The OCaml Platform v0.1

Anil Madhavapeddy, Amir Chaudhry, Thomas Gazagnaire[1],
David Sheets, Philippe Wang, Leo White and Jeremy Yallop
University of Cambridge, OCamlPro SAS[1]

The OCaml Platform combines the core OCaml compiler with a coherent set of tools, documentation, libraries and testing resources. It's a project led by the OCaml Labs group in Cambridge, working closely with OCamlPro in France, and the requirements of the Platform are being guided by the industrial OCaml Consortium (primarily Jane Street, Citrix and Lexifi to start with).

Overall ethos

OCaml Labs has initially taken direction from major industrial users because these groups have a great deal of experience of using the language at scale (e.g. millions of lines of code, thousands of directories with complicated testing and build requirements as well as the concomitant issues in training new developers).

We're of the pragmatic view that examining and solving the problems that such organizations have will ultimately result in a large and positive impact on the wider community through significantly improved tooling, libraries, and documentation.

For the Platform to be considered successful, it has to be a viable product for those heavy users of OCaml. This ensures that the Platform has a long-term future and that it has the ability to scale right from beginner through to industrial use-cases. We believe that having a Platform that can scale in this way also has benefits for establishing workflows and practices that help every community member.

The Platform will initially bundle OCaml 4.01.0 and a specific set of libraries and tools that provide:

- source-based package management via OPAM.
- testing via the OCamlot continuous integration suite, which works both on public and private code (e.g. within company firewalls).
- documentation with global cross-references across all packages via OPAM-doc.
- a coherent ocaml.org online website to glue all these resources together.

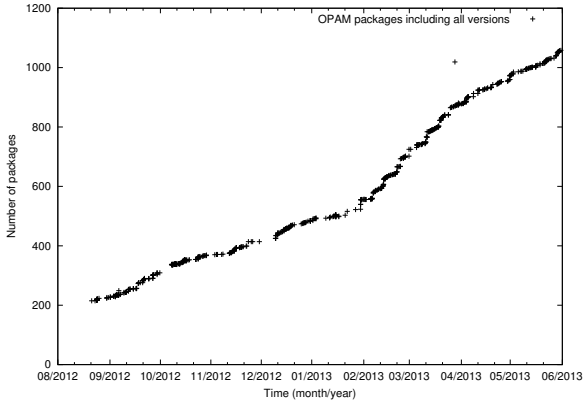

Figure 1: Growth in OPAM packages since we began a public package repository. Note that this plots all unique versions, since OPAM supports installing older revisions of a given package.

Creating and maintaining the Platform is an ambitious, long-term project. The v0.1 is not taking arbitrary decisions about which packages are included. Instead, we are first placing the industrial contributors on a common toolchain and workflow so that it's easier to work together and share code. We expect a standard set of libraries to emerge from consensus over time as this workflow rises in adoption.

OPAM

OPAM plays a key role in the tooling for the Platform by providing a frontend that can control a concurrently installed set of compiler versions and package sets. Since its public release, thousands of packages and revisions have been added to the repository (see Figure 1). An important feature of OPAM is that it tracks multiple revisions of a single package, thereby letting packages rely on older interfaces if they need to. It also supports multiple package repositories, letting companies blend the global stable package set with their internal revisions.

One additional benefit of OPAM's growth and vibrancy

is the capability to run tests over the complete package set to understand the ramifications of proposed compiler changes and any breaking effects they might have. This substantially improves the usage data available to anyone working on compiler modifications.

OCamlot

OCamlot, OCaml Online Testing system, is a continuous integration and quality control system using OPAM and GitHub[1].

This critical piece of infrastructure is the first networked service of the Platform under development. Such a system ensures that patches submitted to the OPAM repository are thoroughly tested on the variety of supported configurations, architectures, and systems before being merged. The resulting improved build and metadata quality in turn speeds up development on other aspects of the Platform through earlier error feedback.

GitHub also forms an important part of the toolchain and, as OPAM has shown, it is a fine tool for collaborative, open source projects. Developers submitting packages to the OPAM repository benefit from regular and extensive tests on every update of their package. In time, we plan to incorporate benchmarking and package constraint exploration to mechanically improve the quality of the OPAM metadata.

OCamlot is currently testing weekly pull requests from Jane Street (for their Core library suite) and Citrix (for their Xen open source releases). Each of these pull requests typically affects hundreds of other packages and is difficult to test by hand.

Opam-doc

The OCaml toolchain has shipped with the ocamldoc tool since 2002. ocamldoc runs over a single OCaml library and generates cross-referenced documentation. It supports a variety of outputs including Latex, HTML, PDF, and even manpages. However, it is starting to show its age for large, complex codebases such as Core and so we are developing a more scalable alternative for the Platform effort. This work is being led by Leo White.

Opam-doc consists of two separate commands:

- bin-doc is a replacement for the ocamldoc lexer (which extracts documentation from source code comments. It uses the OCaml-4.00+ facility for generating .cmt files that contain the typed AST, and generates .cmd files which contain the documentation information. By using a separate file from the AST, we leave open the possibility of having multiple language translations in the future. These .cmd and .cmdi files can be combined with the .cmt files to generate complete documentation directly from the output of the compiler. This command is intended to be temporary, and can be integrated into the upstream in the future.
- opam-doc takes a set of cmt and cmd files and outputs a single JSON representation of all the files. This JSON can then be post-processed (or directly rendered in Javascript) to create a single documentation repository that reliably links references across entire installations. Thus, the entire Platform can have one documentation source rather than having to search across packages.

The ultimate aim is to support the OCaml Platform with interactive tutorials embedded into interface documents by using the js_of_ocaml compiler. A preview of this is already available online at the "Try OCaml" website[2].

ocaml.org

Parallel to the Platform efforts we are also working with Ashish Agarwal and Christophe Troestler on redesigning and adding functionality to the main ocaml.org website. This involves coalescing the outputs from the previously mentioned projects (i.e. OPAM, documentation tools and testing results) and presenting them in a clear and navigable way via ocaml.org.

Summary

The OCaml Platform represents the best way for developers, both new and old, to write software in OCaml.

Our presentation at OCaml 2013 is only the beginning, and we will continue to advance the infrastructure to improve the experience and productivity of OCamlers everywhere.

[1] `http://amirchaudhry.com/wireframe-demos-for-ocamlorg`

[2] `http://try.ocamlpro.com`