

Setting up the DSPCAD Integrative Command-Line Environment: Setup Guide for DICE Version 1.2*

Shuvra S. Bhattacharyya, William Plishker, Kishan Sudusinghe,
Nimish Sane, George Zaki
Department of Electrical and Computer Engineering, and
Institute for Advanced Computer Studies
University of Maryland at College Park, USA
{`ssb`, `plishker`, `kishans`, `nsane`, `gzaki`}@umd.edu

April 3, 2015

This document provides instructions on setting up DICE, the DSPCAD Integrative Command Line Environment. The setup instructions provided here complement the resources provided on the DICE Project Website [1], and the DICE overview report [2]. The overview report provides general background on DICE, and is therefore useful as a preliminary orientation to the DICE package. However, it should be noted that some features and command names have changed since the publication of the overview report; for the most up-to-date information, one should consult other documentation available from the DICE Project Website [1].

1 Setup Instructions

This section provides information on installing DICE. The following steps outline the installation process.

1. Download the DICE package from [1], and unpack the archived download file `dice.tar.gz` (this will result in a singled directory called `dice`). Place this `dice` directory in the directory location where you want it to reside. This location is referred to in the remainder of this document as your *DICE installation directory*. For example, if one has placed the downloaded, unarchived `dice` directory in `/users/me/import/applications`, then the DICE installation directory is:

```
/users/me/import/applications/dice
```

2. Go to the parent directory of your `dice_user` directory (or the directory in which you want `dice_user` to be created if it does not yet exist). The `dice_user` directory is a directory in which user-specific files related to DICE are maintained. For further background about the `dice_user` directory, see the DICE overview report [2].
3. Run the following commands (replacing the right hand sides of the first two assignment statements appropriately for your architecture and DICE installation directory). The options for the `UXARCH` setting are `lin` (Linux), `macos` (Macintosh), `win` (Windows with Cygwin), and `sol` (Solaris).

*Technical Report DSPCAD-TR-2014-01, Maryland DSPCAD Research Group, Department of Electrical and Computer Engineering and Institute for Advanced Computer Studies, University of Maryland at College Park. Originally published on May 24, 2014, revised on April 3, 2015.

```
UXARCH=macos
UXDICE="/users/me/import/applications/dice"
source "$UXDICE"/setup/runme
```

4. If you want to run the three commands above through a script, you can use the following as the contents of your script (again with the appropriate replacements).

```
#!/usr/bin/env bash

set -a
UXARCH=macos
UXDICE="/users/me/import/applications/dice"
source "$UXDICE"/setup/runme
```

2 Startup Instructions

To use DICE in a given `Bash` session, one must first start up DICE within that session. Starting up involves loading necessary environment settings so that you can use all of the features in DICE.

To start up DICE, follow these steps:

1. Start a `Bash` shell.
2. `cd` to your DICE user directory (e.g., run `cd ~/dice_user`).
3. Run

```
source startup/dice_startup
```

IMPORTANT: The `dice_startup` file must be invoked from the `dice_user` directory — e.g., as opposed to running it as

```
source dicemin_startup
```

from `~/dice_user/startup` or running it from your home directory.

As a basic test of the startup process one can run the `dxversion` command, which takes no arguments, from the enclosing `Bash` session after DICE has been started up. If DICE has been properly set up and started up, the `dxversion` command should execute and produce a brief message on standard output that gives the version number and other basic background information for the corresponding installation of DICE.

3 Startup Script

The `Bash` script shown in Listing 1 provides a way to start up (or “load”) DICE conveniently just after starting a new `Bash` session. The comments in Listing 1 provide more details on how to use the script.

```

#
# This script can be placed in your home directory ... e.g., as bashrc-local.
#
# Then you can load dice upon starting the shell by running:
#
# . bashrc-local
#
# (. is a shorthand for invoking the unix "source" command)
#
# You need to update the argument to the pushd command to ensure that
# it finds the location of your dice_user directory.
#

#!/usr/bin/env bash

pushd ~/dice_user
source startup/dice_startup
popd

```

Listing 1: A Bash script for starting up DICE

4 Updating DICE

To update your installation of DICE to a new version (without resetting your `dice-user` directory), use the following steps.

1. Start up DICE (i.e., your currently-installed version of DICE).
2. Go (`cd`) to the directory that contains the new version of DICE.
3. Make sure that the downloaded version has been unarchived (extracted) — i.e., it should appear as the *subdirectory* `dice` (not as the *file* `dice.tar` or `dice.tar.gz`).
4. Run the following command:

```
dxupdate
```

5. This will install the new version of DICE, and save your previous version of DICE in `"$XTMP"/dice-bak.tar.gz`.
6. Exit all open DICE sessions and their enclosing Bash sessions.
7. Build the new version of DICE unless instructions with the new release have been provided indicating that a build is not needed for this update. If in doubt, build the new version of DICE. To build the new version of DICE, run the following command:

```
dxbuild
```

After working through these steps successfully, one can use the new version of DICE by starting up DICE in the usual way. For example, one can start by running the `dxversion` command (see Section 2) to check the version number or revision date of the installed version.

Acknowledgments

This work has been sponsored in part by the U. S. National Science Foundation, the Laboratory for Telecommunication Science, and the US Air Force Research Laboratory.

We are grateful also to the following people who have made valuable contributions to DICE, and earlier software components that have evolved into parts of DICE and `dicelang`: Bishnupriya Bhattacharya, Nitin Chandrachoodan, Chung-Ching Shen, Soujanya Kedilaya, and Robert Ricketts.

References

- [1] “DICE project website,” <http://www.ece.umd.edu/DSPCAD/projects/dice/dice.htm>.
- [2] S. S. Bhattacharyya, W. Plishker, C. Shen, N. Sane, and G. Zaki, “The DSPCAD integrative command line environment: Introduction to DICE version 1.1,” Tech. Rep. UMIACS-TR-2011-10, Institute for Advanced Computer Studies, University of Maryland at College Park, 2011, <http://drum.lib.umd.edu/handle/1903/11422>.