Compiling the Hydrological Simulation Program in Fortran (HSPF) from the Source Code

David J. Lampert

Introduction

The Hydrological Simulation Program in Fortran (HSPF) is a simulation package for watershed hydrology and water quality modeling. The model simulates movement of water and associated constituents on pervious and impervious land segments (PERLNDs and IMPLNDs) and in streams and reservoirs (RCHRESs). The HSPF model is derived from a series of predecessor models including the Hydrocomp Simulation Program (HSP) Model, the Non-Point Source (NPS) Model, the Agricultural Runoff Management (ARM) Model, and the Sediment and Radionuclides Transport (SERATRA) Model. HSPF was organized using a top-down approach designed to accommodate a variety of simulation modules (Bicknell 2005). HSPF is a data-driven model, and its structure is designed around manipulation of time series on HSPF-specific operations (PERLNDs, IMPLNDs, and RCHRESs). HSPF uses a time series-oriented, direct access data system to communicate the input data to HSPF for simulations.

HSPF has undergone a series of updates since its inception over 30 years ago. The original HSPF documentation and code were sponsored by the United States Environmental Protection Agency (EPA) and released in January 1980. Revisions and modifications were made and release a year later in January 1981. Versions 7 and 8 were developed under sponsorship from the EPA's laboratory in Athens, GA. Versions 10, 11, and 12 were developed with support from a variety of sponsors.

The latest version of HSPF (12.2) was released in 2005. The source code for HSPF 12.2 is hosted publicly (Aquaterra 2014). HSPF 12.2 was designed to work in conjunction with the EPA Better Assessment Integrated Point and Non-point Source Pollutants (BASINS), which is a Windows-based interface written in Visual Basic 6. The source code for HSPF12.2 contains build instructions that assume the user has a copy of the commercial Lahey Fortran compiler, but provides no specific details on the particular Fortran files needed to build the HSPF routine beyond opening a command prompt and typing "compile hspf122." Due to the lengthy history of updates and changes in the Fortran language, the latest releases contain many outdated utilities that needlessly complicate the compilation process. The purpose of this document is to describe the HSPF 12.2 source code and the minor modifications needed to build a library of subroutines that can be used to run HSPF and interact with HSPF input files using the free and open source GNU Fortran compiler (Stallman 2014).

Background

The source code for HSPF Version 11 is hosted on the USGS website (USGS 2014a). In HSPF Version 11, the utility subroutines that form the foundation of the HSPF source code are completely separated from the main HSPF program. These subroutines are housed in a group of libraries packed as LIB3.2 (USGS 2014b). HSPF 11 and LIB3.2 contain makefiles for statically-compiling the binaries on Unix or Unix-like operating systems. The LIB3.2 source can be modified for a variety of compilers and explicitly lists all the dependencies for the various HSPF modules and utilities.

Approach

The approach described herein was to use the makefile and documentation from HSPF 11 and LIB3.2 to build a shell script (for Unix-like systems) and batch file (for Windows) for HSPF 11 and then extend the source code to include the updates in HSPF 12.2 while remove interactions with Windows-specific programs. The resulting files together with the HSPF12.2 source code and modifications can be used to compile HSPF using the open source GNU compiler collection into an HSPF dynamic link library (DLL) on Windows or shared object (SO) library for Linux or Unix-like operating systems.

HSPF 11 and LIB3.2

The time series data for HSPF are organized using the Watershed Data Management (WDM) files. WDM files are unformatted (binary), direct-access, files having a fixed record length of 2048 bytes in both HSPF 11 and 12.2. The unformatted, direct-access nature of WDM files increases the speed of reading and writing data to the files. LIB3.2 contains many subroutines for opening, closing, reading, and writing to and from WDM files. WDM files can theoretically be used for time series, tables, text, vector and other data types, although the primary purpose for HSPF is time series.

LIB3.2 was designed to work with a variety of different software packages in addition to HSPF including ANNIE-IDE (AIDE), a system for developing interactive user interfaces for environmental models. AIDE has been used in the past to interact with WDM files both as input and output for HSPF. Because the primary purpose of this study was to build a library for HSPF with only essential utility routines (i.e., without the interactive features of AIDE), many of the non-essential subroutines were removed.

The following is a list and brief description of the subroutines in the different libraries in LIB3.2:

1. ADWDM – Utilities for reading and writing to WDM files essential to both models (i.e., HSPF) and interactive platforms (i.e., AIDE)
2. AIDE – Subroutines for building AIDE interfaces
3. ANN – Subroutines to perform algebra on datasets, modify table datasets, modify dataset attributes, manipulate time series, import and export datasets
4. AWSTAT – Subroutines to perform statistical analyses on datasets
5. GRAPH – Subroutines for drawing axes, curves, and symbols using Fortran Graphical Kernel System (GKS) graphics
6. HSPF – Primary subroutines related to hydrologic and water quality modeling for HSPF
7. HSPNODSS – Subroutines for reading and writing to United States Army Corps of Engineers' Data Storage System (DSS) files
8. NEWAQT – Subroutines to interact with geographic data, perform duration analysis, work with dates, get and put HSPF scenario information
9. STATS – Utility functions and subroutines for statistical analysis
10. UTIL – Primary utility routines for memory allocation, file path listing, terminal input/output, date and time, unit and other number conversions; operating system-specific options
11. WAIDE – Subroutines for interactive plotting and reading WDM data to and from buffers

12. WDIMEX – Program for importing and exporting datasets to WDM files (can be used to build the HSPF message file)
13. WDM – Primary subroutines for managing water resources (i.e., HSPF time series) data files (WDM files)
14. WDMRX – Program for restoring corrupt WDM files

An in-depth description of the details of the LIB3.2 sub-libraries is available elsewhere (Kittle et al., 1991).

To build WDM files for HSPF and perform HSPF simulations only the UTIL, ADWDM, WDM, HSPNODSS, and HSPF libraries are needed. The UTIL routines are the most fundamental (memory allocation, file path locating). The ADWDM routines build new utilities specific to WDM file format that which form the foundation for the HSPF-specific file utilities in the LIB3.2 WDM library. The HSPNODSS library utilizes only a few routines to allow HSPF simulations to read and write from DSS files. The primary HSPF subroutines including the essential hydrology and utility modules are located in the LIB3.2 HSPF library. The primary HSPF program in HSPF 11 is distributed separately from LIB3.2, and contains only a few subroutines used to interact with LIB3.2. Figure 1 shows the directory dependency structure for LIB3.2.

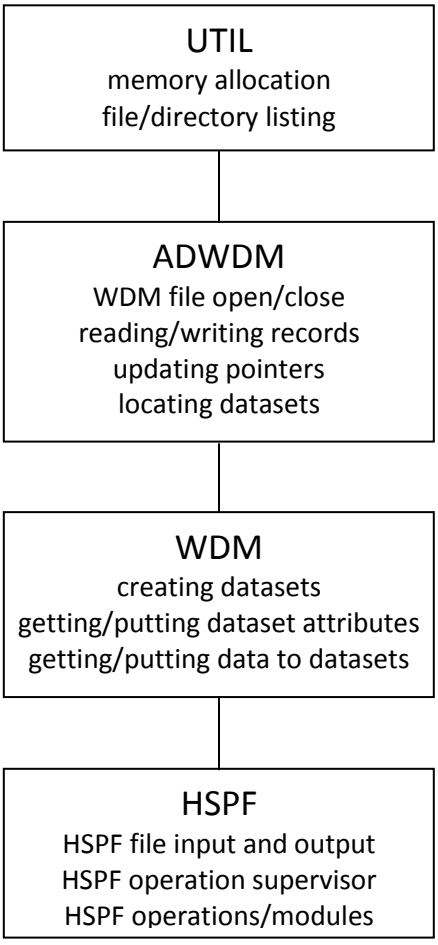

Figure 1. LIB 3.2 library for HSPF 11 dependency structure and basic functionality.

HSPF 12.2 Background

HSPF 12.2 contains most of the same code as version 11. A few additional utilities were added to the primary HSPF subroutines for the following functionality:

1. Wetlands and shallow water table hydrology additions to PERLND module
2. Irrigation capability additions to PERLND module
3. Simplified snow-melt algorithm to PERLND module
4. Best Management Practices module BMPRAC

Most of the other changes between HSPF 11 and HSPF 12.2 are designed to allow calls and interaction to the original Fortran 77 subroutines from Fortran 90 and Visual Basic 6. The other major change in HSPF 12.2 was to rename many of the all lower-case Fortran 77 files designed with .f extensions to all capital files with .FOR extensions.

HSPF12.2 Source Code Overview

The HSPF 12.2 source code is housed in a zip file "HSPF12.2SourceForBASINS4.0.zip" that unpacks into a primary directory with a series of subdirectories. The source code for HSPF is located in a LIB3.0 subdirectory with the structure shown in Figure 2.

HSPF12.2SourceForBASINS4.0
- lib3.0
  - src
    - ADWDM
    - AIDE
    - ANN
    - AWSTAT
    - GRAPH
    - hec
    - HSPDSS
    - hspf122
    - HSPNODSS
    - iowdm
    - Newaqt12
    - PRZM2
    - STATS
    - UTIL
    - WAIDE
    - Wdatfx
    - WDIMEX
    - WDM
    - WDMDSN
    - WDMRX

Figure 2. HSPF Version 12.2 source code directory structure (non-source code files are omitted).

As in the case of HSPF 11, most of the files are non-essential. The essential subroutines needed for HSPF are located in UTIL, ADWDM, WDM, hspf122, and HSPNODSS subdirectories. The files used from these directories are discussed below.

Source Code files from UTIL

The following is a list of the underlying C files in the UTIL directory used for memory allocation and file path location:

DIRLIS_.C
GETDIR_.C
MALLO~BA.C
FILEN~AI.C

Additional C files are located in UTIL for terminal input and output, but these are not essential to run HSPF. The UTIL directory also contains the following Fortran 77 files containing base subroutines and functions for date, character, and number, and array functionality:

UTCHAR.FOR
Utdate.FOR
UTGNRL.FOR
UTNUMB.FOR
UTSORT.FOR
UTCPGN.FOR

The UTIL directory has an operating system (OS) and compiler specific file for identifying the matching filenames meeting given search criteria (CKFSPC.FOR and CKFSDG.FOR). The CKFSDG.FOR was selected and found to compile successfully on both Windows OS and Linux OS with the Gfortran compiler.

Additional platform/OS dependent files are used to move the cursor and clear the screen (USCNVT.FOR and USCNUX.FOR). The USCNVT.FOR file was found to compile successfully on both Windows OS and Linux OS with Gfortran.

The UTSCXX.FOR file contains several subroutines that are used for terminal interaction that are part of the dependency structure of some of the HSPF source code files. The subroutines are not used directly by HSPF, but the compilation procedure was simplified by adding them to the library rather than trying to remove these interactions. The SCPRST, SCPRBF, COLSET, ZFMTWR, ZWRSCR subroutines were copied directly, while the SCPRBN subroutine was replaced with a dummy subroutine containing no commands.

The system date and time are used by HSPF during runtime. HSPF uses subroutines SYDATE and SYTIME in the UTIL folder to retrieve this information. However, the GNU Fortran compiler version of these subroutines takes the date and time as a single array argument for each function, whereas the subroutines in the UTIL folder use individual integer arguments for the year, month, day, hour, minute, and second. The calls in HSPF 12.2 are structured around this format, and so new subroutines were written for consistency with GFortran. An additional utility subroutine was introduced to fetch the system date and time that are different in Fortran 77 and Fortran 90 for HSPF 12.2. The Fortran 90

routines were an addition for HSPF12.2 versus older versions of HSPF. The Fortran 77 file (DATSYS77.FOR) was found to compile successfully on both Linux OS and Windows OS using Gfortran.

Source Code files from ADWDM and WDM

The ADWDM and WDM libraries in the HSPF source code contain utility subroutines to perform various operations on datasets located in WDM files. WDM files contain a series of datasets with a unique integer identifier, an array of data values, and a set of optional attributes that can be used to store the metadata for each dataset. The subroutines in ADWDM and WDM perform functions such as creating a new WDM file, finding dataset numbers in a file, checking the existence of dataset numbers in a file, retrieving and writing data and attributes to datasets, performing arithmetic, aggregation and disaggregation operations of data in a dataset, changing dataset attribute values, and renumbering and deleting datasets.

The makefile for the ADWDM library distributed with LIB3.2 compiles a number of platform-independent files and two OS/compiler-specific files. The directory also contains a number of Fortran include (.inc) files used to communicate optional parameter values such as maximum array sizes to the Fortran compiler. The default values of these two files compiled successfully on both Windows and Linux OS using Gfortran.

The developers of HSPF 12.2 re-named both the Fortran (.f) and the include (.inc) files using upper-case characters and changed the file extensions from .f to .FOR in the HSPF12.2 source code. The HSPF 12.2 versions of the files were found to compile successfully on both Windows and Linux OS using Gfortran after the file names were modified back to their former lower-case LIB3.2 names with the .f file extension. Failure to modify the file names was problematic in Linux (but not Windows) because Linux is case-sensitive and the changes to upper case were not made in the source code. The following are the Fortran files in the ADWDM library in the HSPF 12.2 source code needed to compile HSPF 12.2 on Windows or Linux using Gfortran (additional .inc files are needed for parameters but not listed):

UTWDMD.FOR
UTWDMF.FOR
utwdt1.FOR
WDMESS.FOR
WDATM1.FOR
MSIMPT.FOR
MSEXPT.FOR
MSOPEN.FOR
PRTFIL.FOR
ZTWDMF.FOR
ADUTIL.FOR
USYSUX.FOR
WDOPUX.FOR

As in the case of the ADWDM source files, the source files for the WDM library has modified filenames with all upper-case characters and .FOR extensions. The files for HSPF12.2 were compiled successfully when the file names were changed back to lower-case characters with the .f extension. The following is a list of files from the HSPF12.2 WDM library needed to compile HSPF on Windows OS or Linux OS using Gfortran (additional .inc files are needed for parameters but not listed):

WDBTCH.FOR
WDATRB.FOR
WDATM2.FOR
WDSPTM.FOR
WDIMPT.FOR
WDEXPT.FOR
WDTMS2.FOR
WDTMS1.FOR
WDTBLE.FOR
WDTBL2.FOR
WDDLG.FOR
WDATRU.FOR
TSBUFR.FOR
WDTMS3.FOR
WDMID.FOR

Source Code files from hspf122

The primary HSPF subroutine source code is located in the hspf122 directory. As in the case of the other libraries, to successfully compile the HSPF library required changing filenames to lower-case and changing extensions from .FOR to .f. The following is a list of unmodified files needed to compile HSPF (additional .inc files are needed for parameters but not listed):

HOSUPER.FOR	HDATUT.FOR	HGENUT.FOR	HIMP.FOR
himpgas.FOR	HIMPQUA.FOR	HIMPSLD.FOR	HIMPWAT.FOR
Hioosup.FOR	HIOOSV.FOR	HIOTSIN.FOR	HIOUCI.FOR
HIOWRK.FOR	HPER.FOR	HPERAGUT.FOR	HPERAIR.FOR
HPERGAS.FOR	HPERMST.FOR	HPERPES.FOR	HPERPHO.FOR
HPERQUA.FOR	HPERSED.FOR	HPERTMP.FOR	HPERTRA.FOR
HPERWAT.FOR	HPERNIT.FOR	HPERSNO.FOR	HPRBUT.FOR
HRCH.FOR	HRCHACI.FOR	HRCHCON.FOR	HRCHGQU.FOR
HRCHHTR.FOR	HRCHNUT.FOR	HRCHOXR.FOR	HRCHPHC.FOR
HRCHPLK.FOR	HRCHRQ.FOR	HRCHUT.FOR	HRCHHYD.FOR
HRCHSED.FOR	HRINGEN.FOR	HRINGEUT.FOR	HRINOPUT.FOR
HRINSEQ.FOR	HRINTS.FOR	HRUNUT.FOR	HRUNTSGP.FOR
HRUNTSGQ.FOR	HRUNTSGT.FOR	HRUNTSGW.FOR	HRUNTSUT.FOR
HRINTSS.FOR	HRINWDM.FOR	HRUNTSPT.FOR	HRUNTSPW.FOR
HTSINSI.FOR	HTSINSZ.FOR	HTSSUT.FOR	HUTOP.FOR
HUTOPINP.FOR	HWDMUT.FOR	HUTDURA.FOR	Specact.FOR
HSPF.FOR	HSPFEC.FOR	HSPFITAB.FOR	HRINOPN.FOR
HEXTUTIL.FOR	HBMPRAC.FOR	HREPORT.FOR	HIOSTA.FOR
HIRRIG.FOR	HRCHSHD.FOR	HPESTUT.FOR	

One additional file is needed from this directory, “HFILES.FOR.” This file contains a subroutine “FILBLK” that is used to process the HSPF files block in the UCI file. The subroutine contains an OPEN statement

that includes an “ACTION = ‘DENYONE’” specifier. This specifer resulted in an error during compilation with GNU Fortran and was removed.

Source Code from HSPNODSS

The HDSSX.FOR file contains subroutines that work with DSS files. This file is located in the HSPNODSS subdirectory and is needed for to compile HSPF 12.2.

Additional Needed Files

A few additional essential subroutines and programs for utilizing HSPF exist outside of the UTIL, ADWDM, WDM, and hspf122 directory. These subroutines and programs and the modifications needed to compile HSPF with the GNU compiler collection described above were placed into a new file (djl.f) for simplicity.

THE HSPFBAT.FOR file is located outside the LIB3.0 directory in HSPF 12.2 in the root subdirectory. In HSPF 11 this file was also located outside of the primary HSPF subroutines in LIB3.2. In HSPF 11, this file contains the HSPFBAT program that is called from the command prompt. HSPF uses a file called the User Control Input (UCI) file to communicate the commands for the simulation from the user to HSPF through the HSPFBAT program. The program prompts the user for the path to the UCI file using a system-specific subroutine and acts to open the files used during an HSPF run, pass control to the primary HSPF subroutine, and close the files used during the run. The HSPFBAT program from HSPF 11 was copied into the djl.f file and converted into a subroutine with two arguments, the path to the UCI file and the path to the HSPF message file. Previously the message file location was statically built into the program, which caused errors when calling HSPF from different locations. By adding the message file location and UCI file as arguments to a subroutine, it is possible to seamless call HSPF from outside Fortran. Three includes files are used by HSPFBAT to track the version number and Fortran file number for the message file, FVERSN.INC, VERSN.INC, and PMESFL.INC. The version information in these files was modified slightly to indicate that the compiled version of HSPF is version 13.

The HSPSTA.FOR file is located in the Newaqt12 directory in the HSPF 12.2 source and contains several subroutines that are used to show the status of an HSPF run. In HSPF 11, two versions of this file are available. One version is distributed with LIB3.2 in the newaqt subdirectory. The other version is distributed with the main HSPF 11 program. The two files for HSPF 11 are different, with the version in LIB3.2 utilizing arrays of length 4 for most of the local variables and different nomenclature than the version distributed with the HSPF program. The HSPF 12.2 version of this routine added a new argument IOPT into all the subroutines in the HSPSTA.FOR file. The IOPT parameter is used to position the output when called from Visual Basic 6 (thus deemed not essential for the purposes described herein). The HSPF 11 versions of the subroutines in the HSPSTA.FOR file were copied into djl.f along with other source code changes. These subroutines compiled with the other source code using the GNU compiler when the IOPT argument was added into the HSPSTA subroutine as a dummy parameter for consistency with the subroutine calls in HSPF 12.2. The subroutines include HSPSTA, HDMESC, HDMESI, HDMEST, HDMES2, HDMES3, HDMESN, CKUSER, and SDELAY.

The QTFILE.FOR file is located in the AIDE subdirectory. This file contains subroutines to read from the message files including the QFDPRS subroutine that parses the file name and directory from an input string. The CKFSDG.FOR file in the UTIL directory and all additional system-specific alternatives in UTIL contain an external call to QFDPRS in subroutine CKFSPC. The UTIL library is not supposed to depend on the AIDE library so this is likely an error in the HSPF source code that can be traced by to LIB3.2 and HSPF 11 at least. To avoid compiling the entire AIDE library, the QFDPRS subroutine was copied into the djl.f file.

Summary of Source Code Modifications

All modifications to the HSPF 12.2 source code described above were grouped together for documentation and clarity. The changes include very minor modifications to HFILES.FOR, Fversion.INC, PMESFL.INC, VERSN.INC and all the other changes lumped into the djl.f file. To summarize, the changes in djl.f include:

1. Modification of the primary HSPFBAT program to make it a subroutine
2. Removal of interactions with Visual Basic 6 in HSPF run status subroutines HSPSTA, HDMESC, HDMESI, HDMES2, HDMES3, and HDMESN
3. Re-writing the SYDATE and SYTIME date and time retrieval subroutines to interaction with the GNU complier routines
4. Copying subroutines from files that contained other frivolous subroutines that increased the dependencies including SYDATM, SCPRST, ZFMTWR, ZWFSCR, SCPRBF, COLSET, and QFDPRS
5. Creating dummy subroutines to remove interactions with the terminal and Visual Basic 6 including SCPRBN, CKUSER, UPDWIN, SDELAY, HSPF_INI, EXT_UPDATE, and LOG_MSG

The resulting files containing these changes were placed into a directory named “djl,” containing files HFILES.FOR, Fversn.INC, PMESFL.INC, VERSN.INC, and djl.f.

Compilation of HSPF using the GNU Compiler Collection

To compile HSPF 12.2, the files listed above in the UTIL, ADWDM, WDM, and hspf122 directories are needed in addition to the files in the HSPF13.zip archive. Attempts to compile on Linux Operating Systems demonstrated inconsistencies in the nomenclature used in HSPF 12.2 vs HSPF 11 and LIB3.2 files. Thus the file names for the Fortran source files and all include (“.inc”) files were changed to all lower-case with the “.f” extension as opposed to the “.FOR” extension. The GNU compiler collection utilizes different options depending on the file extension, and this method simplified the procedure the most. The primary HSPF Fortran routines from the LIB3.0/HSPF directory were copied into a new directory “hspf13.” The additional files from the HSPF13.zip archive were then placed into the folder. After performing these manipulations to the source, the HSPF 12.2 compilation is relatively trivial on both Windows and Unix-like operating systems.

Windows

The source C files can be compiled into objects using a single call to the GNU C Compiler:

gcc -ansi -O3 -c <C files>

Where <C files> are the C source files listed above in the UTIL directory. The ansi flag is needed to indicate that the code is consistent with the original 1989 C standard, the O3 flag tells the compiler to optimize for performance as much as possible, and the c flag tells the compiler to create objects from the individual files to be linked later.

The Fortran files can likewise be compiled into objects using a single call the the GNU Fortran compiler:

gfortran -c -O3 -fno-automatic -fno-align-commons <Fortran files>

Where <Fortran files> are all the Fortran source files listed in the UTIL, ADWDM, WDM, hspf122, and the djl directories. The fno-automatic flag was needed for HSPF 12.2 but not for HSPF 11. The GNU documentation indicates that this flag is needed to deal with missing SAVE statements. No attempts were made to find the missing SAVE statements, but in the future this issue should be investigating more thoroughly as it may impact performance. The fno-align-commons flag was also needed to successfully compile the Fortran objects. The GNU documentation indicates that this flag is needed when COMMON blocks are not declared with consistent data types. The source of this issue should be investigated to avoid potential alignment issues and to potentially increase performance. The c and O3 flags tell the compiler to create objects from the individual files for subsequent linking and to optimize the code as much as possible.

After compiling the c and Fortran code, the resulting objects can be linked together in a dynamic library on either Windows or Unix-like operating systems. On Windows, the result is a dynamic link library (DLL) using the following command:

gfortran -shared -O3 -mrtd -o hspflib.dll <object files>

Where <object files> are the object files (.o extension) generated from the source C and Fortran steps above. The O3 flag indicates maximum optimization. The shared and mrtd flags tell the compiler to make a DLL instead of an executable, and the o flag indicates that the name of the DLL should be hspflib.dll. The subroutines in hspflib.dll can then be called from other applications in Windows. The HSPFBAT subroutine can be linked to other applications and used to run HSPF simulations. The other subroutines are also needed to interact with HSPF's input and output data located in WDM files.

To facilitate the compilation process, a batch file name "compile_libhspf.txt" was written. The file contains instructions on how to use it that are summarized here. To use the batch file, the source code file HSPF12.2SourceForBASINS4.0.zip should be downloaded and extracted from the internet. The djl folder from the HSPF13.zip archive should then be extracted into the LIB3.0/src directory. The name of the compile_libhspf.txt file should then be changed to "compile_libhspf.bat" and placed into the LIB3.0/src directory to enable Windows to run the commands located in the file. The GNU compilers must be installed and available as environment variables. The batch file can be run from the command prompt. The commands create a new directory named "HSPF13," copy the needed files from the other directories, and then perform the commands described above to compile the libhspf.dll.

Unix-like Operating Systems

The source C files can be compiled into objects using a single call to the GNU C Compiler:

gcc -ansi -fPIC -c -O3 <C files>

The only difference from the command on Windows is the fPIC flag, which tells the compiler to generate file path independent code that is needed to building a shared library.

The source Fortran files can be compiled into objects using a single call to the GNU Fortran compiler:

gfortran -c -O3 -fno-automatic -fPIC <Fortran files>

The only differences from the command on Windows are the presence of the fPIC flag for file path independent code.

On Linux or other Unix-like systems, the following command can be used to compile the objects into a shared object (SO) library:

gfortran -u -shared -O3 -o libhspf.so <object files>

The u and shared flags are needed to build the library and the o flag tells the compiler to name the library libhspf.so. As in the case of Windows, the subroutines in the shared library can be called by other applications as needed.

To facilitate the compilation process, a shell-script was written named “compile_libhspf.sh.” The file contains instructions on how to use it that are summarized here. The source code should be downloaded from the internet and extracted. The shell script should then be placed in the LIB3.0/src directory along with the djl folder from the HSPF13.zip archive. The GNU Compilers should be installed and available as environment variables. Running the script creates a new directory “HSPF13,” copies the needed files from the other directories, and then compiles the libhspf.so library using the commands above.

Availability of HSPF File Modifications

The files described herein were then zipped into an “HSPF13.zip” archive that is publicly available in the “misc” directory of the PyHSPF repository at:

https://github.com/djlampert/PyHSPF

Important HSPF Library Subroutines

The following are some subroutines from the compiled HSPF library (on either Windows or Unix-like operating systems) that are particularly useful.

hspfbat: calls HSPF for a given UCI file name and message file path (hspfbat.f)
timdif: calculates the number of time steps between two dates (util/utdate.f)

wdbopn: opens a WDM file (adwdm/wdopux.f)
wdbsac: sets a character attribute for a dataset in a WDM file (wdm/wdatrb.f)
wdbsai: sets an integer attribute for a dataset in a WDM file (wdm/wdatrb.f)
wdbsar: sets a real attribute for a dataset in a WDM file (wdm/wdatrb.f)
wdbsgc: gets a character attribute for a dataset in a WDM file (wdm/wdbtch.f)
wdbsgi: gets an integer attribute for a dataset in a WDM file (wdm/wdbtch.f)
wdbsgr: gets a real attribute for a dataset in a WDM file (wdm/wdbtch.f)
wdckdt: checks if a dataset with the supplied number exists in a WDM file (adwdm/utwdmd.f)
wdflcl: closes a WDM file (adwdm/utwdmd.f)
wdlbax: creates a new dataset in the WDM file with the supplied dataset number (adwdm/wdmess.f)
wdtget: get time-series data from a WDM file with the supplied dataset number (wdm/wdtms1.f)
wdtput: write time-series data to a dataset in a WDM file (wdm/wdtms1.f)
wddsrn: renumber a dataset in a WDM file (wdm/wdbtch.f)
wddsdl: delete a dataset in a WDM file (adwdm/wdmess.f)
wddsnx: finds next dataset number in a WDM file (adwdm/utwdmd.f)

References

Bicknell et al., 2005, HSPF 12 User Manual

United States Geological Survey (USGS), 2014a, HSPF website, accessed 08/10/2014, http://water.usgs.gov/software/HSPF/

United States Geological Survey (USGS), 2014b, LIB website, accessed 08/10/2014, https://water.usgs.gov/software/LIB/

Aquaterra, HSPF public access website, accessed 08/10/2014, http://hspf.com/pub/hspf/HSPF12.2SourceForBASINS4.0.zip

Kittle, J., Flynn, K.M., Hummel, P.R., Lumb, A.M, United States Geological Survey (USGS), 1991, "Programmers Manual for the Watershed Data Management (WDM) System."

Stallman, R.M., 1999, "Using and porting the GNU compiler collection," Free Software Foundation.