

# **mashc**: The MaSH Precompiler

Andrew Rock

School of Information and Communication Technology

Griffith University

Nathan, Queensland, 4111, Australia

a.rock@griffith.edu.au

June 14, 2010

## **1 Introduction**

This document describes how to set up and use the **mashc** precompiler. The purpose of **mashc** is to read MaSH programs (in **.mash** files) and produce Java translations (**.java**).

## **2 Installation**

Once compiled, **mashc** may be placed in any directory that is in your shell's path variable so that it may be invoked on the command line.

To compile a MaSH program **mashc** needs to access the MaSH environment file (**.mashenv**) requested by that program. These should be kept in a separate directory or directories from the MaSH programs (**.mash**). To configure **mashc** so that it can find these directories, create a system environment variable **MASH\_ENV\_PATH** containing the paths of the directories to search in the order that you wish them to be searched. For example in my UNIX shell start-up file (for **tcsh**) I have something like:

```
setenv MASH_ENV_PATH .:~/lib/MaSH
```

For **bash** use:

```
export MASH_ENV_PATH=.:~/lib/MaSH
```

Note that with this configuration the current working directory (**.**) will be searched first. If **mashc** fails to find the requested environment in the directories specified in **MASH\_ENV\_PATH** or by command line arguments then it will try the current working directory if it hasn't already done so.

In the above UNIX example, paths are separated by colons. On Windows use a semicolon.

## 3 Invoking

The command:

```
mashc options... files...
```

processes each of the named *files* according to the given *options*. The files must have the extension `.mash`. Output is written to files with the same base name, with extension `.java`. The options are:

- `-e dir` – add *dir* to the list of directories to search for environment files (may be used multiple times).
- `+debug` – add debugging code (for debugging the MaSH program).
- `+search` – display progress as directories are searched for environment files (for debugging the compiler itself).
- `+echo` – echo the source code (for debugging the compiler itself).
- `+lex` – print the lexer results (for debugging the compiler itself).
- `+parse` – print the parser results (for debugging the compiler itself).
- `+st` – print the symbol table (for debugging the compiler itself).
- `+phase` – announce the compiler phases (for debugging the compiler itself).