Data Exploration through Dot-driven Development (Artifact)[*]

Tomas Petricek

**The Alan Turing Institute, London, UK
and Microsoft Research, Cambridge, UK**
`tomas@tomasp.net`

Abstract

This artifact presents The Gamma, a simple programming environment for data exploration that uses member access as the primary mechanism for constructing queries. The artifact consists of two parts. The user facing web-based component allows users to explore a simple dataset of Olympic medal winners while a back-end service provides the data and evaluates queries executed by the user.

The purpose of the artifact is to illustrate the pivot type provider, which provides a simple way for constructing queries in a object-based programming language equipped with member access. The pivot type provider can be use to construct new queries from code or using the user interface, but it also encourages the user to modify existing code.

1 Scope

The artifact is designed to provide a repeatble way of evaluating the user experience that is provided by the pivot type provider. As discussed in the paper, the purpose of the project is to simplify data exploration (by using a simple programming language), make it open and transparent (by revealing the source code) and encourage further data exploration (by making it easy to change the source code). The presented artifact supports this claim.

The artifact web page contains a number of examples that illustrate the pivot type provider. When the page loads, the examples are executed and the results appear. For each example, the web page shows the full source code using the pivot type provider together with several additional libraries not presented in the paper (such as `chart` and `table`), which are used for creating visualizations.

Clicking on the "Edit source code" button opens an editor where the code can be modified. The editor supports auto-completion, which is the key feature made possible by the type system of the language and the pivot type provider. The following three sections briefly describe three scenarios that illustrate interesting aspects of the artifact.

2   Content

The artifact package includes:

- a Docker image (`thegamma-ecoop17.tar`), which contains the necessary client-side and server-side components for running the artifact
- an F# source code (`thegamma-source.zip`) which contains the full source code for the client-side and server-side components behind the artifact

To make evaluating of the user interface easier, we provide a Docker container that can be used to host the system. The Docker container is based on the standard `fsharp` image supplied by the F# Software Foundation.

The included source code provides complete source code for the server-side components (running as F# code on the server) and client-side components (translated to JavaScript using the Fable compiler). When compiling code from scratch, several packages are downloaded prior to the compilation from standard .NET and JavaScript package repositories NuGet and npm.

3   Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS).

The presented artifact consists of a simple web server packaged as a Docker container. When executed the artifact can be viewed by navigating to `http://localhost:8889` as described in the artifact documentation. The artifact can be also obtained and started from Docker hub using:

```
docker run -p 8889:80 tomasp/thegamma-ecoop17
```

The tool implemented as part of the submitted paper is also available as a JavaScript library that is made available through the standard JavaScript package manager npm. The full project soruce code is also hosted on GitHub. The following links can be used to obtain the latest version:

- `https://github.com/the-gamma/thegamma-script`
- `https://npmjs.com/package/thegamma-script`

4   Tested platforms

The artifact is known to work on any platform running Docker version 17 using the Google Chrome browser. It has been specifically tested using Docker version 17.03.1-ce-win12 on the Windows operating system. 5 GB of free space on disk and at least 2 GB of free space in RAM are sufficient for running the artifact.

5   License

MIT (`https://opensource.org/licenses/MIT`)

6   MD5 sum of the artifact

1f68db3bcbd54433a62c9b54a35a5666

7   Size of the artifact

336 MB