Installing Version 8 of Icon under VMS

Gregg M. Townsend
Sandra L. Miller

Department of Computer Science
The University of Arizona

General Notes

VMS Icon has been built and tested using a recent release of VMS as indicated in the status file.

The full Icon tape requires about 11000 blocks when copied to disk. The .EXE files required for running Icon take about 500 blocks, and the optional Icon program library occupies around 1400 blocks.

.OBJ and .EXE files are included on the distribution tape, so recompilation is usually not required. Complete source code and build procedures are included for optional use on systems having a C compiler. A total of approximately 12000 disk blocks are needed for a rebuild.

The Icon Source Code

The source code for the Icon system is in the public domain. You may use and copy this material freely. This extends to modifications, although any modified version of this system given to a third party should clearly identify your modifications as well as the original source.

The responsibility for the use of this material resides entirely with you. We make no warranty of any kind concerning this material, nor do we make any claim as to the suitability of Icon for any application.

Changes Made in This Release

The following features are new since Version 7.5:

- Serial numbers for structures.
- Functions for performing mathematical computations.
- Functions for getting variables from their names and vice versa.
- Support for arithmetic on integers of arbitrarily large magnitude.
- The invocation of functions and procedures with arguments from a list.
- Interfaces for calling C functions from Icon and calling an Icon program from C.
- Instrumentation of storage management.

These changes are described in more detail in an associated technical report [1].

Unloading the Distribution Tape

These instructions assume that your tape drive is named MT:. The drive on your system may have a different name, and if so you should adjust the instructions accordingly.

The Icon distribution tape is a BACKUP tape containing three levels of directories. To unload the tape, first set your default directory to a parent directory that can contain the Icon hierarchy. Copy Icon to disk using these commands:

```
$ MOUNT/FOREIGN MT:
$ BACKUP MT:ICON8 [...]
$ SET DEFAULT [.V8]
```

These commands create a subdirectory V8 containing the Icon system and make it the current directory. For convenience, subsequent sections refer to this directory as [V8], though its actual location is system dependent and it is not necessarily a top-level directory.

Check the current status of the VMS distribution, and the VMS version under which it was prepared, by entering

```
$ TYPE STATUS.
```

This command and all subsequent ones assume that V8 is the default directory. Note that if a command scripts fails, it may leave the default directory set incorrectly.

Configuring the Host Name

The value of &host seen by an Icon program is the logical symbol ICON_HOST if it exists; otherwise SYS$NODE if it exists; otherwise the string "VAX/VMS". Check to see if SYS$NODE is defined on your system by typing

```
$ SHOW LOGICAL SYS$NODE
```

If SYS$NODE is not defined (and you can't persuade your system administrator to define it), edit [V8.BIN]DEFICON.COM to supply a value for ICON_HOST. Insert

```
$ DEFINE/NOLOG ICON_HOST hostname::
```

immediately before the EXIT line.

Rebuilding Icon

The VMS distribution of Icon includes prebuilt binaries of the Icon system in [V8.BIN]. Relinking may be needed if your VMS system is an older version than the one indicated in the status file. Otherwise, this section can be skipped.

Object files included with the distribution allow relinking without requiring recompilation. This is accomplished using these commands:

```
$ SET VERIFY
$ @LINK
```

A complete rebuild is possible if your system has a C compiler. This takes about 15-30 minutes on a lightly loaded VAX 8600. The following commands recompile and relink the Icon system:

```
$ SET VERIFY
$ @MAKE
```

Testing the Installation

A few simple tests usually are sufficient to confirm that Icon is running properly. A suite of nine tests is provided for testing the Icon system. It is run by these commands:

```
$ SET NOVERIFY
$ @SAMPLES
```

Each program's name is printed as it is run. Unless problems are found, there is no other output.

Icon Program Library

The Icon program library [2] contains a variety of programs and procedures. This library not only is useful in its own right, but it provides numerous examples of programming techniques which may be helpful to novice Icon programmers. This optional component is available ready for use in [V8.IPL]. It may be rebuilt, if desired, by

```
$ SET NOVERIFY
$ @IPLMAKE
```

and/or tested by

```
$ SET NOVERIFY
$ @IPLTEST
```

The test procedure generates five DIFF outputs which should report no differences.

Icon Memory-Monitoring System

To build the memory-monitoring programs, do:

```
$ SET NOVERIFY
$ @MEMMONMAKE
```

For testing, do:

```
$ SET NOVERIFY
$ @MEMMONTEST
```

There will be differences in date lines and in some monitoring data because of different memory locations, but there should not be extensive differences.

Benchmarking

Test programs are provided for benchmarking Version 8 of Icon. To perform the benchmarks, do:

```
$ SET NOVERIFY
$ @BENCHMARK
```

See also the material in the subdirectory [V8.BENCH]. It contains a form, IPD116.DOC, that you can use to record your benchmarks with the Icon Project (see the section on Feedback).

Installation for General Use

The directory [V8.BIN] holds all files required for compilation and execution of Icon programs. This directory can be left in place or moved to a system area.

Separate documentation explains the use of Icon, but it is worth noting here that any use of Icon requires the execution of [V8.BIN]DEFICON.COM to define commands and logical names. A system administrator can incorporate DEFICON.COM into the system login procedure, making Icon available to all users automatically and relieving them of the need to run the definition script. If this is done the symbol ICON_BIN, defined in the script, should be set to reference the [V8.BIN] directory.

Cleaning Up

After Icon has been built and installed, type

```
$ @CLEAN
```

to remove duplicate versions, intermediate files, test results, etc.

Only [V8.BIN] is needed for running Icon programs. The library directory [V8.IPL] may be retained if desired. All other subdirectories of [V8] can be deleted unless you wish to keep the source code online.

Feedback

Please let us know of any problems you encounter (and your solutions) so that we may make things easier for others. Our mailing address is

Icon Project
Department of Computer Science
University of Arizona
Tucson, Arizona 85721
U.S.A.

(602) 621−4049

icon−project@cs.arizona.edu (Internet)
uunet!arizona!icon−project (UUCP)

Acknowledgement

Jan Trumbo helped improve the installation process.

References

1. R. E. Griswold, *Version 8 of Icon*, The Univ. of Arizona Tech. Rep. 90-1, 1990.
2. R. E. Griswold, *The Icon Program Library*, The Univ. of Arizona Tech. Rep. 90-7, 1990.