# Brand Objects for Nominal Typing (Artifact)

**Timothy Jones, Michael Homer, and James Noble**

**Victoria University of Wellington**
**New Zealand**
{tim,mwh,kjx}@ecs.vuw.ac.nz

## Abstract

In Brand Objects for Nominal Typing, we describe an implementation of a branding system for both runtime and static types. This artifact provides the extended form of Hopper, an interpreter for the Grace programming language, and extra modules which define both the dynamic objects and the modular static type checker. The extra modules extend the existing structural type checker in the provided version of Hopper, and are capable of statically checking code which interacts with statically determinable declarations of brand objects, including singleton brand constructors, brand sums, and dynamic variables which are known to contain some brand value at runtime. The dynamic brand objects extend this behaviour to the runtime, enforcing non-static contracts and allowing runtime type testing.

**1998 ACM Subject Classification** D.3.3 Classes and objects
**Keywords and phrases** brands, types, structural, nominal, Grace
**Digital Object Identifier** 10.4230/DARTS.1.1.4
**Related Article** Timothy Jones, Michael Homer, and James Noble, "Brand Objects for Nominal Typing", in Proceedings of the 29th European Conference on Object-Oriented Programming (ECOOP 2015), LIPIcs, Vol. 37, pp. 198–221, 2015.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2015.198
**Related Conference** 29th European Conference on Object-Oriented Programming (ECOOP 2015), July 5–10, 2015, Prague, Czech Republic

## 1 Scope

The artifact provides the basic implementation of brands on top of Hopper, a prototype implementation of Grace, as described in the paper. Hopper can run Grace code using either just the dynamic or the dynamic and static behaviour of brands. The exception and singleton case studies in the paper are provided as extra Grace modules, whereas the AST case study is not relevant to the Hopper implementation and so is not present.

## 2 Content

The artifact package includes:
- an installation of Node.js and the required Node packages;
- a modified form of the Hopper interpreter;
- Grace modules implementing brands, static checking, exceptions, and tests;
- in-depth instructions on the contributions of the artifact, and instructions on running the provided tests or arbitrary code, provided as an `index.html` file.

The artifact is provided as a VirtualBox disk image with all of the necessary software preinstalled. The image uses Ubuntu 14.04 LTS (Trusty Tahr), Node.js 0.12.2, and the libraries listed in Hopper's `package.json` file. The interpreter and modules are provided as a Node Package Manager package in the `hopper` directory on the desktop, alongside `index.html`. The primary contribution of the artifact are the modules defined in `src/brands.grace` and `src/branded.grace`. The former defines the dynamic behaviour of the brand objects, including the `brand` constructor, and the latter exports the brand objects as a dialect and includes a static type checker for modules which opt into branding. The artifact also comes with a series of small test modules demonstrating the static checker's behaviour on both statically valid and invalid modules, and a script to execute the tests.

## 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 latest version of Hopper is available on Github: `https://github.com/zmthy/hopper`

## 4 Tested platforms

The artifact is known to work on any platform running Oracle VirtualBox version 4 (`https://www.virtualbox.org/`) with at least 2 GB of free space on disk and at least 1 GB of free space in RAM. Import the image using `File > Import Appliance`.

## 5 License

GPL-3.0+ (`https://www.gnu.org/licenses/gpl.html`)

## 6 MD5 sum of the artifact

10d101b645becf411e242d0a04b9e614

## 7 Size of the artifact

1.2 GB