The SAX WFC Data Analysis System

"How to make your SAX life even better"

— Software Installation Manual —

January 18, 1997

Written by	:	M. Savenije
Revised by	:	
Document ID	:	SRON/SAX/DA-001
Issue no.	:	$Revision: 1.2 $
Status	:	applicable

Revision history

Revision no.	Date	Author	Comments
1.0	1996/12/13	MS	
1.1	1996/12/17	MS	Comments from R. Jager
1.2	1996/12/18	MS	Comments from J. Schuurmans

Contents

1 Introduction **4**

2 SAX-WFC installation requirements **4**

3 Getting the software **4**

3.1 Pre installation tasks 4

3.2 FTP 5

4 Unzipping and untarring **6**

4.1 Full distribution 6

4.2 Update file 6

5 Configuring the Makefile **6**

6 Running the Makefile **7**

7 Running WFC programs and scripts **7**

A Example Makefile **9**

1 Introduction

This paper will lead you through the installation of SAX-WFC software on your computer system. It has been developed on HP workstations, running HP-UX 9.05. It is known to work on HP-UX 10, DEC Alpha's and DEC ULTRIX (stage II only).

Problems should be directed to: helpdesk@sax.sdc.asi.it.

2 SAX-WFC installation requirements

To compile and install WFC software you'll need a UNIX based computer system and the `PATH` should be set in a such way all compilers and UNIX utilities can be found, which is normally the case. Programs like `make`, `tar`, `gunzip` are supposed to be on your system and along the `PATH`.

In order to compile all software you need an ANSI C compiler (`gcc` version $\geq$ 2.6.3 is OK) and a FORTRAN 77 compiler. `gcc` is from the GNU Free Software Foundation and can be ftp'd from `ftp://prep.ai.mit.edu/pub/gnu`.

Stage I software also needs a FTOOLS implementation. We currently use version 3.4. Stage II software only needs the PGPLOT part of the FTOOLS distribution. FTOOLS is from NASA's HEASARC and can be ftp'd from `ftp://legacy.gsfc.nasa.gov/software/ftools`.

3 Getting the software

The official distribution point for all SAX software is the Software Data Center (SDC) in Rome, Italy. All relevant software and calibration components can be obtained by anonymous ftp. Also SRON keeps a mirror of the WFC software. The SDC ftp-address is: `www.sdc.asi.it` (193.204.215.42), and the directory is: `/pub/sax/software/wfc`. An up to date overview of the software and other necessary parts can be found at: `http://www.sdc.asi.it/software/index.html`.

Corresponding information about the SRON service: `purple.sron.ruu.nl` (131.211.40.23), and the directory is: `/sax/Dist`.

Full distributions can be identified by their name: `wfc_full-###.###.tar.gz`. Updates, depending on the contents of the update, can have names like: `wfc_stage2-###.tar.gz` if it concerns an update on the stage 2 software. The `###` corresponds to the release version of the file. If multiple versions are available choose the highest. As you can see from the suffixes the distributions (and updates) are gnu-zipped compressed tar-files.

3.1 Pre installation tasks

In case of a new and full installation of the WFC software a directory has to be created. This can be anywhere and with any name. Make sure there is at least approximately 25 MByte of free disk space left to create the necessary libraries, binaries and calibration files.

When updating existing WFC software you simply go to the main directory.

Let's say user Joe wants to install the WFC software. He already has a directory named `SAX` in his home directory, which is `/home/joe`. He wants to install the WFC software under directory `SAX`.

```
$ cd
$ cd SAX
$ mkdir WFC
$ cd WFC
```

3.2 FTP

Now the actual transfer from whether SDC or SRON can start. An example session to SRON:

```
$> ftp purple.sron.ruu.nl
Connected to purple.sron.ruu.nl.
220 purple FTP server (Version 1.7.111.1 Mon Apr 17 20:26:20 GMT 1995) ready.
Name (purple.sron.ruu.nl:joe): anonymous
331 Guest login ok, send ident as password.
Password: joe@some.institute (Not echoed)
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /sax/Dist
250 CWD command successful.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /usr/bin/ls.
total .....
.....
-rw-r--r--   1 sax      users    5653188 Oct 10 10:18 wfc_full-038.023.tar.gz
.....
226 Transfer complete.
ftp> bin
200 Type set to I.
ftp> get wfc_full-038.023.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for wfc_full-038.023.tar.gz (5653188 bytes).
226 Transfer complete.
5653188 bytes received in 5.66 seconds (975.94 Kbytes/s)
ftp> bye
221 Goodbye.
$
```

4 Unzipping and untarring

If everything went OK, we have `wfc_full-038.023.tar.gz` or any other full or update file in our directory. The main thing is that it is a gnu-zipped compressed tar file. Now depending on whether it is a full distribution or an update file you must do the following.

4.1 Full distribution

In case of full distribution we started with a new directory so we can unpack right away:

$ **gunzip wfc_full-038.023.tar.gz**

Which reveals file `wfc_full-038.023.tar` which can be untarred with:

$ **tar xf wfc_full-038.023.tar**

There are several ways to do the trick. Another is:

$ **gzcat wfc_full-038.023.tar.gz | tar xf -**

or

$ **gtar wfc_full-038.023.tar.gz**

4.2 Update file

In case of a software or calibrationfile update you should use `update` which can be find in the root-directory of the WFC software. The purpose of `update` is to modify the file attributes of the files mentioned in the update file. While source-files are shipped read-only, they must be changed to writable in order the update to succeed. So

$ **./update wfc_stage2-038**

Will do the job. Note `update` is called with the basename only! No suffixes should be given. The `update` script will supply them automatically. Beware the script uses `gunzip`, `chmod`, `tar` and `grep`. In case one of the programs cannot be found the update will fail. Also if your system does not have `gunzip` you will have to modify `update`.

5 Configuring the Makefile

This part is probably the most difficult. There is a script `configure` in the root of the WFC distribution which attempts to make a `Makefile` in order to make the various components of the WFC software. `configure` is **not** the well-known GNU configure but a self-written one.

The purpose of configure is to set values for the names of C and FORTRAN compilers, locations of libraries, etc. Also optimizing and/or debug flags should be set here.

For a number of machines and operating systems the information is present. As expected some settings are operating system specific and some are machine specific, e.g. the location of the FTOOLS directory. The idea is that you only edit `configure` and not the `Makefile`. So, in first instance try to run `configure` to see have far it gets in generating the `Makefile`, so type:

```
$ ./configure
```

If you get responses like 'Your hostname is not known to configure' or it complains it doesn't know your operating system you have to add one or more lines to the `configure` script. Just start an editor and load `configure`. As there are several machines and operating systems in the script you have enough examples how to add your system. The variables mentioned in table 1 should be set in `configure`.

In case your system required much effort in order to create a suitable `Makefile` you can send your `Makefile` to us and we will incorporate it in `configure`. See the E-mail address in the first section. Also in case of real trouble the E-mail address can be used for help.

`configure` will create the first part of `Makefile` by itself. It then appends `Makefile.ORIG` to it where machine and operating system independent information is stored. An example Makefile can be found in appendix A.

6 Running the Makefile

After successful creation of the `Makefile` you can use it to compile programs and to make the calibration database. The `Makefile` has several targets:

`all`, `stage1`, `stage2`, `<program>` Make everything, stage I, stage II, or individual program.

`idb` Makes calibration files (only works if `genFits`, `mask` and `instrument` are available.

`clean`, `clean1`, `clean2` Make clean.

Note that all libraries and separate programs are available as targets in the `Makefile`. In case of a full new distriBution now type `make`. In case of an update consult the `README` file for post install instructions.

There is no separate `install` target. It installs directly to `BIN_INSTALL`, `LIB_INSTALL`, `MAN_INSTALL` or `PAR_INSTALL`.

7 Running WFC programs and scripts

After compilation of the WFC software and installation of all components you ready to use the software. See 'Software User Manual' and 'The SAX-WFC Cookbook' for further instructions. Make sure the `BIN_INSTALL` is in your `PATH` and `PFILES` points to the parameterfiles directory.

ARCH	e.g. hp, alpha, dec, i386.
HOST	hostname (no domain), e.g. barolo, purple, sdc1.
BIN_INSTALL	directory where the binaries will be installed.
LIB_INSTALL	directory where the libraries will occur.
PAR_INSTALL	directory where the parameter files will occur. Should be $PFILES.
MAN_INSTALL	directory where the man pages are strored.
FTOOLS_LIB	location of your FTOOLS host directory, e.g. /usr/local/ftools/host.
CC	Your favorite ANSI compliant C compiler, e.g. gcc, cc on HP-UX, acc on SUN.
GCC	If you have GNU C compiler then set GCC to gcc else set GCC to cc.
FC	Name of your F77 FORTRAN compiler, e.g. f77, g77.
CFLAGS	The idea is that the CFLAGS contains the obligatory flags and EXTRA_CFLAGS the optional flags. CFLAGS normally contains an set-ANSI switch, e.g. -ansi for gcc and -Aa for HP-UX C compiler. It also should contain -I../../include.
EXTRA_CFLAGS	can contain optimizing and debug switches, but the user is free to add flags at will.
GCFLAGS	See above for CFLAGS, now for gcc.
PGFLAGS	Flag for PGPLOT. This flag is empty for FORTRAN compilers which don't produce underscores (All DECS) and -DFORTRAN_UNDERSCORE for compilers who do, e.g. HP, Linux.
FFLAGS	Only two are obligatory: A source line extend flag so sources with more then 80 characters are permissible, and -I../../include. The compiler should produce underscores on externals (only relevant for HP).
FVECFLAGS	Some very specific flags for vector optimizing flags for HP. Empty on other machines.
LDFLAGS	$CC is used for ld. Normally the ANSI switch and the library location LIB_INSTALL are needed.
RANLIB	is ranlib on DECS's and SUN's. touch on other computers.
INSTALL	is install where available. In case it is not available a script is supplied, installer, which does the same thing.
EXTRA_CLIBS	may be needed in case EXTRA_CFLAGS is used.
XLIB	is the X11 library (PGPLOT routines need it).
SYSLIBS	is the FORTRAN library and the mathematical library.

Table 1: Variables set by configure.

A Example Makefile

```
#
# Makefile generated by configure for ruurq2 running HP-UX-A.09.05
#
# System has NO data access layer
# System is NO HEAD system
#
# Take a glance at directories and options (till line of #'s)
#
ARCH=hp
HOST=ruurq2
BIN_INSTALL=/usr3/home/sax/sources/DIST/bin
LIB_INSTALL=/usr3/home/sax/sources/DIST/lib
PAR_INSTALL=/usr3/home/sax/sources/DIST/bin
MAN_INSTALL=/usr3/home/sax/sources/DIST/help
FTOOLS_LIB=/home/head/library/ftools/host
CC=cc
GCC=gcc
FC=f77
CFLAGS= -Aa -Dunix -D_HPUX_SOURCE -DHAVE_FFUNCTIONS -DFORTRAN_UNDERSCORE -I../../inclu
GCFLAGS= -ansi -Dunix -D_HPUX_SOURCE -I../../include
PGFLAGS=-DFORTRAN_UNDERSCORE
FFLAGS=+ppu +es +U77 +E6 +E1 +A -O
FVECFLAGS=+OP +O3 +es -w -I../../include
LDFLAGS=-Aa -z -L/usr3/home/sax/sources/DIST/lib
RANLIB=touch
INSTALL=/etc/install
SYSLIBS= -L/usr/lib/X11R5 -lX11 -lU77 -lcl -lisamstub -lvec -lm -lf
#
##########################################################
# Rest of this makefile is from Makefile.ORIG
#
SHELL=/bin/sh

STAGE1 = wfotfits wmakgti wattana wshift watproc wgain \
 wgtitest wmrggti wgtiflt whkproc wqplot wslew wspiros

STAGE2 = msk ins sim cor iros cdi lcv tools wop eci

LIBRARIES2-a = dta fits util log pcl
LIBRARIES2-b = evt PSF exp COR INS ANA APS

LIBRARIES = saxwfclib $(LIBRARIES2-a) $(LIBRARIES2-b)

CC_PROGRAMS = msk ins sim cor iros cdi lcv tools genFits eci

GCC_PROGRAMS = wfotfits wmakgti wattana wshift watproc wfmerge wgain \
       wgtitest wmrggti wgtiflt whkproc wqplot wslew wspiros

DIRS = "BIN_INSTALL=$(BIN_INSTALL)" "LIB_INSTALL=$(LIB_INSTALL)" \
       "PAR_INSTALL=$(PAR_INSTALL)" "MAN_INSTALL=$(MAN_INSTALL)"
```

```
SHIPLIST  = Makefile.ORIG configure \
    installer cal/*.asc cal/Makefile cal/wfc_config \
    lib/catalogs/XRAYSOURCESCAT README doc update

SHIPLIST1 = src/saxwfclib src/wattana src/wshift src/wfmerge src/wfotfits \
    src/wgtiflt src/whkproc src/wgain src/wmakgti src/wmrggti src/wgtitest \
    src/wqplot src/wslew src/wspiros src/watproc help \
    etc version/etc/stage1

SHIPLIST2 = src/ana src/exp src/iros src/pcl src/aps \
    src/cdi src/dft src/fits src/lcv src/psf src/tools \
    src/dta src/genFits src/log src/util \
    src/cor src/evt src/ins src/msk src/sim src/eci src/wop \
    include version/etc/stage2


FTOOLS = "FTOOLS_LIB=$(FTOOLS_LIB)"
FLIBS = -lftools -lhost -lxanlib -lfitsio -ltermcap -lpgplot

all : stage1 stage2

1 : stage1

2 : stage2

stage1 : saxwfclib $(STAGE1)

stage2 : $(LIBRARIES2-a) dft genFits CAL $(LIBRARIES2-b) $(STAGE2) idb

$(LIBRARIES): checkHost
cd `echo "src/$@" | tr "A-Z" "a-z"`;\
make lib "LIB_INSTALL=$(LIB_INSTALL)" \
"CC=$(CC)" "CFLAGS=$(CFLAGS)"  \
"FC=$(FC)" "FFLAGS=$(FFLAGS)"  \
"LDEST=$(LIB_INSTALL)" \
"RANLIB=$(RANLIB)" "INSTALL=$(INSTALL)"

$(CC_PROGRAMS): checkHost
cd src/$@;\
make install $(DIRS) $(FTOOLS) \
"CC=$(CC)" "CFLAGS=$(CFLAGS)"  \
"FC=$(FC)" "FFLAGS=$(FFLAGS)" \
"FLIBS=$(FLIBS)" "DEST=$(BIN_INSTALL)" \
"SYSLIBS=$(SYSLIBS)" "INSTALL=$(INSTALL)" \
        "LD=$(CC)" "LDFLAGS=$(LDFLAGS)" "RANLIB=$(RANLIB)"

$(GCC_PROGRAMS): checkHost
cd src/$@;\
make install $(DIRS) $(FTOOLS) \
"CC=$(GCC)" "CFLAGS=$(GCFLAGS)"  \
"FC=$(FC)" "FFLAGS=$(FFLAGS)" \
"FLIBS=$(FLIBS)" "DEST=$(BIN_INSTALL)" \
```

```
"SYSLIBS=$(SYSLIBS)" "INSTALL=$(INSTALL)" \
"LD=$(GCC)" "LDFLAGS=$(LDFLAGS)" "RANLIB=$(RANLIB)"

dft: checkHost
@if [ `hostname | awk -F. '{print $$1}'` != $(HOST) ]; then echo "ERROR!! Makefile is c
cd src/dft; \
make lib "LIB_INSTALL=$(LIB_INSTALL)" \
"CC=$(CC)" "CFLAGS=$(CFLAGS)"  \
"FC=$(FC)" "FFLAGS=$(FVECFLAGS)"  \
"LDEST=$(LIB_INSTALL)" \
"RANLIB=$(RANLIB)" "INSTALL=$(INSTALL)"

CAL: checkHost
cd cal;\
make install $(DIRS) \
"CC=$(CC)" "CFLAGS=-I../include $(CFLAGS)"  \
"DEST=$(BIN_INSTALL)" \
"SYSLIBS=$(SYSLIBS)" "INSTALL=$(INSTALL)" \
"LDFLAGS=$(LDFLAGS)" "RANLIB=$(RANLIB)"

wop: checkHost
cd src/wop;\
make install $(DIRS) \
"CC=$(CC)" "CFLAGS=$(CFLAGS) $(PGFLAGS)"  \
"FC=$(FC)" "FFLAGS=$(FFLAGS)"  \
"DEST=$(BIN_INSTALL)" \
"SYSLIBS=-L$(FTOOLS_LIB) -lpgplot $(SYSLIBS)" \
"INSTALL=$(INSTALL)" \
"LDFLAGS=$(LDFLAGS)"

lin: checkHost
cd src/lin;\
make install $(DIRS) \
"CC=$(CC)" "CFLAGS=$(CFLAGS) $(PGFLAGS)"  \
"FC=$(FC)" "FFLAGS=$(FFLAGS)"  \
"DEST=$(BIN_INSTALL)" \
"SYSLIBS=$(SYSLIBS)" \
"INSTALL=$(INSTALL)" \
"LDFLAGS=$(LDFLAGS)"

idb: checkHost
cd cal; \
WFC_CAL=`pwd`; export WFC_CAL; \
cp -f ./wfc_config $(BIN_INSTALL); \
$(BIN_INSTALL)/mask -p -Lidb.log wfc "pr33,[256],my" 1.0 1.0 0.9 0.9; \
$(BIN_INSTALL)/genFits -p -Lidb.log det_wfc1.asc; \
$(BIN_INSTALL)/genFits -p -Lidb.log det_wfc2.asc; \
$(BIN_INSTALL)/instrument -p -n -Lidb.log wfc1 wfc wfc1 700.3; \
$(BIN_INSTALL)/instrument -p -n -Lidb.log wfc2 wfc wfc2 700.3;

veryclean:
@rm -f $(BIN_INSTALL)/*
@rm -f $(LIB_INSTALL)/*.a
```

```
@rm -f $(PAR_INSTALL)/*.par
@rm -f cal/*.h cal/*.det cal/*.c cal/*.o cal/*.log cal/*.ins cal/*.msk
@make clean1
@make clean2

clean:
@make clean1
@make clean2

clean1:
@rm -f src/*/*.o src/*/*.a
@rm -f `find . -name 'y.tab.h' -print`
@rm -f include/detector.h include/detector_keys.h
@rm -f $(LIB_INSTALL)/libsaxwfc.a
@rm -f src/wattana/wattana src/wshift/wshift src/wfmerge/wfmerge \
      src/wfotfits/wfotfits src/wgtiflt/wgtiflt \
      src/whkproc/whkproc src/wgain/wgain src/wmakgti/wmakgti \
      src/wmrggti/wmrggti src/wgtitest/wgtitest \
      src/wqplot/wqplot src/wslew/wslew src/watproc/watproc \
      src/wspiros/wspiros

clean2:
@rm -f src/*/*.o src/*/*.a
@rm -f `find . -name 'y.tab.h' -print`
@rm -f include/detector.h include/detector_keys.h
@rm -f cal/*.h cal/*.det cal/*.c cal/*.o cal/*.log cal/*.ins cal/*.msk
@rm -f src/ana/analyse src/cdi/cdi \
       src/cor/correlate src/genFits/genFits \
       src/ins/instrument src/iros/iros \
       src/lcv/lcv src/msk/mask \
       src/sim/simulate src/tools/fstructure src/tools/fextract \
       src/tools/fautocor src/tools/fcautocor src/wop/wopi \
       src/lin/det_lin
@rm -f $(LIB_INSTALL)/libpcl.a $(LIB_INSTALL)/libwfc.a $(LIB_INSTALL)/libutil.a \
       $(LIB_INSTALL)/libdft.a

dist:   clean
@version/bin/inc_version 1
@version/bin/inc_version 2
@echo making distribution ...
@tar cf wfc_full-`./version/bin/version 1`.`./version/bin/version 2`.tar \
$(SHIPLIST) $(SHIPLIST1) $(SHIPLIST2)
@ident wfc_full-`./version/bin/version 1`.`./version/bin/version 2`.tar | \
sort | uniq | grep "Header: " > \
./version/lib/full-`./version/bin/version 1`.`./version/bin/version 2`.ident
@gzip -f wfc_full-`./version/bin/version 1`.`./version/bin/version 2`.tar
@echo created: wfc_full-`./version/bin/version 1`.`./version/bin/version 2`.tar.gz

dist1:  clean1
@version/bin/inc_version 1
@echo "making distribution (stage1) ..."
@tar cf wfc_stage1-`./version/bin/version 1`.tar $(SHIPLIST) $(SHIPLIST1)
@ident wfc_stage1-`./version/bin/version 1`.tar | sort | uniq | \
```

```
grep "Header: " > ./version/lib/stage1-`./version/bin/version 1`.ident
@gzip -f wfc_stage1-`./version/bin/version 1`.tar
@echo created: wfc_stage1-`./version/bin/version 1`.tar.gz

dist2:  clean2
@version/bin/inc_version 2
@echo "making distribution (stage2) ..."
@tar cf wfc_stage2-`./version/bin/version 2`.tar $(SHIPLIST) $(SHIPLIST2)
@ident wfc_stage2-`./version/bin/version 2`.tar | sort | uniq | \
grep "Header: " > ./version/lib/stage2-`./version/bin/version 2`.ident
@gzip -f wfc_stage2-`./version/bin/version 2`.tar
@echo created: wfc_stage2-`./version/bin/version 2`.tar.gz

checkHost:
@if [ `hostname | awk -F. '{print $$1}'` != $(HOST) ]; then \
  echo "ERROR!! Makefile is configured for $(HOST)."; \
  exit 1;\
fi;
```