

UNITED STATES DEPARTMENT OF THE INTERIOR  
GEOLOGICAL SURVEY

Preparing Software Summary Forms on a Text Processor

by

Don L. Sawatzky

Open-File Report 89-286  
1989

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards. Any use of trade names in this report is for descriptive purposes only and does not imply endorsement by the U.S. Geological Survey.

## Introduction

Scientific investigations by U. S. Geological Survey personnel for some Department of Energy programs, such as the National Nuclear Waste Site Investigation (NNWSI), require compliance with rules of Quality Assurance. Certification of computer software used in these investigations is one form of compliance.

The first activity is the completion and submittal of software summary forms (SSF). These forms contain a brief description of a computer program, its usage, date of last modification, and summary date. In addition, the program must have had version number and identification information inserted in it, so that a user will be informed of this on invocation. Because of the large number of programs to be certified, a personal computer was used to prepare completed SSFs.

Software summary forms are prepared with a text processor on a personal computer by merging program information with the standard SSF form during printing. Program information is extracted from previously prepared help files or manuals available interactively to program users. Other information on dates of modification and length required in the SSF is taken from file directories on the computer storing the programs to be summarized.

## Program Preparation

Programs written in Fortran are modified to include a PROGRAM statement, a system header, a required version number and identification, and required usage documentation. These modifications and inclusions are at the beginning of the source code, as shown in the sample partial listing in listing 1. The REMAPP.HDR inclusion is not essential to the SSF, but identifies the origin and location of the program. The NNWSI.INT inclusion is essential in later submittals and is from the help file for this program. The REMLOG routine initiates logging of program operations to a file, but is not essential for the SSF. Version information is essential in the program, and in an executable statement that informs the user the program has been documented and approved through submittal of the SSF. Writing the version information to the log file is not essential.

The help file for each program has the format given in listing 2. Key lines beginning with the words "Name:", "Purpose:", "Algorithm:", "Functions:", or "Invocation:" and their accompanying texts provide essential information for sections of the SSF. "Name" fills the name line. "Purpose" goes in the brief description, and "Algorithm" or "Functions" go in the narrative.

### Standard document

The main document that is the skeleton of the SSF was prepared in the style of the word processing program, Microsoft Word<sup>1</sup>. The procedure described here can be extended to any text processor that can merge variable files with a standard form.

The main document contains the essential words and lines of the SSF and is shown in listing 3. In addition, there are filled-in fields that are the same for all programs. Variables to be filled in from a variable text file (described below) are enclosed in double angle brackets. The name of the variable text file is given in the DATA statement on line one. The variable text file can be constructed to contain data for several programs.

### Variable text

Variable text is formatted in the style of the word processor (listing 4). In this case, the first line is a list of the variable names in order of their use in the merged document, the SSF. Following this are the ordered lists of variable values for each SSF, i.e., each program being summarized. This list can be generated on the word processor. However, program information and documentation existing on a minicomputer was transferred and reformatted into the variable text file by programs on the personal computer. Program SSFI (appendix I) is a Turbo Prolog<sup>1</sup> program that reads the help file and creates a similar but indexed help file, indexed to the position of program names. A second file containing the indexes is also created by SSFI. Program SSFP (appendix II) is a Turbo Prolog program that creates the variable text file for programs in the indexed help file. One additional file was prepared on the host minicomputer and contains a list of program names,

---

<sup>1</sup>Use of this copyrighted name or tradename neither intends nor implies endorsement of this product by the U. S. Geological Survey.

number of lines, creation dates, and system requirements. An example of this file is in listing 5.

#### Merged document

The merged standard document and variable text become the completed Software Summary Form shown in listing 6.

## Appendix I

```
/*
Turbo Prolog Program SSFI
Converts NNWSI Remapp and Linanl Documentation file to an indexed file for use in program SSFP
*/
domains
    file=infile;outfile;tempfile
    name=string
    ssf=ssf(name,real)

database
    ssf(name,real)
    h(string)

predicates
    endpurpose(string,real)
    findend(string)
    findname(name)
    instr(string,string,real,real)
    pwrite(dbasedon)
    repeat
    testend(string)

clauses

    repeat.
    repeat if repeat.

    findname(Name) if
        repeat,readln(Line),
        frontstr(6,Line,S,Name),S="Name: ",
        write(Line),nl,!.

    findend(Eop) if repeat,readln(Line),
        write(Line,"\\n"),
        frontstr(1,Line,Eop,_),
        testend(Eop),!.

    testend(Eop) if Eop="\\12";
        Eop="\\4".

    instr(S,T,_,_) if
        str_len(S,Ns),str_len(T,Nt),
        Ns<Nt,!,fail.
    instr(S,T,N,N) if
        str_len(T,Nt),
        frontstr(Nt,S,Sub,_),
        Sub=T,!.
    instr(S,T,N,M) if
        frontstr(1,S,_,Next),N1=N+1,
        instr(Next,T,N1,M).

    endpurpose(S,N) if
```

```

instr(S,"Functions:",1,M),N=M-1,!;
instr(S,"Algorithm:",1,M),N=M-1,!;
instr(S,"Invocation:",1,M),N=M-1,!.

```

```
ppwrite(X) if write(X),nl.
```

goal

```

openread(infile,"c:/nnwsi.int"),
readdevice(infile),
openwrite(outfile,"nnwsi.ssf"),
writedevic(outfile),
repeat,
openwrite(tempfile,"temp"),
writedevic(tempfile),
findname(Name),
findend(EOP),
closefile(tempfile),
file_str("temp",S),
writedevic(outfile),
filepos(outfile,Nst,0),
ppwrite(h(S)),
writedevic(screen),
write("ssf",Name,',',Nst,')'),nl,
assertz(ssf(Name,Nst)),
EOP="\4",save("ssf.idx"),
closefile(outfile).

```

## Appendix II

```
/*
  Turbo Prolog Program SSFP
  Converts NNWSI indexed program documentation file into a data file for printer merging with an SSF on
  MSWORD word processing
*/
project"ssfp"

domains
  h=h(string)
  file=infile;outfile
  strlist=string*

database
  ssf(string,real)
  h(string)
  nstates(string,integer,string,string)

global predicates
  _instr(string,string,integer) - (i,i,o) language c

predicates
  alias(string,strlist)
  do
  endpurpose(string,string)
  head(string,string,string)
  head2(string,integer,string)
  narrative(string,string)
  nstates(strlist,integer)
  purpose(string,string,string)
  repeat
  tail(string,string,string)
  tail2(string,integer,string)
  writenstates(string)
  writeuop(string)
  writetext(string,integer,integer,string)

clauses
  repeat.
  repeat if repeat.

  alias(Name,[Name]).

  purpose(S1,Name,P) if
    ssf(Name,T1),
    filepos(infile,T1,0),
    readterm(dbasedon,h(S)),
    tail(S,"Purpose:",S1),
    endpurpose(S1,P).

  tail(S1,S2,S3) if
```

```

_instr(S1,S2,N),N>0,
tail2(S1,N,S3).

tail2(S,1,S) if !.
tail2(S1,N,S2) IF
    N1=N-1,frontstr(N1,S1,_,S2),!.

endpurpose(S1,S2) if
    head(S1,"Functions:",S2),!;
    head(S1,"Algorithm:",S2),!;
    head(S1,"Invocation:",S2),!.

narrative(P,N) if
    tail(P,"Functions:",T),
    head(T,"Invocation:",N0),
    frontstr(11,N0,_,N),!;
    tail(P,"Algorithms:",T),
    head(T,"Invocation:",N0),
    frontstr(11,N0,_,N),!;
    head(P,"Invocation:",N0),
    frontstr(9,N0,_,N),!.

head(S1,S2,S3) if
    _instr(S1,S2,N),N>0,
    head2(S1,N,S3).

head2(S,1,S) if !.
head2(S1,N,S2) if
    N1=N-1,frontstr(N1,S1,S2,_,),!.

do if
    nostates(Name,_,LW,_),
    purpose(S1,Name,P),
    writedevic(screen),write(Name),nl,writedevic(outfile),
    write(Name),write(","),
    date(Year,Mon,Day),
    str_int(SYear,Year),
    frontstr(2,SEar,_,Year1),
    write("__",Mon,"_m/___",Day,"_d/___",Year1,"y__"),write(","),
    fronttoken(LW,MW,LW1),
    str_int(MW,NMW),
    fronttoken(LW1,_,LW2),
    fronttoken(LW2,DW,LW3),
    str_int(DW,NDW),
    fronttoken(LW3,_,LW4),
    fronttoken(LW4,YW,_,),
    write("__",NMW,"_m/___",NDW,"_d/___",YW,"y__"),write(","),
    frontstr(9,P,_,P0),
    Application="Used for processing of geologic information in digital images.",
    concat(Application,P0,P1),
    write("\\"),writetext(P1,1,27,P2),write("\\"),
    write("\\"),writetext(P2,1,87,P3),write("\\"),
    write("\\"),writetext(P3,1,87,P4),write("\\"),
    write("\\"),writetext(P4,1,87,_,),write("\\"),

```



```

narrative(S1,N1),
write("\n"),writetext(N1,1,77,N2),write("\n"),
write("\n"),writetext(N2,1,87,N3),write("\n"),
write("\n"),writetext(N3,1,87,N4),write("\n"),
write("\n"),writetext(N4,1,87,N5),write("\n"),
write("\n"),writetext(N5,1,87,N6),write("\n"),
write("\n"),writetext(N6,1,87,_),write("\n"),
writenostates(Name),write("\n"),
writeup(Name),nl,
fail.

writetext(S,N,N,S).
writetext("",N,M,S1) if
    write(" "),N1=N+1,
    writetext("",N1,M,S1),!.
writetext(S,N,M,S1) if
    frontstr(1,S,C,S2),
    C="\n",
    writetext(S2,N,M,S1),!.
writetext(S,N,M,S1) if
    frontstr(1,S,C,S2),
    C="\t",write(" "),N1=N+1,
    writetext(S2,N1,M,S1),!.
writetext(S,N,M,S1) if
    frontstr(1,S,C,S2),
    write(C),N1=N+1,
    writetext(S2,N1,M,S1),!.

writenostates(Name) if
    alias(Name,Alias),
    nstates(Alias,N),
    write(N).

nstates([],0) if !.
nstates([First:Rest],Total) if
    nostates(First,N,_),
    nstates(Rest,M),
    Total=N+M.

writeup(Name) if
    nostates(Name,_,_,UOP),UOP<>"",
    write("_",UOP),!;
    write("X,_____").

goal
    clearwindow,
    openread(infile,"nnwsi.ssf"),
    openwrite(outfile,"a:ssfs.dat"),
    consult("ssf.idx"),
    consult("ssfprt.db"),
    readdevice(infile),
    writedevicel(outfile),
    write("progname,sundate,modddate,p1,p2,p3,p4,n1,n2,n3,n4,n5,n6,nostates,uop1,uop2\n"),
    do

```

Listing 1 - Partial Listing of Example Source Code

```
PROGRAM ARITH
$INCLUDE REMAPP.HDR
$INCLUDE NNWSI.INT,**ARITH-(NEND)
C...VARIABLE DECLARATIONS
    INTEGER*4 IBUF(8000)
    INTEGER*2 XBUF(8000)
    EQUIVALENCE(XBUF,IBUF)
    INTEGER FCBA(256),FCBB(256),FCBC(256),AL60R
    REAL LNUM,LDEN,C(7)
C...LOG IT
    CALL REMLOG(' * * * *  AP ARITH  * * * * ')
C...USGS Computer Program "ARITH";Version 1.000
C...Technical Contact: D. L. Sawatzky
    WRITE(4,*) 'USGS Computer Program "ARITH";Version 1.000'
    WRITE(4,*) 'Technical Contact: D. L. Sawatzky'
    WRITE(6,*) 'USGS Computer Program "ARITH";Version 1.000'
    WRITE(6,*) 'Technical Contact: D. L. Sawatzky'
C...PROGRAM STARTS
    ....
    ....
    END
```

## Listing 2 - Sample program help file

Name: ARITH

Purpose: Does arithmetic operations on registered data numbers  
of two REMAPP files and writes the result to a REMAPP file.

Functions: fileA and fileB are input files, fileC is the  
output file, alpha and beta are constants.  
1.  $\text{fileC} = (\text{alphaA} * \text{fileA}) + (\text{alphaB} * \text{fileB}) + \text{beta}$ .  
2.  $\text{fileC} = \text{alpha} * \text{fileA} * \text{fileB} + \text{beta}$ .  
3.  $\text{fileC} = \text{alpha} * (\text{fileA} / \text{fileB}) + \text{beta}$ .  
4.  $\text{fileC} = \text{alpha} * \text{fileA} + \text{beta}$ .

Invocation: GO ARITH

Interactions:

1) Select algorithm 1,2,3 or 4 \*

?enter number of one of the above

2) fileA:

Diskio: ENTER INPUT FILE NAME:

?enter REMAPP file name

a) ENTER 1st scanline, no. scanlines, skips, band:

?k,l,m,n = subset parameters

0,0,0,n = all scanlines, no skips

b) ENTER 1st pixel, no. pixels, skips:

?k,l,m = subset parameters

0,0,0 = all pixels, no skips

3) fileB:

Diskio: ENTER INPUT FILE NAME:

?enter REMAPP file name

a) ENTER 1st scanline, no. scanlines, skips, band:

?k,l,m,n = subset parameters

0,0,0,n = all scanlines, no skips

b) ENTER 1st pixel, no. pixels, skips:

?k,l,m = subset parameters

0,0,0 = all pixels, no skips

4) fileC:

Diskio: enter output file name:

?enter REMAPP file name

when algorithm 1 is selected:

6) Enter alphaA,alphaB, and beta \*

?alphaA,alphaB,beta

when algorithms 2,3, or 4 are selected:

6) Enter alpha and beta \*

?alpha,beta

7) continue(y,n)?

?yes = return to 1) for more processing

no = stop

Listing 3 - Sample of the main document to be merged with a variable text to produce the SSF

Software Summary Form (SSF)

USGS Computer Program "<<programe>>"; Version: 1.000 ;

Technical Contact: Don L. Sawatzky

address: MS 964 DFC

Box 25046

Denver, CO 80225

telephone: ( 303 ) 236-1387 x

FTS 776 - 1387 x

Organization: U.S. Department of the Interior, U.S. Geological Survey,  
| Geologic Division, Branch of Geophysics |

DR Contractor:

Summary prepared by (X) TC or : \_\_\_\_\_

Summary Date:<<sumdate>>; Software last modified:<<moddate>>

Action: (X)1st summary; ( ) Revision; ( ) Error; ( ) New TC; Other \_\_\_\_\_

If not "1st summary", version number of previous SSF submitted: \_\_\_\_\_

Brief description of software function and application area: <<p1>> |

| <<p2>> |

| <<p3>> |

| <<p4>> |

Software is : ( ) Interactive; ( ) Batch; (X) Combination

Software is: ( ) Scientific and Engineering or (X) Auxiliary. Justification:

| Software is not used only for Quality Level III work.

| Software is not an operating system, standard library, or utility.

| Software performs calculations that are simple statistics, coordinate transformations,  
| trivial calculations, data reformatting, or graphics.

Software may used for NNWSI work at Quality Level (X) I; ( ) II; ( ) III.

Narrative: <<n1>> |

| <<n2>> |

| <<n3>> |

| <<n4>> |

| <<n5>> |

| <<n6>> |

Keywords (circle appropriate): Computer Program; SES ; (Auxiliary); Numerical Method ;

| Modeling ; Inversion ; Data Acquisition ; (Data Reduction); Other: \_\_\_\_\_ ;

| \_\_\_\_\_ ; \_\_\_\_\_ ; \_\_\_\_\_ ;

Programming Language(s): Fortran 77

Number of source statements/(lines)(circle): <<nostates>>

Any unusual operation requirements(<<uop1>>none): <<uop2>>

Software is: (X) Available; ( ) In Testing; ( ) In Preparation.

Documentation is: ( ) Available; (X) Partly Available; ( ) In Preparation; ( ) Not Required.

For QA Manager use:

SCIF received:

Listing 4 - Variable text to be merged with Microsoft Word  
main document

```

progname,sumdate,mddate,p1,p2,p3,p4,n1,n2,n3,n4,n5,n6,nostates,uop1,uop2
ARITH, __2_m/ __28_d/ __89y__, __2_m/ __27_d/ __89y__, "Does arithmetic operations ",
"on registered data numbers of two RENAPP files and writes the result to
", " a RENAPP file.
",
"fileA and fileB are input files, fileC is the
", "    output file, alpha and beta are constants.
", "    1. fileC = (alpha*fileA) + (alphaB*fileB) + beta.
", "    2. fileC = alpha*fileA*fileb + beta.
", "    3. fileC = alpha*(fileA/fileB) + beta.
", "    4. fileC = alpha*fileA + beta.
",105, _Requires FPS AP-120B array processor

```

Listing 5 - File containing program name, number of lines, creation date, and system requirements.

```
nostates("APPEND",77,"2/27/89","")
nostates("ARITH",218,"2/27/89","Requires FPS AP-120B array processor")
nostates("CONCAT",67,"2/27/89","")
nostates("DSST",176,"2/27/89","Requires FPS AP-120B array processor")
nostates("DSTRIP",92,"2/27/89","Requires FPS AP-120B array processor")
nostates("EDGES",159,"2/27/89","Requires FPS AP-120B array processor")
nostates("ENLARGE",52,"2/27/89","")
nostates("ERTSGEOM",57,"2/27/89","")
nostates("ERTSIN",297,"2/27/89","")
nostates("EXPAND",127,"2/27/89","Requires FPS AP-120B array processor")
nostates("GEOMX2",125,"2/27/89","")
nostates("HCMMIN",221,"2/27/89","")
nostates("HISTMAIN",211,"2/27/89","")
nostates("HUENAP",132,"2/27/89","Requires FPS AP-120B array processor")
nostates("LANDSBIL",159,"2/27/89","")
nostates("LANDSBSQ",119,"2/27/89","")
```

Listing 6 - Merged standard and variable documents in SSF



## Software Summary Form (SSF)

USGS Computer Program "ARITH"; Version: 1,000 ;Technical Contact: Don L. Sawatzkyaddress: MS 964 DFCBox 25046Denver, CO 80225telephone: ( 303 ) 236-1387 xFTS 776 - 1387 x

Organization: U.S. Department of the Interior, U.S. Geological Survey,

Geologic Division, Branch of Geophysics

OR Contractor:

Summary prepared by ☒ TC or :Summary Date: 5 m/ 6 d/ 88y ; Software last modified: 4 m/ 7 d/ 88yAction: ☒ 1st summary; ☐ Revision; ☐ Error; ☐ New TC; OtherIf not "1st summary", version number of previous SSF submitted: .

Brief description of software function and application area: Used for processing of geologic information in digital images. Does arithmetic operations on registered data numbers of two REMAPP files and writes the result to a REMAPP file.

Software is : ☐ Interactive; ☐ Batch; ☒ CombinationSoftware is: ☐ Scientific and Engineering or ☒ Auxiliary. Justification:

Software is not used only for Quality Level III work.

Software is not an operating system, standard library, or utility.

Software performs calculations that are simple statistics, coordinate transformations, trivial calculations, data reformatting, or graphics.

Software may used for NNWSI work at Quality Level ☒ I; ☐ II; ☐ III.

Narrative: fileA and fileB are input files, fileC is the output file, alpha and beta are constants. 1. fileC = (alpha\*fileA) + (alphaB\*fileB) + beta. 2. fileC = alpha\*fileA\*fileB + beta. 3. fileC = alpha\*(fileA/fileB) + beta. 4. fileC = alpha\*fileA + beta.

Keywords (circle appropriate): Computer Program; SES ; (Auxiliary); Numerical Method ; Modeling ; Inversion ; Data Acquisition ; (Data Reduction); Other: \_\_\_\_\_

Programming Language(s): Fortran 77Number of source statements/(lines) (circle): 1636Any unusual operation requirements (none): Requires FPS AP-120B array processorSoftware is: ☒ Available; ☐ In Testing; ☐ In Preparation.Documentation is: ☐ Available; ☒ Partly Available; ☐ In Preparation; ☐ Not Required.

For QA Manager use:

SCIF received: