[redacted]

535

PIONEER 10 AND 11

72-012A-13B 73-019A-13A

SOLAR WIND PROTON BULK SPEED DATA

[redacted]

PIONEER 11

SOLAR WIND PROTON BULK SPEED DATA

73-019A-13A

THIS DATA SET HAS BEEN RESTORED. THERE WERE ORIGINALLY FIVE 9-TRACK, 800 BPI TAPES, WRITTEN IN BINARY. THERE IS ONE RESTORED TAPE. THE DR TAPE IS A 3480 CARTRIDGE AND THE DS TAPE IS 9-TRACK, 6250 BPI. THE ORIGINAL TAPES WERE CREATED ON AN IBM 360 COMPUTER. THE DR AND DS NUMBERS ALONG WITH THE CORRESPONDING D NUMBERS AND TIME SPANS ARE AS FOLLOWS:

DR#	DS#	DD#	FILES	TIME SPAN
DR004179	DS004179	D041558	1	04/21/73 - 12/31/73
		D041559	2	01/01/74 - 12/31/74
		D041560	3	01/01/75 - 04/12/75
		D041561	4	05/05/78 - 12/30/78
		D047257	5	01/01/79 - 12/31/79

PIONEER 10

SOLAR WIND PROTON BULK SPEED DATA

72-012A-13B

This data set has been restored. There were originally 8 9-track, 800 BPI tapes written in Binary. There are two restored tapes. The DR tape are 3480 cartridges and the DS tape are 9-track, 6250 BPI. The original tapes were created on a 360 computer. The DR, DS, and DD numbers along with the time spans are given as follows:

DR #	DS #	DD #	FILES	TIME SPAN
DR02952	DS02952	D-47249	1	04/18/72 - 06/23/72
		D-47250	2	06/24/72 - 12/31/72
		D-47251	3	01/01/73 - 12/31/73
		D-47252	4	01/01/74 - 12/31/74
DR02953	DS02953	D-47253	1	01/01/75 - 12/31/75
		D-47254	2	01/01/76 - 12/31/76
		D-47255	3	01/15/77 - 12/31/77
		D-47256	4	01/01/78 - 12/31/79

REQ. AGENT	RAND NO.	ACQ. AGENT
LSM	V0121	WSC

PIONEER 10 and 11

SOLAR WIND PROTON BULK SPEED DATA

72-012A-13B

73-019A-13A

This data set catalog consists of 13 tapes. The tapes are 800, binary, 9 track and single filed. The tapes were created on the IBM 360 computer. The D and C numbers with the time span are as follows:

D#	C#	Time Span	S/C
D-47249	C-22121	04/18/72 - 06/23/72	PIONEER 10
D-47250	C-22122	06/24/72 - 12/31/72	↓
D-47251	C-22123	01/01/73 - 12/31/73	↓
D-47252	C-22124	01/01/74 - 12/31/74	↓
D-47253	C-22125	01/01/75 - 12/31/75	↓
D-47254	C-22126	01/01/76 - 12/31/76	↓
D-47255	C-22127	01/15/77 - 12/31/77	↓
D-47256	C-22128	01/01/78 - 12/31/79	↓
D-41558	C-21077	04/21/73 - 12/31/73	PIONEER 11
D-41559	C-21078	01/01/74 - 12/31/74	↓
D-41560	C-21079	01/01/75 - 04/12/75	↓
D-41561	C-21080	05/05/78 - 12/30/78	↓
D-47257	C-22129	01/01/79 - 12/31/79	↓

National Aeronautics and
Space Adr tration

Ames Research Center
Moffett Field, California 94035

Cpo - 13
D.S. name Solar wind Proton Bulk Speed data
Pioneer 10/72-012A-13B 8 tapes
Pioneer 11/73-019A-13A 1 tape

NASA

Reply to Attn of: SST:245-3

January 13, 1982

Mrs. W. Cameron
NSSDC, Code 601
NASA-Goddard Space Flight Center
Greenbelt, MD 20771

Dear Mrs. Cameron:

With this letter are 9 magnetic tapes that contain solar wind proton bulk speeds in the spacecraft reference frame from Pioneer 10 from experiment turn-on (1972) through 1979, and for Pioneer 11 for 1979, all according to the tape list that is also included. The tape format is the same as for the four Pioneer 11 proton speed tapes sent to Jim Vette in late July, 1980, who also requested late last month a description of the Pioneer 10 and 11 proton flow direction, temperature and density data sets. Those parameters are calculated, along with the aforementioned speeds, from a least-squares fit of a convecting, isotropic temperature, Maxwellian proton velocity distribution to the proton flight data. The flow direction angles refer to the (moving) spacecraft reference frame, and are defined with respect to the spacecraft spin axis and the ecliptic plane.

Sincerely,

John D. Mihalov

John D. Mihalov
Research Scientist
Theoretical and Planetary Studies Branch

Enclosures:
Tape list
Magnetic tapes (9 each)

Pn 10 & 11 L.S. Velocities for Space Science Center

FOR DOCUMENTATION SEE CATALOG #470 Pioneer 11 73-019A-13A

	1	2	3	4	5	6	Cylindex 8.	9
1	Pioneer	11	1979					
1979 2	103B	90979A	D 001-	365			19958/2	
3								
4								
5	Pioneer	10						
6								
1972	242B	90980A	D176	366			8259/4	
8								
1972 9	363A	90981A	D109	176			15362/1	
10								
1973 11	204B	90982	D001	365			12174 ac 12187/2	
12								
1974 13	198B	90983A	001-033,	56-365			12441/2	
14								
1975 15	158B	90984A	001	365			10227/2	
16								
1976 17	157B	90985A	001	366			3816/2	
18								
1977 19	2891B	90986A	015-125,	128-365			6806/2	
20								
1978 21	2034B	90987A	78/001	79/365			8644/2	
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

73-019A-13

NASA

Reply to Attn of: SST:245-3

July 23, 1980

Dr. J.I. Vette, Director
National Space Science Data Center
NASA-Goddard Space Flight Center
Greenbelt, MD 20771

Dear Dr. Vette:

Under separate cover four magnetic tapes are being sent to you that contain a solar wind proton bulk speed record from the Ames plasma analyzer on Pioneer 11 for 1974, 1973 beginning April 21, 1975 ending April 12, and 1978 beginning May 5. The variables on the tapes are TI01, 2, and 6, CS01, EP07 and PV07. Copies of two reports are included with the tapes; one report gives the tape description and format, and a description of the variables, and the other gives a listing of an IBM 360 program that will read the tapes.

Sincerely yours,

John D. Mihalov

John D. Mihalov 8.448-5516
Research Scientist
Theoretical and Planetary Studies Branch

WORD 3	TI01	YEAR	INTEGER
" 4	TI02	DAY OF YEAR	INTEGER
" 5	TI06	MILLISECONDS OF DAY	INTEGER
" 6	CS01	CHI-SQUARE	FLOATING POINT
" 7	EP07	ERROR WITH FREE PROTON BULK VELOCITY (KM/SEC)	FLOATING POINT
" 8	PV07	FREE PROTON BULK VELOCITY (KM/SEC)	FLOATING POINT

WORDS 1 & 2 - 360 CONTROL WORDS

Project 1335-2 - Technical Note 6

Originator: Alice Crema

Subject: Plasma Parameter
Data Format
Block II and Block III

Date: December 28, 1973
April 26, 1974 Revision 1
Oct. 20, 1975 Revision 2

TABLE OF CONTENTS

Part 1 PLASMA PARAMETER DATA FORMAT

Part 2 PLASMA PARAMETER VARIABLE CODE NAMES AND DESCRIPTIONS

Part 3 PLASMA PARAMETER DATA DESCRIPTOR RECORDS

The Plasma Parameter Data exists on 9-track magnetic tape and/or disk dataset. The data on a 9-track magnetic tape is written using density of 800 BPI and variable sequential record format. The logical record length equals 4112 and the block size equals 4116.

The Plasma Parameter data contains an infinite number of variables within ID information, proton parameter and alpha parameter categories. New variables may be added at any time. The Plasma Parameter datasets are set up with descriptor records which contain the four letter code names of the variables existing in that type of data record. The descriptor records are at the beginning of the dataset describing the data existing on that dataset. The purpose of the descriptor records is to give flexibility to the data records. If, for example, a new proton variable is to be added to the proton data records, a new descriptor record would be added and no change would be needed to the existing data records. For example, there is a descriptor record for the Block III one-temperature proton parameter model and also a descriptor for the Block II one-temperature alpha parameter model.

The Plasma Parameter Data records are read with routine SDTAPE. Refer to Technical Note 1335-2-7 for the description of routine SDTAPE.

Part 2

PLASMA PARAMETER VARIABLE CODE NAMES AND DESCRIPTIONS

T01 = FREE ALPHA TEMPERATURE (DEGREES KELVIN)
AT02 = FREE ALPHA TEMPERATURE PERPENDICULAR (DEGREES KELVIN)
AT03 = FREE ALPHA TEMPERATURE PARALLEL (DEGREES KELVIN)
AT04 = SPARE
AT05 = SPARE
AT06 = SPARE
AV07 = FREE ALPHA BULK VELOCITY (KM/SEC)
AA08 = FREE ALPHA BULK AZIMUTHAL ANGLE (DEGREES)
AA09 = FREE ALPHA BULK POLAR ANGLE (DEGREES)
AA10 = FREE ALPHA NUMBER DENSITY (PROTONS/CC)
AA11 = FREE ALPHA ANISOTROPIC AZIMUTHAL ANGLE (DEGREES)
AA12 = FREE ALPHA ANISOTROPIC POLAR ANGLE (DEGREES)

AF51 = FIXED ALPHA TEMPERATURE (DEGREES KELVIN)
AF52 = FIXED ALPHA TEMPERATURE PERPENDICULAR (DEGREES KELVIN)
AF53 = FIXED ALPHA TEMPERATURE PARALLEL (DEGREES KELVIN)
AF54 = SPARE
AF55 = SPARE
AF56 = SPARE
AF57 = FIXED ALPHA BULK VELOCITY (KM/SEC)
AF58 = FIXED ALPHA BULK AZIMUTHAL ANGLE (DEGREES)
AF59 = FIXED ALPHA BULK POLAR ANGLE (DEGREES)
AF60 = FIXED ALPHA NUMBER DENSITY (PROTONS/CC)
AF61 = FIXED ALPHA ANISOTROPIC AZIMUTHAL ANGLE (DEGREES)
AF62 = FIXED ALPHA ANISOTROPIC POLAR ANGLE (DEGREES)

Key: △ Data on Pn 8-9 tapes
○ Data on Pn 10-11 tapes

CS01 = Chi-Square

EA01 = ERROR WITH ALPHA TEMPERATURE (DEGREES IN KELVIN)
EA02 = ERROR WITH ALPHA TEMPERATURE PERPENDICULAR (DEGREES IN KELVI
EA03 = ERROR WITH ALPHA TEMPERATURE PARALLEL (DEGREES IN KELVIN)
EA04 = SPARE
EA05 = SPARE
EA06 = SPARE
EA07 = ERROR WITH ALPHA BULK VELOCITY (KM/SEC)
EA08 = ERROR WITH ALPHA BULK AZIMUTHAL ANGLE (DEGREES)
EA09 = ERROR WITH ALPHA BULK POLAR ANGLE (DEGREES)
EA10 = ERROR WITH ALPHA NUMBER DENSITY (PROTONS/CC)
EA11 = ERROR WITH ALPHA ANISOTROPIC AZIMUTHAL ANGLE (DEGREES)
EA12 = ERROR WITH ALPHA ANISOTROPIC POLAR ANGLE (DEGREES)
EP01 = ERROR WITH FREE PROTON TEMPERATURE (DEGREES KELVIN)
EP02 = ERROR WITH FREE PROTON TEMPERATURE PERPENDICULAR (DEGREES KELVIN)
EP03 = ERROR WITH FREE PROTON TEMPERATURE PARALLEL (DEGREES KELVIN
EP04 = ERROR WITH FREE PROTON TEMPERATURE PERPENDICULAR (DEGREES KELVIN)
EP05 = ERROR WITH FREE PROTON TEMPERATURE INWARD (DEGREES KELVIN)
EP06 = ERROR WITH FREE PROTON TEMPERATURE OUTWARD (DEGREES KELVIN)
EP07 = ERROR WITH FREE PROTON BULK VELOCITY (KM/SEC)
EP08 = ERROR WITH FREE PROTON BULK AZIMUTHAL ANGLE (DEGREES)
EP09 = ERROR WITH FREE PROTON BULK POLAR ANGLE (DEGREES)
EP10 = ERROR WITH FREE PROTON NUMBER DENSITY (PROTON/CC)
EP11 = ERROR WITH FREE PROTON ANISOTROPIC AZIMUTHAL ANGLE (DEGREES)
EP12 = ERROR WITH FREE PROTON ANISOTROPIC POLAR ANGLE (DEGREES)
EP13 = ERROR WITH FREE PROTON TEMPERATURE PERPENDICULAR FOR COMPONENT 1 (DEGREES KELVIN)
EP14 = ERROR WITH FREE PROTON ANISOTROPIC AZIMUTHAL ANGLE OF 'EP13' (DEGREES)
EP15 = ERROR WITH FREE PROTON ANISOTROPIC POLAR ANGLE OF 'EP13' (DEGREES)
EP16 = ERROR WITH FREE PROTON TEMPERATURE PERPENDICULAR FOR COMPONENT 2 (DEGREES KELVIN)
EP17 = ERROR WITH FREE PROTON ANISOTROPIC AZIMUTHAL ANGLE OF 'EP16' (DEGREES)
EP18 = ERROR WITH FREE PROTON ANISOTROPIC POLAR ANGLE OF 'EP16' (DEGREES)

ID01 = MODE: 0 = UNSUPPRESSED/1 = SUPPRESSED
ID02 = SPACECRAFT ID NUMBER
ID03 = BIT RATE
ID04 = MODE: 0 = FSM/1 = MFM
ID05 = CALIBRATION IDENTIFICATION FOR BLOCK 2
IDA5 = FIRST PART OF CALIBRATION IDENTIFICATION FOR BLOCK 3
IDB5 = SECOND PART OF CALIBRATION IDENTIFICATION FOR BLOCK 3
ID06 = TYPE OF PROTON OR ALPHA MODEL
ID07 = MODEL NUMBER
ID08 = DETECTOR: 0=A/1=B
ID09 = MODE: 1,2,3,4
ID10 = NUMBER OF STEPS/CYCLE
ID11 = FOR DETECTOR A: SECTOR INTEGRATION PERIOD
ID12 = NUMBER OF HALF-REVOLUTIONS PER STEP
ID13 = COLLECTOR NUMBER OF PEAK REPORT
ID14 = ORDINAL NUMBER FOR THIS CYCLE IN COMBINED CYCLE (E.G.,1)
ID15 = NO. OF CYCLES COMBINED WITH AND INCLUDING THIS CYCLE (E.G.,2)
ID16 = SPARE
ID17 = SPARE
ID18 = SPARE
19 = SPARE
ID20 = SPARE
ID21 = SPARE
ID22 = SPARE

PF51 = FIXED PROTON TEMPERATURE (DEGREES KELVIN)
PF52 = FIXED PROTON TEMPERATURE PERPENDICULAR (DEGREES KELVIN)
PF53 = FIXED PROTON TEMPERATURE PARALLEL (DEGREES KELVIN)
PF54 = FIXED PROTON TEMPERATURE PERPENDICULAR (DEGREES KELVIN)
PF55 = FIXED PROTON TEMPERATURE INWARD (DEGREES KELVIN)
PF56 = FIXED PROTON TEMPERATURE OUTWARD (DEGREES KELVIN)
PF57 = FIXED PROTON BULK VELOCITY (KM/SEC)
PF58 = FIXED PROTON BULK AZIMUTHAL ANGLE (DEGREES)
PF59 = FIXED PROTON BULK POLAR ANGLE (DEGREES)
PF60 = FIXED PROTON NUMBER DENSITY (PROTONS/CC)
PF61 = FIXED PROTON ANISOTROPIC AZIMUTHAL ANGLE (DEGREES)
PF62 = FIXED PROTON ANISOTROPIC POLAR ANGLE (DEGREES)
PF63 = FIXED PROTON TEMPERATURE PERPENDICULAR FOR COMPONENT 1 (DEGREES KELVIN)
PF64 = FIXED PROTON ANISOTROPIC AZIMUTHAL ANGLE FOR COMPONENT 1 (DEGREES)
PF65 = FIXED PROTON ANISOTROPIC POLAR ANGLE FOR COMPONENT 1 (DEGREES)
PF66 = FIXED PROTON TEMPERATURE PERPENDICULAR FOR COMPONENT 2 (DEGREES KELVIN)
F67 = FIXED PROTON ANISOTROPIC AZIMUTHAL ANGLE FOR COMPONENT 2 (DEGREES)
PF68 = FIXED PROTON ANISOTROPIC POLAR ANGLE FOR COMPONENT 2 (DEGREES)

PT01 = FREE PROTON TEMPERATURE (DEGREES KELVIN)
PT02 = FREE PROTON TEMPERATURE PERPENDICULAR (DEGREES KELVIN)
PT03 = FREE PROTON TEMPERATURE PARALLEL (DEGREES KELVIN)
PT04 = FREE PROTON TEMPERATURE PERPENDICULAR (DEGREES KELVIN)
PT05 = FREE PROTON TEMPERATURE INWARD (DEGREES KELVIN)
PT06 = FREE PROTON TEMPERATURE OUTWARD (DEGREES KELVIN)
PV07 = FREE PROTON BULK VELOCITY (KM/SEC)
PA08 = FREE PROTON BULK AZIMUTHAL ANGLE (DEGREES)
PA09 = FREE PROTON BULK POLAR ANGLE (DEGREES)
PN10 = FREE PROTON NUMBER DENSITY (PROTON/CC)
PA11 = FREE PROTON ANISOTROPIC AZIMUTHAL ANGLE (DEGREES)
PA12 = FREE PROTON ANISOTROPIC POLAR ANGLE (DEGREES)
PT13 = FREE PROTON TEMPERATURE PERPENDICULAR FOR COMPONENT 1 (DEGREES KELVIN)
PA14 = FREE PROTON ANISOTROPIC AZIMUTHAL ANGLE OF 'PT13' (DEGREES)
PA15 = FREE PROTON ANISOTROPIC POLAR ANGLE OF 'PT13' (DEGREES)
PT16 = FREE PROTON TEMPERATURE PERPENDICULAR FOR COMPONENT 2 (DEGREES KELVIN)
PA17 = FREE PROTON ANISOTROPIC AZIMUTHAL ANGLE OF 'PT16' (DEGREES)
PA18 = FREE PROTON ANISOTROPIC POLAR ANGLE OF 'PT16' (DEGREES)

RN01 = NOISE LEVEL FOR COLLECTOR 1 (1.E-14 AMPS)
RN02 = NOISE LEVEL FOR COLLECTOR 2 (1.E-14 AMPS)
RN03 = NOISE LEVEL FOR COLLECTOR 3 (1.E-14 AMPS)
RN04 = NOISE LEVEL FOR COLLECTOR 4 (1.E-14 AMPS)
RN05 = NOISE LEVEL FOR COLLECTOR 5 (1.E-14 AMPS)
RN06 = SPARE
RN07 = SPARE
RN08 = SPARE
RN09 = SPARE
RN10 = SPARE
RN11 = SPARE
RN12 = SPARE
RN13 = SPARE
RN14 = SPARE
RN15 = SPARE
RN16 = SPARE
RN17 = SPARE
RN18 = SPARE
RN19 = SPARE
RN20 = SPARE
RN21 = SPARE
RN22 = SPARE
RN23 = SPARE
RN24 = SPARE
RN25 = SPARE
RN26 = SPARE
RP01 = RATIO OF PEAK REPORT/NOISE (1.E-13 AMPS)
RP02 = PEAK REPORT (1.E-14 AMPS)

O△ TI01 = YEAR
O△ TI02 = DAY OF YEAR
O△ TI06 = MILLISECONDS OF DAY OF YEAR

O△ TI91 = YEAR DATA WAS GENERATED
O△ TI92 = DAY OF YEAR DATA WAS GENERATED

Part 3

PLASMA PARAMETER DATA DESCRIPTOR RECORDS

TI01	TI01	TI01	TI01	TI01	TI01	TI02		TI02	
TI02	TI02	TI02	TI02	TI02	TI02	TI06		TI06	
TI06	TI06	TI06	TI06	TI06	TI06	ID01		ID01	
ID01	ID01	ID01	ID01	ID01	ID01	ID02		ID02	
ID02	ID02	ID02	ID02	ID02	ID02	ID03		ID03	
ID03	ID03	ID03	ID03	ID03	ID03	ID04		ID04	
ID	ID04	ID04	ID04	ID04	ID04	IDA5		IDA5	
ID05	ID05	ID05	IDA5	IDA5	IDA5	IDB5		IDB5	
ID06	ID06	ID06	IDB5	IDB5	IDB5	ID07		ID07	
TI91	TI91	TI91	ID07	ID07	ID07	ID08		ID08	
TI92	TI92	TI92	ID08	ID08	ID08	ID09		ID09	
CS01	CS01	CS01	ID09	ID09	ID09	ID10		ID10	
PT01	PT02	PT04	ID10	ID10	ID10	ID11		ID11	
PV07	PT03	PT05	ID11	ID11	ID11	ID12		ID12	
PA08	PV07	PT06	ID12	ID12	ID12	ID13		ID13	
PA09	PA08	PV07	ID13	ID13	ID13	ID14	AS	ID14	AS
PN10	PA09	PA08	ID14	ID14	ID14	ID15		ID15	
EP01	PN10	PA09	ID15	ID15	ID15	TI91	FOR	TI91	FOR
EP07	PA11	PN10	ID16	ID16	ID16	TI92		TI92	
EP08	PA12	PA11	ID17	ID17	ID17	PT01	7	PT02	9
EP09	EP02	PA12	ID18	ID18	ID18	PT03		PT03	
EP10	EP03	EP04	ID19	ID19	ID19	PV07		PV07	
PF51	EP07	EP05	ID20	ID20	ID20	PA08		PA08	
PF57	EP08	EP06	ID21	ID21	ID21	PA09		PA09	
PF58	EP09	EP07	ID22	ID22	ID22	PN10		PN10	
PF59	EP10	EP08	TI91	TI91	TI91	EP01		PA11	
PF60	EP11	EP09	TI92	TI92	TI92	EP07		PA12	
	EP12	EP10	CS01	CS01	CS01	EP08		EP02	
	PF52	EP11	RP01	RP02	RP02	EP09		EP03	
	PF53	EP12	RN01	RN01	RN01	EP10		EP07	
	PF57	PF54	RN02	RN02	RN02	CS01		EP08	
	PF58	PF55	RN03	RN03	RN03	RP02		EP09	
	PF59	PF56	RN04	RN04	RN04	RN01	RN01	EP10	
	PF60	PF57	RN05	RN05	RN05	RN02	RN02	EP11	
	PF61	PF58	PT01	PT02	PT01	.	.	EP12	
	PF62	PF59	PV07	PT03	PV07	.	.	CS01	
		PF60	PA08	PV07	PA08	.	RN05	RP02	
		PF61	PA09	PA08	PA09	.		RN01	RN01
		PF62	PN10	PA09	PN10	.		RN02	RN02
			EP01	PN10	EP01	.		.	.
			EP07	PA11	EP07	.		.	.
			EP08	PA12	EP08	.		.	RN05
			EP09	PT13	EP09	.		.	
			EP10	PA14	EP10	.		.	
				PA15		.		.	
				PT16		RN26		.	
				PA17				.	
				PA18				RN26	
				EP02					
				EP03					
				EP07					
				EP08					
				EP09					
				EP10					
				EP11					
				EP12					
				EP13					
				EP14					
				EP15					
				EP16					
				EP17					
				EP18					

27	36	39	44	53					38	NWDS
TI01	TI01	TI01	TI01						TI01	
TI02	TI02	TI02	TI02	FIRST					TI02	
TI06	TI06	TI06	TI06						TI06	
ID01	ID01	ID01	ID01	53					ID01	
ID02	ID02	ID02	ID02						ID02	
ID03	ID03	ID03	ID03	ITEMS					ID03	
ID04	ID04	ID04	ID04						ID04	
ID05	ID05	ID05	IDA5	OF					IDA5	
ID06	ID06	ID06	IDB5						IDB5	
TI91	TI91	TI91	ID07	JTYPE					ID07	
TI92	TI92	TI92	ID08						ID08	
CS01	CS01	CS01	ID09	7					ID09	
PT01	AT02	AT04	ID10						ID10	
PV07	AT03	AT05	ID11	(LAST					ID11	
PA08	AV07	AT06	ID12						ID12	
PA09	AA08	AV07	ID13	5					ID13	
PN10	AA09	AA08	ID14						ID14	
EP01	AN10	AA09	ID15	MISS-					ID15	
EP07	AA11	AN10	ID16						ID16	
EP08	AA12	AA11	ID17	ING)					ID17	
EP09	EA02	AA12	ID18						ID18	
EP10	EA03	EA04	ID19						ID19	
AF51	EA07	EA05	ID20						ID20	
AF57	EA08	EA06	ID21						ID21	
AF58	EA09	EA07	ID22						ID22	
AF59	EA10	EA08	TI91						TI91	
AF60	EA11	EA09	TI92						TI92	
	EA12	EA10	CS01						CS01	
	AF52	EA11	RP02						PT01	
	AF53	EA12	RN01						PV07	
	AF57	AF54	RN02						PA08	
	AF58	AF55	RN03						PA09	
	AF59	AF56	RN04						PN10	
	AF60	AF57	RN05						EP01	
	AF61	AF58	AT01						EP07	
	AF62	AF59	AV07						EP08	
		AF60	AA08						EP09	
		AF61	AA09						EP10	
		AF62	AN10							
			EA01							
			EA07							
			EA08							
			EA09							
			EA10							

						NWDS
TI01						
TI02						
TI06						
ID01						
ID02						
ID03						
ID04						
IDA5						
IDB5						
ID07						
ID08						
ID09						
ID10						
ID11						
ID12						
ID13						
ID14						
ID15						
ID16						
ID17						
ID18						
ID19						
ID20						
ID21						
ID22						
TI91						
TI92						
CS01						
PT02						
PT03						
PV07						
PA08						
PA09						
PN10						
PA11						
PA12						
PT13						
PA14						
PA15						
PT16						
PA17						
PA18						
EP02						
EP03						
EP07						
EP08						
EP09						
EP10						
EP11						
EP12						
EP13						
EP14						
EP15						
EP16						
EP17						
EP18						

Page 3 of 3

Project 1335-2 - Technical Note 7.

Originator: Nelson Beebe
Alice Crema

Date: June 19, 1973

Subject: Program Description of Subroutine SDTAPE
Revision 1

TABLE OF CONTENTS

	Page
Program Description of Subroutine SDTAPE	1
Purpose	1
Calling Sequence	1
Common Blocks	1
Input/Output	2
Key Variables	5
Explanatory Record	8
Method	9
Program Listing	11
Cross Reference List	17
Appendix A (Example Program Using SDTAPE)	23

PROGRAM DESCRIPTION

of

SUBROUTINE SDTAPE

PURPOSE

To be used by all programs that need input/output from Plasma parameter data sets on disk or tape.

CALLING SEQUENCE

CALL SDINIT to initialize routine

CALL SDTAPE to obtain data

SDINIT must be called before any call to SDTAPE in a run, and must be called again whenever the user wishes to change the list of desired variables, keys, or other user-supplied parameters.

COMMON BLOCKS

COMMON/SDVAR/IDVAR (100), VARVAL(100)

COMMON/SDKEYS/IDKEY(5), KEYVAL(2, 5)

COMMON/SDINFO/NDATRC, NDESRC, KTAPUN, KPRTUN, JTYPE, NWDS, ISPARE(12)

COMMON/SDNEWV/LVNEW(100)

COMMON/SDFLAG/LEOF, LERR, LOVFL, LBADIN, LPRTEX, LRD1, LBADDR, LSPARE(13)

COMMON/SDDATA/DATA(1024)

INPUT/OUTPUT

User Input

a. Variable identifiers of all desired variables should be put in successive words of array IDVAR. At least the first unused word (if any) must be set = -1.

b. IDKEY should be filled with the variable numbers (subscript of IDVAR) of variables to be checked to determine acceptability of a record. KEYVAL should contain the lowest and highest values within which that variable must lie to accept a record containing it. To be accepted, a record must satisfy all the applicable keys. Unused positions in IDKEY should be padded with -1's (first unused one is enough). Only variables appearing on the tape in integer format should be used. To use JTYPE as a key, IDKEY should contain a 0.

c. FORTRAN data set reference number for the input tape (KTAPUN) and the printout data set (KPRTUN) should be supplied.

d. If the user desires explanatory tape records to be printed out, he should set LPRTEX to .TRUE. Otherwise, it should be set to .FALSE. Explanatory tape records are numbered in the record type by IRT+1000, where IRT is the record type of the record being explained. Each element in an explanatory record contains an 80-byte verbal explanation of the variable occupying its position in the explained record.

Subroutine Output

a. For each call to SDTAPE, the tape will be read until a data record is encountered which contains at least one desired variable, and which satisfies all the keys which appear in it. All desired variables in the record will have their values stored in VARVAL, in the same relative position as their identifiers in IDVAR.

b. Upon return from SDTAPE, NDATRC will contain the number of data records read in the last call, NDESRC will contain the number of descriptor records read, and JTYPE will contain the type of the last record read.

c. Upon return from SDTAPE, LVNEW will contain a .TRUE. in every position corresponding to variables in VARVAL receiving new values, and a .FALSE. elsewhere.

d. Upon return from SDTAPE, LEOF will be .TRUE. if an end-of-file was reached; LERR will be .TRUE. if an error condition occurred in reading the tape, LOVFL will be .TRUE. if more record types contain desired variables than the table can handle (25).

e. Upon return from SDINIT, LBADIN will be set to .TRUE. if some erroneous condition was found in the user's input. (e.g., no variables requested).

f. The subroutine reads into array DATA, so upon return, it contains the last tape record, and NWDS contains the number of words in it.

g. If the user sets LRD1 to .TRUE., SDTAPE will return after processing one record, whether or not it contains any relevant variable.

h. LBADDR will be set .TRUE., by SDTAPE if:

1. a second descriptor for a record type is read
2. a data record shorter than its descriptor is read
3. an undescribed data record is read

KEY VARIABLES

Common Variables

Name	Type	Description
DATA(1024)	Real	Array containing all data from record read
IDKEY(5)	integer	Filled with the variable numbers (subscripts of IDVAR) of variables to be checked to determine acceptability of a record.
IDVAR(100)	integer	User input of code names of parameters desired
ISPARE(12)	integer	Spare array
JTYPE	integer	Type of data record read
KEYVAL(2, 5)	integer	Contains the lowest highest values for variables in IDVAR to accept a record containing it
KPRTUN	integer	Unit number of print output definition
KTAPUN	integer	Unit number of data definition
LBADIN	logical	Flag for bad user input
LBADDR	logical	Flag for bad descriptor record
LEOF	logical	Flag when end of file is reached
LERR	logical	Flag when there exist an error in data read
LOVFL	logical	Flag for an overflow of descriptor records (25 allowed)
LPRTEX	logical	Flag for printing explanatory records
LRD1	logical	Flag used to return after each record read
LSPARE(13)	logical	Spare array
LVNEW(100)	logical	Flag set for new values put into array VARVAL
NDATRC	integer	The number of data records read since last call
NDESRC	integer	The number of descriptor records read since last call
NWDS	integer	The number of words in the last record read
VARVAL(100)	integer	Parameter data corresponding to code names in IDVAR

Internal Variables of Importance

VPTR(2, 100, 25) Array linking IDVAR subscript to record subscript for each record type. Thus for record type j, with RTPTR (j) = k, VPTR(1, i, k) contains the subscript of the record where the desired variable is stored.

VPTR (2, i, k) contains the subscript of VARVAL to which that value (i. e., data (VPTR(1, i, k))) should be moved.

The subscript i is used to store the cross-reference pairs for a record. Thus if this record contains n desired variables the program will loop from i=1, n moving data from DATA (VPTR(1, i, k)) to VARVAL (VPTR(2, i, k)).

KPTR(2, 5, 25) Same as VPTR, but links to KEYVAL pairs instead of VARVAL.

RTPTR(500) For each record type code i, RTPTR (i) will contain:

- -1 if no descriptor for that type has been encountered
- 0 if a descriptor record has been encountered, and the record has no desired variables
- k> 0 points to that value of the 3rd subscript of VPTR, KPTR, RTLEN, NVP, and NKP where references to that record type (i) are found

Internal Variables of Importance

MCRT =(25)	The largest allowed number of record types with desired data which the program can handle. Equals the dimension of NVP, NKP, RTLEN, and the third dimension of VPTR and KPTR
RTLEN(25)	RTLEN (j) contains the length (not counting length and rec. type words) of records of type i, where RTPTR (i) = j
NKP(25)	Contains the no. of IDKEY variables contained in the record
NVP(25)	Contains the no. of IDVAR variables contained in the record
NCRT	Contains the current no. of active record types (i.e., the last j used where RTPTR (i) = j for the last record type i processed with desired variables).

EXPLANATORY RECORD

Note that explanatory records still must be less than 1024 words long. Therefore, any record type of length >51 words cannot presently be "explained".

One variation on the explanation record is to have only the first 4 bytes of explanation for each variable in record type IRT+1000 then have additional records with types IRT+2000, IRT+3000, etc. each containing an additional 4 bytes of explanation. What the maximum number of such records should be, and whether or not a number less than the max. is considered an error, should be decided. I suggest a max. of 32 with less being permitted and encouraged.

METHOD

There are two parts to the routine, the initialization,(SDINIT entry) and the read (SDTAPE entry).

When entry is via SDINIT the user initialization information (IDVAR, IDKEY, KEYVAL) is checked for consistency and, if bad, LBADIN is set .TRUE. , a message is printed, and return is made. If consistent we just return.

When entry is via SDTAPE, a record is read from KTAPUN. The record type is checked. If it is an explanatory record, (type >1000) the content is printed if LPRTEX is .TRUE., and in either case we read the next record from KTAPUN.

If the record type is negative (descriptor record), array RTPTR is checked with the subscript of the absolute value of the record type. If that value is 0 or positive, an error exit is taken, otherwise (that is, if this record type has not been encountered before) RTPTR (type) is set to NCRT + 1, NCRT is incremented, and the record is scanned for matches to one of the IDVAR variable names. When a match is found, NVP(j) (where j = RTPTR (type) = NCRT) is incremented, and VPTR (1, i, j), where i = NVP(j) is set to k, where k is the current subscript of DATA being examined, and VPTR (2, i, j) is set to m, where m is the subscript of IDVAR at which the match was found. If no matches are found, NCRT is decremented and RTPTR (type) is set to 0. The scan goes from 1 to NRP(j) (= length of the record) and is repeated for IDKEY filling KPTR.

If the record type is positive (data record) an error exit is taken if RTPTR (type) is -1, and the next record is read if RTPTR (type) = 0.

If RTPTR (type) is positive, the program loops from i = 1 to NKP (j) where j = RTPTR (type) checking DATA (KPTR (1, i, j)) against IDVAL (k, KPTR(2, i, j)), k=1, 2, and if all checks are successful, it loops from 1 to NVP(j) moving DATA (VPTR(1, i, j)) to VARVAL (VPTR(2, i, j)).

If LRD1 is .TRUE. the routine returns after one record is processed regardless of type.

```
      SUBROUTINE SDTAPE
C
C     *          ROUTINE TO READ SELF-DESCRIBING TAPES COMPOSED OF VARIABLE LENGTH RECORDS PRECED
C     *          RECORDS PROVIDING KEYS TO LINK EACH POSITION IN RECORDS OF A SPECIFIC 'TYPE' TO
C     *          VARIABLE IDENTIFIERS.  USER PROVIDES LIST OF VARIABLES DESIRED, AND IS RETURNED
C     *          ALL SUCH VARIABLES IN EACH RECORD.
C
      IMPLICIT INTEGER(A-K,M-Z), LOGICAL (L)
C
      COMMON /SDVAR/ VARID(100), VAR(100)
C
      COMMON /SDKEYS/ KNU(5), KVAL(2,5)
C
      COMMON /SDINFO/ NDATR, NDESR, TAPE, PRT, JRT, NW, ISPAR1(12)
C
      COMMON /SDNEWV/ LVN(100)
C
      COMMON /SDFLAG/ LEOF, LERR, LOVFL, LBADIN, LPRTX, LRD1, LBADDR, LSPAR(13)
C
      COMMON /SDDATA/ REC(1024)
C
C
      INTEGER ERR(5,5)/' NO ','VARI','ABLE','S AS','KED.',' KEY',' > L','IST ','LENG','TH. ', -
     1          ' KVA','L2 <',' KVA','L1. ','    ', 'DUPL','ICAT','E KE','YS. ',6*'    '/
C
      INTEGER MCRT/25/, VPTR(2,100,25), KPTR(2,5,25), RTPTR(500), NVP(25), NKP(25)
      INTEGER RTLEN(25)
C
C
C     *          ENTRY SDTAPE - COMES AFTER INITIALIZATION - CODE IS THUS LATER ON
      GO TO 50
C
      ENTRY SDINIT
      LBADIN=.FALSE.
      LCKPT=.FALSE.
C
C     *          CLEAR THE DECKS FOR ACTION
```

```
      NDATR=0
      NDESR=0
      DO 10 I=1,500
      RTPTR(I)=0
   10 CONTINUE
      NCRT=0
C
C     *            CHECK CALLER PARAMETERS
      DO 15 I=1,100
      IF(VARID(I).EQ.-1)GO TO 16
   15 CONTINUE
      NVAR=100
      GO TO 18
   16 IF(I.EQ.1)GO TO 40
      NVAR=I-1
   18 DO 20 I=1,5
      IF(KND(I).EQ.-1)GO TO 21
      IF(KND(I).GT.NVAR)GO TO 41
      IF(KVAL(2,I).LT.KVAL(1,I))GO TO 42
      IF(KND(I).NE.0)GO TO 20
      IF(LCKRT)GO TO 43
      LCKRT=.TRUE.
      JRTCK=I
   20 CONTINUE
      NK=5
      GO TO 900
   21 NK=I-1
      GO TO 900
C
C     *            BAD INPUTS
   40 IERR=1
      GO TO 45
   41 IERR=2
      GO TO 45
   42 IERR=3
      GO TO 45
   43 IERR=4
      GO TO 45
```

```
C
   45 WRITE(PRT,1000)IERR,(FRR(J,IERR),J=1,5)
 1000 FORMAT(//' INPUT ERROR',I2,',',5A4//)
      LBADIN=.TRUE.
      GO TO 900.
C
C     *            SDTAPE ITSELF - FIRST READ THE RECORD, OBVIOUSLY.
   50 LEOF=.FALSE.
      NDESR=0
      NDATR=0
      LERR=.FALSE.
      LOVFL=.FALSE.
      LBADDR=.FALSE.
      DO 51 I=1,100
      LVN(I)=.FALSE.
   51 CONTINUE
C
   52 READ(TAPE,END=810,ERR=820)NW,JRT,(REC(I),I=1,NW)
C
C     *            CHECK RECORD TYPE
      IF(JRT)200,830,54
   54 IF(JRT.GE.100)GO TO 700
C
C     *            DATA RECORD
C
C     *            CHECK IF RELEVANT
      NDATR=NDATR+1
      IF(RTPTR(JRT))500,840,55
C
C     *            RECORD TYPE WITH INFO WE WANT
   55 IRT=RTPTR(JRT)
C
C     *            CHECK KEYS, IF ANY
      IF(.NOT.LCKRT)GO TO 60
C
C     *            CHECK RECORD TYPE AGAINST KEYS
      IF(JRT.LT.KVAL(1,JRTCK).OR.JRT.GT.KVAL(2,JRTCK))GO TO 500
   60 IF(NKP(IRT).LE.0)GO TO 70
```

```
      M1=NKP(IRT)
      DO 56 I=1,M1
      M2=KPTR(1,I,IRT)
      M3=KPTR(2,I,IRT)
      IF(REC(M2).LT.KVAL(1,M3).OR.REC(M2).GT.KVAL(2,M3))GO TO 500
C
C     *            PASSES KEY TEST
   56 CONTINUE
C
C     *            FILL VAR ARRAY
   70 IF(RTLEN(IRT).GT.NW)GO TO 860
      M4=NVP(IRT)
      DO 75 I=1,M4
      M5=VPTR(1,I,IRT)
      M6=VPTR(2,I,IRT)
      VAR(M6)=REC(M5)
      LVN(M6)=.TRUE.
   75 CONTINUE
C
C     *            ALL DONE FOR A GOOD DATA RECORD
      GO TO 900
C
C     *            DESCRIPTOR RECORD
  200 KRT=-JRT
      NDESR=NDESR+1
C
C     *            NEW RECORD TYPE - IS THERE ROOM?
      IF(NCRT.GE.MCRT)GO TO 850
      IF(RTPTR(KRT).NE.0)GO TO 870
      NCRT=NCRT+1
      RTPTR(KRT)=NCRT
      RTLEN(NCRT)=NW
C
      NVP(NCRT)=0
      NKP(NCRT)=0
C
C     *            FILL VPTR, THEN KPTR
      DO 220 I=1,NW
```

```
      DO 205 J=1,NVAR
      IF(REC(I).EQ.VARID(J))GO TO 210
  205 CONTINUE
      GO TO 220
  210 NVP(NCRT)=NVP(NCRT)+1
      M7=NVP(NCRT)
      VPTR(1,M7,NCRT)=I
      VPTR(2,M7,NCRT)=J
      IF(NK.LE.0)GO TO 220
      DO 215 K=1,NK
      IF(KNO(K).EQ.J)GO TO 216
  215 CONTINUE
      GO TO 220
C
C     *              FOUND A KEY
  216 NKP(NCRT)=NKP(NCRT)+1
      M8=NKP(NCRT)
      KPTR(1,M8,NCRT)=I
      KPTR(2,M8,NCRT)=K
C
  220 CONTINUE
      IF(NVP(NCRT).GT.0)GO TO 500
      NCRT=NCRT-1
      RTPTR(KRT)=-1
      GO TO 500
C
C     *              DONE WITH THIS RECORD - READ NEXT UNLESS LRD1 IS SET
  500 IF(LRD1)GO TO 900
      GO TO 52
C
C     *              EXPLANATORY RECORD - PRINT IF LPRTEX SET
  700 IF(.NOT.LPRTX)GO TO 500
      J=JRT-1000
      N=NW/20
      WRITE(PRT,1003)J,N
 1003 FORMAT(//' EXPLANATORY RECORD FOR RECORD TYPE',I4,' - ',I5,' ITEMS'/)
      DO 710 I=1,N
      M9=(I-1)*20+1
```

```
      M10=I*20
      WRITE(PRT,1001)(REC(J),J=M9,M10)
 1001 FORMAT(1X,20A4)
  710 CONTINUE
      GO TO 500
C
C     *              ERROR CONDITIONS
C
C     *              EOF
  810 LEOF=.TRUE.
      GO TO 900
C
C     *              ERROR ON READ
  820 LERR=.TRUE.
      GO TO 900
C
C     *              RECTYP OF ZERO
  830 LBADDR=.TRUE.
      GO TO 900
C
C     *              DATA RECORD NOT PREVIOUSLY DESCRIBED
  840 GO TO 830
C
C     *              RECORD TYPE TABLE OVERFLOW
  850 LOVFL=.TRUE.
      GO TO 900
C
C     *              RECORD SHORTER THAN DESCRIPTION
  860 GO TO 830
C
C     *              DESCRIPTOR RECORD FOR TYPE ALREADY DESCRIBED
  870 GO TO 830
C
C
  900 RETURN
      END
```

Cross Reference List

SOTAPE15.08/31/72 12:06:27

17

SYMBOL CROSS-REFERENCE LIST

SYMBOL	DEFINITIONS	REFERENCES
CHCBD1		0
CHCIA1		6800 8200 15900 16200
CHCIE1		6800 8200 15900 16200
CHCIU1		6800 8200 15900 16200
ERR		2400 6800
I	3700 4300 5000 7800 8200 9900 10700 12900 16100	3800 4400 4800 4900 5100 5336 5700 7900 8200 9916 13500 14400 16125 16143
IERR	6100 6300 6500 6612	6800 6800
IRT	5500	9800 9825 9916 9932 106
ISPAR1		1500
J	6800 13000 15700 16200	6800 13100 13600 13800
JRT	8200	1500 8500 8600 9200 95
JRTCK	5336	9736 9736
K	3700	13800 14500
KNO		1300 5100 5200 5312 1
KPTR	14400 14500	2700 9916 9932
KRT	11600	12025 12200 14832
KVAL		1300 5300 5300 9736 9

LBADDR	7700	17800	1900	
LBADIN	3400	7000	1900	
LCKPT	3425	5324	5315	9712
LEOF	7400	17000	1900	
LERR	7500	17400	1900	
LOVFL	7600	18500	1900	
LPRTX			1900	15600
LRD1			1900	15200
LSPAR			1900	
LVN	7900	10900	1700	
MCRT			2700	12000
M1	9825		9900	
M10	16143		16200	
M2	9916		10000	10000
M3	9932		10000	10000
M4	10625		10700	
M5	10725		10800	
M6	10743		10800	10900
M7	13425		13500	13600

MP	14325			
MP	16125			
N	15800			
NCRT	4000	12100	14816	
NDATP	3616	7432	9100	
NDESP	3632	7416	11700	
NK	5500	5700		
NKP	12600	14300		
NVAR	4600	4900		
NVP	12500	13400		
NW	8200			
PRT				
REC	8200			
RTLEN	12300			
RTPTR	3800	12200	14832	
SDDATA	2100			
SDFLAG	1900			
SDINFO	1500			
SDINIT	3300			

14400	14500		
16200			
15900	16100		
12000	12100	12200	12
13425	13500	13600	14
14800	14816		
1500	9100		
1500	11700		
13625	13700		
2700	9800	9825	14300
5200	13000		
2700	10625	13400	1342
1500	8200	10600	12300
1500	6800	15900	16200
2100	10000	10000	10800
2750	10600		
2700	9200	9500	12025

SDTAPE$$,08/31/72 12:06:27

30

SDKEYS	1300	
SDNFWV	1700	
SDTAPE	100	
SDVAR	1100	
TAPE		1500 8200
VAR	10800	1100
VARID		1100 4400 13100
VPTR	13500 13600	2700 10725 10743

SDTAPES$.08/31/72 12:06:27

LABEL CROSS-REFERENCE LIST

LABEL	DEF	REFERENCES
00010	3900	3700
00015	4500	4300
00016	4800	4400
00018	5000	4700
00020	5400	5000 5312
00021	5700	5100
00040	6100	4800
00041	6300	5200
00042	6500	5300
00043	6612	5315
00045	6800	6200 6400 6600 6624
00050	7400	3100
00051	8000	7800
00052	8200	15300
00054	8600	8500
00055	9500	9200
00056	10300	9900
00060	9800	9712
00070	10600	9800
00075	11000	10700
00200	11600	8500
00205	13200	13000
00210	13400	13100
00215	13900	13700
00216	14300	13900

00220	14700	12900	13300	13625	14000					
00500	15200	9200	9736	10000	14800	14900	15600	16500		
00700	15600	8600								
00710	16400	16100								
00810	17000	8200								
00820	17400	8200								
00830	17800	8500	18200	18900	18936					
00840	18200	9200								
00850	18500	12000								
00860	18900	10600								
00870	18936	12025								
00900	19200	5600	5800	7100	11300	15200	17100	17500	17900	18600
01000	6900	6800								
01001	16300	16200								
01002	16000	15900								

Appendix A

EXAMPLE PROGRAM USING SDTAPE

Example program using routine SDTAPE to read parameter data tapes:

```
C***        THIS PROGRAM WILL PRINT PROTON PARAMETERS FOR
C           CYCLES WITH CHI-SQUARES LESS THAN 100.
C
            DIMENSION INPT(11), XVARVL(100)
C
C***        THE FOLLOWING COMMONS USED WITH SDTAPE
            COMMON/SDVAR/IDVAR (100), VARVAL (100)
            COMMON/SDINFO/NDATRC, NDESR, KTAPUN, KPRTUN,
                    JTYPE, NWDS, ISPARE(12)
            COMMON/SDFLAG/LEOF, LERR, LOVFL, LBADIN, LPRTEX,
                    LRD1, LBADDR, LSPARE(13)
            COMMON/SDDATA/DATA(1024)
            COMMON/SDKEYS/IDKEY(5), KEYVAL(2, 5)
C
            LOGICAL *4 LEOF, LERR, LOVFL, LBADIN, LPRTEX, LRD1
            LOGICAL *4 LBADDR, LSPARE
            INTEGER VARVAL
            EQUIVALENCE (VARVAL, XVARVL)
C
C***        CODE NAMES TO BE USED BY SDTAPE
            DATA INPT/'TI01','TI02','TI06', TI91','TI92','CS01',
                    'PT01','PV07','PA08','PA09','PN10' /
C
C***        PUT DATA INPT INTO ARRAY IDVAR FOR SDTAPE
            DO 5 I = 1, 11
            IDVAR (I) = INPT (I)
      5     CONTINUE
C***        NEXT ARRAY MEMBER FOLLOWING INPT CODE NAMES
C           IS SET TO -1
            IDVAR (12) = -1
C
C***        DEFAULTS TO BE SET TO RETURN AFTER EACH CYCLE
            READ FROM SDTAPE
            IDKEY (1) = -1
            LRD1 = .TRUE.
```

```
C***        OUTPUT PRINT UNIT
            KPRTUN = 6
C***        INPUT TAPE UNIT
            KTAPUN =12
C***        PRINT HEADER LINE
            WRITE (KPRTUN, 500)
500         FORMAT
            (1H1,2X,'YEAR DAY TM-IN-MS GEN DATE        CHI',
             6X,'PT01',6X,'PV07',6X,'PA08',6X,'PA09',6X,'PN10')
C***        ROUTINE SDTAPE NEEDS TO BE INITIALIZED FIRST TIME THRU
            CALL SDINIT
100         CALL SDTAPE
C***        CHECK FOR END OF FILE
            IF (LEOF) GO TO 200
C***        CHECK FOR READ ERROR
            IF(LERR) GO TO 210
C***        CHECK FOR CHI-SQUARE TO BE LESS THAN 100.
            IF (XVARVL(6).GT.100.) GO TO 100
C
C***        PRINT CYCLE INFO
            WRITE (KPRTUN,505) (VARVAL(I),I=1,11)
505         FORMAT (1H,2X,2I4,I9,I4,'/',I3,1X,6(6X,F10.5))
C
            GO TO 100
C
200         WRITE (6,201)
201         FORMAT (1H0,2X,'END OF FILE REACHED')
            GO TO 999
210         WRITE (6,211)
211         FORMAT (1H0,2X,'ERROR ON TAPE')
C
999         STOP
            END
```

A parameter tape is a one file 9-track tape with DCB as follows:

DEN = 2 (800 BPI)
RECFM = VS (Variable)
LRECL = 4112
BLKSIZE = 4116

NOTE: In program, code names can be put in any consecutive order in array IDVAR and their values are put in the corresponding array member in VARVAL. e.g.,

```
DIMENSION INPT(6)
DATA INPT/'TI01','TI02','PN10','CS01','TI06','PT01' /
```

( ) 002[illegible] 001C[illegible] [illegible] 48 [illegible] 6D 0226669 43137F29 40C78ED8 41524EE

72-012A-13B

FILE 1 RECORD 2 LENGTH 32BYTES
( ) 002[illegible] 001C[illegible] 00[illegible]48 00000[illegible]6D 0025BEE8 43135CE1 4114C7A2 41611DE9

FILE 1 RECORD 1538 LENGTH 32BYTES
( ) 00200000 001C0000 00000048 00000BAF 04F0D51C 47187B9C 40A2A882 4117F974

DR02952
DS02952

FILE	INPUT RECS.	DATA RECORDS INPUT	MAX. SIZE	READ ERROR SUMMARY PERM	ZERO B	SHORT	UNDEF.	INPUT RETRIES #RECS.	TOTAL#
1	1538	1539	32	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

FILE 4 RECORD 1 LENGTH 74 32BYTES 01
( ) 002[illegible] 001C[illegible] [illegible]4A [illegible]01 0044DD8A 431C72EF 4192E4DB 41F8C4E6

FILE 4 RECORD 12441 LENGTH 32BYTES 16D
( ) 002[illegible] 001C[illegible] [illegible]4A [illegible]6D 04F2D5A8 43249959 4112CA98 41128A0A

74 365

FILE	INPUT RECS.	DATA RECORDS INPUT	MAX. SIZE	READ ERROR SUMMARY PERM	ZERO B	SHORT	UNDEF.	INPUT RETRIES #RECS.	TOTAL#
4	12441	12442	32	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

EOJ DUMP STOPPED AFTER FILE 4 # OF PERMANENT READ ERRORS [illegible]

START TIME 01/21/92 17:17:41 STOP TIME 01/21/92 17:22:11

```
(      ) 002[illegible] 001C[illegible] [illegible] 4B [illegible] 1 045B2D8E 432443CE 4135E9C2 42117E16

FILE      1  RECORD      2  LENGTH      32BYTES
(      ) 002[illegible] 001C[illegible] [illegible]4B [illegible]01 045E92B4 432443CE 4135E9C2 42117E16

          FILE  INPUT   DATA RECORDS   MAX.        READ ERROR SUMMARY        INPUT RETRIES
                RECS.     INPUT        SIZE      PERM  ZERO B   SHORT  UNDEF.  #RECS.  TOTAL#
            1   10727   10727          32         0      0       0      0      0      0

FILE      4  RECORD    8643  LENGTH      32BYTES
(      ) 002[illegible] 001C[illegible] [illegible]4F [illegible]16D 4EF2B1A 43166519 41237ED2 401F74E9

                                  79          365
FILE      4  RECORD    8644  LENGTH      32BYTES
(      ) 002[illegible] 001C[illegible] [illegible]4E [illegible]16D 4517B292 4316637B 412A2A7D 40175445

          FILE  INPUT   DATA RECORDS   MAX.        READ ERROR SUMMARY        INPUT RETRIES
                RECS.     INPUT        SIZE      PERM  ZERO B   SHORT  UNDEF.  #RECS.  TOTAL#
            4    8644    8645          32         0      0       0      0      0      0

EOJ       DUMP STOPPED AFTER FILE   4        # OF PERMANENT READ ERRORS   0

START TIME 1/21/92  18:13:44        STOP TIME  1/21/92  18:17:57
```

DR02953
DS02953

```
(      0)  00200000  001C0000  00000049  0000006F  00206100  432F73F3  4093F269  411AA303
FILE      1  RECORD      2  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  00546FF8  432C3109  412ADCFA  4079128C
FILE      1  RECORD      3  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  00BD0960  432FC379  411DB63B  4226D59D
FILE      1  RECORD      4  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  00C0E8C5  432F2233  40FD59C6  41FD2950
FILE      1  RECORD      5  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  00E9C5A4  432CCD1F  411DC026  4223FBEE
FILE      1  RECORD      6  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  012E9AF8  432CB8F3  41179017  41FF5DC
FILE      1  RECORD      7  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  0166EAFF  432C50C0  41372015  4291C292
FILE      1  RECORD      8  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  019EDA57  432C32F9  41199039  4227[illegible]782
FILE      1  RECORD      9  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  01DE45CE  432C4B73  40FAAB93  4218E01D
FILE      1  RECORD     10  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  0000006F  020F4C22  432B6596  41165973  42112D[illegible]5
FILE      1  RECORD   9403  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  00000160  03B1DF42  431AC1D5  40E220D1  42B5[illegible]3D3
FILE      1  RECORD   9404  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  00000160  03D93172  431AD1D5  40E220D1  42B503D3
FILE      1  RECORD   9405  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  00000160  03F11D3A  431A911B  4081D33F  429AEE[illegible]3
FILE      1  RECORD   9406  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  00000160  03E46908  431A911B  4081D33F  429A0E[illegible]3
FILE      1  RECORD   9407  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  00000160  042[illegible]824E  431A87FE  40857069  4293[illegible]020
FILE      1  RECORD   9408  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  00000160  042F1FF8  431A87E5  40857069  4293C020
FILE      1  RECORD   9409  LENGTH      32BYTES
(      0)  00200000  001C0000  00000049  00000160  04B92330  431A1750  406C031E  429BC160
FILE      1  RECORD   9410  LENGTH      32BYTES
(      0)  00200000  [illegible]
```

2,122 > 3418 sec

5,540 > 3703 sec

9,243 > 3400 "

12,643 > 3678 "

16,321 > 3511 "

19,832 > 3678 "

23,510 > 3678 "

27,188 > 3566 "

30,754 > 3813 "

34,567

61,988

CHEN PRINTING & SUPPLY CO.

```
(      0)   00200000  001C0000  0000004F  00000001  016D37AC  43333384  411FA07E  427ECB00

FILE      1  RECORD   19958  LENGTH      32BYTES
(      0)   00200000  001C0000  0000004F  00000016D  03966412  43187394  414EDFD5  411A3FAE
```

365

FILE	INPUT RECS.	DATA RECORDS INPUT	MAX. SIZE	READ ERROR SUMMARY PERM	ZERO	SHORT	UNDEF.	INPUT RETRIES #RECS.	TOTAL#
1	19958	19959	32	0	0	0	0	0	0

```
EOJ        DUMP STOPPED AFTER FILE   1          # OF PERMANENT READ ERRORS  0

START TIME 02/09/82  13:00:34        STOP TIME 02/09/82  13:04:56
```