NOAA Techniques Development Laboratory
Computer Program NWS TDL CP 89-1

STRUCTURE FLOW DIAGRAM GENERATOR

Silver Spring, Md.
March 1989

U.S. DEPARTMENT OF COMMERCE / National Oceanic and Atmospheric Administration / National Weather Service

PREFACE

The Techniques Development Laboratory's (TDL's) computer program (CP) series is a subset of TDL's technical memorandum series. The CP series documents computer programs written at TDL primarily for the Automation of Field Operations and Services (AFOS) computers.

The format for the series follows that given in the AFOS Handbook 5, Reference Handbook, Volume 6: Applications Programs, Part 1: Policy and Procedures, published by the Office of Technical Services/AFOS Operations Division.

NOAA Techniques Development Laboratory
Computer Program NWS TDL

CP 83-1 Cross Sectional Analysis of Wind Speed and Richardson Number. Gilhousen, Kemper, and Vercelli, May 1983. (PB83 205062)
CP 83-2 Simulation of Spilled Oil Behavior in Bays and Coastal Waters. Hess, October 1983. (PB84 122597)
CP 83-3 AFOS-Era Forecast Verification. Heffernan, Newton, and Miller, October 1983. (PB84 129303)
CP 83-4 AFOS Monitoring of Terminal Forecasts. Vercelli, December 1983.
CP 83-5 Generalized Exponential Markov (GEM) Updating Procedure for AFOS. Herrmann, December 1983.
CP 84-1 AFOS Display of MDR Data on Local Map Background. Newton, July 1984.
CP 84-2 AFOS Surface Observation Decoding. Perrotti, September 1984.
CP 84-3 AFOS-Era Forecast Verification. Miller, Heffernan, and Ruth, September 1984.
CP 85-1 AFOS Monitoring of Terminal Forecasts. Vercelli and Norman, May 1985.
CP 85-2 AFOS Terminal Forecast Decoding. Vercelli, Norman, and Heffernan, October 1985.
CP 85-3 AFOS-Era Forecast Verification. Ruth, Miller, and Heffernan, October 1985.
CP-87-1 AFOS Terminal Aerodrome Forecast Formatting. Wantz and Eggers, July 1987.
CP-87-2 AFOS-Era Forecast Verification. Ruth and Alex, July 1987.
CP-87-3 Forecast Review. Wolf, July 1987.
CP-87-4 AFOS Monitoring of MDR Data Using Flash Flood Guidance. Norman and Newton, October 1987.
CP-87-5 AFOS Terminal Forecast Quality Control. Vercelli and Leaphart, December 1987.
CP-88-1 AFOS Terminal Forecast Decoding. Vercelli and Leaphart, August 1988.

NOAA Techniques Development Laboratory
Computer Program NWS TDL CP 89-1

STRUCTURE FLOW DIAGRAM GENERATOR

Susan M. Adams

Techniques Development Laboratory
Silver Spring, Md.
March 1989

UNITED STATES
DEPARTMENT OF COMMERCE
Robert A. Mosbacher
Secretary

National Oceanic and
Atmospheric Administration
William E. Evans, Under Secretary

National Weather Service
Elbert W. Friday, Jr.
Assistant Administrator

TABLE OF CONTENTS

STRUCTURE FLOW DIAGRAM GENERATOR

Susan M. Adams

1. INTRODUCTION

The Automation of Field Operations and Services (AFOS) system is presently running on the Data General S/230 Eclipse minicomputer operating under the Real-Time Disk Operating System (RDOS). The Data General S/230 minicomputers have 32K bytes of addressable memory (NWS, 1987). This means that all programs running must be able to fit within the 32K byte limit of main memory. Occasionally, a program will require more memory than is available on the machine. One of the more widely used memory management tools is user overlays, which effectively extends the main memory with disk space. Due to wide use of such overlays, there is sometimes a need to see the program structure, along with the segment and overlay numbers, when overlay mapping is planned. Previously, the structure and overlay mapping was done by hand. This becomes long and tedious, and errors often occur.

The Structure Flow Diagram Generator provides the user with a convenient, easy-to-use, and less time-consuming way of diagramming the overall structure of a program. It allows for several options including segment and overlay numbers, and the inclusion of repetitive calls within a subroutine.

This software can also be useful in preparing external documentation and publications which require an overall program structure diagram as part of the supporting documents.

2. METHODOLOGY AND SOFTWARE STRUCTURE

The Structured Flow Diagram Generator consists of a single program, SFD. SFD is designed to create a software structure flow diagram showing the executional hierachy of non-library routines called by a program. Routines to be included are determined from the program load line. SFD accesses the program FORTRAN file, load line file, and the FORTRAN or Assembly Language files of its subordinate routines. Functions and library routines are not included in the diagram. Tasks initiated within the program with a call to "ITASK" are printed, along with their subroutines, at the end of the diagram in the order in which they are initiated by the program.

The structure flow diagram created by SFD is output to a file with the .FD extension on the program name. Fig. 1 illustrates the data flow for the Structure Flow Diagram Generator. The overall program structure of SFD is shown in Fig. 2.

3. PROCEDURES AND USE OF SWITCHES

SFD should be installed in an applications directory with links from the directory where the source code and load line for the program to be used

reside. Several switches are available to pass information to the program at runtime and to vary the output from the program.

The SFD program is initiated from the terminal by entering:

SFD/A/O progname/N

The required local switch, /N, passes the name of the program to be used in generating the flow diagram to SFD. The program to be used must have a load line (.LD) file and a FORTRAN (.FR) file in the directory from which SFD is initiated. The two optional global switches, /A and /O, are used to vary the output of SFD. The /A switch is used to include repetitive calls to a routine within the same subroutine (see Fig. 3). The /O switch can be used to print the segment and overlay numbers of those routines not stored in main memory to the right of the routine name (see Fig. 4). The segment and overlay numbers are determined from the program load line. If no optional switches are used, neither the segment and overlay numbers nor the repetitive subroutine calls are printed (see Fig. 5).

4. CAUTIONS AND RESTRICTIONS

Due to the limited amount of main memory available on the Data General S/230 computers, several limitations have been set. Programs that exceed these limitations should not be used with SFD. These limitations are:

1. The program load line may not exceed 50 lines in length. If this occurs, the error message "FATAL ERROR - LOAD LINE TOO LONG" will be printed to the terminal from which the software was initiated.

2. The program load line may contain no more than 300 subroutines. If this limitation is exceeded, the error message "FATAL ERROR - TOO MANY SUBROUTINES IN LOAD LINE" will be printed at the terminal.

3. Program and subroutine filenames may contain no more than 12 characters, excluding the extension. Files with names exceeding 12 characters in length will be ignored by the program and will not be included in the diagram.

4. The program may have no more than 5 calls to "ITASK". If this occurs, the error message "FATAL ERROR - TOO MANY TASKS" will be printed to the terminal.

5. The diagram may not exceed 5000 lines in length. A line is placed in the diagram for the main program, a task initialization statement, or a call to a subordinate routine. When the 5000 line limit is exceeded, the error message "FATAL ERROR - DIAGRAM EXCEEDS MAXIMUM LENGTH" will be printed to the terminal. The diagram up to that point will be in the output file.

6. The diagram may not exceed 30 imbedded levels of calls. If this happens, the error message "FATAL ERROR - EXCEEDS NUMBER OF LEVELS ALLOWED" will be printed to the terminal. The diagram up to that point will be in the output file.

SFD is designed to be used as a tool in maintaining and updating other software. The running time for the program varies slightly depending on the number of subroutines called, the length of those subroutines, and the general flow of the program. If the program includes a large number of repetitive calls, the running time with the optional /A switch will be longer than if the switch is not used. When the optional /O switch is used, the running time is not noticeably longer. In general, the running time should not deviate more than a few seconds from 5 seconds per 100 lines of source code.

5. REFERENCES

National Weather Service, 1987: National Weather Service AFOS Handbook, No. 5, Vol. 6, Pt. 2, National Weather Service, NOAA, U.S. Department of Commerce, (in preparation).

6. PROGRAM INFORMATION AND PROCEDURES FOR INSTALLATION AND EXECUTION

STRUCTURE FLOW DIAGRAM GENERATOR

PART A: PROGRAM INFORMATION and INSTALLATION PROCEDURE

PROGRAM NAME: SFD

AAL ID: MSC012
Revision No.: 01.00

FUNCTION: This software creates a software structure flow diagram showing the executional hierarchy of non-library routines called by a program. The program to be diagrammed is specified on the command line. Routines to be included in the diagram are determined from the program load line file. Optional global switches may be used to include repetitive calls to subroutines within the program, and to print segment and overlay numbers next to the name of subroutines not stored in main memory. Segment and overlay numbers for the subroutines are determined from the program load line. Tasks initiated by the program with a call to "ITASK", and their subordinate routines, are printed after the main program in the sequence initiated. Functions and library routines are not included in the diagram.

PROGRAM INFORMATION:

Development Programmer:

Susan M. Adams

Location: Techniques Development Laboratory

Phone: FTS 427-7639

Maintenance Programmer:

Harry Lebowitz

Location: Techniques Development Laboratory

Phone: FTS 427-8065

Language: FORTRAN IV/Revision 5.57
Macro Assembler/Revision 6.30

Type: Standard program

Save file Creation dates: SFD.SV
Original release/Revision 01.00 - December 1988

Running Time: Approximately 5-6 seconds per 100 lines of source code. Time varies with number and length of procedures called.

Disk space: Program files - 29 RDOS blocks

PROGRAM REQUIREMENTS

Program files:

NAME
SFD.SV

Data files:

NAME	LOCATION	READ/WRITE	COMMENTS
<progname>.FR	User's Directory	R	
<progname>.LD	User's Directory	R	
<subrtn>.FR	User's Directory	R	
<subrtn>.SR	User's Directory	R	
<progname>.FD	User's Directory	W	Created by SFD

LOAD LINE:

```
RLDR FINDER FINDREV CONVRT INITAR ISRCH RDARY OCHN TROUBL WMOV BG.LB
  UTIL.LB FORT.LB SYS.LB AFOSE.LB
```

PROGRAM INSTALLATION

1. Move SFD.SV to an applications directory and create links to this directory from the directories containing the required source code and load line for the program and subroutines.

STRUCTURE FLOW DIAGRAM GENERATOR

PART B: PROGRAM EXECUTION and ERROR CONDITIONS

PROGRAM NAME: SFD AAL ID: MSC012
Revision No.: 01.00

PROGRAM EXECUTION:

1. From the terminal, enter:

   SFD/A/O progname/N

   Definition of required switches:

   xxx/N - Program name to be diagrammed. SFD reads the load line file for the program, <progname>.LD, to determine files to be included in the diagram.

   Definition of optional switches:

   /A - Include repetitive calls to routines within the same subroutines in the diagram. Without this switch, calls to the same routine within a subroutine are printed only once.

   /O - Prints the segment and overlay number for subroutines immediately to the right of the subroutine name for those subroutines that are not in main memory. The overlay and segment numbers are determined from the program load line, <progname>.LD.

2. Before execution, be sure that the program specified on the command line does not exceed the following limitations:

   a. The load line for the program may be no longer than 50 lines in length.

   b. The load line for the program may contain no more than 300 subroutines.

   c. Filenames may contain no more than 12 characters, not including the extension. Subroutines with names exceeding 12 characters will be ignored.

   d. The program may have no more than 5 calls to "ITASK" to initialize a task for multi-tasking programs.

   e. The diagram cannot exceed 5000 lines, each line representing the main routine, a task, or a call to a subordinate routine. When the 5000 line limitation is exceeded, the program is halted.

   f. The diagram cannot exceed 30 imbedded levels of calls.

3. During execution, the name of the procedure currently being processed is written to the terminal from which SFD was initiated.

4. Upon completion, the output file, <progname>.FD can be viewed either by displaying it on the terminal (enter TYPE <progname>.FD), or by printing a hard copy (enter PRINT <progname>.FD).

ERROR CONDITIONS

	TERMINAL MESSAGES	MEANING
1.	"INCORRECT SWITCH"	/N switch was not found. Re-enter the command and rerun.
2.	"OPENING ..."	Cannot open either program FORTRAN file or load line file from /N switch. Restore the file or link and rerun.
3.	"CREATING ..."	Cannot create output file. Clear file and rerun.
4.	"FATAL ERROR - TOO MANY TASKS"	More than 5 calls to "ITASK" were found.
5.	"FATAL ERROR - LOAD LINE TOO LONG"	Load line contains more than 50 lines.
6.	"FATAL ERROR - TOO MANY SUBROUTINES IN LOADLINE"	Load line contains more than 300 unique subroutine names.
7.	"FATAL ERROR - DIAGRAM EXCEEDS MAXIMUM LENGTH"	Diagram is more than 5000 lines in length.
8.	"FATAL ERROR - EXCEEDS NUMBER OF LEVELS ALLOWED"	Diagram has more than 30 embedded levels of calls.

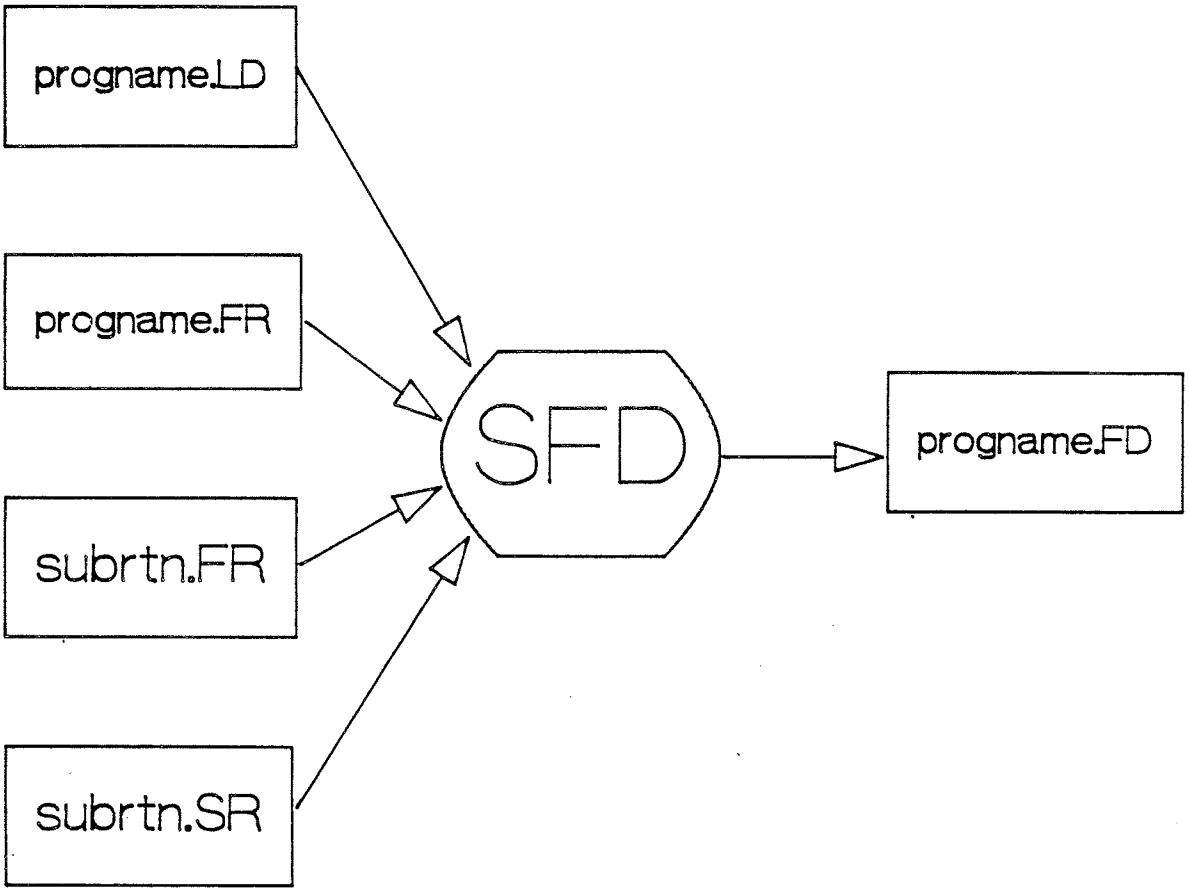

Figure 1. Data flow for the Structure Flow Diagram Generator.

MAIN PROGRAM	SUBROUTINES	
SFD		
•	INITAR	
•	WMOV	
•	TROUBL	
•	OCHN	
•	•	TROUBL
•	RDARY	
•	•	INITAR
•	•	ISRCH
•	•	WMOV
•	FINDPER	
•	STRING	
•	ISRCH	
•	CONVRT	

LOAD LINE

```
RLDR SFD SFDREV CONVRT FINDPER INITAR ISRCH OCHN RDARY STRING TROUBL WMOV
  BG.LB UTIL.LB FORT.LB SYS.LB AFOSE.LB
```

Figure 2. Overall program structure and load line for the Structure Flow Diagram Generator software.

```
COMMAND:  SFD/A TIMELINE/N

OUTPUT IN  TIMELINE.FD

TIMELINE
.           OCHN
.           .           TROUBL
.           .           TROUBL
.           OCHN
.           .           TROUBL
.           .           TROUBL
.           OCHN
.           .           TROUBL
.           .           TROUBL
.           GGDM
.           .           GDMID
.           .           .           GCID
.           .           OCHN
.           .           .           TROUBL
.           .           .           TROUBL
.           .           TROUBL
.           .           VALID
.           .           OCHN
.           .           .           TROUBL
.           .           .           TROUBL
.           .           TROUBL
.           .           TROUBL
.           INTCLS
.           INTLFT
.           .           CONVRT
.           ILOCS
.           INTRT
.           .           CONVRT
.           .           CONVRT
.           .           CONVRT
.           .           CONVRT
.           .           WBRDR
.           INTTIME
.           .           CONVRT
.           LINE
.           .           WBRDR
.           .           WBRDR
.           .           WBRDR
.           .           WBRDR
.           WRCLS
.           .           GDCLR
.           TIMEWX
```

Figure 3. Sample output of SFD with optional global /A switch.

```
COMMAND:  SFD/O GMOD/N

OUTPUT IN  GMOD.FD

.          MODIFY (0,1)
.          .          RCUR (0,1)
.          .          .          RDPCUR (0,1)
.          .          INTZOPT
.          .          INITAR
.          .          DISDD(1,3)
.          .          .          WRCLS (0,1)
.          .          .          .          GDCLR
.          .          .          INTCLS
.          .          .          WMOV
.          .          .          INTIDD (1,3)
.          .          .          .          ADDHR (1,3)
.          .          .          .          DAYWK (1,3)
.          .          .          .          WMOV
.          .          .          .          CONMD (1,3)
.          .          .          .          .          CONVRT
.          .          RWXCL (1,3)
.          .          .          TROUBL
.          .          .          WMOV
.          .          DISWX (1,5)
.          .          .          ERASE (0,1)
.          .          .          .          WRCLS (0,1)
.          .          .          .          .          GDCLR
.          .          .          INTRC (0,1)
```

```
RLDR/P 7/C 2/K GMOD.LM/L CONVRT CTASK GDCLR INTCLS INITAR INTZOPT ^
  RDCUR TROUBL WMOV 2000/N GMOD ^
[GGDM CONDAT FBRDR FILBOX FILVEC FMINT GCID GDIR GDMID GTMDC GSET INTCOM ^
  MKTIME OCHN RBRDR RDARNS RDIR SHRINK VALID, ^
 MODIFY DISPER ENDIT ERASE FBOX FILBKG GBOX INTRC ^
  MOVCUR MSHADE RCUR RDPCUR RODNA SCROLL WCURSR WRCLS] ^
[FILL FILLIT ID INCDEC, ^
 UGRD CHPRO FDUPTR FINIT FPRO PCHR RDGRD SHADE STGRD, ^
 ADVECT ADV ASCON CONVAL DETADV DGBOX DINCVAL FRBCK GTDIG PERCENT ^
  RSTADV SGN, ^
 DISDD ACK ADDHR CLEAR CONMD DAYWK GTPRO INTIDD LOCS RWXCL SWXCL ^
  TWICE, ^
 SRDATA ASCICLD CHCLD RSGRID SELGEN, ^
 SELWX ADWX BUBBLE DCDWX DISWX FWXBX STRING, ^
 BRITNS DISDAT FWORD THRESH] GMOD/S ^
<XMEM BG UTIL FMT FORT SYS AFOSE>.LB
```

Figure 4. Sample output of SFD with optional /O switch and load line of the program used in creating the diagram. Included this diagram are segment and overlay numbers (e.g., MODIFY (segment no., overlay no.)).

```
COMMAND:  SFD TIMELINE/N

OUTPUT IN  TIMELINE.FD

TIMELINE
.           OCHN
.           .           TROUBL
.           GGDM
.           .           GDMID
.           .           .           GCID
.           .           OCHN
.           .           .           TROUBL
.           .           TROUBL
.           .           VALID
.           INTCLS
.           INTLFT
.           .           CONVRT
.           ILOCS
.           INTRT
.           .           CONVRT
.           .           WBRDR
.           INTTIME
.           .           CONVRT
.           LINE
.           .           WBRDR
.           WRCLS
.           .           GDCLR
.           TIMEWX
.           .           RWXCL
.           .           .           TROUBL
.           .           .           WMOV
.           .           GTPRO
.           .           .           WMOV
.           .           .           TROUBL
.           .           RDDTA
.           .           PKWXSM
.           .           TIMERSL
.           TIMEPOP
.           .           GTPRO
.           .           .           WMOV
.           .           .           TROUBL
.           .           RDDTA
.           .           WBRDR
.           TEMPGRD
.           .           GTPRO
.           .           .           WMOV
.           .           .           TROUBL
```

Figure 5. Sample output of SFD without optional global switches.