

Spool Queue Line #: 26
IRIS LU/Filename : 18/L.TIPD81.9291

Printed on/at : FEB 7, 1990 14:48:36
For Group/User: 0, 1
On Port No: 5

Print control parameters :
Printer Class code : 0
Form Code/paper type : ?
Print Priority (0-9) : 5
Starting Page Number : 1
This is copy number : 1
Keep file (Y/N) : Y
Notify User when done: N
Comments, optional : For RELSE CNTRL

```
.EOT ; "TIP81" (DISCSUBS GROUP 5) FOR IRIS R9. xx  
.EOT  
.EOT ; "DSUBDEFS" FOR IRIS  
.END
```

ASM 18/A. TIPDB1. 9291!, @18/L. TIPDB1. 9291!, B050, -B051, B052
FEB 7, 1990 11:39:57

; Batchfile: R95JCL. TIPDB1

; ; A = 9291

; -R95DEFSPZ
; -R95DSUBDEFSD
; R92TIPDSB81SA

.EOT ; "TIP81" (DISCSUBS GROUP 5) FOR IRIS R9. xx

<< BI = R92TIPDSB81SA; BO = 1B/A.TIPDB1.9291! >>

4 REVIS= 4 ;the revision number

CALL B1,A,D\$,C\$,D,E

A= mode - 0 - return revision number
 mode - 1 - single character search
 searches for first char of B\$ in C\$
 mode - 2 - search for string B\$ within C\$
 mode - 3 - search for next word in C\$, position and length
 returned in D and E respectively, words are
 two or more upper case alpha numeric chracters

 mode - 5 - search for B\$ in C\$, starting at begining, the
 routine assumes that both B\$ and C\$ are terminated
 with zero-word and that both strings and even the
 section to be found are aligned on word boundaries.

B\$= target - string to be found, mode 1 only 1st char used
 mode 2 entire match required

C\$= dest. - string to be searched

D= result - mode 0: D<= revision number
 1: D<= first occurance of B\$[1,1] in C\$
 2: D<= first occurance of B\$ in C\$ after
 start position of D
 3: D<= begining of next word, 0 if none found
 5: D<= begining of B\$ in C\$ if present

E= length of word, mode 3 only, 0 if no word found

			.TXTM 1	
	102400		.LOC LTPO1	
102400	160	DSB160:	TIP01	
102401	3		START-DSB160	
102402	177426		DSB160-DSBEND	
102403	54515	START:	STA 3,RET	;the return address
102404	50513		STA 2,APT	;pointer to the arguments
102405	4437		JSR PICK	;go and get the value of the switch
102406	4412		JSR LSEND	;check size of switch and branch
				;pointers to the various internal routines
102407	46	LSTRT:	V0-LSTRT	;version of this discsub
102410	60		V1-LSTRT	;single character search
102411	131		V2-LSTRT	;string search
102412	131		V3-LSTRT	;spelling dictionary mode
102413	14		EX-LSTRT	
102414	131		V5-LSTRT	;word search mode
102415	14		EX-LSTRT	
102416	14		EX-LSTRT	
102417	14		EX-LSTRT	

```

102420 40417 LSEND: STA 0,SWTCH ;store switch for reference
102421 126420 SUBZ 1,1 ;should not be less than 0
102422 106032 SGE 0,1
102423 2475 EX: JMP @RET ;it is so do an error return
102424 24410 LDA 1,SIZE ;the valid maximum switch
102425 106433 SLE 0,1 ;it should be less than equal
102426 2472 JMP @RET ;its not so do an error return
102427 171000 MOV 3,2 ;A3 has the address of LSTART
102430 117000 ADD 0,3 ;it now has the address of the pointer
102431 35400 LDA 3,0,3 ;load the pointer value
102432 173000 ADD 3,2 ;add the pointer to address of LSTART
102433 1000 JMP 0,2 ;jump to this, the real address

102434 11 SIZE: LSEND-LSTRT ;the number of routines here
102435 0 RETS: 00000 ;used for subroutine RTN addresses
102436 4 RNUM: REVIS ;the revision number
102437 0 SWTCH: 0 ;temporary storage of A

```

```

(;;)PICKN returns the value of the switch in A0:;;
102440 30457 PICKN: LDA 2,APT
102441 24002 LDA 1,C2
102442 123000 ADD 1,2
102443 50454 STA 2,APT

```

```

(;;)PICK returns the value of the next parameter in A0:;;
A2 points to the parameters table entry

```

```

102444 54771 PICK: STA 3,RETS ;store the return address
102445 25001 LDA 1,1,2 ;number type of the parameter
102446 31000 LDA 2,0,2 ;address of the parameter
102447 102520 SUBZL 0,0 ;make a one as a so DECIMAL loads
102450 6120 DECIMAL ;DA gets the parameter
102451 6121 FIX ;A1 gets the value of the DA
102452 2446 JMP @RET ;fix had an error so return
102453 121000 MOV 1,0 ;put the parameter value in
102454 2761 JMP @RETS ;return

```

```

(;;)return value of this discsub in second parameter

```

```

102455 24761 VO: LDA 1,RNUM ;revision number
102456 102420 PUTD: SUBZ 0,0 ;sign of number is +ve
102457 6122 FLOAT ;convert to a decimal
102460 30437 LDA 2,APT ;the pointer to the arguments
102461 25007 LDA 1,7,2 ;number type of D
102462 31006 LDA 2,6,2 ;address of D
102463 102420 SUBZ 0,0 ;tells Decimal to do a store
102464 6120 DECIMAL ;store the Revision in B
102465 34433 EXIT: LDA 3,RET ;the return address
102466 1401 JMP 1,3 ;do a skip return

```

```

(;;) look for the first occurrence of B#[1,1] in C# ;;;

```

```

102467 30430 V1: LDA 2,APT ;the argument pointers
102470 35002 LDA 3,2,2 ;A3 gets the address of B$
102471 21400 LDA 0,0,3 ;A0 gets the first two bytes of B$
102472 101320 MOVZS 0,0 ;move first byte to lsbits
102473 24064 LDA 1,C377 ;to mask out the least sig character
102474 123400 AND 1,0 ;A0 now contains B#[1,1]

```

```

102475 40425 STA 0,B1 ;the first character of B$
102476 35005 LDA 3,5,2 ;the dimension and type of C$
102477 175100 MOVL 3,3 ;shift out the Msb
102500 175220 MOVZR 3,3 ;remove the type bit
102501 54420 STA 3,CSZE ;store as the size of C$
102502 31004 LDA 2,4,2 ;A2 gets the address of C$
102503 126420 SUBZ 1,1 ;A1 is the counter in string C$
102504 4420 LOOP1: JSR GET ;get the character in A2+A1 into A0
102505 101015 GNZ 0,0 ;skip if not end of string
102506 430 JMP NOTF ;not found
102507 125400 INC 1,1 ;increment the substring count
102510 34411 LDA 3,CSZE ;the size of C$
102511 166032 GCE 3,1 ;skip if not last character
102512 424 JMP NOTF ;not found
102513 34407 LDA 3,B1 ;B#[1,1]
102514 162414 SEQ 3,0 ;skip if we have found a match
102515 767 JMP LOOP1 ;try again
102516 740 JMP PUTD ;we have found the character

102517 0 APT: 0 ;used for storing the pointer to arguments
102520 0 RET: 0 ;stores the return address

102521 0 CSZE: 0 ;the dimension of C$
102522 0 B1: 0 ;the first character of B$
102523 0 RTN2: 0

102524 54777 GET: STA 3,RTN2 ;note the return address
102525 135000 MOV 1,3 ;the byte pointer
102526 175220 MOVZR 3,3 ;convert to a word pointer
102527 157000 ADD 2,3 ;A3 now points at the bytes
102530 31400 LDA 0,0,3 ;get them
102531 125213 SKO 1,1 ;don't swap on even
102532 101300 MOVS 0,0 ;swap since odd
102533 34064 LDA 3,C377 ;the byte mask
102534 163400 AND 3,0 ;mask out the wrong byte
102535 2766 JMP @RTN2 ;return

102536 126420 NOTF: SUBZ 1,1 ;make a zero
102537 717 JMP PUTD ;put the result and return

102540 V5: ;mode 5 uses same input parameters
102540 V3: ;modes two and three share much code
102540 V2: ;string search proper
102541 30757 LDA 2,APT ;address of B$
102542 35002 LDA 3,2,2 ;store for future reference
102543 54464 STA 3,BADR ;the dimension and type of B$
102544 35003 LDA 3,3,2 ;shift out the msb
102545 175100 MOVL 3,3 ;shift back without the bit
102546 175220 MOVZR 3,3 ;the size of B$
102547 54457 STA 3,BSZE ;the address of C$
102548 35004 LDA 3,4,2 ;remember it
102550 54457 STA 3,CADDR ;the size and type
102551 35005 LDA 3,5,2 ;the address only
102552 175100 MOVL 3,3 ;the type bit is removed
102553 175220 MOVZR 3,3 ;the size of C$
102554 54745 STA 3,CSZE

102555 34026 LDA 1,C6 ;the address of D
102556 130000 ADD 1,2 ;A2 now gets the address for D

```

102557	4665		JSR PICK	;get the original value of D for POS
102560	40444		STA 0,POS	;nothing found
102561	24656	NMTCH:	LDA 1,SWTCH	;what was the mode switch on entry
102562	20002		LDA 0,C2	;was it 2
102563	106414		SEQ 0,1	;???? was the mode 2
102564	446		JMP MODE3	;NO, therefore assume mode 3
102565	126420		SUBZ 1,1	;since no match set string counter to 0
102566	30440	LOOP:	LDA 2,BADR	;the address of B\$
102567	4735		JSR GET	;get the character
102570	101015		SNZ 0,0	;skip if not end of B\$ and match found yet
102571	430		JMP FND1	;we have found a match
102572	40730		STA 0,B1	;store the character we have just got
102573	30432		LDA 2,BSZE	;the dimensioned size of B\$
102574	132433		SLE 1,2	;skip if this is not reached
102575	424		JMP FND1	;we have hit the end of B\$, so match
102576	30431		LDA 2,CADDR	;the address of C\$
102577	34425		LDA 3,POS	;the position we are investigating
102600	44431		STA 1,ACC1	;make a note of A1
102601	167020		ADDZ 3,1	;increment 1 so that we take into account POS
102602	4722		JSR GET	;get the character from C\$
102603	24425		LDA 1,ACC1	;restore the accumulator
102604	101015		SNZ 0,0	;skip if not at end of C\$
102605	731		JMP NOTF	;we are at the end of C\$
102606	30714		LDA 2,B1	;restore the value for B\$
102607	142414		SEQ 2,0	;skip if they match
102610	403		JMP FAIL1	;the match failed
102611	125400		INC 1,1	;increment the substring counter
102612	754		JMP LOOP	;check the next characters
102615	10411	FAIL1:	ISZ POS	;look at the next position
102614	20410		LDA 0,POS	;we are going to check it
102615	34704		LDA 3,CSZE	;the size of C\$
102616	142032		SOE 3,0	;continue if it fits
102617	717		JMP NOTF	;not found
102620	741		JMP NMTCH	;restart the index counter
102621	24403	FND1:	LDA 1,POS	;pos is the position where we matched
102622	125400	FND51:	INC 1,1	;no displacement is called 1 etc
102623	632		JMP PUTD	;put this value and return
102624	0	POS:	0	;where we find the value
102625	0	BSZE:	0	;the size of B\$
102626	0	BADR:	0	;the address of B\$
102627	0	CADDR:	0	;the addr of C\$
102630	0	C1:	0	;temporary storage of char from C\$
102631	0	ACC1:	0	;temporary storage of A1
102632	101120	MODE3:	MOVZL 0,0	;A0 contained 2 now contains 4
102633	122433		SLE 1,0	;??? A1 contains the switch, are we in MODE3,4
102634	465		JMP MDE5	;NO, therefore it is MODE 5
102635	30662		LDA 2,APT	;we are going to find the value of LENW
102636	24030		LDA 1,C10	;the argument displacement
102637	133020		ADDZ 1,2	;A2 contained the address for the 4th parameter
				;we increment this to point at LENW
102640	4604		JSR PICK	;get the value of the LENW parameter
102641	40510		STA 0,LENW	;note it
102642	24762	AGAIN:	LDA 1,POS	;as passed by the caller, or by us if we

102643	39505		LDA 0, LENW	;are skipping short words
102644	135050		ADD 1, 0	;the specified length of previous word
102645	24454		LDA 1, CSZE	;move pointer to next word
102646	106403		SLE 0, 1	;the dimensioned size of C\$
102647	440		JMP BYOND	;???? are we within the line still
102650	40500		STA 0, SCAN	;NO, set everything to 0 for not found
				;YES, set up the scan position as start of
				;this newly found word
102651	40753		STA 0, POS	;this is also the beginning of the word
102652	102420		SUBZ 0, 0	;the words length is currently zero
102653	40476		STA 0, LENW	;save this value
102654	30753	LOOP3:	LDA 2, CADDR	;the address of the second parameter
102655	24473		LDA 1, SCAN	;character number to scan
102656	4546		JSR GET	;get the SCANth character from C\$
102657	24467		LDA 1, CHARA	;the character A
102660	106032		SGE 0, 1	;???? is the specified character greter then A
102661	407		JMP FNDWD	;NO, therefore we have found the end of word
102662	24469		LDA 1, CHARZ	;the greatest legal character
102663	106433		SLE 0, 1	;???? character >="A" and <="Z"
102664	404		JMP FNDWD	;NO, its greater than Z, so end of word
102665	10463		ISZ SCAN	;ready for the next character
102666	10466		ISZ LENW	;one more character has been found in the word
102667	765		JMP LOOP3	;check the next character
102670	24461	FNDWD:	LDA 1, LENW	;see how big the word is
102671	30002		LDA 2, C2	;words less than 2 don't count
102672	132032		SGE 1, 2	;???? is length greter than or equal to 2
102673	411		JMP SHORT	;NO, the word is to short
102674	102420	FINS:	SUBZ 0, 0	;required by FLOAT
102675	6122		FLOAT	;place LENW in DA
102676	30521		LDA 2, APT	;the arguement pointer
102677	25011		LDA 1, 11, 2	;the number type of LENW
102700	31010		LDA 2, 10, 2	;the address of LENW
102701	102420		SUBZ 0, 0	;the store DA parameter
102702	6120		DECIMAL	;store the DA in LENW
102703	716		JMP FND1	;place incremented POS in D as we have
				;already stored LENW
102704	30056	SHORT:	LDA 2, C240	;check that we have not hit end of line
102705	142432		SGR 2, 0	;???? is the character a control character
102706	405		JMP RPT	;NO, so try again
102707	126000	BYOND:	ADC 1, 1	;generate -1 for incrementing to zero
102710	44714		STA 1, POS	;the position of the start of this word
102711	136420		SUBZ 1, 1	;generate a zero as not found generates
				;zero for LENW as well as POS
102712	762		JMP FINS	;restore these values for return
102713	102420	RPT:	SUBZ 0, 0	;try for another word as this was too short
102714	40403		STA 0, LENW	;reset length to 0
102715	20933		LDA 0, SCAN	;start at last investigated character
102716	101400		INC 0, 0	;step beyond it
102717	40708		STA 0, POS	;this is start of the word
102720	752		JMP AGAIN	;try whole thing again
102721	126460	ND5:	SUBZ 1, 1	;we are starting so position=0
102722	44702		STA 1, POS	;reset the position

```

102723 30700 VSRPT: LDA 2, BADR ;the address of B$, A2 will be used as pointer
102724 34700 LDA 3, CADDR ;the address of C$, A3 will be used as pointer
102725 21000 LOOP5: LDA 0, 0, 2 ;A0 gets next word from B$
102726 101015 SNZ 0, 0 ;??? is it the terminator
102727 411 JMP FND5 ;YES, therefore we have a match
102730 25400 LDA 1, 0, 2 ;NO, so get next word from C$
102731 125015 SNZ 1, 1 ;??? is it the end
102732 604 JMP NOTF ;YES, therefore there is no match
102733 106414 SEQ 0, 1 ;??? is the C$ and B$ words equal
102734 497 JMP FAIL5 ;NO, so start at the next position
102735 151400 INC 2, 2 ;YES, so step both strings
102736 170400 INC 3, 3
102737 766 JMP LOOP5 ;check the next pair

102740 24664 FND5: LDA 1, POS ;we have hit end of B$ so we have a match,
102741 185120 MOVZL 1, 1 ;we must convert POS to a pointer and return
102742 660 JMP FND51 ;convert to a byte pointer
;return this value

102743 10661 FAIL5: ISZ POS ;we failed this section, move up dand try again
102744 10663 ISZ CADDR ;step in C$
102745 756 JMP VSRPT ;repeat the procedure

102746 301 CHARA: 301 ;the character "A"
102747 332 CHARZ: 332 ;the character "Z"

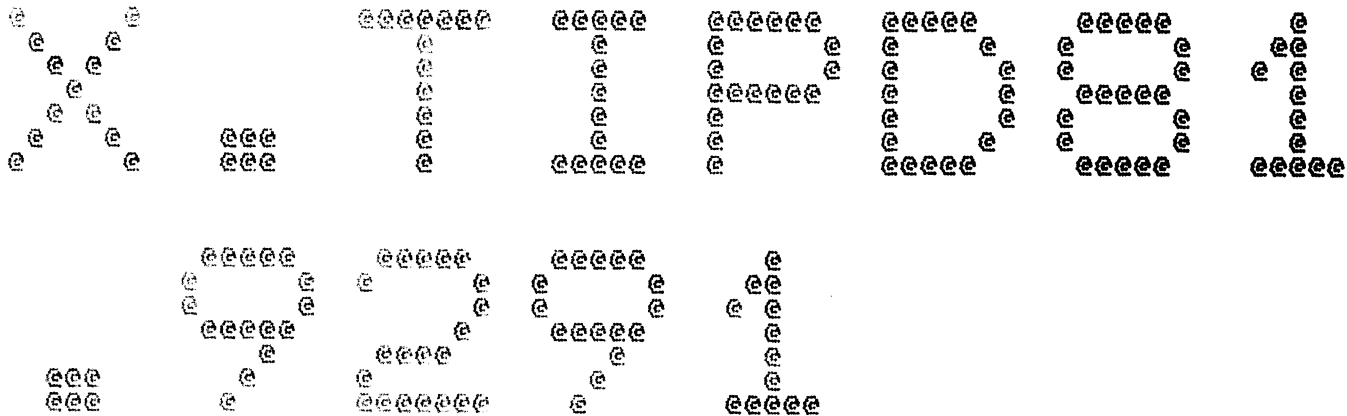
102750 0 SCAN: 0 ;the current character to investigate
102751 102751 LENW: LENW ;lenght of previous word

102752 DSBEND=
0 .ERR DSB160+400C.; OVERFLOW CHECK

.END

```

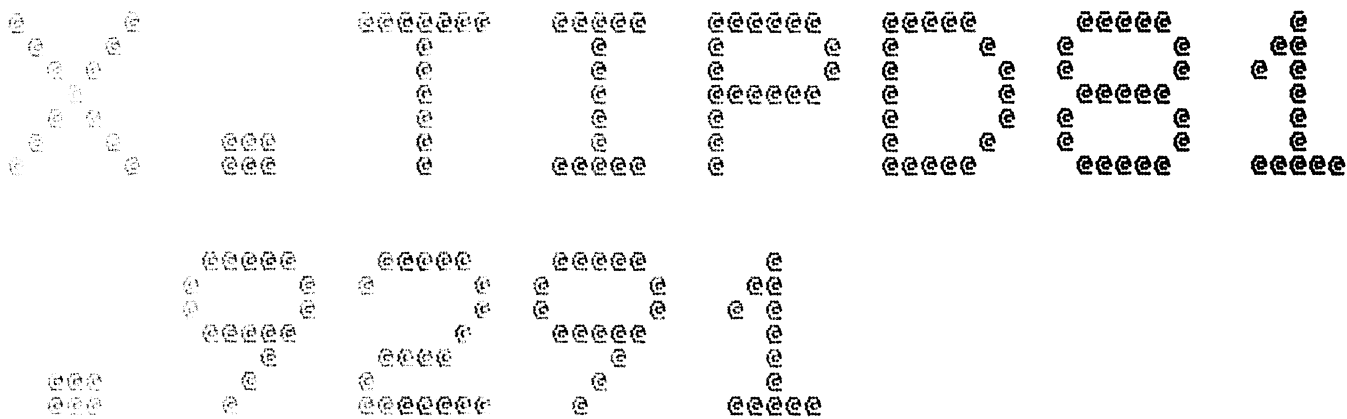
ACC1	102557	AGAIN	102542	APT	102517	B1	102522	BADR	102626
BINDI	6113	BJNMJ	6116	BPI	16	BSACF	75	BSZE	102625
BUMPU	6117	BYRND	102707	C1	102630	C10	30	C100	51
C1000	67	C11	31	C12	32	C13	33	C14	34
C15	35	C16	36	C160	174	C163	175	C166	176
C17	37	C170K	21	C171	177	C177	52	C1777	70
C2	2	C20	42	C200	53	C2000	71	C205	54
C215	55	C240	56	C244	57	C260	60	C271	61
C3	0	C300	62	C334	63	C37	43	C377	64
C4	24	C40	44	C400	65	C4000	72	C5	25
C6	26	C500	100	C7	27	C77	50	C774C	22
C777	66	CADDR	102627	CALL	6101	CHANN	6106	CHARA	102746
CHARZ	102747	CM400	23	CSZE	102521	DA	160	DAC	164
DAS	165	DATAP	6110	DB	166	DBA	41	DBC	172
DBS	172	DECIM	6120	DFTCA	34106	DMCAL	34110	DQUEUE	6105
DSB16	102400	DSBEN	102752	ERRF	76	ESCF	73	ETSF	74
EX	102433	EXIT	102465	FAIL1	102613	FAIL5	102743	FIN3	102674
FINDL	6123	FIX	6121	FLAGC	6102	FLOAT	6122	FND1	102621
FND5	102740	FND51	102622	FNDWD	102670	FREEN	6107	GET	102524
GETBY	6124	HALTS	6153	INBYT	6125	INSTB	6126	IOCAL	34103
IOP	6	ISA2D	6127	ISA2L	6130	JFLT0	151	LACNT	4000
LAFSE	13000	CALCO	47400	LALLO	1400	LATDE	36000	LBAKU	106000
LBILD	5000	LBUIL	4400	LCALL	75000	LCHAN	41000	LCHFL	30000
LCHSU	61000	LCLFA	7400	LCLDS	7000	LDLPY	76000	LCNVA	11400
LCNVD	12000	LDDMM	33400	LDALC	2000	LDALL	1000	LDB7A	114000
LDB7B	114400	LDB7C	115000	LDB7D	115400	LDB7E	116000	LDB7F	116400
LDB7G	117000	LDB7H	117400	LDB7I	120000	LDEKE	52400	LDELE	3400
LDIRE	50400	LDLTP	20400	LDREN	37400	LDSB1	400	LDSB2	22400
LDSB3	47000	LDSB4	65000	LDSB5	77000	LDSB6	106400	LDSB7	113400
LECHO	37000	LENW	102751	LEOB7	105400	LERRO	23000	LFAUL	400
LFFIL	2400	LFIXD	57400	LFNDC	112000	LFNDL	20000	LFOFI	17000
LGETR	10000	LGHQP	107400	LGHQS	107000	LGMUX	16000	LHCON	17400
LIBCA	44400	LIBEN	45000	LIBTR	45400	LIDAT	103000	LLINK	35400
LLDAD	34400	LLDGI	32000	LLUIN	112400	LMAPB	73000	LMDE0	65000
LMDE1	66000	LMDES	71400	LMRCS	56400	LMRFH	57000	LMRFI	54000
LMTAP	55400	LMTAS	54400	LMTFP	56000	LMTFY	60400	LMTNX	55000
LMTPL	60000	LBADD	6131	LOOP	102566	LOOP1	102504	LOOP3	102654
LOOPE	102729	LOPEN	6000	LOPNM	13400	LPATQ	110000	LPEXP	23400
LPPAB	72000	LPFLN	73400	LPPNA	3000	LPPRL	72400	LPFSE	67000
LPPSH	70000	LPFSX	70400	LPLQG	24400	LPPWR	33000	LPRAN	36400
LPRCD	71000	LPSIN	25400	LPSQR	22400	LPTAN	25000	LQIBF	63400
LQICL	63000	LQIQP	62400	LRDFH	26400	LRDIS	31400	LRDSE	110400
LRDCC	50000	LREDI	11000	LREDM	14000	LREDP	74000	LRENA	15000
LREDP	53000	LRESO	42000	LRWIT	113000	LRWMB	14400	LRWSX	111400
LS105	77000	LS152	102000	LS153	101000	LS154	100400	LS156	101400
LS157	100000	LSAVE	43000	LSAVP	43400	LSEAB	64000	LSEAR	51000
LSEND	102420	LSETF	40000	LSHUF	52000	LSIGP	12400	LSING	40400
LSMCS	106400	LSPEC	27000	LSTRI	32400	LSTRT	102407	LSYSC	30400
LTP01	102400	LTP03	104000	LTP04	104400	LTP05	105000	LVMUX	42400
LWRIT	47000	LXMIN	62000	MDE5	102721	MODE3	102632	NMTCH	102561
NOTE	102536	DUTBY	6132	OUTTE	6133	PIB	4	PICK	102444
PICKN	102440	FDS	102624	PUTBY	6134	PUTD	102456	QCHAR	6103
QUEUE	6104	READB	6135	RELJM	6136	RET	102520	RETS	102435
REVIS	4	RJSR	6136	RNUM	102436	RPT	102713	RTN2	102523
RTP	7	RUP	5	SBA	40	SCAN	102750	SCDCA	34147
SHORT	102704	SIZE	102434	SPINP	6146	START	102403	STINP	6140
STINT	6147	STORD	6137	STOUT	6141	SWTCH	102437	TASKQ	15
TRAPP	6142	VO	102455	V1	102467	V2	102540	V3	102540



Spool Queue Line #: 27
IRIS LU/Filename : 18/X.TIPD81.9291

Printed on/at : FEB 7, 1990 14:50:13
For Group/User: 0, 1
On Port No: 5

Print control parameters :
Printer Class code : 0
Form Code/paper type : ?
Print Priority (0-9) : 5
Starting Page Number : 1
This is copy number : 1
Keep file (Y/N) : Y
Notify User when done: N
Comments, optional : For RELSE CNTRL



Spool Queue Line #: 27
IRIS LU/Filename : 18/X.TIPDB1.9291

Printed on/at : FEB 7, 1990 14:50:23
For Group/User: 0, 1
On Port No: 5

Print control parameters :
Printer Class code : 0
Form Code/paper type : ?
Print Priority (0-9) : 5
Starting Page Number : 1
This is copy number : 1
Keep file (Y/N) : Y
Notify User when done: N
Comments, optional : For RELSE CNTRL

***** JOB STATISTICS *****

2	TOTAL # DUPLICATE KEYS
0	TOTAL # DIR. RE-ORGS
204	TOTAL # KEYS INSERTED
0	TOTAL # ASSEMBLY ERRS

ERR	2. 939						
ACCU	2. 200	2. 203	2. 228:				
ADAIN	2. 240:	2. 295					
API	2. 048 2. 233	2. 083 2. 273	2. 086	2. 106	2. 117	2. 142:	2. 166
BI	2. 123	2. 137	2. 147:	2. 194	2. 206		
BADR	2. 168	2. 190	2. 225:	2. 300			
BSZS	2. 172	2. 195	2. 224:				
PYDIT	2. 246	2. 284:					
CI	2. 227:						
CIT	2. 234						
CE	2. 084	2. 186	2. 267				
C240	2. 281						
C370	2. 121	2. 157					
CS	2. 180						
CATIT	2. 174	2. 198	2. 226:	2. 253	2. 301	2. 320	
CHARA	2. 256	2. 328:					
CHARC	2. 259	2. 329:					
CSIS	2. 127	2. 134	2. 146:	2. 178	2. 214	2. 244	
DEFIN	2. 095	2. 110	2. 277				
DEXT	2. 043:	2. 044	2. 045	2. 339			
DSSES	2. 045	2. 333:					
EX	2. 058	2. 060	2. 061	2. 062	2. 067:		
EXIT	2. 111:						
FAITH	2. 208	2. 212:					
FAITH	2. 309	2. 319:					

FINC	2.271:	2.283					
FIR	2.096						
FLDAP	2.105	2.272					
FNDL	2.193	2.197	2.219:	2.278			
FNDG	2.304	2.314:					
FNDRA	2.220:	2.317					
FNDRC	2.258	2.261	2.266:				
GET	2.130	2.150:	2.191	2.202	2.255		
GENA	2.238	2.242	2.251	2.263	2.266	2.291	2.332:
LOSP	2.190:	2.210					
LOGPT	2.130:	2.137					
LOGRD	2.253:	2.264					
LOGRE	2.302:	2.312					
LOGRI	2.050	2.064	2.077				
LOGRI	2.054:	2.055	2.056	2.057	2.058	2.059	2.060
	2.061	2.062	2.077				
LTCDI	2.042						
MDFR	2.232	2.298					
MOCS	2.188	2.230:					
MOCLA	2.185:	2.217					
NOFR	2.132	2.136	2.161:	2.205	2.216	2.307	
PICR	2.049	2.091:	2.182	2.237			
PICRA	2.083:						
POB	2.183	2.199	2.212	2.213	2.219	2.223:	2.240
	2.249	2.285	2.294	2.299	2.314	2.319	
POTB	2.104:	2.140	2.162	2.221			
RET	2.047	2.067	2.070	2.097	2.111	2.143:	

REIS	2.078:	2.091	2.099		
REVIS	2.008=	2.077			
RNUM	2.079:	2.103			
RPT	2.283	2.290:			
RTN2	2.148:	2.150	2.159		
SCAN	2.247	2.254	2.262	2.292	2.331:
SHORT	2.269	2.281:			
SIZE	2.068	2.077:			
START	2.044	2.047:			
SWTUP	2.064	2.080:	2.185		
TIPD:	2.043				
V0	2.054	2.103:			
V1	2.055	2.117:			
V2	2.056	2.136:			
V3	2.057	2.165:			
V4	2.059	2.164:			
V5RPT	2.000:	2.321			