Installing the Icon Compiler

Ralph E. Griswold

Department of Computer Science, The University of Arizona

Introduction

The Icon compiler [1] is available in object format for a variety of UNIX platforms. Installation involves only a few simple steps:

(1) Obtaining a copy of the compiler for your platform.

(2) Unloading the distribution files.

(3) Editing a C header file that defines the path for support files that the compiler needs.

(4) Rebuilding the compiler so that it can find the support files.

(5) Performing some simple tests.

(6) Moving the compiler itself to the desired location.

Obtaining the Compiler

The Icon compiler is available via anonymous FTP to cs.arizona.edu. After getting connected,

```
cd /icon/v8/Compiler/Packages
```

In that area you will find subdirectories for the different platforms that presently are supported. For example, the subdirectory sun4 contains the Icon compiler in object format for the Sun 4 workstation.

Each subdirectory contains a README file. Get the one for your platform and read it before proceeding.

When you are ready to download the Icon compiler, set the binary (image) mode in FTP and get the compiler (named comp.cpio.Z or comp.tar.Z).

Next uncompress the file, as in

```
uncompress comp.cpio.Z
```

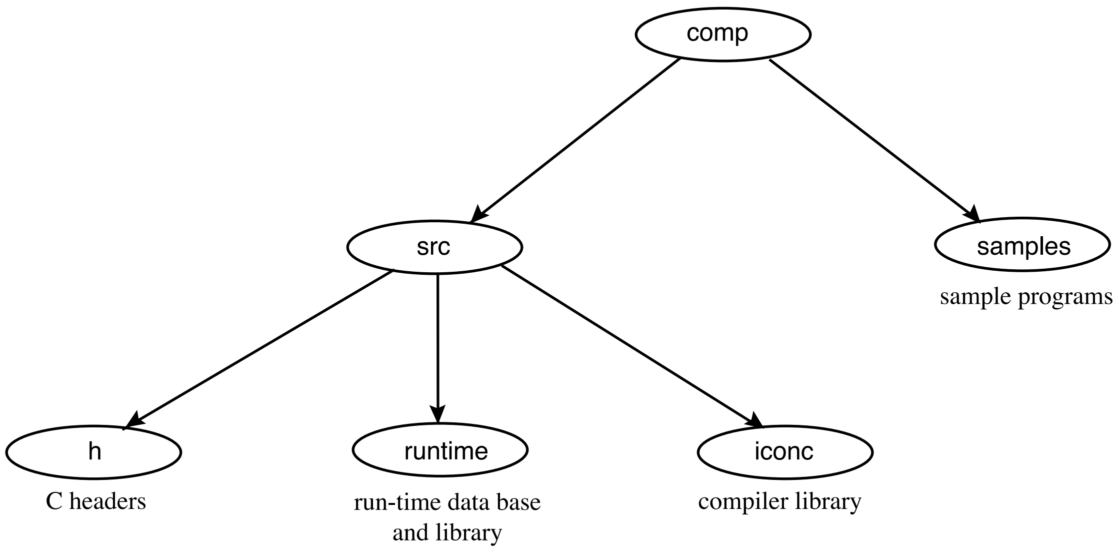
which leaves the cpio file comp.cpio.

Unloading the Distribution Files

As indicated above, the compiler must be able to locate support files, including a data base that contains information about Icon operations, C header files that are included by the C program produced by the Icon compiler, and an object library of routines needed during linking to produce an executable program.

The hierarchy containing these files should be unloaded at the location where they are expected to reside after installation of the compiler. (If this hierarchy is moved subsequently, it is necessary to rebuild the compiler itself.)

The compiler hierarchy looks like this if it is unloaded in comp:



If necessary, create the directory to hold the Icon compiler. Copy the cpio or tar file into that directory, and cd to it.

For a cpio file, do

```
cpio −icd <comp.cpio
```

For a tar file, do

```
tar xf comp.tar
```

Setting Up the Path to the Support System

The file src/h/paths.h in the distributed hierarchy contains a definition for RunTime. The default definition, as provided, is

```
#define RunTime "/usr/icon/comp/src"
```

where /usr/icon/comp is the default location of the distributed hierarchy.

Edit this definition so that it specifies the path to where you installed the src directory.

Building the Compiler

Next, at the top level of the Icon compiler hierarchy, do

```
make Iconc
```

This recompiles a small root file for the Icon compiler and links it with object-code files provided in the distribution to produce iconc, the Icon compiler, which is then placed at the top level of the Icon compiler hierarchy.

Testing the Icon Compiler

To be sure the Icon compiler is working properly, do

```
make Samples
```

This assumes iconc is in the top level of the Icon compiler hierarchy as a result of the preceding step.

The output of the tests should indicated sample programs are being compiled, run, and their output compared with the expected output. There should be no differences.

Moving the Icon Compiler

As indicated in the introduction, iconc can be placed anywhere. You may wish to move it to a place where it is publicly accessible.

Cleaning Up

To remove the byproducts of installing the Icon compiler, do

```
make Clean
```

in the top level of the Icon compiler hierarchy. If you do not expect to need the sample programs again, you can remove the samples directory. If you do not expect to reconfigure the Icon compiler again, you also can remove the src/iconc directory.

Trouble Shooting

The Icon compiler has been installed and tested on all the platforms for which the object distribution is provided. Nonetheless, the Icon compiler and its installation system are new and there may be problems.

If the Icon compiler doesn't install properly as described above or if it doesn't work at all, there may be object-code incompatibilities. Check the README at the top level of the Icon compiler hierarchy to make sure your computer and operating system are compatible with the version of the Icon compiler you are trying to install.

If the Icon compiler passes the sample tests but has problems with other Icon programs, check the user's guide [1] carefully. Some features of Icon are available only via command-line options to iconc, a few features of Icon are not available in the compiler, and the present Icon compiler corresponds approximately to Version 7.6 of Icon, while the current Icon interpreter is 8.0.

You may also have trouble with your C compiler; the Icon compiler has exposed previously unknown bugs in several C compilers that are in widespread production use.

Another possibility is lack of sufficient temporary file space during C compilation; not all C compilers handle this situation gracefully.

If you are unable to resolve the problem yourself, contact the Icon Project:

Icon Project
Department of Computer Science
Gould-Simpson Building
The University of Arizona
Tucson, AZ 85721
U.S.A.

(602) 621-8448

icon-compiler@cs.arizona.edu (Internet)
… {uunet, allegra, noao}!arizona!icon-compiler (uucp)

References

1. K. Walker, *Using the Icon Compiler*, The Univ. of Arizona Icon Project Document IPD157, 1991.