Installing Version 9 of Icon on UNIX Platforms

Gregg M. Townsend, Ralph E. Griswold, and Clinton L. Jeffery

Department of Computer Science
The University of Arizona
Tucson, Arizona

IPD243f
May 17, 1999
http://www.cs.arizona.edu/icon/docs/ipd243.htm

1. Introduction

Version 9 [1] is the current version of Icon, superseding Version 8. Version 9 contains new features and major changes to the implementation. This report provides the information necessary to install Version 9 of Icon on computers running UNIX.

The implementation of Icon is designed so that it can be installed, largely automatically, on a variety of UNIX platforms. This is accomplished by configuration information that tailors the installation to specific platforms.

The distribution contains configuration information for many UNIX platforms. These are listed in the appendix. Some of these originated under earlier versions of Icon. The platforms marked with an asterisk in the appendix have been tested since the major source code revision of Version 9.3.2. Installation on a tested platform should be routine, although minor configuration adjustments may be necessary for local conditions.

If there is configuration information for your platform, you may be able to install Icon without modification, but if problems show up, you may have to modify configuration files [2]. In some cases, there may be partial configuration information. If the configuration information for your platform is partial or lacking altogether, you still may be able to install Version 9 of Icon by providing the information yourself, using other configurations as guides.

If your platform is not listed in the appendix, it may have been added since this report was written. See Section 2 for information on how to check for a configuration for a specific platform.

2. The Installation Process

There are only a few steps needed to install Icon proper.

There are `Makefile` entries for most steps. Those steps are marked by asterisks. Steps that are optional are enclosed in brackets:

```
     1.    Decide where to unload Icon.
```

```
2.    Unload the Icon hierarchy at the selected place.
3*    Check the status of the configuration for your system.
4*    Configure the source code for your system.
5*    Compile Icon.
6*    Run simple tests.
[7*]  Run extensive tests.
[8*]  Run benchmarks.
[9.]  Install Icon at the desired place.
```

Step 1: Deciding Where to Unload Icon

You can build Icon at any place you wish. The executable binaries can be moved to another place later.

In the balance of this report, relative paths and the location of files are given with respect to the location at which the Icon hierarchy is unloaded. For example, a reference to `make` is with respect to the `Makefile` at the top level of this hierarchy.

Step 2: Unloading the Files

The distribution consists of a hierarchy, which is rooted in `"."`. Icon is distributed in a variety of formats. It requires about 20 MB of disk space when unloaded. The amount of space it takes to build Icon depends on the platform, what components are built, and whether intermediate files are deleted between building components.

If the root of the Icon hierarchy is icon, the resulting hierarchy should look like this after the distribution files are unloaded:

```
          |-bin------              executable binaries and support files
          |
          |-config---|-unix------  UNIX configuration directories
          |
          |          |-common----  common source
          |          |-h---------  header files
          |          |-iconc-----  Icon compiler source
|-icon----|-src------|-icont-----  Icon translator source
          |          |-preproc---  preprocessor source
          |          |-rtt-------  run-time translator source
          |          |-runtime---  run-time source
          |          |-xpm-------  XPM support
          |
          |          |-bench-----  benchmarks
          |          |-calling---  calling C functions
          |          |-general---  general tests
          |-tests----|-graphics--  graphics tests
                     |-samples---  sample programs
```

There are additional subdirectories that are not shown above.

Step 3: Checking the Status of the Configuration for Your Platform

Check the status of the configuration for your platform before attempting an installation; it may contain essential information. This can be done by

```
make Status name=name
```

where name is one of those given in the table in the appendix at the end of this report. For example,

```
make Status name=intel_linux
```

lists the status of the configuration for a PC running Linux.

In many cases, the status information was provided by the person who first installed Icon on the platform in question. The information may be obsolete and possibly inaccurate; use it as a guide only.

There are some configurations for which not all features of Icon are implemented. If the status information shows this for your platform, proceed with the installation, but you may wish to implement the missing features later. See Reference 2 for this.

Step 4: Configuring Icon for Your Platform

Configuring Icon creates several files for general use. Before starting the configuration, be sure your umask is set so that these files will be accessible.

There are two configuration possibilities: with or without graphics facilities.

To configure Icon without graphics facilities, do

```
make Configure name=name
```

where `name` is the name of your platform as described above. For example,

```
make Configure name=intel_linux
```

configures Version 9 of Icon for Linux, but without graphics facilities.

To configure Icon with the X Window System graphics facilities, use `X-Configure` instead of `Configure`, as in

```
make X-Configure name=intel_linux
```

Note: On some platforms, error exit codes from installation processes may be intercepted by `make` and result in warning messages. These messages can be safely ignored.

If you first configure without graphics facilities and later decide to add them, you will need to re-install Icon starting with this step.

If errors occur because the X include files or libraries are not found where they are expected, modify the appropriate files in the subdirectory of `config/unix` (see Reference 2) and restart from the make `X-Configure` step.

Step 5: Building the Icon Interpreter

Next, compile the Icon interpreter by

```
make Icon
```

There may be warning messages on some platforms, but there should be no fatal errors.

Step 6: Performing Simple Tests

If Icon compiles without apparent difficulty, a few simple tests usually are sufficient to confirm that Icon is running properly. The following does the job:

```
make Samples
```

This test compares local program output with the expected output. There should be no differences. If there are no differences, you presumably have a running installation of Icon.

Step 7: Extensive Testing

If you want to run more extensive tests, do

```
make Test
```

Some differences are to be expected, since tests include date, time, local host information, and platform-specific formats for floating-point numbers. In addition to `Test` there are some individual tests of optional features. See the main `Makefile` for more information about the tests.

To test Icon's graphic facilities, use `gpxtest.icn` in `test/graphics`. It should build and run without error, producing a window similar to the GIF image `gpxtest.gif` in the same area.

Step 8: Benchmarking

Programs are provided for benchmarking Version 9 of Icon. To perform the benchmarks, do

```
make Benchmark
```

See also the other material in the subdirectory `tests/bench`. It contains a form that you can use to record your benchmarks with the Icon Project (see Section 9).

Step 9: Installing Icon

The files needed to run Icon are placed in `bin` in the Icon hierarchy as the result of building the Icon interpreter:

```
icont Icon translator
iconx Icon interpreter
```

Some other files related to installing Icon and the optional components mentioned earlier also are placed in `bin`. The executable files needed to run Icon -- `icont` and `iconx` -- can be copied or moved to any desired place, and they need not be in the same directory.

Since `icont` must know the location of `iconx`, it is necessary to patch `icont` if `iconx` is moved. The program `patchstr`, also installed in `bin`, is provided for this purpose. It is used as follows:

```
patchstr icont-location iconx-location
```

For example, if `icont` is moved to `/usr/local/icont` and `iconx` is moved to `/usr/local/icon/iconx`, the patching step is

```
patchstr /usr/local/icont /usr/local/icon/iconx
```

Patching can be repeated if necessary. The patch value can be checked by using `patchstr` without a second argument, as in

```
patchstr /usr/local/icont
```

which prints the path to `iconx` in `/usr/local/icont`.

3. Cleaning Up

You can remove object files and test results by

```
make Clean
```

If you copied components of Icon to other places, you can delete the copies left in the Icon hierarchy.

You also can remove source files, but think twice about this, since source files maybe useful to persons studying or modifying the implementation. In addition, you can remove files related to the option components of the Icon system that you do not need. If you are tight on space, you may wish to remove documents as well.

4. Communicating with the Icon Project

If you run into problems with the installation of Version 9 of Icon, contact the Icon Project:

Icon Project
Department of Computer Science
The University of Arizona
P.O. Box 210077
Tucson, AZ 85721-0077
U.S.A.

(520) 621-6613 (voice)
(520) 621-4246 (fax)

icon-project@cs.arizona.edu

Please also let us know if you have any suggestions for improvements to the installation process or corrections or refinements to configuration information.

References

1. R. E. Griswold, C. L. Jeffery and G. M. Townsend, *Version 9.3 of the Icon Programming Language*, The Univ. of Arizona Icon Project Document IPD278, 1996.

2. G. M. Townsend, R. E. Griswold and C. L. Jeffery, *Configuring the Source Code for Version 9 of Icon*, The Univ. of Arizona Icon Project Document IPD238, 1995.

Appendix -- UNIX Icon Configurations

Configuration information for the platforms listed below is provided in Version 9 of Icon. Asterisks identify configurations that have been tested since Version 9.3.2.

	computer	operating system	name
*	Compaq/DEC Alpha	Linux	alpha_linux
*	Compaq/DEC Alpha	Tru64, Digital Unix, OSF/1	alpha_tru64
	Hewlett-Packard	HP-UX, cc	hp_risc
	Hewlett-Packard	HP-UX, Gnu C	hp_risc_gcc
*	Intel-based PC	BeOS	intel_beos
	Intel-based PC	BSD/OS	intel_bsdos
	Intel-based PC	FreeBSD	intel_freebsd
*	Intel-based PC	Linux	intel_linux
	Intel-based PC	Solaris	intel_solaris
	Intel-based PC	System V	intel_sysv
*	IBM RS/6000	AIX	ppc_aix
*	Silicon Graphics	Irix	sgi_irix
	Sun SPARC	Linux	sun_linux
*	Sun SPARC	Solaris 2.x, SunPro C	sun_sunc
*	Sun SPARC	Solaris 2.x, Gnu C	sun_gcc
*	Sun SPARC	Solaris 2.x, CenterLine C	sun_clcc

Icon home page