks in a single MPI job will show up in Breeze as Network I/O, shown in the Network View and Profile Data View.

- For more information on using Breeze on MPI applications see the Breeze user manual
- For information on optimizing MPI I/O see the [DL\\_POLY white paper](#) that we wrote with the Hartree Centre.

## 2.11 Benchmarking storage performance and troubleshooting

Breeze contains detailed I/O performance metrics. The data is collected at the application level so it is not possible in general to say where the bottleneck is. However, it is often possible to diagnose complex I/O issues by looking at the pattern of I/O latency in the Profile Data View in Breeze.