

```

=====
KLINK load map size size RYND1 11:59 AM TUE., 13 DEC., 1983
file start end hex/oct decimal version message page 1
=====
RYZ101 6000 7FFC 1FFC 8188
RYZ112 7FFC 7FFC 0000 0
=====

```

total size=1FFC + 8188 decimal

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

=====
KLINK load map RYND1 12:00 PM TUE., 13 DEC., 1983 page 3
=====

```

```

APICLK 6864
APIDEL 6372
APIDSP 6418
APIERR 642C
APIFMD 644A
APIGET 646C
APINIS 6482
APINRG 735D
APIP 64A4
APIR 648C
APIRIG 73B3
ARDOH 73FD
ARITH 73A9
ARARY 105B
ASC113 6069
ASCNAM 2E2D
ASCNHI 2E29
ASINHI 664C
ASNPAN 49CF
ASPACK 64D7
ASSIGM 1303
ASSIM 64BF
ASTBLD 742D
ATMPLT 650D
ATM2 74F4
ATM2. 7501
ATSTGM 7E2D
ATINH 084B
AVADR1 5666
AVADR2 566B
AVCALL 2B03
AVVAL1 567E
AVVAL2 5683
AWAKE 09C9
B. INIT 064C
BALK10 0E2B
BADDIV 6A0B
BASEND 209A
BASIC 5245
BASLOC 6002
BATS 358C
BCDINT 2E77
BCKSTO 43BD
BEEP 7EB4
BEEP. FED1 7BD4
BEEP. 7D6C
BEEPER FED7 62C8 7BF4
BEEPR 7DBB
BININI 2ED8
BLANKS 09FB 79AB 79F8
BLEBUF 4BCA
BLIMP 4100 7F61 7FA4
BLKFIL 5FRE
BOUND 0862
BRTIO 7549

```

```

=====
KLINK load map RYND1 12:00 PM TUE., 13 DEC., 1983 page 2
=====

```

```

A. ZERO EAC7
ABORT FBED
ABSS 3AM3
ABSS- 3AM6
ABTHRD F8EA
ACMMU 3572
ACREAT 7210
ADD15 3B0A
ADD20 3B0F
ADD9 3B04
ADDR F042
ADDR01 3704 7039 7CE6
ADJUST 233F
ATACHK 67B0
ALARM 7430
ALOEOP 7217
ALCALL 1F93
ALFA 0E02
ALFAM 0E03 6973 749A 74C8
ALINI 2E39
ALLOD 1F9F 7E26
ALMCHK 6C07
ALOSNV 27F9
ALVENT 2BEB
ALWAYS E9D1
AND 4E6A
AND. 4A0A
AND. 4415
AND. 43E4
AND. 43E9
AND. 43EE
AND. 43F3
AND. 4400
AND. 4405
ANNUNG 43ED
ANOTE 3607
ANYER 4B3A
OPEN 7220
OPEN 7224
APDEL 6401
APEKIT 668A
APTIO 09EF
APFILE 09F7
APFMD 644E
APINFO 6361
APMSKE 6537
APMSKY 653F
APPROD 72BC
APPL 4993
APPIND 6514
APPIRS 7229
APPIM 6552
APPROD 64C6 7992
APSTAT 7300
APTRAC 731E

```

```

=====
KLINK load map RYND1 12:00 PM TUE., 13 DEC., 1983 page 4
=====

```

```

ARIS 7541
ARIS15 069F
ARIS40 06A1
ATAB. R E022
AYE 0889
C. INIT E8AB
CALCSZ 2EFF
CALEM 540F
CALL 50EE
CALL 514F
CALNAM 0044
CARD. 49BB
CAT 7E80
CAT. 593B
CAT. 58CA
CATAL 590D
CATALL 7EEB
CAIBUF 7F62
CATHED 7F88
CAILJM 58EB
CAIIMP 7F5E
CDPT 1B08
CELL10 37AB
CH2LED 5AFD
CHANGE 5C02 6E0F
CHECK 087A
CINEDI 7AE4
CINEND 5E8D
CHKEDL 42FA
CHRA 7F7A
CHSROJ 80 7BC4
CHTRIG 6772
CL. AC1 EBF6
CLDEV 5E4F
CLERN 5BF2 6E28
CLIF 7186
CLMGAL 2C57
CLOOP 5E46
CLCOP 6A93
CLRENY 509E
CLREAR 4B12 633D 6E12 6F61 75BF 7EAD
CLRTM 7A05
CLRLCO EFA4
CLROFF 5E16 62E0 6DF6
CLRTM 5E38
CLRV 003F
CMOREP 1E1F 6D6B 6E6E
CMUSHD E815 62E9
CMPCHK 052C
CMPDSR 04F7
CMPENR 04ER
CMPEM 0501
CMPLP 05BF
CMPSET 0585
CMPSRV 040A

```

KLINK load map RYNDI 12:00 PM TUE., 13 DEC., 1983 page 5

ENPSTR	408A				
ENRST	0489				
ENRIG	7448				
ENRMA	E781				
ENRBM	018A	6707	7C80	7CE0	
ENRCA	FC30	733F			
ENRCA	FC32				
ENRIG	5807				
ENRIM	1037				
ENRIL	51CC				
ENRIP	22E7				
ENRIP	1A43				
ENRIS	39E5				
ENRIS	3905				
ENRIM	107F				
ENRIP	F6CA				
ENRIP	F696				
ENRIP	F664				
ENRIP	F66C				
ENRIP	4550				
ENRIP	F82C				
ENRIP	7528	70C7			
ENRIP	0090				
ENRIP	398A				
ENRIP	39D1				
ENRIP	676E				
ENRIP	EF8F				
ENRIP	068F	63E8			
ENRIP	06C4	63F4			
ENRIP	EF89	6304			
ENRIP	F35A				
ENRIP	4848				
ENRIP	2E93				
ENRIP	2E0C				
ENRIP	159C				
ENRIP	6008				
ENRIP	6027				
ENRIP	7C74				
ENRIP	7C83				
ENRIP	EE24	6AF3			
ENRIP	6787				
ENRIP	700A	6919	6E9E		
ENRIP	7008				
ENRIP	60AD				
ENRIP	6E23				
ENRIP	0A56				
ENRIP	6E0E				
ENRIP	6930				
ENRIP	6E3F				
ENRIP	32A8				
ENRIP	3E03				
ENRIP	6FFC				
ENRIP	6782				
ENRIP	7001				
ENRIP	2F4D				

KLINK load map RYNDI 12:00 PM TUE., 13 DEC., 1983 page 7

ENRIP	2A8B				
ENRIP	28E7				
ENRIP	0E7C				
ENRIP	EA90				
ENRIP	455C	669A			
ENRIP	0835				
ENRIP	3ABE				
ENRIP	40DF				
ENRIP	40EF				
ENRIP	7472				
ENRIP	4340				
ENRIP	7870				
ENRIP	FD46				
ENRIP	4177				
ENRIP	4C45				
ENRIP	4C4A	67A2	68A7	68C7	68D6
ENRIP	773C	7A50		6C29	6E47
ENRIP	64E8	64F4	65B2	663E	66D9
ENRIP	6980	6988	6C87	6CEA	6D34
ENRIP	7110	7369	7422	7402	759C
ENRIP	7607	7788	7708	7976	798B
ENRIP	7078	70BC	7E18	7E4C	7E58
ENRIP	7E6A				7E7C
ENRIP	104E				
ENRIP	FCAB				
ENRIP	4D70				
ENRIP	4C91	67F5	78B0		
ENRIP	4C99	67B4			
ENRIP	4C8C				
ENRIP	E2A0				
ENRIP	4841				
ENRIP	5CFO	7575	79E8		
ENRIP	00BA				
ENRIP	EE1D				
ENRIP	4E56				
ENRIP	3068				
ENRIP	3053				
ENRIP	386B				
ENRIP	3576				
ENRIP	2104				
ENRIP	2114				
ENRIP	2011	7E0D			
ENRIP	206D				
ENRIP	2078	7E1E			
ENRIP	221F				
ENRIP	20A5				
ENRIP	10AD				
ENRIP	1E43				
ENRIP	4459				
ENRIP	10E6				
ENRIP	10B8				
ENRIP	4F3B				
ENRIP	10B5	783C			
ENRIP	10D0				
ENRIP	10C3	7862			

KLINK load map RYNDI 12:00 PM TUE., 13 DEC., 1983 page 6

ENRIP	2F46				
ENRIP	EF8F				
ENRIP	13E1				
ENRIP	4E3E				
ENRIP	4ECB				
ENRIP	79FB				
ENRIP	3A0A				
ENRIP	3A0B				
ENRIP	703F	6490			
ENRIP	1C01				
ENRIP	1FB3				
ENRIP	21F3				
ENRIP	2220				
ENRIP	07A3				
ENRIP	7830				
ENRIP	5260				
ENRIP	0E1C	6997	7844		
ENRIP	1358				
ENRIP	1373				
ENRIP	E38A				
ENRIP	376E	7C8F	7C1A		
ENRIP	3771				
ENRIP	3768				
ENRIP	3786				
ENRIP	3744				
ENRIP	136A				
ENRIP	1365				
ENRIP	5919				
ENRIP	0121				
ENRIP	E383	7287			
ENRIP	58FC	675B			
ENRIP	F01C				
ENRIP	516A				
ENRIP	588E				
ENRIP	458F				
ENRIP	0A4B				
ENRIP	0A4C				
ENRIP	7547				
ENRIP	7816	7E33			
ENRIP	6E7B				
ENRIP	491A				
ENRIP	5C5A				
ENRIP	74AF				
ENRIP	5293				
ENRIP	4482				
ENRIP	4486				
ENRIP	1986				
ENRIP	597A				
ENRIP	EE55				
ENRIP	6A2A				
ENRIP	5204				
ENRIP	E30E				
ENRIP	503E				
ENRIP	00CC				
ENRIP	2749				

KLINK load map RYNDI 12:00 PM TUE., 13 DEC., 1983 page 8

ENRIP	1EDF				
ENRIP	10B2				
ENRIP	10A6				
ENRIP	6016				
ENRIP	EA80				
ENRIP	1989				
ENRIP	4A3C				
ENRIP	4A22				
ENRIP	4A28				
ENRIP	6AF8				
ENRIP	FBAS				
ENRIP	6B6D				
ENRIP	6814				
ENRIP	60DF				
ENRIP	4B31				
ENRIP	622F				
ENRIP	6EC2				
ENRIP	67E8				
ENRIP	67E8				
ENRIP	63FB				
ENRIP	67E3				
ENRIP	6809				
ENRIP	6819				
ENRIP	6823				
ENRIP	1F1B				
ENRIP	6888				
ENRIP	198F				
ENRIP	2212	78CA	7E77		
ENRIP	4A45				
ENRIP	4A47				
ENRIP	6392				
ENRIP	6225				
ENRIP	620C				
ENRIP	63CF				
ENRIP	64B3				
ENRIP	6A09				
ENRIP	6886				
ENRIP	683A	70FO	79B6		
ENRIP	63EC				
ENRIP	49FE				
ENRIP	63B5				
ENRIP	7FD7				
ENRIP	685F				
ENRIP	56CC				
ENRIP	56CC				
ENRIP	214C				
ENRIP	214C				
ENRIP	2140				
ENRIP	2146	6733			
ENRIP	ED0F				
ENRIP	682B				
ENRIP	2A9C				
ENRIP	146F				
ENRIP	1696				
ENRIP	5839				

KLINK load map RYWI01 12:00 PM TUE., 13 DEC., 1983 page 9

FNREI.	5891			
FNRTN.	5858			
FNWU	10F8			
FOPAC?	247F			
FOPEN	2480	6341		
FOPEN?	2492	7061	7071	70C0
FOPENG	2494			
FDR	1760			
FDR.	ED64			
FORMAR	1618	6604		
FORMN	F368			
FORMN+	F363			
FPS	3A81			
FPURGE	1FD6	60F3	6E25	
FREHAM	2014			
FREHAM	1C5F			
FREPLN	2259			
FREPLS	224C			
FREI	26A1			
FRMARM	2EE3			
FRMARM	2EE0			
FSEK	2154	7E53	7ECE	
TSREPL	2104	6F25		
FTRDR	56A1			
FTR12	3807			
FTR13	380D			
FTR53	38E4			
FTR88	3C63			
FISTIL	54B1			
FISTILS	54B8			
FISVL	549E			
FUMI	0FC1			
FKALRM	6ED8			
FKRPP1	6C49			
FKDATE	6EF3			
FKDRY	6F27			
FKRENU	FC07			
EXTIME	6F37			
FKYERR	6F6F			
GA*012	1200			
GA012N	12C6			
GA18	1250			
GA18/*	5887			
GA12N	12DC			
GA1N	12D7			
GA1N8	1289			
GA104N	12E1			
G180R*	587F	6593		
G10R2N	12F3			
GEHAR	0DEB	7895		
GEHARM	0DE0			
GE08.	40CD			
GEQ.	4E1C			
GEIN1	5094			
GEIN2	509H			

NOMAS
Not Manufacturer Supported
recipient agrees NOT to contact manufacturer

KLINK load map RYWI01 12:00 PM TUE., 13 DEC., 1983 page 11

ICOMST	54CE			
ICOS.	7535			
IDYSND	E80D			
IF	18D5			
ILE1	1644			
INEMV?	53EE			
INFTO	39LB			
INFR2	3A7C			
INFR3	3A4E			
INFR4	3A51			
INFR9	3A90			
INIPR-	0806			
INIPRS	0802			
INISIZ	675D			
INIT.	666B			
INI16L	00F5			
INPU4.	4373			
INPUH.	4309			
INPUT	1557			
INPUT.	4276			
INSERT	1FFC			
INSTRM	23LE			
INTS	3A68			
INTCON	57D7			
INTDIV	3A6F	7C35		
INTEGR	0D16			
INTMUL	394E			
INTORL	3F16			
INIRP	00BE			
INMPOP	ED23			
INMRTN	ED04			
INMIAD	EFFB			
IO.OFF	F015			
IO.OH	100C			
IO?	E99F	7E3C		
IOF1LE	09E7			
IOOPEN	218E			
IPS	3AEB	7CC2		
ISCHAI	1075			
ISETUP	43D5			
ISIM.	752D			
ITAIL.	43DF			
ITAM.	753D			
ITHEM	16DF			
IZOPEN	2189			
JHPLM	EC54			
JHPSUB	EC44			
JSBCRT	702D			
JTRUEM	EC35			
KEY.	EC60			
KEY?	0854	7A2A		
KEYOSP	419A			
KEYNAM	4E8D			
KEYS.	499F			
KEYSRV	0277			

KLINK load map RYWI01 12:00 PM TUE., 13 DEC., 1983 page 10

GE1IM	416D			
GE118	128E	64FD		
GE11N	1261			
GE12N	1258			
GETADN	1F12	684E	700E	70E5 7EF8
GETAD+	1F14			
GETADR	1F11	683B	71C2	7999
GETALM	00E6			
GETCHR	0750	63EE		
GETCLK	745B			
GETCM?	106B			
GETCMA	1066			
GETFC	EDCC			
GETFST	4AF8	686A		
GETINT	2C6F			
GETLIN	15D0	65N8		
GETLN	06C9			
GETLN+	06E9			
GETLNK	64C7			
GETNAM	4A77	6338	7D55	7D6F 7DB3 7DF4
GETNI	3133			
GETNKT	2E85			
GETPR?	125D			
GETPAD	5070	689B	710B	79DA
GETPAR	1264	6523		
GETREP	4158	7F87		
GETRIU	24F8			
GETRM2	475E			
GETRON	4761			
GETRUN	18AA			
GETSTP	1296	6547		
GETTD	746C			
GETTEN	4110			
GETTER	412E			
GEZLNM	18BB			
GEORD	F868			
GOPEN	241C			
GOPEN?	2497	6749		
GORTN	568E			
GOTAPR	17BB			
GOTOSU	17BE			
GR.	4DE6			
GR.	4E25			
GRANGE	18D8			
GTHKT1	2E85			
GXX3	1289			
HALT?	1889			
HANDI	5419	7182	7E63	
HANDIO	544C			
HANDIR	541C			
HANG	016A			
HLFLIN	07F1			
HLFOUT	07D6			
INPUSH	57DA			
I/ROFF	7CE6			

KLINK load map RYWI01 12:00 PM TUE., 13 DEC., 1983 page 12

KEYARM	4195			
KOPY	0A5F			
KYBDR	E457			
KYOPEN	4E87			
LA	7DDF	6429		
LADDEV	45FE			
LADREP	EC18			
LBL4	7260			
LDDU1	59DB			
LDDRY	F027			
LDDST	F037			
LEAPYR	6F94			
LEM.	FCBE			
LEMCAL	E939			
LE8.	4DC4			
LEQ.	4E0B			
LE1	1649			
LE160	079A			
LIFI.	524D			
LJMEOR	44EF			
LJMEED	FE56			
LJMHED	0C15			
LJMLEN	21E5			
LJST.	1848			
LIT?	06B3			
LM20	362A			
LM30	3668			
LM38	3654			
LMS	3732			
LM77	3676			
LOAD	F7CC			
LOCK.	79B6			
LOCK?	79CB			
LOGCOM	399A			
LOG15	378A			
LOOKUP	0A26			
LPDFF	5E20	62E3	6D99	
LPON	5E28			
LSTIO.	5D8B			
LT8.	4D46			
LT.	4E3C			
MAKEIT	06A8	6E54		
MALLOC	741C			
MARGN.	7F67			
MAXIO	3034			
MAXOAI	6AFB			
MBLIM-	7446			
MBLIMP	7440			
MLJSB	4661	6917	696D	6E3B 6E9C 70EE 79B4
MEM.	FCC6			
MERGE.	5339			
MINFO	3025			
MIMOD	6FAB			
MIMHH	6FAE			
MIMHH	6FAB			

MINIM 6F80
 MINYY 6F86
 MKPRV 6F90
 MOD10 7705
 MODE 7 648B
 MODEKY 6422
 MOVE 2270
 MOVE11 5E96
 MOVENW 2C16
 MOVEUP 2218
 MPV28 38E2
 MPV30 38E5
 MPYR 7F9A
 MPYR01 393D
 MRE1H 6C01
 MREOUT 07E3
 MULLGO 7473
 MVBYS 3F43 7077
 MVRD 3090
 MYCE1W 7418
 MYSE12 7418
 MYSTPR 7412
 NAME 5253
 NAME1W 6C55
 NEXT 1785
 NEXT 6DD7
 NFR 388A
 NOCLC 17FC 6516
 NOT 4E61
 NOTASH 71ED
 NULCST 0A03
 NULOLD 4108
 NUL 6C05 701F
 NULBER 0D40
 NULCHK 0A69
 NULOUT 6985
 NULREP 1086
 NULVA 0E6D 6581
 NULVAL 0E70 6666
 NULPK 6F85
 NULPT 747D
 NULUT 4783
 NULIN 2E82
 NULTRM 47AC
 OF ALRN 7444
 OFF10 5E11
 OFF10 5E06
 OFF1A 3F50
 OFFUML 5E09
 OF10K 5FEC
 ON 1796
 ON 6C0D
 ON/OFF 5F07 6511
 ON? 1 6283
 ON?R 6287

PAINT 18C1 6607
 PAINT 6398
 PALINE 642E
 PAMNT 6DE5
 PAMOTE 7492
 PAMTW 18E8
 PAMTCH 4575
 PAMUP 5F66
 PASE 1 1343
 PASEN 6763
 PASTR 5F1F
 PSETUP 69CB
 PIALD 2207
 PIRALO 2583
 PUS3SC 1248
 PU 33D0
 PUING 6CE4 600D 6F86
 PURGE 197E
 PUSHIA 122F
 PUSHIF 481B 652A
 PUSH32 1240
 PUSHMS 1237
 PUT 07B0
 PUTABR 1F07 6335
 PUTBIN 6E89
 PUTCAL 2C14
 PUTCHR 7451
 PUTKEY 07C1 7800
 PUTREL 1F06 6F58 710F
 PUTWIN 5806
 PURSK 062B
 PURSRV 05EE
 QMARK 348A
 QUIP 1807
 QWARY 68D9
 RA 7E0D 6422 647E 6483
 RAD 79FF
 RAD10 3828
 RADTRY 67F3
 ROYSD 67BC
 ROYSDH 6B18 68EC 680C 6830 68A7 6E77
 READ 15F3
 READM 15E5
 READM 655A
 READ 6593
 READER 690E
 READM 6E64
 READM 6E69
 RECDM 7483
 RECDVR 5005
 REFNUM 168C 65EC
 RELJMP 26A1
 RELMEM 2453 7668 7A34 7A7F
 REN 118E
 REN10 76FF

ONE7B 70EE
 ONE7PR 6375
 ONEB 3E88 671B 6F8A 6FAA 7156 71A0 7209 722E
 ONEB 3E88 7966 7AF7 7887 78D9 78E4 7C38 70D4
 ONE1 3EAC 6727 7880 7C13 7E42
 ONE 3E0D 7C15
 ONE- 3E0D
 ONE1 3E11
 ONE01 3E0E
 ONE00 1340
 DNOT2 5F00
 DNR12 1F3A 6FE2 7204 7229 7279 770B 7724 7903
 DNR 3F57
 DNTOK 5FF6
 DPTON 131F
 DR 4E4B
 DTER 0140
 DUTCH 0810
 DUTC40 0814
 DUTCHR 080C
 DUTEOL 082C
 DUTERB 494A
 DUTESC 0819
 DUTLIN 022B
 DUTMUM 5F6C
 DUTSTR 07EA
 DWRAY 687E
 DWRK 122E
 DWRK 64E6
 DWR1 09D7
 DWRSE 088A
 DWRSER 080E
 DWRST 08A7
 DWR 7465
 DWRUPR 2164
 DWRMUM 43FB
 DWR 39F6
 DWR1 6B03
 DWR1 6F8B
 DWR1 45E5 6C30 6E80
 DWR1 1906 65F5
 DWR1 1841
 DWR 6D1E
 DWRMUM 5040
 DWR 6F0F
 DWRMUM 6F57
 DWR10 3D16
 DWRMUM 6737
 DWRVAL 67B1
 DWR 4596
 DWRMUM 216A
 DWRMUM 654E
 DWRMUM 58B7
 DWRMUM 188E
 DWRMUM 18E5

REMDR 51F1
 REHGSB 6D54
 REHAR 1989
 REHUM 1C2E
 REP 1 40D7
 REPLAC 23E8
 REPLIN 21A7
 REPPOR 0261
 REPPOR 4B4A
 REPPRM 0262
 REPRAT 4B54
 REPRAT 401D
 REPRITM 658E
 RESCOH 2418 716C 7086 7EE5
 RESFIL 5F86
 RESPUT 6C25 7F90
 RESW 6542
 REST 50D9
 RESIO 6580
 RESIEM 4FF6
 RESIM 6586
 RESIOR 15D5
 RESTR 43C6
 RESUL 6CE0
 RETRM 6D8A
 RETURN 09E6
 REVBY 7033
 REVPSH 703D
 RINCHK 67F7
 RMEAR 471C
 RND10 77F4
 RND11 3C89
 RND12 775E
 RND1Z 3C82
 RNLST 491F
 RNDGO 4721
 RNDCL 257B
 RNDCLA 2574
 RNDICPY 2770
 RNDJSB 4694 6553 7338 7990 7E31 7E95
 RNDMUM 481D
 RNDSET 47ED
 RNDTPA 2787
 RND 3A87
 RND 1F8A
 RND 1F61
 RND 24D1
 RND10 3C8E
 RND 5941
 RND10 67E8
 RND10 7072
 RND10 7105
 RND10 1F20
 RND10 69C7
 RND10 241F 6788 706B 7140 7395 7715 772F 7749

ASSEM= 2420
 ASIBUF 6409
 ASIREG 0E51
 ATDIM 1E6A
 RUN. 50AC
 RUN11 50E7
 RUNKEE 50B9
 RUNROM 4754
 S.DIF 6A15
 S.OW 6AD1
 SAFE1 5390
 SAFMUM 6663
 SAFSTR 6669
 SALT 180E
 SAVBUF 64E2
 SAVENE 4741
 SAVENV 47C4
 SAVREG 0E33
 SAVRG+ 0E39
 SDCSR 10B7
 SEAM 0C83 6540
 SEAM+ 0C80
 SEAME1 1232
 SEANLX F193
 SEANM F048
 SECSI 304F
 SECSI1 54D5
 SEARLH 0805
 SECIO 39AA
 SEMIW. C781
 SEP10 3EE3
 SEP15 3C0F
 SEP20 3CC8
 SEQM 17E7
 SEDMO 1708
 SET 1E80
 SETCPL 0067
 SETCLM 522A
 SETCOM 4096
 SETED 5308
 SET11R 7CE8
 SET11 5E30
 SET130 7458
 SET11M 0704
 SETOFF 5E11 62EC 6E0C
 SETPR 1F5C
 SETPSB 09ED
 SET112 00E1
 SETAN 1F47
 SETUPB 408A
 SETUOB 7E55
 SETCAN 0C9A
 SGWS 397C
 SHIELD? 7ECF
 SHF10 3789

NOV 1983
 Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

STREXP 10FD 666C
 STRIND 1EF8
 STRREF 1156 65C5
 STRIME 7901
 STRK-1 129C
 SUBRO1 37CA
 SUBSCR 339A
 SUBST1 57E2
 SUBST2 5B21
 SUBSTF 6C64 6756
 SUM11 6AF6
 SUM11+ 6AF9
 SVARCK 580A
 SVAROR 54AA
 SVAROR+ 54A9
 SVCCCHK 4620 7A39
 SVCCLR 063F
 SVCCSE1 0634
 SVVAL3 54A4
 SYCALL 50EC
 SYSINI 7A20
 SYSJSB 4660 6420 6427 645F 6466 647C 648E 6495
 SYSJSB 4660 64A7 64BC 64FB 6505 650F 6521 6528
 SYSJSB 4660 6545 654B 657F 6591 65A6 65C3 65CA
 SYSJSB 4660 6605 66D2 60F1 6E23 6E52 6F23 6F4A
 SYSJSB 4660 70C5 7F8E
 TAB 58DB 6DEE 6E38
 TAB. FD91
 TADREP 6C0F
 TAWIO 39ED
 TBL3 3E53
 TBL3B 3E8B
 TCKTGL 75D0
 TENRIT 7047
 TEXT. 523E
 THERE1 5E7D 714E
 THERE1 734C
 THERE2 5E74
 TJCK 6844
 TJCKHK 71A7
 TJNDIV 64ED
 TIME. 7C6D
 TIME. 7CFA
 TIMEID 673A
 TIMASK 654A
 TIMCHK 67F7
 TIMGET 3FDC
 TIMEID 757A
 TIMEAL 3F3B
 TIMESET 76B2
 TIMECL 409E
 TIMEDE 4020
 TIMEGO 3FF2
 TIMEIL 407F
 TIMEOF 4049

SHFET 7F6F
 SHLA 7F46 6461
 SHRA 7F58
 SHROF 39A4
 SHIGH- 24FE
 SHIGHF 24FB
 SHIO 39D0
 SHIFUM 3425
 SKIP1 FEE4
 SKIPW 35E2
 SKIPD FEEA
 SKIPEH FEE7
 SKIP1 FEEB
 SKIP1 FEED
 SKIPL 2DD4
 SKIPR FEEA
 SKIPS FEE1 717D 7AE5
 SKPCHK 5E8A
 SKPCOM 2E1A
 SKPLN 21B5
 SKPLM 21B7
 SKPLOP EE95
 SKPVR 27C2
 SNLJN1 10A3
 SHDFRM E01B 6DA5
 SPY 4617
 SOR30 38A5
 SOR5 3890
 STALL 51C3
 STALL+ 51C8
 STALLS 51BA
 STALRM 7400
 STAM+ EADE 694C 6958
 STAM- EAC2
 STARD- EABF
 STAR? 50C3
 START 7A20
 START? 7A0E
 STBEEP FECA
 STBEP? 7054
 STORTE 6811
 STEP. EE56
 STMSK 65A5
 STOP. 51B9
 STOP? 06C6 7A2F 7C67
 STORF 7208
 STOST 54E5 7798
 STOSTM 54E4 778E
 STOSV 5598
 STOSVM 5597
 STRICH 7C7D
 STR?? 5899
 STRCN+ 1193
 STREX+ 1299
 STREX+ 10FA 6507

THROH. 3F78
 THROPH 402F
 THRLV 3FFE
 THRSS 408E
 THFAG 40A3
 TO. E234
 TOP1 E2A4
 THASC2 0A91 761A 7629
 THOD2 0A9C 7612
 THOD8 0A8A
 THOD1? E6D0
 THOD2 0A8C
 THOD8 0A8B
 THOD 602E
 THOPRO 6000
 THAP1? E6F7
 THO1 3C9F
 THO2 3C95
 THOFP 48E5
 THOED 4957
 THOCP 48D0
 THOVL 489C
 THOLO. 4887
 THOFL. 7312
 THJUMP 4926 6746
 THKEY 02A6
 THSLP 7056
 THOFF. 488C
 THVAR. 4890
 THYTM 12FC
 THHALD 2207
 THEND 2925
 THIRN 2919
 THPLT 6507
 THOB 3EBE
 THOR 30D2
 THORD1 3EE9
 THOPK? 2478
 THPSM 13C7
 THLNM. E413
 THAVIO 01F3
 THED. 408D
 THED. 4DFA
 THREP 6BF6
 THREP? EBE0
 THLSD 6BE5 6E03
 THQUOT 1212
 THROM 47E2
 THSEE 6021
 THSTAK 3C04 6969
 THIREP 6F0A
 THISD 6B12
 THIUHL 7026
 THPC. 6DCC
 THDISP 67FF

```

*****
KLINK load map      RYV101 12:00 PM TUE., 13 DEC., 1983 page 21
*****
UPENT 5913
UPRC* 1B35 6C10
UPRLSE 7989
UPSET 005C
UROOM* 1F85
USING. E3F9
U1210 3D0B
VAL* FC3B
VAL. FCF5
VCCALL 5101
VCREST 4FFE
VER. FBFR
VERB 6B47
VERIFY F829
VERANO 069C
VERANI 358A
VERAN2 5417
VERAN3 720E 64B4
VERAN4 F5C5
VERANS 6F44
VERSUB FC20
VF1102 6FDB
VFADCL 6CE8
VFADDR 6B80
VFBSY 6D06
VFBSY+ 67C1 696F
VFBYE 6DAD
VFCD46 6DBE
VFCDCE 6D97
VFCEDE 6D86
VFCELEH 6CED
VFDD12 6CBE
VFDECL 6DBF
VFDIR+ 6B8C
VFDIR* 67AD
VFIDUE 6EEF
VFIDU? 6DF7
VFERR 6D48
VFEXCH 6E7A
VFGE1 6A44
VFGLOC 6E19
VFHI 6A0F
VFHI+ 67A8
VFIAD 6C18
VFLAD+ 67A3
VFLD? 6E09
VFLIF? 6EB3
VFLIBY 6CFF
VFLTY+ 67CB
VFMP? 6E25
VFMI? 6A9A
VFMOVE 6E3B
VFMSG 6ED9
VFND- 6DD8
VFNDDE 6DCD

```

```

2 0000
3 0000
4 0000
5 6000

```

```

ABS
LOC 24K

```

```

*****
KLINK load map      RYV101 12:00 PM TUE., 13 DEC., 1983 page 22
*****
VFME+ 67E6
VFPE? 6DF1
VFRLX 6D76
VFRE 6F68
VFRNA 6E2B
VFRLF? 6EBA
VFRD? 6E90
VFRREC 6C80
VFRVDE 6EC3
VFRUO+ 67D0
VFRUK+ 679E
VFRURD 6BF3
VFRUSO 6BB2
VFRUSB 6BE0
VFRUSK 6B88
VFRUO 6D7F
VFRUR 6C92
VFSECI 6FE3
VFSKFL 6EE3
VFSTAI 6D0F
VFID 6C0F
VFID+ 67D5
VFTEAN 6DA2
VFTH 6F46
VFTRL 6F0F
VFUT1+ 67DA
VFVAC2 6C3D
VFVACH 6C2F
VFVBUO 6E84
VFVHOP 6C8A
VFVR 6C8B
VFVARK 6CC3
VFVACL 6C7B
VFVRD- 6F0F
VFVRD1 6DED
VFVRDE 6DEB
VFVRNEC 6C86
VSRCH 27A0
VSIOR 27E2
WBJT. E366 7BCB
WBJTCK 7E64
WBJTKY 0755
WKEUP 7666 6E3D
WBRN 4B03
WBRN.R 4C34
WCKJSB E36F 7BEF
WRTICR F913
YEARIM 659F
YINLHK 67F7
YTK5 3974
YTK5* 778B
Z36.56 2A67
ZMOD1- 2A79
ZMONEM 2A6C
Z7777 0E99

File RYV101 had 2 files 1144 symbols (1019 unreferenced, 0 undefined)
1 0000

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RYV101 OBJ=RYV101 12/13/1983 11:50 AM PG 1
*****
>>>>>> exts, ents, equs, etc <<<<<<<<
7 6000
8 6000 ENT SAFSTR get strexp
9 6000 ENT SAFNUM get numval
10 6000
11 6000 EXT ADDROI add numbers on stack
12 6000 EXT ALFAM upper case register
13 6000 EXT ALLOC get ram
14 6000 EXT APROM open var file
15 6000 EXT BEEP beep the beeper
16 6000 EXT BEEPER beep the beeper
17 6000 EXT BLANKS blanks
18 6000 EXT BLIMP blank input
19 6000 EXT CHANGE tell rons of dev change
20 6000 EXT CHSROI change number sign
21 6000 EXT CLEAN clean up devs
22 6000 EXT CLRERR clear pending error
23 6000 EXT CLOFF clear offio
24 6000 EXT CNDREP send and report command
25 6000 EXT CMDSND send command
26 6000 EXT COMBIN convert to binary
27 6000 EXT CONCA move string to reshen
28 6000 EXT CTRNDT create time
29 6000 EXT CURSE cursor on
30 6000 EXT CURSE- cursor off
31 6000 EXT CURSE? set cursor
32 6000 EXT DATREP send data and report
33 6000 EXT DATSD- send data
34 6000 EXT DEL delete char
35 6000 EXT DIGIT check for digit
36 6000 EXT DIVIO divide
37 6000 EXT DOCOM do output setup
38 6000 EXT DONE finish assignio
39 6000 EXT DTHL decimal to hex
40 6000 EXT EOL14? check for end tokens
41 6000 EXT ERR1 error report E=1
42 6000 EXT ERR1+ error report E=1, return
43 6000 EXT ERROR error report
44 6000 EXT ERROR+ error report return
45 6000 EXT EVIL save registers
46 6000 EXT FCRALO create file
47 6000 EXT FCRNUL create null file
48 6000 EXT FETSET get var
49 6000 EXT FEINUM get var value
50 6000 EXT FLOPEN open a file
51 6000 EXT FLSTAK get pil m device
52 6000 EXT FNDLW find line in file
53 6000 EXT FOPEN open a file
54 6000 EXT FOPEN+ open file, error
55 6000 EXT FORMAR get formal array
56 6000 EXT FPURGE purge file
57 6000 EXT FSEEK get a line
58 6000 EXT FSREPL replace line in file
59 6000 EXT GIOR* get parameter string

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RYA101 OBJ=RYZ101 12/13/1983 11:50 AM PG 2
=====
>>>>>> exts, ents, equs, etc <<<<<<<<
60 6000 EXT GENAR get a char
61 6000 EXT GETIB get a string, push token
62 6000 EXT GETIDM get address to drp
63 6000 EXT GETADR get address
64 6000 EXT GETCHR get input char-IV
65 6000 EXT GETFST get file string
66 6000 EXT GETLIN get line number
67 6000 EXT GETNAM get file name
68 6000 EXT GETPAD get pil address
69 6000 EXT GETPAR get parameters
70 6000 EXT GETREP get input
71 6000 EXT GETSIP get a string
72 6000 EXT GORTH goto process
73 6000 EXT HMDI do a handi call
74 6000 EXT INTDIV integer divide
75 6000 EXT IDI get idfile
76 6000 EXT IPS integer part
77 6000 EXT KEY? key there?
78 6000 EXT LA do left arrow
79 6000 EXT LPOFF set test bits
80 6000 EXT MAKEIT make file
81 6000 EXT RELJSB jsb to meltron
82 6000 EXT MBVYS move bytes around
83 6000 EXT NOCLE1 insure prog mode
84 6000 EXT NUM convert char to num
85 6000 EXT NUMVA+ scan, get number
86 6000 EXT NUMVAL get number
87 6000 EXT ON/OFF on/off parsing
88 6000 EXT ONEB get one binary
89 6000 EXT ONEI get one integer
90 6000 EXT ONER get one real
91 6000 EXT ONK12? anything on stack?
92 6000 EXT PILREP report PIL errors
93 6000 EXT PLIST print list pars
94 6000 EXT TYPEM1 position pointer
95 6000 EXT PRINT+ print using parse
96 6000 EXT PUJING output integer
97 6000 EXT PUSHIF parse file name
98 6000 EXT PUTADR put relative addr (R34)
99 6000 EXT PUTKEY put key hit
100 6000 EXT PUTREL put relative addr (drp)
101 6000 EXT RA do right arrow
102 6000 EXT RDYSHD send data
103 6000 EXT REFNUM get number ref
104 6000 EXT RELMEM release memory
105 6000 EXT RESCON reserve space
106 6000 EXT RESPUT reserve space
107 6000 EXT RONJSB JSB to rom >IV
108 6000 EXT RSHEN reserve memory
109 6000 EXT SCAM scan input
110 6000 EXT SETOFF do offio
111 6000 EXT SHLA shift left arrow
112 6000 EXT SKPCHK skip to next line (devfile)

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RYA101 OBJ=RYZ101 12/13/1983 11:50 AM PG 4
=====
>>>>>> exts, ents, equs, etc <<<<<<<<
166 6000 REOSBH EQU 0E05BH
167 6000 OFFBOM EQU OFFBOM
168 6000 OFFEOM EQU OFFEOM
169 6000 09+1 EQU '9'+1
170 6000 01-0 EQU '1'-0
171 6000 0A-0 EQU 'A'-0
172 6000 0A-1 EQU 'A'-1
173 6000 ASCIID EQU 0092H
174 6000 BEEPIM EQU 0004H Standard beep duration
175 6000 CNDTK EQU 0018H
176 6000 DAB EQU 0000H
177 6000 DATASY EQU 0086H
178 6000 DATAME EQU 0049H
179 6000 DDL EQU 00A0H
180 6000 DDLK EQU 001DH
181 6000 DDI EQU 00C0H
182 6000 DMMY EQU 00B0H
183 6000 ERR EQU 1800H
184 6000 END EQU 0040H
185 6000 EMOY EQU 00B8H
186 6000 ENME EQU 004BH
187 6000 EOLTK EQU 001AH
188 6000 EIE EQU 41A0H
189 6000 ETO EQU 40A0H Beep frequency = 220 Hz
190 6000 FREQ EQU 112D
191 6000 GIL EQU 0001H
192 6000 GILTK EQU 0024H
193 6000 JFC EQU 0090H
194 6000 IDY EQU 00C1H
195 6000 IDYCB EQU 00C0H
196 6000 IMWIK EQU 002FH
197 6000 IEMTK EQU 0032H
198 6000 IMYIK EQU 0030H
199 6000 IOE*IK EQU 0031H
200 6000 LAD EQU 0020H
201 6000 LLO EQU 0011H
202 6000 LPD EQU 003BH Loop power down frame
203 6000 MYLEMM EQU 12D
204 6000 MYROM EQU 4053H
205 6000 MYROM2 EQU 4064H
206 6000 MYROM3 EQU 4065H
207 6000 MRD EQU 42A0H
208 6000 MRE EQU 0093H
209 6000 R0Y EQU 00A0H
210 6000 REN EQU 0092H
211 6000 REVCDE EQU 'a'
212 6000 SAD EQU 0060H
213 6000 SAI EQU 63A0H
214 6000 SDA EQU 60A0H
215 6000 SDR' EQU 0060H
216 6000 SDC EQU 00A0H
217 6000 JAD EQU 0040H
218 6000 TALKTK EQU 0023H

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RYA101 OBJ=RYZ101 12/13/1983 11:50 AM PG 3
=====
>>>>>> exts, ents, equs, etc <<<<<<<<
113 6000 EXT SHDFRM send frame
114 6000 EXT STAND+ stand by set
115 6000 EXT SIOP? check attn and power
116 6000 EXT SIOSI store strang
117 6000 EXT SIOSV store number
118 6000 EXT STIREX+ scan, get string
119 6000 EXT STIREXP get string
120 6000 EXT STIRREF get string ref
121 6000 EXT SUBSTF do gosub stuff
122 6000 EXT SVCLMK check machines
123 6000 EXT SYSJSB jab to alt rom
124 6000 EXT TAB devfile name
125 6000 EXT THERE1 demand devfile
126 6000 EXT TORASC2 convert to ASCII
127 6000 EXT TUBED2 convert to BCD
128 6000 EXT TRJUMP trace a jump
129 6000 EXT UNLSHD send unl
130 6000 EXT UNIREP untalk and report
131 6000 EXT UPRC+ uppercase registers
132 6000 EXT VERNM3 version message rom 3
133 6000 EXT VFBYE untalk, unlisten
134 6000 EXT UCKJSB wait some
135 6000 EXT WAIT wait some
136 6000 EXT WAKEUP do RAD
137 6000
138 6000 TYPEM1 DAD 6723H
139 6000
140 6000 02 EQU 2D
141 6000 03 EQU 3D
142 6000 04 EQU 4D
143 6000 05 EQU 5D
144 6000 07 EQU 7D
145 6000 08 EQU 8D
146 6000 09 EQU 9D
147 6000 016 EQU 16D
148 6000 018 EQU 18D
149 6000 030 EQU 30D
150 6000 086 EQU 86D
151 6000 094 EQU 94D
152 6000 0256 EQU 256D
153 6000 0000EH EQU 0000EH
154 6000 0001FH EQU 001FH
155 6000 00020H EQU 0020H
156 6000 00030H EQU 0030H
157 6000 000C0H EQU 00C0H
158 6000 003FFH EQU 03FFH
159 6000 043A0H EQU 43A0H
160 6000 06000H EQU 6000H
161 6000 064A0H EQU 64A0H
162 6000 07FFFH EQU 7FFFH
163 6000 08000H EQU 8000H
164 6000 0C0FFH EQU 0C0FFH
165 6000 0E001H EQU 0E001H

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RYA101 OBJ=RYZ101 12/13/1983 11:50 AM PG 5
=====
>>>>>> exts, ents, equs, etc <<<<<<<<
219 6000 UNL EQU 003FH
220 6000 UNL EQU 005FH
221 6000 UNLTK+ EQU 002FH
222 6000
223 6000 E3 1C DEF RHHEAD header for all rows
224 6002
225 6002 16 60 DEF LENBGM address of 1st file
226 6004 E6 1F DEF (LENEND-LENBGM) size of the rom
227 6006 8D BY1 10001101B type (access)
228 6007 4C VAL 1VNLKX this is a system file
229 6008 20 7D 04 9D BY1 020H,070H,0C4H,09DH creation date
230 600C 69 6F 72 6F ASE 'ioron1' name of the file
231 6014
232 6014 BSZ 2D end of directory
233 6016

```

NOMAS
 recipient Not Manufacturer Supported
 agrees NOT to contact manufacturer

[illegible][illegible]

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA101 OBJ=RYX101 12/13/1983 11:50 AM PG 10
*****
***** ROM tables *****
411 6203 40 D0
412 6205 42 49 4E 45 ASP 'BIMOR' 36
413 6209 4F D2
414 620B 42 49 4E 49 ASP 'BIMOR' 37
415 620F 4F D2
416 6211 45 4E 5A 49 ASP 'ENTOR' 38
417 6215 4F 4A
418 6217 52 49 CF ASP 'RIO' 39
419 621A 53 50 4F 4C ASP 'SPOLL' 3A
420 621E 4C 4A
421 6220 53 50 4F 4C ASP 'SPOLL' 3B
422 6224 CC
423 6225 50 50 4F 4C ASP 'PPOLL' 3C
424 6229 CC
425 622A 41 44 4A 52 ASP 'ADDRESS' 3D
426 622E 45 53 D3
427 6231 44 45 56 49 ASP 'DEVIOB' 3E
428 6235 44 4A
429 6237 44 45 56 41 ASP 'DEVIOB' 3F
430 623B 49 4A 4A
431 623E 44 45 56 41 ASP 'DEVADDR' 40
432 6242 44 4A D2
433 6245 44 45 56 4E ASP 'DEVNAME' 41
434 6249 41 40 45 4A
435 624B 48 45 59 42 ASP 'KEYBOARD' 42
436 6251 4F 41 52 44
437 6255 4A
438 6256 54 4F 4E C5 ASP 'TONE' 43
439 625A 43 52 45 41 ASP 'CREATED' 44
440 625E 54 45 4A
441 6261 41 43 43 45 ASP 'ACCESS' 45
442 6265 53 53 4A
443 6268 4C 49 4E 50 ASP 'LIMITE' 46
444 626C 55 54 4A
445 626F 54 59 50 45 ASP 'TYPE' 47
446 6273 4A
447 6274 FF VAL FF 48 END OF TABLE
448 6275 44 41 54 C1 ASP 'DATA' 49
449 6279 80 VAL DUMMY 4A
450 627A 45 4E C4 ASP 'END' 4B
451 627D 80 VAL DUMMY 4C
452 627E
453 627F 61 VAL REVECODE revision number
454 627F 02 BYT 00000010B ROM LEX access bits
455 6280
456 6280

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA101 OBJ=RYX101 12/13/1983 11:50 AM PG 12
*****
***** HANDI call processing *****
457 62E2 FO D0 JMP AUTODN clean up
458 62F0
459 62F0
460 62F0
461 62F0
462 62F0
463 62F0
464 62F0
465 62F0
466 62F0
467 62F0
468 62F0
469 62F0
470 62F0
471 62F0
472 62F0
473 62F0
474 62F0
475 62F0
476 62F0
477 62F0
478 62F0
479 62F0
480 62F0
481 62F0
482 62F0
483 62F0
484 62F0
485 62F0
486 62F0
487 62F0
488 62F0
489 62F0
490 62F0
491 62F0
492 62F0
493 62F0
494 62F0
495 62F0
496 62F0
497 62F0
498 62F0
499 62F0
500 62F0
501 62F0
502 62F0
503 62F0
504 62F0
505 62F0
506 62F0
507 62F0
508 62F0
509 62F0
510 62F0
511 62F0
512 62F0
513 62F0
514 62F0
515 62F0
516 62F0
517 62F0
518 62F0
519 62F0
520 62F0
521 62F0
522 62F0
523 62F0
524 62F0
525 62F0
526 62F0
527 62F0
528 62F0
529 62F0
530 62F0
531 62F0
532 62F0
533 62F0
534 62F0
535 62F0
536 62F0
537 62F0
538 62F0
539 62F0
540 62F0
541 62F0
542 62F0
543 62F0
544 62F0
545 62F0

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA101 OBJ=RYX101 12/13/1983 11:50 AM PG 11
*****
***** HANDI call processing *****
441 6280 40 C8 3B INILEX CMB R00,=V.LFTY translating R00 type?
442 6283 F6 D0 IFEQ yes,
443 6285 51 C8 47 CMB R21,='G' general?
444 6288 F6 49 RNE no, leave
445 628A 66 A9 58 E0 LDM R46,=R05BH yes, get LIF type
446 628E 44 A9 AF 63 1H GTD HANDLE finish up
447 6292
448 6292
449 6292 C8 3C CMB R00,=V.RFTY translating LIF type?
450 6294 F6 D0 IFEQ yes,
451 6296 50 C9 58 E0 CMB R20,=R05BH general?
452 629A F6 37 RNE no, leave
453 629C 51 A8 47 LDB R21,='G' yes, get R00 type?
454 629F FO ED JMP 1B go on
455 62A1
456 62A1
457 62A1 C8 20 CMB R00,=V.WAIT doing waitkey?
458 62A3 F6 07 IFEQ yes,
459 62A5 06 E4 PUBD R00,=R06 save event
460 62A7 CE E8 79 JSB =DOKBD do it
461 62AA FO 21 JMP AUTODN clean up
462 62AC
463 62AC
464 62AC C8 02 CMB R00,=2D Cold or warm start?
465 62AC FO 24 IFNE yes,
466 62B0 06 E4 PUBD R00,=R06 save event number
467 62B2 80 40 84 LDBD R00,=AUTOLP get flag
468 62B5 F6 16 IFZR on,
469 62B7 54 E5 PUBD R24,=R06 save the term key
470 62B9 CE E1 6D JSB =RSHLP do the loop
471 62BC F6 0C IFEZ success, no beep
472 62BE 50 A9 70 00 LDM R20,=FREQ get frequency
473 62C2 62 93 CLM R42 build
474 62C4 64 88 04 LDB R44,=BEEPLN duration
475 62C7 CE FF FF JSB =BEEPER beep
476 62C9
477 62CA 54 06 E3 PUBD R24,=R06 get back key
478 62CB
479 62CB 40 06 E2 AUTODN PUBD R00,=R06 restore event
480 62DD 82 90 82 STBD R00,=R06L number
481 62D3 9E RTN
482 62D4
483 62D4
484 62D4 C8 03 CMB R00,=V.SLEEP going to sleep?
485 62D6 F6 18 IFEQ yes,
486 62D8 06 E4 PUBD R00,=R06 save event
487 62DA 80 40 84 LDBD R00,=AUTOLP get flag
488 62DD F6 D0 IFZR on,
489 62DF CE FF FF JSB =LROFF clear offio and
490 62E2 CE FF FF JSB =LPDFF set test loop bit
491 62E5 6F 88 98 LDB R57,=LPD loop power down
492 62EB CE FF FF JSB =LMSMD send the command
493 62ED CE FF FF JSB =SETOFF set offio bit

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA101 OBJ=RYX101 12/13/1983 11:50 AM PG 13
*****
***** HANDI call processing *****
546 6365 89 ICM R00 for neg push
547 6366 42 89 ICM R02 back up 1 char
548 6368 79 A8 D0 LDB R71,=CR get CR
549 636B 00 A6 STBD R71,R00 put it out
550 636D
551 636D 42 12 C1 CMB R02,R22 move loop
552 6370 F6 03 IFEQ at end of name?
553 6372 7F 00 E6 PUBD R77,=R00 yes,
554 6375
555 6375 79 02 E2 PUBD R71,=R02 put out '
556 6378 00 E6 PUBD R71,=R00
557 637A 42 C9 80 81 CMB R02,=IMPBUF get char
558 637E F6 D0 WHCY move it
559 6380 7A B3 80 81 SIND R72,=IMPBUF at beginning?
560 6384 9E RTN no, until done
561 6385
562 6385 40 C8 38 CHAR.V CMB R00,=V.CHAR add CALL '
563 6388 F6 13 IFEQ done
564 638A 42 B0 C8 82 LDBD R02,=ROUTE doing char out?
565 638E C8 03 CMB R02,=3D yes,
566 6390 F6 D0 IFEQ get the route
567 6392 CE C4 64 JSB =OUTSUP is it us?
568 6395 6F 1A 80 LDB R57,R32 yes,
569 6398 CE A5 71 JSB =DOED setup, save regs
570 639B FO 13 JMP HANDLE get the char
571 639D
572 639D
573 639D
574 639D 40 C8 39 CMB R00,=V.ENDL output the char
575 639A F6 14 IFEQ handle it
576 63A2 42 B0 C8 82 LDBD R02,=ROUTE doing an eol?
577 63A6 C8 03 CMB R02,=3D yes,
578 63A8 F6 D0 IFEQ get the route
579 63AA CE C4 64 JSB =OUTSUP is it us?
580 63AD CE D4 71 JSB =EOL setup, save regs
581 63B0 42 92 HANDLE CLR R02 do the eol
582 63B2 82 75 83 STBD R02,=HANDLE clear
583 63B5 9E RTN to say handled
584 63B6
585 63B6
586 63B6
587 63B6 40 C8 14 CMB R00,=V.CHED chedit?
588 63B9 F6 7A RNE no, leave
589 63BB 42 B0 D8 82 LDBD R02,=JUSTSO leave alone?
590 63BF F6 74 RNE yes, leave
591 63C1 60 18 88 CMB R40,=JKEY? I/R key?
592 63C4 F6 13 IFEQ yes,
593 63C6 42 B0 5A 82 LDBD R02,=IOBYTE our version?
594 63CA F3 D0 IFOD yes,
595 63CC 40 D8 82 LDB R02,=I/RLG get the flag
596 63CF 8E RLB R02 toggle bit
597 63D0 82 D8 82 STBD R02,=I/RLG put it back
598 63D3 CE FF FF JSB =CURSE? set cursor

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYZ101 12/13/1983 11:50 AM PG 14
*****
>>>>>> HMMOI call processing *****
599 6306 FO 57 JMP MOPDNE finish up
600 6308 ENDF
601 6308 9E RTN no, leave
602 6309 ENDF
603 6309 42 92 CLD R02 assume leftarrow
604 6308 60 C8 E6 CMB R40,=DOEGH cntl/shift/left arrow?
605 630E F7 06 JEQ DMOHVE
606 63E0 C8 E7 CMB R40,=DOE7H cntl/shift/right arrow?
607 63E2 F6 E8 IFEQ yes,
608 63E4 42 8A DCB R02 make at right arrow
609 63E6 42 82 84 #4 DMOHVE STBD R02,=INPM3 save flag
610 63EA CE FF FF JSB =CURSE cursor on
611 63ED CE FF FF JSB =GETCHR get the char
612 63F0 56 02 80 LDB R26,R02 save char
613 63F3 CE FF FF JSB =CURSE cursor off
614 63F6 50 B1 60 83 LDM R20,=INPM3 get cursor
615 63FA 40 B0 64 84 LDBD R00,=INPM3 get dir flag
616 63FE F5 09 IFNG right,
617 6400 50 89 ICM R20 move one right
618 6402 52 B1 58 83 LDM R22,=LASTCH get end
619 6406 89 ICM R22 adjust for loop
620 6407 FO 07 ELSE left,
621 6409 50 88 DCM R20 move one left
622 640B 52 A9 80 B1 LDM R22,=INPM3 get start
623 640F 80 DCM R22 adjust for loop
624 6410 ENDF
625 6410 57 92 CLD R27 clear accum
626 6412 LOOP search loop
627 6412 57 88 ICB R27 inc count
628 6414 56 10 08 CMBD R26,R20 char wanted?
629 6417 F6 1F IFEQ yes,
630 6419 LOOP move loop
631 6419 40 B0 64 84 LDBD R00,=INPM3 get dir flag
632 641D F5 07 IFNG right,
633 641F CE FF FF JSB =SYSJSB move right
634 6422 F7 FF DEF RA
635 6424 FO 05 ELSE left,
636 6426 CE FF FF JSB =SYSJSB move left
637 6429 F7 FF DEF RA
638 642B ENDF
639 642B 57 80 DCM R27 dec count
640 642D F6 EA UHM7 until done
641 642F 60 88 14 MOPDNE LDB R40,=V.LHED get event number
642 6432 B2 90 82 STBD R40,=ROMFL restore it
643 6435 88 FF LDB R40,=NOPKEY put out a nopkey
644 6437 9E RTN done
645 643B ENDF
646 643B 40 B0 64 84 LDBD R00,=INPM3 get dir flag
647 643C F5 06 IFNG right,
648 643E 50 89 ICM R20 inc pointer
649 6440 12 C1 LCM R20,R22 at end?
650 6442 FO 04 ELSE left,
651 6444 50 8B DCM R20 dec pointer

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYZ101 12/13/1983 11:50 AM PG 16
*****
>>>>>> HMMOI call processing *****
705 640F ENDF
706 640F 9E RTN
707 64B0 VER? A get code for call
708 64B0 LDM R43,=
709 64B0 63 A9 LDBD R42,=VERM3
710 64B2 62 B0 FF FF RTN
711 64B6 9E STBD R43,=IMP3+1 put code out
712 64B7 63 B3 65 84 JSB =SYSJSB call it
713 64B8 CE FF FF DEF IMP3+1 to get version
714 64BE 65 84 CMB R42,='d' test version
715 64C0 62 C8 64 RTN
716 64C3 9E
717 64C4
718 64C4 42 06 E3 DUISUR POMB R02,=-R06 get return addr
719 64C7 60 E5 PUMB R40,=R06 push R40/47
720 64C9 18 A1 LDM R40,R30 get R30/47
721 64CB 06 E5 PUMB R40,=R06 push em
722 64CD 58 02 A1 LDM R30,R02 move rln addr
723 64D0 42 A8 01 LDB R02,=10 get data flag
724 64D3 B2 64 84 STBD R02,=INPM3 set it
725 64D6 18 C6 00 00 JSB N30 go do rest of stuff
726 64D8 60 06 E3 POMB R40,=-R06 get R30/37
727 64DB 18 A3 STM R40,R30 restore em
728 64D1 06 E3 POMB R40,=-R06 restore R40/47
729 64E1 9E RTN
730 64E2

```

NOMAS
 Not Manufacturer Supported
 Recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYZ101 12/13/1983 11:50 AM PG 15
*****
>>>>>> HMMOI call processing *****
652 6446 52 C1 CMB R22,R20 at start?
653 6448 ENDF
654 6448 FA C8 UHNC continue until
655 644A FO E3 JMP MOPDNE not found
656 644C ENDF
657 644C
658 644C C8 C8 CMB R40,=DOCAH cntl/del key?
659 644E F6 5F IFEQ yes,
660 6450 42 B1 60 83 LDM R02,=INPM3 get position
661 6454 C0 80 B1 STB R02,=INPM3 compute count
662 6457 17 A2 STB R02,R27 save it
663 6459 CE B0 64 JSB =VER? which version?
664 645C FA 07 IFEQ new,
665 645E CE FF FF JSB =SYSJSB move to far left
666 6461 FF FF DEF SHLA
667 6463 FO 05 ELSE old,
668 6465 CE FF FF JSB =SYSJSB move to far left
669 6468 42 7F DEF SHLA+2
670 646A ENDF
671 646A LOOP number loop
672 646A 43 B8 60 83 LDBI R03,=INPM3 get char
673 646E C8 30 CMB R03,='0' < 0?
674 6470 FA 04 JMC DELLP yes, continue
675 6472 C8 3A CMB R03,='9'+1 > 9?
676 6474 FA 27 IFCY yes,
677 6476 43 C8 20 DELLP CMB R03,=' ' on a blank?
678 6479 F6 07 IFEQ yes,
679 647B CE FF FF JSB =SYSJSB move one more
680 647E FF FF DEF RA
681 6480 57 8A DCM R27 adjust length
682 6482 ENDF
683 6482 57 8A DCM R27 dont del cursor
684 6484 LOOP del loop
685 6484 57 90 TSB R27 check count
686 6486 F4 A7 JMC MOPDNE done
687 6488 CE B0 64 JSB =VER? which version?
688 648B FA 07 IFCY new,
689 648D CE FF FF JSB =SYSJSB do a del char
690 6490 FF FF DEF DEL
691 6492 FO 05 ELSE old,
692 6494 CE FF FF JSB =SYSJSB do a del char
693 6497 41 70 DEF DEL+2
694 6499 ENDF
695 6499 57 8A DCM R27 dec count
696 649B FO E7 UHM7 go on
697 649D ENDF
698 649D 42 B1 60 83 LDM R02,=INPM3 get pointer
699 64A1 D1 58 83 CMB R02,=LASTCH check for end
700 64A4 FB 89 JCY MOPDNE done
701 64A6 CE FF FF JSB =SYSJSB move right
702 64A9 FF FF DEF RA
703 64AB 57 8A DCM R27 adjust count
704 64AD FO B8 UHM7 go on

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYZ101 12/13/1983 11:50 AM PG 17
*****
>>>>>> Parse routines *****
732 64E2 CE 44 65 ROM3S JSB =GTSTP+ get a string
733 64E5 F7 04 IFNE comma?
734 64E7 CE FF FF JSBN =ERR1+ no, err out
735 64EA 51 BYT B1D
736 64EB ENDF
737 64EB 4C 1E A0 LDB R14,R36 move token
738 64EE
739 64EE ROM2S
740 64EE CE 44 65 JSB =GTSTP+ get a string
741 64F1 F7 04 IFNE comma?
742 64F3 CE FF FF JSBN =ERR1+ no, errout
743 64F6 51 BYT B1D
744 64F7 ENDF
745 64F7 4C 1E A0 LDB R14,R36 move token
746 64FA
747 64FA ROM1S
748 64FA CE FF FF JSB =SYSJSB get a string
749 64FD FF FF DEF GET1B and put out token
750 64FF FO 30 JMP RONTOK go clean up
751 6501
752 6501 4C 1E A2 ROM1S? STB R14,R36 save token
753 6504 CE FF FF JSB =SYSJSB scan, see if
754 6507 FF FF DEF STREX+ string there
755 6509 5E 0A E4 PUMB R36,=R12 token to stack
756 650C FO 23 JMP RONTOK finish up
757 650E
758 650E CE FF FF ON/OFF+ JSB =SYSJSB get the on/off
759 6511 FF FF DEF ON/OFF word
760 6513 FO 1C JMP RONTOK finish up
761 6515
762 6515 CE FF FF ROM1H+ JSB =NOCLC+ not an calc mode
763 6518
764 6518 50 A8 01 ROM1H LDB R35,=1 want one number
765 651B FO 03 JMP GET1NB get it
766 651D
767 651D 50 A8 02 ROM2H LDB R35,=2 we want two numbers
768 6520
769 6520 CE FF FF GET1NB JSB =SYSJSB go get the
770 6523 FF FF DEF GETPAR parameters
771 6525 FO 0A JMP RONTOK finish up
772 6527
773 6527 CE FF FF ROM1F JSB =SYSJSB go get a filename
774 652A FF FF DEF PUSHIF
775 652C FO 03 JMP RONTOK clean up
776 652E
777 652E 4C ROM14+ DRP 14 for some
778 652F 0A E4 RONTK+ PUMB RN,=R12 shove token
779 6531 6C A9 RONTOK LDM R54,=
780 6533 B4 VAL EROTKM get the rom switch token
781 653A 63 40 GET NYAOTM and ny rom number
782 6536 00 BYT 0 and garbage
783 6537 6F 0A E2 POMB R57,=R12 get ny token
784 653A 6C E5 PUMB R54,=R12 put it all back

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYZ101 12/13/1983 11:50 AM PG 18
-----
>>>>>> Parse routines <<<<<<<
785 653E 9E RTN
786 653D
787 653D CE 2E 65 ROM1A JSB =ROM14+ adjust output
788 6540 CE 4A 65 JSB =SSCAN do a post scan
789 6543 9E MYRTM RTM
790 6544
791 6544 CE FF FF GISTP+ JSB =SYSJSB do a sysjsb
792 6547 FF FF DEF GETSTP to getstb
793 6549 9E RTN
794 654A
795 654A CE FF FF SSCAN JSB =SYSJSB do a sysjsb
796 654D FF FF DEF SCAN to scan
797 654F 9E RTN
798 6550
799 6550
800 6550 68 A9 SAFJSB + LDM R50,+ get a ROMJSB
801 6552 CE FF FF JSB =ROMJSB
802 6555 00 00 DEF (0)
803 6557 00 00 DEF (0)
804 6559 9E RTN
805 655A 00 06 E3 POMB R00,-R06 get rtindr (data)
806 655D 02 00 E1 POMB R02,-R00 get the address
807 6560 00 06 E5 POMB R00,-R06 put rtindr (data-)
808 6563 42 2B A3 STM R02,R53 set the address
809 6566 93 CLM R02 assume rom 0
810 6567 67 18 04 CNB R47,-O4H number constant?
811 656A F7 03 IFNE no,
812 656C 42 21 A1 LDM R02,R41 get rom #
813 656F ENDF
814 656F 42 2D A3 STM R02,R55 set the rom #
815 6572 68 83 64 84 STMB R50,+THPM3 put it out
816 6576 CE 64 84 JSB =THPM3 call it
817 6579 9E RTN
818 657A
819 657A CE 82 60 84 G11M7 STBD R14,-THP3+9 save our token
820 657E CE FF FF JSB =SYSJSB scan and
821 6581 FF FF DEF MUMVA+ try a number
822 6583 F8 03 IFEZ nope,
823 6585 CE 69 66 JSB =SAFSTR go and try $
824 6588 ENDF
825 6588 42 80 60 84 LDBD R02,-THP3+9 get back our token
826 658C CE 2F 65 JSB =ROM1K+ put it out
827 658F 9E RTN
828 6590
829 6590 CE FF FF PRSKBD JSB =SYSJSB get the parameter
830 6593 FF FF DEF G160R+
831 6595 CE 31 65 JSB =ROM1K+ put out our stuff
832 6598 9E RTN
833 6599
834 6599 CE 7A 65 PRSMIR JSB =G11M7 get the device
835 659C 4C 18 08 CNB R14,-USTOK USING?
836 659F F6 17 IFEQ yes,
837 65A1 1E A2 STB R14,R35 save token

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYZ101 12/13/1983 11:50 AM PG 20
-----
>>>>>> Parse routines <<<<<<<
831 660F 9E RTN
832 6610
833 6610 4C PRSSMD DRP 14 token here
834 6611 CE 2F 65 PRSSMD+ JSB =ROM1K+ token out
835 6614 CE 4A 65 JSB =SSCAN scan for next
836 6617 CE 94 66 PRSSMD- JSB =CHKSEN check it out
837 661A F8 F3 REN error, leave
838 661C F7 F1 RZR done, leave
839 661E 4C 18 2F CNB R14,-UNITK+ END or DATA?
840 6621 F8 1E IFMC no,
841 6623 18 24 CNB R14,-G11K no param type?
842 6625 F8 E9 JCY PRSSMD yes, go again
843 6627 18 23 CNB R14,-TALKTK talk?
844 6629 F6 05 IFEQ yes,
845 662B CE 18 65 JSB =ROM1K get the addr
846 662E F0 E7 JMP PRSSMD- ok, go on
847 6630 ENDF
848 6630 CE 1D CNB R14,-DOLTK multi # type?
849 6632 FA 0D IFCY yes,
850 6634 50 92 CLB R35 get as many as there
851 6636 CE 20 65 JSB =GE1M7 get the numbers
852 6639 5C 90 TSB R34 get any?
853 663B F6 0A JNZ PRSSMD- ok, go on
854 663D CE FF FF MOPARM JSB =ERR1+ no, error out
855 6640 5B BYT 91D
856 6641 ENDF
857 6641
858 6641 82 6D 84 STBD R14,-THP3+9 save token
859 6644 83 ICB R14 get inv token
860 6645 CE 2E 65 JSB =ROM1K+ token out
861 6648 CE 6F 66 JSB =GE1M7 get number or strings
862 664B 42 80 6D 84 LDBD R02,-THP3+9 get token
863 664F CE 2F 65 JSB =ROM1K+ put it out
864 6652 42 18 18 CNB R02,-CMDTK was it CMD?
865 6655 F7 10 JEQ PRSSMD- go on
866 6657 6D 21 A1 LDM R55,R41 get rom#/token
867 665A C9 CNM R55,-
868 665B 63 40 DEF MYROMM my rom?
869 665D 1A VAL EOLIK EOL token?
870 665E 6F DRP 57 point to token
871 665F F7 80 JEQ PRSSMD- EOL, put out
872 6661 F0 84 JMP PRSSMD- no, check scan
873 6663
874 6663
875 6663 CE 50 65 SAFNUM JSB =SAFJSB do a safe job
876 6666 FF FF DEF MUMVAL to numval
877 6668 9E RTN
878 6669
879 6669 CE 50 65 SAFSTR JSB =SAFJSB do a safe job
880 666C FF FF DEF STREXP to strexp
881 666E 9E RTN
882 666F
883 666F CE 4A 65 GE1M7 JSB =SSCAN do pre-scan

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYZ101 12/13/1983 11:50 AM PG 19
-----
>>>>>> Parse routines <<<<<<<
838 65A3 8E 80 LDB R14,-ULMTOK assume line#
839 65A5 CE FF FF JSB =SYSJSB try to get it
840 65A8 FF FF DEF GETLIM no,
841 65AA F8 0C IFEZ
842 65AC CE 69 66 JSB =SAFSTR get the string
843 65AF F8 04 IFEZ no,
844 65B1 CE FF FF BADENT JSB =ERR1+ error out
845 65B4 59 BYT 89D
846 65B5 ENDF
847 65B5 5E 0A E4 PUMB R36,-R12 push string token
848 65B8 ENDF
849 65B8
850 65B8 4C 18 27 CNB R14,-SEMI semicolon?
851 65B8 F6 1E IFEQ yes,
852 65B8 0A E4 PUMB R14,-R12 push it
853 65B8 LOOP param list
854 65B8 CE 4A 65 JSB =SSCAN scan for next
855 65C2 CE FF FF JSB =SYSJSB try a string
856 65C5 FF FF DEF STAREF
857 65C7 F8 07 IFEZ no,
858 65C9 CE FF FF JSB =SYSJSB try a number
859 65CC FF FF DEF REFNUM
860 65CE F9 E1 JEZ BADENT error out
861 65D0 ENDF
862 65D0 42 8B 32 LDB R02,-IENITK get inv token
863 65D3 CE 2F 65 JSB =ROM1K+ add it
864 65D6 4C 18 2C CNB R14,-COMMA more?
865 65D9 F7 E4 UNED until done
866 65DB ENDF
867 65DB 42 8B 32 LDB R02,-IENITK get inv token
868 65DE CE 2F 65 JSB =ROM1K+ put it out
869 65E1 F0 26 JMP DEPEND finish up
870 65E3
871 65E3 CE 7A 65 PRSOUT JSB =G11M7 get number or string
872 65E6 5A 92 CLB R32 clear for later
873 65E8 4C 18 27 CNB R14,-SEMI semicolon?
874 65EB F6 0C IFEQ yes,
875 65ED 0A E4 PUMB R14,-R12 push it
876 65EF CE 4A 65 JSB =SSCAN scan
877 65F2 CE 50 65 JSB =SAFJSB go do the list
878 65F5 FF FF DEF PLIST
879 65F7 F0 10 ELSE no,
880 65F9 CE 1B CNB R14,-USTOK using?
881 65FB F7 07 IFMC no,
882 65FD 42 8B A2 LDB R02,-CRITOK get wol token
883 6600 0A E4 PUMB R02,-R12 push it out
884 6602 F0 05 ELSE yes,
885 6604 CE FF FF JSB =SYSJSB go process using/list
886 6607 FF FF DEF PRINT+
887 6609 ENDF
888 6609
889 6609 42 8B 31 DEPEND LDB R02,-IODETK add invisible token
890 660C CE 2F 65 JSB =ROM1K+ to output

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYZ101 12/13/1983 11:50 AM PG 21
-----
>>>>>> Parse routines <<<<<<<
944 6672 LOOP get # or $
945 6672 CE 63 66 JSB =SAFNUM scan for #
946 6675 F9 08 IFEM yes,
947 6677 42 8B 2F LDB R02,-INVMTK get token
948 667A CE 2F 65 JSB =ROM1K+ put it out
949 667D F0 08 ELSE no,
950 667F CE 69 66 JSB =SAFSTR go and
951 6682 F9 89 JEZ MOPARM no, error out
952 6684 42 8B 30 LDB R02,-INVMTK get token
953 6687 CE 2F 65 JSB =ROM1K+ put it out
954 668A ENDF
955 668A 4C 18 2C CNB R14,-COMMA more?
956 668D F6 0F RNE no, done
957 668F CE 4A 65 JSB =SSCAN scan next
958 6692 F0 DE WIMP go on
959 6694
960 6694 42 21 A1 CHKSEN LDM R02,R41 get rom# found
961 6697 F6 1F IFZR if system
962 6699 CE FF FF JSB =EOL14? CN, EOL, @, ', ELSE
963 669C F8 03 IFEZ yes,
964 669E 42 92 CLB R02 set zero flag
965 66A0 9E RTN
966 66A1 ENDF
967 66A1 4C 18 86 CNB R14,-DATASY DATA?
968 66A4 F6 04 IFEQ yes,
969 66A6 8B 49 LDB R14,-DATASY make it mine
970 66A8 F0 0C ELSE no,
971 66AA 18 8A CNB R14,-ENDSY END?
972 66AC F6 04 IFEQ yes,
973 66AE 8B 4B LDB R14,-ENDME make it mine
974 66B0 F0 04 ELSE no,
975 66B2 CE FF FF BADTKM JSB =ERR1+ error out
976 66B5 5B BYT 88D
977 66B6 ENDF
978 66B6
979 66B6 F0 11 ELSE no,
980 66B8 19 63 40 CNB R02,-MYROMM my rom?
981 66BB F6 F5 JNE BADTKM no, error
982 66BD 63 18 18 CNB R43,-CMDTK < range?
983 66C0 F8 F0 JMC BADTKM yes, error
984 66C2 CE 2F 65 CNB R43,-UNITK+ > range?
985 66C4 F8 E1 JCY BADTKM yes, error
986 66C6 4C 23 A0 LDB R14,R43 get token
987 66C9 ENDF
988 66C9
989 66C9 90 CLE clear error flag
990 66CA 9E RTN
991 66CB
992 66CB 4C 1E A2 PRSMDI STB R14,R36 save our token
993 66CE CE 4A 65 JSB =SSCAN scan next
994 66D0 CE FF FF JSB =SYSJSB go get an array
995 66D3 FF FF DEF FORNAR
996 66D6 F8 04 IFEZ none found,
997 66D8 CE FF FF JSB =ERR1+ error out

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 22
-----
>>>>>> Parse routines <<<<<<<
997 660B 59 BYT 89D
998 660C EMDIF
999 660D 5E DRP 36 our token here
1000 660D CE 2F 65 JSB =RONTK+ finish up
1001 660E 9E RIN
1002 66E1

```

NOMAS

Not Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 24
-----
>>>>>> goto/gosub <<<<<<<
1044 6725 A1 BYT 241
1045 6726 CE FF FF GOTOX. JSB =ONEI get the number
1046 6729 98 BIN for sure
1047 672A 65 91 TSM R45 test for -
1048 672C F5 01 IFNG yes,
1049 672E 93 CLN R45 make it 0
1050 672F EMDIF
1051 672F 7E 25 A1 LDM R76,R45 R76 gets line number
1052 6732 CE FF FF JSB =FNDLRH find it in the runfile
1053 6735 5E 08 A3 STM R36,R10 setup new R10
1054 6738 42 A9 99 A9 LDM R02,=EMOLIM get the last line#
1055 673C D9 CMD R02,R10 last line?
1056 673D 5E DRP 36 for branch
1057 673E F6 05 IFEQ yes,
1058 6740 83 40 82 STMD R36,=PCR set PCR
1059 6743 F0 03 ELSE no,
1060 6745 CE FF FF JSB =TRJUMP do trace
1061 6748 EMDIF
1062 6748 CE FF FF JSB =GORTN go to this line
1063 674B 9E RIN
1064 674C
1065 674C A1 BYT 241
1066 674D 98 GOSBK. BIN to be safe
1067 674E 5C A9 00 80 LDM R34,=8000H get normal flag
1068 6752 5E 08 A1 LDM R36,R10 get return address
1069 6755 CE FF FF JSB =SUBSTF do the sub stuff
1070 6758 F9 CC JEZ GOTOX. ok, finish up
1071 675A 9E RIN
1072 675B

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 23
-----
>>>>>> bit twiddling stuff <<<<<<<
1004 66E1 20 2D BYT 40,55
1005 66E3 CE 0D 67 BAND. JSB =BINSUP do some setup
1006 66E6 5E 10 C7 AND R36,R20 do the and
1007 66E9 F0 1D JNP BINEND finish up
1008 66EB
1009 66EB 20 2D BYT 40,55
1010 66ED CE 0D 67 BEOR. JSB =BINSUP do some setup
1011 66F0 5E 10 97 XOR R36,R20 do the xor
1012 66F3 F0 1D JNP BINEND finish up
1013 66F5
1014 66F5 20 2D BYT 40,55
1015 66F7 CE 0D 67 BIOR. JSB =BINSUP do some setup
1016 66FA 5E 10 95 ORP R36,R20 do the or
1017 66FD F0 07 JNP BINEND finish up
1018 66FF
1019 66FF 10 2D BYT 20,55
1020 6701 CE 13 67 BCMP. JSB =BENSP- do some setup
1021 6704 5E 8F NCM R36 do the comp
1022 6706
1023 6706 BINEND
1024 6706 CE FF FF MUMOUT JSB =CONBIN convert
1025 6709 60 0A E5 PUMD R40,=R12 push to stack
1026 670C 9E RIN
1027 670D
1028 670D CE 1A 67 BINSIP JSB =ONEB+ get one number
1029 6710 50 26 A1 LDM R20,R46 save it
1030 6713 CE 1A 67 BINSIP JSB =ONEB+ get another number
1031 6716 5E 26 A1 LDM R36,R46 move it
1032 6719 9E RIN
1033 671A
1034 671A CE FF FF ONEB. JSB =ONEB get one number
1035 671D F9 05 IFEN overflow,
1036 671F 66 91 TSM R46 neg?
1037 6721 F5 01 IFNG yes,
1038 6723 8D CLN R46 adjust for 8000H
1039 6724 EMDIF
1040 6724 EMDIF
1041 6724 9E RIN
1042 6725

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 25
-----
>>>>>> shell on/off toggle <<<<<<<
1074 675B A1 BYT 241
1075 675C 42 0A E2 SHLIG. POBO R02,=R12 get user input
1076 675F B2 59 E2 STBD R02,=SHELK save the new status
1077 6762 9E RIN
1078 6763

```



```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RVY&101 12/13/1983 11:50 AM PG 26
*****
***** ENTIO& enter a string *****
1080 6763 2A 2E      BYT 52,56      interpreter leaves BCD
1081 6765 9B      N100&. BIN      get the var file
1082 6766 CE 8F 79      JSB =AROP+      error out
1083 6769 F8 F7      REN      get the IOSIZE
1084 6768 6E 1E 05 10      LDND R56,R36,R16
1084 676F 00      IFZR      not set,
1085 6770 F6 03      LDN R56,-R256      get default size
1086 6772 A9 00 01      ENDIF
1087 6775 50 A9 00 1F      LDN R20,-0,31D      set flags
1088 6779 61 83 A5 84      STD R56,=THEDIE      save it
1089 6770 CE 1F 69      JSB =STSHD      standby on and set
1091 6780      A      return addr for CLSTND
1092 6780 7D A8 40      LDND R75,=IAD      default command for ENIO
1093 6783 CE 33 68      JSB =STIR10      send all commands in list
1094 6786 F8 70      JEN MULEM1      errors in commands
1095 6788 70 93      CLM R75      clear char count
1096 678A 6E CF E0 FF      RMM R56,=OFFEOH      returned frame
1097 678E 05 E5      PUMD R56,=R06      save frame
1098 6790 80 90 84      LDND R56,=FLHEAD      test no data flag
1099 6793 F6 15      IFZR      not set,
1100 6795 80 A7 84      LDND R56,=(THEDIE+2D) get upper byte
1101 6798 F6 07      JNZ IMVSIZ      too big, error
1102 679A 81 A5 84      LDND R56,=THEDIE      get size
1103 6790 F4 02      JNG IMVSIZ      too big
1104 679F F6 06      IFZR      0,
1105 67A1 CE FF FF      JSB =ERR1      error
1106 67A4 59      BYT B9D      ok,
1107 67A5 F0 03      ELSE      get ram
1108 67A7 CE FF FF      JSB =RSEN-
1109 67A8      ENDIF
1110 67A8      ENDIF
1111 67A8 6E 06 E3      PUMD R56,-R06      get frame
1112 67A9 F9 06      IFEN      no room,
1113 67A8 92      CLB R56      clear IL
1114 67B0 8D 69 84      SIBD R56,=IMP3+5      flag to shut up talker
1115 67B3 F0 43      JNP MULEM1      null string out
1116 67B5      ENDIF
1117 67B5 42 B1 67 84      LDND R02,=IMP3+3      get last command sent
1118 67B9 C9 C1 00      CMN R02,=IDV      was it IDV
1119 67BC F7 14      JEQ ONECHR      yes, put the char
1120 67BE 43 C8 80      CMN R03,=OOROH      could it be ARG
1121 67C1 FA 20      IFY      yes,
1122 67C3 42 C8 A0      CMN R02,=OOROH      really,
1123 67C6 F6 10      IFEQ      yes,
1124 67C8 43 CF 1F      RMM R03,=001FH      check that
1125 67C8 C8 1F      ENB R03,=001FH      for IAG
1126 67CD F7 29      JEQ MULEM1      yes, leave
1127 67CF 6F CF 1F      RMM R57,=001FH      clear for count
1128 67D2 42 B0 9D 84 ONECHR      LDND R02,=FLHEAD      no data set?
1129 67D6 F6 20      JNZ MULEM1      yes, null out
1130 67D8 6F 16 A6      SIBD R57,R26      move to output
1131 67D8 6E 93      CLM R56      build a 1

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RVY&101 12/13/1983 11:50 AM PG 28
*****
***** ENTIO& enter a string *****
1184 684F F0 02      ELSE      no,
1185 6851 40 92      CLB R00      clear loc
1186 6853      ENDIF
1187 6853 B2 99 84      STBD R00,=(WRIPRO+2D) save new loc
1188 6856      ENDIF
1189 6856 43 24 A0      LDB R03,R44      get frame back
1190 6859 CF 40      RMM R03,=END      isolate END bit
1191 685B C8 40      CMN R03,=END      end bit set?
1192 685D F6 33      IFEO      yes,
1193 685F 80 6A 84      LDND R03,=IMP3+6      was CL+ signified
1194 6862 F7 0B      IFNZ      yes,
1195 6864 A8 00      LDB R03,=CR      get a CR
1196 6866 CE C5 68      JSB =TOOBIG      send it
1197 6869 43 A8 0A      LDB R03,=LF      get a LF
1198 686C CE C5 68      JSB =TOOBIG      send it
1199 686F      ENDIF
1200 686F 43 B0 9B 84      LDND R03,=(WRIPRO+4D) was EL+ signified
1201 6873 F7 13      IFNZ      yes,
1202 6875 B0 C4 82      LDND R03,=EOLLEN      get endline size
1203 6878 F7 0E      IFNZ      if none
1204 687A 40 A9 C1 82      LDN R00,=EOL      get pointer
1205 687E      LOOP      move loop
1206 687E 42 00 E0      POBD R02,=R00      get character
1207 6881 CE C5 68      JSB =TOOBIG      check size, put out
1208 6884 43 A8      DCB R03      dec count
1209 6886 F6 F6      UNWZ      until done
1210 6888      ENDIF
1211 6888      ENDIF
1212 6888 80 9E 84      LDND R03,=(FLHEAD+1D) end terminate?
1213 688B F7 05      IFNZ      yes,
1214 688D 40 A8 FF      LDB R00,=FF      get term flag
1215 6890 F0 85      JNP M1ERN      finish up
1216 6892      ENDIF
1217 6892      ENDIF
1218 6892 67 B0 6C 84      LDND R47,=IMP3+8      get a terminator
1219 6896 F7 04      IFNZ      specified,
1220 6898 2F C0      CMN R47,R57      does it match
1221 689A F7 49      JEQ FIXEND      yes, finish up
1222 689C      ENDIF
1223 689C 80 90 84      LDND R47,=(HPHEAD+1D) get IR+ size
1224 689F F7 1B      IFNZ      set,
1225 68A1 40 93      CLM R00      clear for load
1226 68A3 B0 9A 84      LDND R00,=(WRIPRO+3D) get current loc
1227 68A6 66 00 B4 91      LDND R46,MOO,(HPHEAD+2D) get char
1228 68A8 2F C0      CMN R46,R57      match?
1229 68AD F6 08      IFEQ      yes,
1230 68AF 40 88      ICB R00      inc loc
1231 68B1 27 C0      CMN R00,R47      done?
1232 68B3 FB 30      JCY FIXEND      yes, finish up
1233 68B5 F0 02      ELSE      no,
1234 68B7 40 92      CLB R00      clear loc
1235 68B9      ENDIF

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RVY&101 12/13/1983 11:50 AM PG 27
*****
***** ENTIO& enter a string *****
1132 67D0 89      ICM R56      for length
1133 67D6 DA E5      PUMD R56,=R12      length out
1134 67D0 56 E5      PUMD R26,=R12      address out
1135 67E2 9E      RTN      all done
1136 67E3      ENDIF
1137 67E3      ENDIF
1138 67E3 42 C9 A0 64      CMN R02,=R6AROH      send range?
1139 67E7 FB 0F      JCY MULEM1      no,
1140 67E9 C9 A0 43      CMN R02,=R6AROH      send range?
1141 67EC FA 0A      JNC MULEM1      no, go on
1142 67EE 2E C1      CMN R02,R56      frame same as sent
1143 67F0 F6 0B      JNE HEXAND      no, enter string
1144 67F2 F0 04      JNP MULEM1      yes, give null string
1145 67F4      ENDIF
1146 67F4 CE FF FF      M100K JSB =ERROR      report the error
1147 67F7 39      BYT 57D      bad transmission
1148 67F8 6C 93      MULEM1 CLM R54      put a null string
1149 67FA DA E5      PUMD R54,=R12      on the stack
1150 67FC 9E      RTN      (to clean up stuff)
1151 67FD      ENDIF
1152 67FD 6E 24 A2      HEXAND STB R56,R44      save frame for END check
1153 6800 C9 A0 40      CMN R56,=ETO      end of transmission?
1154 6803 F6 1D      IFEQ      yes,
1155 6805 42 B0 9C 84      LDND R02,=SUMARM      disabled?
1156 6809 F7 0D      IFNZ      yes,
1157 680B 6F B0 68 84      LDND R57,=IMP3+4      get ready frame
1158 680F CE A0 6C      JSB =SNDROY      send it again
1159 6812 6E CF E0 FF      RMM R56,=OFFEOH      mark out garbage
1160 6816 F0 E5      JNP HEXAND      try some more
1161 6818      ENDIF
1162 6818 40 A8 FF      LDB R00,=FF      get term flag
1163 681B B2 9F 84      STBD R00,=TRKSZE      set it
1164 681E 44 A9 FF 68      GTO OKEND      no, finish up
1165 6822      ENDIF
1166 6822 C9 A0 41      CMN R56,=ETE      error end?
1167 6825 F7 C0      JEQ M100K      yes, give up
1168 6827 6F      DRP 57      set the drp
1169 6828 CE C5 68      JSB =TOOBIG      check overflow and save char
1170 682B 67 B0 8F 84      LDND R47,=HPHEAD      get IR+ size
1171 682F F7 25      IFNZ      set,
1172 6831 40 93      CLM R00      clear for load
1173 6833 B0 9A 84      LDND R00,=(WRIPRO+2D) get current loc
1174 6836 66 00 B4 C1      LDND R46,MOO,EOL      get char
1175 683A B2      CMN R46,R57      match?
1176 683D F6 12      IFEQ      yes,
1177 683F 40 88      ICB R00      inc loc
1178 6841 27 C0      CMN R00,R47      done?
1179 6843 FA 0A      IFY      yes,
1180 6845 A8 01      LDB R00,=1      get flag
1181 6847 40 B2 9F 84 M100K      STBD R00,=TRKSZE      set EOL term
1182 684A 44 A9 E4 68      GTO FIXEND      clean up
1183 684F      ENDIF

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RVY&101 12/13/1983 11:50 AM PG 29
*****
***** ENTIO& enter a string *****
1236 68B9 82 9A 84      STBD R00,=(WRIPRO+3D) save new loc
1237 68BC      ENDIF
1238 68BC CE 13 69      JSB =DIRASH      send frame back
1239 68BF F8 38      JEN M100K+      error, leave
1240 68C1 44 A9 FC 67      GTO HEXAND      go again
1241 68C5      ENDIF
1242 68C5 9A      TOOBIG SAD      save arp and drp
1243 68C6 65 B1 A5 84      LDND R45,=THEDIE      get size
1244 68CA F6 02      IFZR      0,
1245 68CC 9F      MODRIA PRD      clean stack
1246 68CD 9E      RTN      go on
1247 68CE      ELSE      no,
1248 68CE 7D 89      ICM R75      inc char count
1249 68D0 25 C1      CMN R75,R45      > or = size?
1250 68D2 FB 0A      IFNC      no,
1251 68D4 50 B0 9D B4      LDND R20,=FLHEAD      no data set?
1252 68D8 F6 F2      JNZ MODRIA      yes, go on
1253 68DA 9F      PRD      restore arp drp
1254 68DB 16 E4      PUMD R4,=R26      push an character
1255 68DD 9E      RTN      return for next
1256 68DE      ENDIF
1257 68DE      ENDIF
1258 68DE 9F      PRD      restore arp drp
1259 68DF 42 06 E3      PUMD R02,-R06      trash return address
1260 68E2 6F 16 E4      PUMD R57,=R26      push in the last char
1261 68E5 6E 06 E5      FIXEND PUMD R56,=R06      save incoming frame
1262 68E8 A9 A0 42      LDN R56,=NRD      send out NOT READY
1263 68EB CE FF FF      JSB =RDYSD      for DIRA
1264 68EE 6E 06 E3      PUMD R56,-R06      get back
1265 68F1 CE 13 69      JSB =DIRASH      send the frame
1266 68F4 F8 06      JEN M100K+      error, give up
1267 68F6 6E C9 A0 40      CMN R56,=ETO      end of transmission?
1268 68FA F7 04      IFNE      no,
1269 68FC 44 A9 F3 67 M100K+      GTO M100K      error out
1270 6900      ENDIF
1271 6900      ENDIF
1272 6900 42 B0 9D 84 OKEND      LDND R02,=FLHEAD      check no data
1273 6904 F7 02      IFNZ      yes,
1274 6906 7D 93      CLM R75      clear count
1275 6908      ENDIF
1276 6908 54 D0 A1      LDN R24,R75      get size
1277 690B A8 E5      PUMD R24,=R12      string len to stack
1278 690D 56 14 C5      SBH R26,R24      find string start
1279 6910 A8 E5      PUMD R26,=R12      string address to stack
1280 6912 9E      RTN      clean up
1281 6913      ENDIF
1282 6913 7D 06 E5      DIRASH PUMD R75,=R06      save counter
1283 6916 CE FF FF      JSB =M1JSB      do a call to
1284 6918 FF FF      DEF DR1SD-      send data around
1285 691B 7D 06 E3      PUMD R75,-R06      restore counter
1286 691E 9E      RTN
1287 691F      ENDIF
1288 691F 42 93      S1STND CLN R02      clear for store

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 30
-----
>>>>>> EMI01 enter a string <<<<<<<
1289 6921 B3 69 B4 SIND R02, *TMP3+5 clear IL and CL
1290 6924 B2 6C B4 SIND R02, *TMP3+8 clear IR flag
1291 6927 B3 8F B4 SIND R02, *TMPHEAD clear termflag
1292 6928 B3 99 B4 SIND R02, *(IRIPRO+20) clear termcount
1293 6929 B3 9B B4 SIND R02, *(URIPRO+40) clear EL, ET flag
1294 6930 B3 8D B4 SIND R02, *FLHEAD clear DM, IR flags
1295 6933 B3 87 B4 SIND R02, *(MEDIE+20) clear upper size
1296 6936 B2 9F B4 SIND R02, *IRASIZE clear term cause
1297 6939 B6 C3 POND R02, *R06 get return addr
1298 6938 40 89 4F 69 LDM R00, *CLSTMD load CLSTMD as last rtn
1299 693F E5 PUND R00, *R06 put on the stack
1300 6940 42 E5 PUND R00, *R06 put return back
1301 6942 B0 83 B3 LDBD R02, *STAND? get current standby
1302 6945 B2 68 B4 SIND R02, *TMP3+7 save it
1303 6948 43 88 FF LDB R03, *OFFH set standby on
1304 694B CE FF FF JSB *STAND? turn on chip
1305 694E 9E RIN
1306 694F
1307 694F 42 B0 B3 B3 LSTMD LDBD R02, *STAND? get current standby status
1308 6953 43 B0 68 B4 LDBD R03, *TMP3+7 get original standby status
1309 6957 CE FF FF JSB *STAND? and restore it
1310 695A 42 B0 69 B4 LDBD R02, *TMP3+5 UMI UML ???
1311 695E F7 0C IFWZ IL was signified
1312 6960 43 B0 B1 B2 LDBD R03, *PLSTAT get PLSTAT
1313 6964 CF 14 RND R03, *14H any DISP or PR devices
1314 6966 F7 E6 RZR nope, so return u/o UMI, UML
1315 6968 CE FF FF JSB *UMIREP else, untalk the talker
1316 696B 9E RIN
1317 696C
1318 696C CE FF FF LPKILL JSB *MELJSB nope leave loop intact
1319 696F FF FF DEF VBYE tell everybody
1320 6971 9E RIN to shut up
1321 6972
1322 6972 CE FF FF BINDGI JSB *ALFWM uppercase it
1323 6975 SE CC 30 SBB R36, *0 shift to 0
1324 6978 FA 10 JNG BRDDGI bad character
1325 697H C8 0A CNB R36, *-0 greater than 9?
1326 697C FA 06 IFVY yes,
1327 697E C8 11 CNB R36, *-0 less than A?
1328 6980 FA 08 JNL BRDDGI bad character
1329 6982 CC 07 SBB R36, *-A- shift to 10
1330 6984
1331 6984 14 C0 CNB R36, R24 greater than base?
1332 6986 FA 02 JCV BRDDGI bad character
1333 6988 9D CLE clear error flag
1334 6989 9E RIN
1335 698A
1336 698A CE FF FF BRDDGI JSBN *ERR1+ a bad value to
1337 698D 58 BYI 89D input
1338 698E
1339 698E 5A 92 GETWM CLD R32 clear counter
1340 6990 5C 16 A1 LDM R34, R26 get pointer
1341 6993 LOOP digit check

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 31
-----
>>>>>> EMI01 enter a string <<<<<<<
1342 6993 50 16 84 LDBD R20, R26 get char
1343 6996 CE FF FF JSB *DIG1? digit?
1344 6999 F9 0A JEZ DONUM no, go on
1345 699B 56 89 JCM R26 inc pointer
1346 699D 5A 88 JCB R32 inc count
1347 699F 52 8B DCM R22 dec length
1348 69A1 F4 0C JNG IMVNM overflow, error
1349 69A3 F6 EE WHNZ until done
1350 69A5 SE 1A C0 DONUM CNB R36, R32 too many digits?
1351 69A8 FA 05 JMC IMVNM yes, error
1352 69AA 58 93 CLM R30 clear for load
1353 69AC A0 LDB R30, R32 get count, check 0
1354 69AD F6 04 IFZR yes,
1355 69AF CE FF FF IMVNM JSBN *ERR1+ error out
1356 69B2 59 BYI 89D
1357 69B3
1358 69B3 5A 8B 0A LDB R24, *10D get base
1359 69B6 60 93 CLM R40 clear accum
1360 69B8 LOOP accum loop
1361 69B8 65 85 LLM R45
1362 69B8 22 A3 STM R45, R42 save
1363 69B8 85 LLM R45 4X
1364 69B8 85 LLM R45 8X
1365 69B8 C3 RDM R45, R42 10X
1366 69BF SE 93 CLM R36 clear for load
1367 69C1 1C E0 PDBD R36, *R34 get digit
1368 69C3 CE 72 69 JSB *BINDGI convert
1369 69C6 F8 11 REN error out
1370 69C8 65 1E C3 RDM R45, R36 add to accum
1371 69CB 5A 8A DCB R32 dec count
1372 69CD F6 E9 WHNZ until done
1373 69CF 9E RIN
1374 69D0

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 32
-----
>>>>>> command frame table <<<<<<<
1376 69D0 44 43 4C CMTBI ASC 'DCL' device clear (all dev's)
1377 69D3 14 BYI 0014H 20D
1378 69D4 45 44 4E ASC 'EDN' enable device not ready for dat
1379 69D7 0F BYI 000FH 150
1380 69D8 45 4C 4E ASC 'ELN' another name for EDN
1381 69D8 0F BYI 000FH 150
1382 69D8 49 46 43 ASC 'IFC' interface clear
1383 69D8 90 BYI 0090H 144D
1384 69E0 4E 4F 50 ASC 'NOP' non-operable command
1385 69E3 10 BYI 0010H 160
1386 69E4 52 45 4E ASC 'REN' remote enable
1387 69E7 92 BYI 0092H 146D
1388 69E8 53 44 43 ASC 'SDC' selected device clear
1389 69E8 04 BYI 0004H 40
1390 69EC 55 4E 4C ASC 'UML' unlisten
1391 69EF 3F BYI 003FH 63D
1392 69F0 55 4E 54 ASC 'UNT' untalk
1393 69F3 5F BYI 005FH 95D
1394 69F4 47 45 54 ASC 'GET' group execute trigger
1395 69F7 08 BYI 0008H 80
1396 69F8 47 54 4C ASC 'GIL' go to local
1397 69FB 01 BYI 0001H 10
1398 69FC 4C 4C 4F ASC 'LLD' local lockout
1399 69FF 11 BYI 0011H 17D
1400 6A00 4C 50 44 ASC 'LPD' loop power down
1401 6A03 9B BYI 009BH 155D
1402 6A04 41 4F 55 ASC 'RAU' auto address unconfigure
1403 6A07 9A BYI 009AH 154D
1404 6A08 4E 52 45 ASC 'NRE' not remote enable
1405 6A0B 93 BYI 0093H 147D
1406 6A0E 50 50 44 ASC 'PPD' parallel poll disable
1407 6A0F 05 BYI 0005H 50
1408 6A10 50 50 55 ASC 'PPU' parallel poll unconfigure
1409 6A13 15 BYI 0015H 21D
1410 6A14 BSZ 4D table end
1411 6A18
1412 6A18 44 44 4C ASC 'DDL' device dependent list
1413 6A1B 80 BYI 0080H 160D
1414 6A1C 44 44 54 ASC 'DDI' device dependant talk
1415 6A1F 0C BYI 000CH 192D
1416 6A20 50 50 45 ASC 'PPE' parallel pole enable
1417 6A23 80 BYI 0080H 128D
1418 6A24 53 41 44 ASC 'SAD' secondary address
1419 6A27 60 BYI 0060H 56D
1420 6A28 BSZ 4D table end
1421 6A2C
1422 6A2C 4C 41 44 ASC 'LAD' listener address
1423 6A2F 20 BYI 0020H 32D
1424 6A30 54 41 44 ASC 'TAD' talker address
1425 6A33 40 BYI 0040H 64D
1426 6A34 BSZ 4D table end
1427 6A38
1428 6A38 4E 52 44 ASC 'NRD' not ready for data

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 33
-----
>>>>>> command frame table <<<<<<<
1429 6A38 42 BYI 0042H 66D
1430 6A3C 53 44 41 ASC 'SDR' send data
1431 6A3F 60 BYI 0060H 96D
1432 6A40 53 53 54 ASC 'SSI' send status
1433 6A43 61 BYI 0061H 97D
1434 6A44 53 44 49 ASC 'SDI' send device id
1435 6A47 62 BYI 0062H 98D
1436 6A48 53 41 49 ASC 'SRI' send access id
1437 6A4B 63 BYI 0063H 99D
1438 6A4C 54 43 54 ASC 'ICI' take control
1439 6A4F 64 BYI 0064H 100D
1440 6A50 49 45 50 ASC 'IEP' illegal ext primary
1441 6A53 8F BYI 008FH 191D
1442 6A54 49 45 53 ASC 'IES' illegal ext secondary
1443 6A57 0F BYI 000FH 223D
1444 6A58 49 41 41 ASC 'IAR' illegal auto address
1445 6A5B 9F BYI 009FH 159D
1446 6A5C 49 4D 50 ASC 'IMP' illegal multiple primary
1447 6A5F FF BYI 00FFH 255D
1448 6A60 5A 45 53 ASC 'ZES' zero extended secondary
1449 6A63 C0 BYI 000CH 192D
1450 6A64 BSZ 4D table end
1451 6A68
1452 6A68 52 44 3A ASC 'RD:' ready class control bits
1453 6A6B A1 BYI 00A1H
1454 6A6C 43 44 3A ASC 'CD:' command class control bits
1455 6A6F 81 BYI 0081H
1456 6A70 49 44 3A ASC 'ID:' identify class control bits
1457 6A73 C1 BYI 00C1H
1458 6A74 44 41 3A ASC 'DR:' data class control bits
1459 6A77 01 BYI 0001H
1460 6A78 45 4E 3A ASC 'EW:' data end control bits
1461 6A7B 41 BYI 0041H
1462 6A7C 49 53 3A ASC 'IS:' identify SRQ control bits
1463 6A7F E1 BYI 00E1H
1464 6A80 44 53 3A ASC 'DS:' data SRQ control bits
1465 6A83 21 BYI 0021H
1466 6A84 45 53 3A ASC 'ES:' data end SRQ control bits
1467 6A87 61 BYI 0061H
1468 6A88 54 52 3A ASC 'TR:' terminator
1469 6A8B 00 BYI 0000H
1470 6A8C BSZ 4D end of table
1471 6A90
1472 6A90 54 4C 2B ASC 'TL:' no UML or UML
1473 6A93 01 BYI 0001H
1474 6A94 43 4C 2B ASC 'CL:' add CR or LF
1475 6A97 02 BYI 0002H
1476 6A98 BSZ 4D
1477 6A9C
1478 6A9C
1479 6A9C 53 5A 3D ASC 'SZ:' set buffer size
1480 6A9F 01 BYI 0001H
1481 6AA0 54 52 2A ASC 'TR:' terminate EOL

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 34
*****
>>>>>>> command frame table <<<<<<<<<
1482 6AB3 02 BYT 0002H terminate string
1483 6AB4 54 52 58 RSC 'TL'
1484 6AB7 03 BYT 0003H terminate on END
1485 6AB8 54 52 21 RSC 'TL'
1486 6AB8 04 BYT 0004H
1487 6AB8 44 41 20 RSC 'DA-' disable data load
1488 6AB8 05 BYT 0005H
1489 6AB8 45 54 20 RSC 'EI-' disable EIO terminate
1490 6AB3 06 BYT 0006H
1491 6AB4 45 4C 20 RSC 'EL-' insert EOL
1492 6AB7 07 BYT 0007H
1493 6AB8 BSZ 4D
1494 6AB8
1495 6AB8 41 41 44 RSC 'RAD' auto address
1496 6AB8 80 BYT 0008H 128D
1497 6AB8 41 45 50 RSC 'REP' auto ext primary
1498 6AB3 80 BYT 0008H 160D
1499 6AB4 41 45 53 RSC 'RES' auto ext secondary
1500 6AB7 00 BYT 0008H 192D
1501 6AB8 41 4D 50 RSC 'RNP' auto multiple primary
1502 6AB8 00 BYT 0008H 224D
1503 6AB8 BSZ 4D table end
1504 6AB8

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 36
*****
>>>>>>> device/command string process <<<<<<<<<
1552 6B33 7E 06 E3 STRIO POND R76,-R06 get our return addr
1553 6B36 7D 83 64 84 STRIO R75,-R06 save defacon, rtnaddr
1554 6B3A LOOP get address loop
1555 6B3A CE FF FF JSB =GETADR get address from stack
1556 6B3D 5A 0A E3 POND R32,-R12 get length
1557 6B40 64 1A A1 LDM R44,R32 load for store
1558 6B43 06 E5 POND R44,-R06 save on the stack
1559 6B45 50 8A DCB R20 dec counter
1560 6B47 F5 F1 MHP until done
1561 6B49 90 CLE set okay flag
1562 6B4A 7F 92 CLB R77 clear dev counter
1563 6B4C 56 DRP 26 get
1564 6B4D CE FF FF JSB =GETADR devlist addr
1565 6B50 5E 0A E3 POND R36,-R12 get the len
1566 6B53 7F 58 IFNZ if device specified,
1567 6B55 52 A9 60 84 LDM R22,-R06 get storage pointer
1568 6B59 LOOP parse device list
1569 6B59 5E 8B DCM R36 dec for :
1570 6B5B F7 49 JZR INVPD can't end with :
1571 6B5D F4 47 JNG INVPD can't end with ,
1572 6B5F 42 16 E0 POSD R02,+R26 get char
1573 6B62 C8 3A CNB R02,':' better be :
1574 6B64 F6 40 JNE INVPD no, give up
1575 6B66 59 A8 04 LDB R31,-4 get max len
1576 6B69 CE FF FF JSB =GETFST get name
1577 6B6C F8 38 JEM INVPD error, give up
1578 6B6E 50 90 TSB R20 end?
1579 6B70 F7 04 IFNZ no,
1580 6B72 C8 2C CNB R20,':' better be ,
1581 6B74 F6 30 JNZ INVPD no, error out
1582 6B76
1583 6B76 74 1C A5 ENDP get name
1584 6B79 7F 88 JCB R77 inc count
1585 6B7B 11 C0 CNB R77,R21 check dev count
1586 6B7D F8 27 JCY INVPD too many
1587 6B7F 5A A9 02 00 LDM R32,-R2 set minimum size
1588 6B83 76 C8 20 CNB R66,':' more?
1589 6B86 F7 08 IFNE yes,
1590 6B88 5A 89 ICM R32 inc char count
1591 6B8A 77 C8 20 CNB R67,':' more?
1592 6B8D F7 02 IFNE yes,
1593 6B8F 5A 89 ICM R32 inc char count
1594 6B91 ENDP
1595 6B91 F0 02 ELSE no,
1596 6B93 34 A1 LDM R66,R64 get name
1597 6B95 ENDP
1598 6B95 5E 06 E5 POND R36,-R06 save len
1599 6B98 50 E5 POND R20,-R06 and flags
1600 6B9A CE FF FF JSB =GETPAD get the pil address of dev
1601 6B9D 50 12 E4 POND R20,-R22 save address
1602 6B9D 06 E3 POND R20,-R06 get flags
1603 6B9E 5E E3 POND R36,-R06 and len
1604 6B9A F9 06 IFEN error,

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 35
*****
>>>>>>> SENDIO to pil device <<<<<<<<<
1506 6B8D A1 BYT 241
1507 6B8D 98 SMDIO, BLN interpreter leaves BCD
1508 6B8D CE FF 69 JSB =SISIND standby on set return CLSTND
1509 6B8D 50 A9 01 1F LDM R20,-R1,31D set flags
1510 6B8D 7D A8 20 LDB R75,-RAD default command is RAD
1511 6B8D CE 33 68 JSB =STRIO send CMD list
1512 6B8F 5C 93 CLM R34 clear for flag
1513 6B8E 83 9E R2 STRIO R34,-RSEADR set RAD loc flag
1514 6B8E 06 E3 POND R34,-R06 get DATA loc
1515 6B8E 1E R3 STM R34,R36 save pointer
1516 6B8E 5A 06 E3 POND R32,-R06 get DATA len
1517 6B8E F8 45 REM error, give up
1518 6B8D F7 43 IFNZ none there,
1519 6B8F LOOP send DATA loop
1520 6B8F 6F 1E 00 POND R57,-R34 get a data frame
1521 6B8F CE FF FF JSB =DATREP send it (report error)
1522 6B8F F8 38 REM error, give up
1523 6B8F 43 2E A0 LDB R03,R56 get status flags
1524 6B8F CF 04 ANN R03,-R4N check the FRAY bit
1525 6B8F F7 29 IFNZ if set, might be RAD
1526 6B8E 6E CF E0 FF ANN R56,-RFFEOH isolate control bits
1527 6B8D C9 A0 42 CNB R56,-RAD compare to RAD
1528 6B8D F7 04 IFNE that ain't it,
1529 6B8D CE FF FF JSB =ERR1+ report
1530 6B8D 39 BYT 57D bad transmission
1531 6B8D ENDP
1532 6B8D CE FF FF JSB =RDYSMD yes, send it back
1533 6B8E F8 22 REM bad trans
1534 6B8D 6E CF 80 FF ANN R56,-RFFBOH isolate to data bits
1535 6B8E 43 93 CLM R02 clear to receive data
1536 6B8E 43 1E E2 POND R03,-R34 get last data sent
1537 6B8E 42 2E C1 CNB R02,R56 did we get that back
1538 6B8E F6 E9 JNE WRDERR no, then report error
1539 6B8E 5E 1E E5 SBM R34,R36 calc len sent
1540 6B8E 89 ICM R32 bias by one
1541 6B8E 83 9E R2 STRIO R34,-RSEADR save how many sent
1542 6B8D F0 04 JMP SMDIO send the EIO
1543 6B8D ENDP
1544 6B8E 5A 8B DCM R32 any more frames
1545 6B8D F6 E4 UNH until done
1546 6B8D 6E A9 A0 40 SMDIO LDM R56,-EIO load up EIO frame
1547 6B8E CE FF FF JSB =RDYSMD send the EIO
1548 6B8D ENDP
1549 6B8E 9E REM
1550 6B8D

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 37
*****
>>>>>>> device/command string process <<<<<<<<<
1605 6B8E CE FF FF INVPD JSB =ERR1 report error
1606 6B89 3F BYT 63D invalid filespec
1607 6B8A F0 04 JMP PDDN get out
1608 6B8C ENDP
1609 6B8C 50 90 TSB R20 done?
1610 6B8E F6 A9 UNH go on
1611 6B8D ENDP
1612 6B8D 56 06 E3 PDDN POND R26,-R06 get CMD loc
1613 6B8D 52 E3 POND R22,-R06 get the len
1614 6B8E F8 78 JEM SIOEND error, get out
1615 6B8E F6 49 IFZR no CMD string,
1616 6B89 6F 80 64 84 LDB R57,-R06 load default command
1617 6B8D 7F 90 TSB R77 test the device count
1618 6B8F F6 08 IFZR no dev also,
1619 6B8C 6F C8 20 CNB R57,-RAD coming from SENDIO
1620 6B8C F7 04 IFNE no,
1621 6B8C CE FF FF JSB =ERR1 this is an error
1622 6B8C 3F BYT 63D bad device spec
1623 6B8C ENDP
1624 6B8C F0 34 ELSE device,
1625 6B8C 6F C8 40 CNB R57,-RAD a RAD?
1626 6B8C F6 18 IFEQ yes,
1627 6B8D 7F 8A DCB R77 more than one device?
1628 6B8D F7 06 IFNZ yes,
1629 6B8D CE FF FF JSB =ERR1 error
1630 6B8D 3F BYT 63D
1631 6B8D F0 57 JMP SIOEND give up
1632 6B8D ENDP
1633 6B8D 5E 80 60 84 LDB R36,-R06 get device N
1634 6B8F CE 61 6D JSB =SMDIO send default command
1635 6B8E F8 4E JEM SIOEND error, leave
1636 6B8E 6F 80 60 84 LDB R57,-RAD do an SDR
1637 6B8E CE A0 6C JSB =SMDIO send the SDR
1638 6B8E F0 14 ELSE no,
1639 6B8E 52 A9 60 84 LDM R22,-R06 get dev list
1640 6B8D LOOP lad devices
1641 6B8D 5E 12 E0 POND R36,-R22 get dev addr
1642 6B8F 6F 80 64 84 LDB R57,-R06 get command
1643 6B8F CE 61 6D JSB =SMDIO send it
1644 6B8E F8 36 JEM SIOEND error out
1645 6B8E 7F 8A DCB R77 dec dev count
1646 6B8E F6 F0 UNH until done
1647 6B8D ENDP
1648 6B8D ENDP
1649 6B8D F0 30 ELSE send CMD string,
1650 6B8D 7F E4 POND R77,-R06 save the dev count
1651 6B8D LOOP send CMD loop
1652 6B8D 65 16 E1 POND R45,-R26 get user CMD
1653 6B8D 52 C0 03 00 SBM R22,-R3 adjust string len
1654 6B8D 40 A9 25 F0 LDM R00,-R45,375 UPAC R45, 3 bytes
1655 6B8D CE FF FF JSB =UPAC cap command
1656 6B8E CE 46 6C JSB =ENDSEH locate and send CMD
1657 6B8E F8 15 JEM GOOM error!

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RVY101 12/13/1983 11:50 AM PG 38
>>>>>>> Device/command string process <<<<<<<<
1658 6C17 52 91 JSR R22 any left
1659 6C19 F7 11 JSR G00M nothing left
1660 6C18 50 16 E0 POBD R20,*R26 get next char
1661 6C1E C8 2E CMB R20,*" should be a "."
1662 6C20 F6 06 JMC PCERRR nope, then error
1663 6C22 52 88 DCM R22 decrement count
1664 6C24 F7 02 JSR PCERRR if trailing,
1665 6C26 F5 DC WHPS while some left
1666 6C28 CE FF FF PCERRR JSB =ERR1 report error
1667 6C2B 59 BVI B9D bad pil command
1668 6C2C 7F 06 C2 G00M POBD R77,*R06 restore dev count
1669 6C2F CE FF FF JSB =PILREP shat was the problem
1670 6C32 EMOIF
1671 6C32 42 B1 65 84 SIOEND LDND R02,*IMP3+1 get return address
1672 6C36 06 E5 PUND R02,*R06 put on stack
1673 6C3B 9E RTH
1674 6C39

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RVY101 12/13/1983 11:50 AM PG 40
>>>>>>> CMDSCH search for command string <<<<<<<<
1686 6C46 58 A9 DO 69 CMDSCH LDM R30,*CMOTBI load rpl value for table
1687 6C4A CE 39 6C JSB =SEARCH table 1
1688 6C4D F9 04 IFEN found, go do it
1689 6C4F 44 A9 63 6D GTO SMOCDM
1690 6C53 EMOIF
1691 6C53 CE 39 6C JSB =SEARCH table 2
1692 6C56 F9 04 IFEN found
1693 6C58 44 A9 58 6D GTO CETHUM go get number
1694 6C5C EMOIF
1695 6C5C CE 39 6C JSB =SEARCH table 3
1696 6C5F F9 04 IFEN found,
1697 6C61 44 A9 1B 6D GTO SMOIAD go do it
1698 6C65 EMOIF
1699 6C65 CE 39 6C JSB =SEARCH table 4
1700 6C68 F8 36 JEN SMDRDY found, go do it
1701 6C6A CE 39 6C JSB =SEARCH table 5
1702 6C6D F9 04 IFEN found,
1703 6C6F 44 A9 71 6D GTO GENPIL go do it
1704 6C73 EMOIF
1705 6C73 CE 39 6C JSB =SEARCH table 6
1706 6C76 F9 04 IFEN found,
1707 6C78 44 A9 0C 6D GTO IL+ go do it
1708 6C7C EMOIF
1709 6C7C CE 39 6C JSB =SEARCH table 7
1710 6C7F F8 25 JEN SZ,ETC found SZ=,IR*,IR[
1711 6C81 CE 39 6C JSB =SEARCH table 8
1712 6C84 F8 04 IFEZ not found,
1713 6C86 CE FF FF JSBH =ERR1+ error out
1714 6C89 59 BVI B9D
1715 6C8A EMOIF
1716 6C8A CE A8 6D JSB =NUMBER
1717 6C8D F8 B6 REN bad number, so error
1718 6C8F 6F 1E C2 ROB R57,R36 add offset start
1719 6C92 CE A0 6C JSB =SMDRDY send the RAGroup frame
1720 6C95 F8 04 IFEZ no response
1721 6C97 6E B1 67 84 LDND R56,*IMP3+3 get frame sent
1722 6C9B EMOIF (cover for the pil chip)
1723 6C9B 9B DCE was FRMS set
1724 6C9C F9 A7 REZ FRMS was set so rtn, okay
1725 6C9E 9C ICE error, recover E
1726 6C9F 9E RTH
1727 6CA0 SMDRDY LDB R56,*OOROH set up RDY frame
1728 6CA2 6E A8 A0 SIND R56,*IMP3+3 save the command
1729 6CA3 B3 67 84 JSB =RDYSHD send RDY frame
1730 6CA6 CE FF FF RTH since FRMS will be set
1731 6CA9 9E SZ,ETC DCB R57 SZ=?
1732 6CAB IFZR yes,
1733 6CAB 6F 8A LDB R36,*70 get max digits
1734 6CAC F6 0C JSB =GETINA get the size
1735 6CAE 5E A8 07 SIND R45,*IMEDIE save for later
1736 6CB1 CE 8E 69 JMP BACK3- finish up
1737 6CB4 65 B3 A5 B4
1738 6CBB F0 5D

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RVY101 12/13/1983 11:50 AM PG 39
>>>>>>> search through pil command tables <<<<<<<<
1676 6C39 90 SEARCH CLE set failure
1677 6C3A LOOP command search loop
1678 6C3A 6E 18 E1 POBD R54,*R30 get frame from table
1679 6C3B F7 F9 RZR end of table
1680 6C3F 65 2C C1 CMB R45,R54 compare to input
1681 6C42 F6 F6 WHNE while not found
1682 6C44 9C ICE set success flag
1683 6C45 9E RTH
1684 6C46

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RVY101 12/13/1983 11:50 AM PG 41
>>>>>>> CMDSCH search for command string <<<<<<<<
1739 6CBA EMOIF
1740 6CBA 8A DCB R57 TR=?
1741 6CBB F6 09 IFZR yes,
1742 6CBB 42 B0 C4 82 LDND R02,*EOLLEN get EOL length
1743 6CC1 B2 8F B4 STBD R02,*HPHEAD set flag
1744 6CC4 F0 50 JMP BACK3 finish up
1745 6CC6 EMOIF
1746 6CC6 8A DCB R57 TR=?
1747 6CC7 F6 24 IFZR yes,
1748 6CC9 40 A9 91 B4 LDND R00,*HPHEAD+2 get save pointer
1749 6CCD LOOP save loop
1750 6CCD 42 16 E0 POBD R02,*R26 get a char
1751 6CD0 52 88 DCM R22 dec string len
1752 6CD2 F4 15 JMC INVPC+ neg. err out
1753 6CD4 42 C8 5D CMB R02,*"]" end of paran?
1754 6CD7 F6 08 JFEO yes,
1755 6CD9 6F B2 90 B4 STBD R57,*HPHEAD+1 save size
1756 6CD0 F7 0A JSR INVPC+ must have something
1757 6CDF F0 35 JMP BACK3 finish up
1758 6CE1 EMOIF
1759 6CE1 00 E4 PUBD R02,*R00 save char
1760 6CE3 6F 88 ICB R57 anc size
1761 6CE5 C9 07 CMM R57,*#7 size too large?
1762 6CE7 FA E4 WHNE until is
1763 6CE9 CE FF FF INVPC+ JSBH =ERR1+ error out
1764 6CEC 59 BVI B9D
1765 6CED EMOIF
1766 6CED 8A DCB R57 TR=?
1767 6CEE F6 06 IFZR yes,
1768 6CF0 88 ICB R57 set flag
1769 6CF1 B2 9E B4 STBD R57,*FLHEAD+1D for later
1770 6CF4 F0 20 JMP BACK3 finish up
1771 6CF6 EMOIF
1772 6CF6 8A DCB R57 DR=?
1773 6CF7 F6 06 IFZR yes,
1774 6CF9 88 ICB R57 set flag
1775 6CFA B2 9D B4 STBD R57,*FLHEAD for later
1776 6CFD F0 17 JMP BACK3 finish up
1777 6CFF EMOIF
1778 6CFF 8A DCB R57 E1=?
1779 6D00 F6 06 IFZR yes,
1780 6D02 88 ICB R57 set flag
1781 6D03 B2 9C B4 STBD R57,*SUBFRM for later
1782 6D06 F0 0E JMP BACK3 finish up
1783 6D08 EMOIF
1784 6D08 B2 9B B4 STBD R57,*WRIPRO+4D must be E1+, set flag
1785 6D0B F0 09 JMP BACK3 finish up
1786 6D0D EMOIF
1787 6D0D 6F 2E A2 TL+ STB R57,R56 save the value
1788 6D10 92 CLB R57 clear for add
1789 6D11 6E CB 68 B4 ROM R56,*IMP3+4 if 1 (TL+) store in (IMP3+5)
1790 6D15 " " if 2 (EL+) store in (IMP3+6)
1791 6D15 A6 STBD R56,R56 store approp.

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 42
-----
>>>>>> CMDSCH search for command string <<<<<<<
1792 6D16 9D BACK3 CLE clear errors
1793 6D17 6E B1 67 84 BACK3- LDND R56,*TMP3+3 get last valid frame
1794 6D18 9F RTN
1795 6D1C
1796 6D1E 50 16 F0 SMD1RD PDBD R20,*R26 get next char
1797 6D1F CE 23 CMB R20,*" " is this LADM or TADM
1798 6D21 F6 37 IFED if so.
1799 6D23 62 88 DCM R22 dec string count
1800 6D25 F4 47 JNG INVPCH off end
1801 6D27 50 06 E3 PDBD R20,*R06 get rtn addr
1802 6D28 7F E2 PDBD R27,*R06 get device count
1803 6D2C E4 PDBD R27,*R06 get device count back
1804 6D2D 50 E5 PDBD R20,*R06 put rtn addr back
1805 6D2F 7F 90 TSD R27 test for no dev
1806 6D31 F6 04 IFZR error TADM or LADM
1807 6D33 CE FF FF JSBN =ERR1+ with no dev
1808 6D36 3F BYT 63D bad device spec
1809 6D37
1810 6D37 50 A9 6D 84 LDND R20,*TMP3+9 get pointer
1811 6D38 7F 8A DCM R27 pre dec count
1812 6D39 F7 05 IFNZ more than one dev.
1813 6D3F 6F CE 40 CMB R57,*1RD doing TADM?
1814 6D42 F7 EF JCO BADIOM can't do that
1815 6D44 ENDF
1816 6D44 LOOP send loop
1817 6D44 5E 10 E0 PDBD R36,*R20 get dev addr
1818 6D47 7F 8A DCM R27 dec count
1819 6D49 F4 16 JNG SMD1RD set LND frame
1820 6D4B 6F 06 E4 PDBD R57,*R06 save frame
1821 6D4E CE 61 6D JSB =SMD1RD send it
1822 6D51 6F 06 E2 PDBD R57,*R06 restore frame
1823 6D54 F8 C5 REN error out
1824 6D56 F0 EC UNMP until done
1825 6D58 F0 07 ELSE else,
1826 6D5A 56 8B DCM R26 back to the number
1827 6D5C CE A8 6D GETNUM JSB =NUMBER get the number
1828 6D5F F8 8A REN bad number, so error
1829 6D61 ENDF
1830 6D61 6F 1E C2 SMD1RD RDB R57,R36 add to the frame
1831 6D64 6E A8 80 SMDLND LDB R56,*R0H set LND frame
1832 6D67 B3 67 84 STND R56,*TMP3+3 save command
1833 6D6A CE FF FF JSB =LNDREP send LND frame
1834 6D6D 9E RTN
1835 6D6E
1836 6D6E CE FF FF INVPCH JSBN =ERR1+ report error
1837 6D71 59 BYT 89D bad command
1838 6D72
1839 6D72 50 16 E1 GENPIL POND R20,*R26 get hex value for frame
1840 6D75 52 8B DCM R22 decr cmd count by two
1841 6D77 8B DCM R22
1842 6D78 F4 F4 JNG INVPCH neg. error bad pal cmd
1843 6D7A CE 8B 6D JSB =HEXBIN convert to binary
1844 6D7D F8 EE REN rtn if error

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 44
-----
>>>>>> locate numeric suffixes to PIL commands <<<<<<<
1872 6D88 5E A8 02 NUMBER LDB R36,*2D set max digits
1873 6D8B CE 8E 69 JSB =GETNM go get number
1874 6D8E 65 19 A2 STB R45,R31 move for output
1875 6D91 C9 1F 00 00 CMM R45,*31D,0,0 number too big?
1876 6D95 FA F0 RNC no, go on
1877 6D97 CE FF FF JSBN =ERR1+ error out
1878 6D9A 59 BYT 89D
1879 6D9B
1880 6D9B
1881 6D9B 54 A8 10 HEXBIN LDB R24,*16D base is 16
1882 6D9E 5E 10 A0 LDB R36,R20 get first digit
1883 6D9F CE 72 69 JSB =BINDDGT convert to binary
1884 6D9F F8 E1 REN
1885 6D9F 5E 84 LLB R36 * by 2
1886 6D9F 84 LLB R36 * by 4
1887 6D9F 84 LLB R36 * by 8
1888 6D9F 84 LLB R36 * by 16
1889 6D9B 10 A2 STB R36,R20 digit base 16
1890 6D9D 11 A0 LDB R36,R21 get 2nd digit
1891 6D9F CE 72 69 JSB =BINDDGT convert to binary
1892 6D9D 50 1E C2 RDB R20,R36 add in to first
1893 6D95 9E RTN
1894 6D96
1895 6D96 00 20 BYT 0,55
1896 6D9B 5E B1 9E B2 SEND? LDND R36,*BSEADR retrieve # sent from SENDIO
1897 6D9C CE FF FF JSB =PUIMIG convert binary to real
1898 6D9F 9E RTN
1899 6D9D

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 43
-----
>>>>>> CMDSCH search for command string <<<<<<<
1845 6D7F 6F 2E A2 SIB R57,R56 save in r56
1846 6D82 F6 06 IFZR is this a terminator
1847 6D84 50 B2 6E 84 STBD R20,*TMP3+8 store terminator
1848 6D88 F0 8C JMP BACK3 jump out
1849 6D8A ELSE no,
1850 6D8A 6E 8A DCM R56 remove flag bit
1851 6D8C 03 A2 SIB R56,R03 save in r3
1852 6D8E 6F 10 A0 LDB R57,R20 get value in r57
1853 6D91 43 CF C0 SMDGEN RMM R03,*R00COH and out garbage
1854 6D94 C8 80 CMB R03,*R0H compare to cmd frame
1855 6D96 F6 05 IFED
1856 6D98 6D A8 50 LDB R55,*50H set LR
1857 6D9B F0 03 ELSE
1858 6D9D 6D A8 60 LDB R55,*60H set TR
1859 6D9D ENDF
1860 6D9D 6E B3 67 84 STND R56,*TMP3+3 save frame
1861 6D9A
1862 6D9A *****
1863 6D9A ** let em do it, if they want to that bad A
1864 6D9A * CMM R56,*ERR here we trap ERR A
1865 6D9A * REQ don't send A
1866 6D9A *****
1867 6D9A CE FF FF JSB =SMDFRM send frame
1868 6D97 ENDF
1869 6D97 9E RTN
1870 6D9B

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 45
-----
>>>>>> Auto assign loop <<<<<<<
1901 6E00 A1 BYT 241
1902 6E01 98 ASHLP. BIN for sure
1903 6E02 4C 92 CLB R14 clear function flag
1904 6E04 F0 06 JMP DORTLP go do it
1905 6E06
1906 6E06 00 2E BYT 0,56
1907 6E08 98 ASLPB. BIN for sure
1908 6E09 4C A8 01 LDB R14,*1 set function flag
1909 6E0C
1910 6E0C 60 B1 FF FF DORTLP LDND R40,*TAB devfile name
1911 6E0F CE FF FF JSB =SYSJSB go purge
1912 6E13 FF FF DEF FPURGE old devfile
1913 6E15 CE FF FF JSB =CLROFF clear offio
1914 6E18 CE FF FF JSB =LPDFF set testloop bit
1915 6E1B 50 92 CLB R20 clear for
1916 6E1D B2 BC 82 STBD R20,*DEVCMT DEVMET
1917 6E00
1918 6E00 *****
1919 6E00 * Send up to 32 IDYs to wake up Wombat and A
1920 6E00 * any devices on standby A
1921 6E00 *****
1922 6E00 A8 20 LDB R20,*32D R20=counter
1923 6E02 LOOP IDY loop
1924 6E02 CE FF FF JSB =UNLSND send UNL
1925 6E05 F9 25 JEZ GOTIT all awake now
1926 6E07 50 8A DCM R20 decr counter
1927 6E09 F6 F7 WHNZ until done
1928 6E0B CE FF FF JSB =SETOFF clear up
1929 6E0E CE FF FF CMARG. JSB =CHANGE tell of change
1930 6E11 CE FF FF JSB =CLRERR clear errors
1931 6E14 CE A9 79 ASLPDN JSB =KBDOFF undo keyboard
1932 6E17 90 9C STE set fail flag
1933 6E19 4C 90 ISB R14 doing function?
1934 6E1B F7 04 IFNZ yes,
1935 6E1D 6C 93 CLM R54 get null &
1936 6E1F 0A E5 PUND R54,*R12 put out
1937 6E21 ENDF
1938 6E21 9E RTN
1939 6E22
1940 6E22 CE FF FF CLEAR* JSB =SYSJSB purge the
1941 6E25 FF FF DEF FPURGE dev file
1942 6E27 CE FF FF CLEN- JSB =CLEARN- clean up
1943 6E2A F0 E8 JMP ASLPDN finish up
1944 6E2C
1945 6E2C 60 93 GOTIT CLM R40 clear for store
1946 6E2E B2 64 84 STBD R40,*TMPNM3 clear dev count
1947 6E31 B3 15 82 STND R40,*{TMPNM2+8D} clear 8 bytes of TMPNM2
1948 6E34 B3 1D 82 STND R40,*{TMPNM2+16D} clear next 8 bytes
1949 6E37 B1 FF FF LDND R40,*1AB get devfile name
1950 6E3A

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 46
=====
>>>>>>> Auto assign loop <<<<<<<<<
1951 6E3A CE FF FF JSB =MELJSB do auto address
1952 6E3D FF FF DEF UNKEXP
1953 6E3F F7 CD JZR CHANG+ failed, bail out
1954 6E41 F8 DF JEN CLEH+ loop error
1955 6E43 42 B1 BC B2 LDND R02,-DEVCHT get number of devices
1956 6E47 A clear PLSTAT and PLSTAT
1957 6E47 CF FF E0 RMI R02,-BCOFFH clear PLSTAT
1958 6E4A B3 BC *2 STMD R02,-DEVCHT put them back
1959 6E4D 2B A2 SIB R02,R51 save DEVCHT in R51
1960 6E4F F7 DD JZR CHANG+ no devices, to clean up
1961 6E51 CE FF FF JSB =SYSJSB go make
1962 6E54 FF FF DEF NAME11 a new devfile
1963 6E56 F8 EF JEN CLEH+ error out
1964 6E58 7E 93 CLM R76 BCD line 0
1965 6E5A 68 92 CLM R50 initialize device counter
1966 6E5C 6A 92 CLM R52 set HP-IB flag to false
1967 6E5E
1968 6E5E 68 88 ERCHND ICB R50 increment device addr
1969 6E60 6A 90 STB R52 test HP-IB flag
1970 6E62 F7 04 IFM2 if HP-IB device
1971 6E64 11 A2 STB R52,R21 fake accessory ID for
1972 6E66 A HP-IB device
1973 6E66 F0 44 ELSE if not HP-IB device
1974 6E68 6F A8 40 LDB R57,-40H construct
1975 6E6B 28 E2 RDB R57,R50 IAD1 in R57
1976 6E6D CE FF FF JSB =CHDREP send command
1977 6E70 F8 00 JEN CLEH+ error, clean up and exit
1978 6E72 6E A9 63 LDN R56,-SRI load SRI
1979 6E76 CE FF FF JSB =RDYSND send it
1980 6E79 6E CF E0 FF RMI R56,-BFFEON clear garbage
1981 6E7D F9 05 IFEM error,
1982 6E7F CE FF FF JSB =PILREP report it
1983 6E82 F0 9E JHP CLEH+ clean up
1984 6E84
1985 6E84 C9 40 41 CMN R56,-ETE error?
1986 6E87 F7 1D JEQ BADSRI yes, leave
1987 6E89 C9 40 40 CMN R56,-E1D no response?
1988 6E8C F7 03 IFNE no,
1989 6E8E C9 40 63 CMN R56,-SRI another form?
1990 6E91
1991 6E91 F6 05 IFED device did not respond
1992 6E93 51 A8 50 LDB R21,-50H fudge acc ID for J468A
1993 6E96 F0 14 ELSE responded
1994 6E98 51 2F 80 LDB R21,R57 use this ID
1995 6E9B CE FF FF JSB =MELJSB send it
1996 6E9E FF FF DEF DATSD- back
1997 6E9F 6E C9 40 41 CMN R56,-ETE error?
1998 6E9A F6 04 IFED yes,
1999 6E9C CE FF FF BADSRI JSB =ERR1 report it
2000 6E99 39 BYT 57D
2001 6E9A
2002 6E9A F8 D6 CLEH+ JEN CLEH+ error, give up
2003 6E9C ENDOF

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 48
=====
>>>>>>> Auto assign loop <<<<<<<<<
2054 6E6B ENDOF
2055 6E6B 54 93 CLM R24 init PLSTAT and setup
2056 6E6D 52 CD 15 B2 SBN R22,-(INPM2+8D) restore class
2057 6E6F 76 B4 65 6F LDND R66,R22,MNEMON get the mnemonic
2058 6E75 83 0D B2 STND R66,-(INPM2+DNHME-3) put name away
2059 6E78
2060 6E78 C9 44 31 CMN R66,-'D1' is this 1st display device?
2061 6E7A F6 04 IFEQ yes,
2062 6E7D 54 A9 10 01 LDN R24,-20,1 get setup, PLSTAT
2063 6F01 ENDOF
2064 6F01 76 C9 50 31 CMN R66,-'P1' is this 1st print device?
2065 6F05 F6 04 IFEQ yes,
2066 6F07 54 A9 04 00 LDN R24,-4,200 get setup, PLSTAT
2067 6F08 ENDOF
2068 6F08 55 B2 0F 82 STBD R25,-(INPM2+SETUP-3) put setup byte away
2069 6F0F 54 D2 0D 82 ARDB R24,-PLSTAT adjust PLSTAT
2070 6F13 82 0D 82 STBD R24,-PLSTAT put new PLSTAT back
2071 6F16 5A B1 8F 82 LDND R32,-DEVLEN record length
2072 6F1A 5C A9 0D 82 LDN R34,-(INPM2 record addr
2073 6F1E 68 B2 10 82 STBD R50,-(INPM2+DADDR-3) put loop addr away
2074 6F22 CE FF FF JSB =SYSJSB replace the
2075 6F25 FF FF DEF FSREPL devfile line
2076 6F27 F8 81 JEN CLEH+ no room, give up
2077 6F29 76 C8 49 CMN R66,-'I' is this an interface?
2078 6F2C F6 11 IFEQ yes,
2079 6F2E 51 C8 03 CMN R21,-3 is it HP-IL to HP-IB?
2080 6F31 F6 0C IFED yes,
2081 6F33 56 A9 15 B2 LDN R26,-(INPM2+8D) get table address
2082 6F37 68 16 B6 0C STBD R50,R26,8000CH initialize count=loop addr
2083 6F3C 6A AB 4F LDB R52,-4FH set the HP-IB flag
2084 6F3F ENDOF
2085 6F3F
2086 6F3F 68 29 C0 CMN R50,R51 done?
2087 6F42 F8 04 IFNE not yet,
2088 6F44 44 A9 5D 6E G10 ERCHND start of loop
2089 6F48 ENDOF
2090 6F48 4C 90 JSB R14 doing function?
2091 6F4A F7 0E IFNZ yes,
2092 6F4C 42 93 CLM R02 clear for load
2093 6F4E B0 64 84 LDND R02,-INPM3 get device count
2094 6F51 0A E5 PUND R02,-R12 put out length
2095 6F53 5C A9 65 84 LDN R34,-(INP3+1 put out
2096 6F57 CE FF FF JSB =PUTREL address
2097 6F5A ENDOF
2098 6F5A CE FF FF JSB =DONE finish up
2099 6F5D CE A9 79 JSB =KBDOFF undo keyboard
2100 6F60 CE FF FF JSB =CLRERR clear errors
2101 6F63 9D CLE clear failed flag
2102 6F64 9E RIN
2103 6F65
2104 6F65
2104 6F65

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 47
=====
>>>>>>> Auto assign loop <<<<<<<<<
2004 6E9E ENDOF
2005 6E9E
2006 6E9E 42 93 CLM R02 clear for add
2007 6E9E B0 64 84 LDND R02,-INPM3 get dev count
2008 6EB1 88 ICB R02 inc count
2009 6EB2 B2 64 84 STBD R02,-INPM3 save new count
2010 6EB5 EB 64 84 RMI R02,-INPM3 point to loc for id
2011 6EB8 51 02 A6 STBD R21,R02 save id
2012 6EBB
2013 6EBB C8 90 CMN R21,-90H id >= 90H
2014 6EBD FA 00 IFEQ yes,
2015 6EBF C8 E0 CMN R21,-EOH id < EOH
2016 6EC1 FB 04 IFNE yes,
2017 6EC3 A8 90 LDB R21,-90H make it 90H
2018 6EC5 F0 02 ELSE no,
2019 6EC7 CC 40 SDB R21,-40H reduce for table
2020 6EC9 ENDOF
2021 6EC9
2022 6EC9 C8 4F CMN R21,-4FH if HP1B
2023 6ECB F6 02 IFEQ yes,
2024 6ED0 AB C0 LDB R21,-OCHH load for table
2025 6EEF ENDOF
2026 6EEF
2027 6EEF
2027 6EEF
2028 6EEF
2029 6EEF
2030 6EEF
2031 6EEF
2032 6EEF
2033 6EEF
2034 6EEF
2035 6EEF
2035 6EEF
2035 6EEF
2036 6EEF
2037 6EEF 99 BFD ready to shift 4 bits
2038 6EB0 80 EIB R21 shift class to e
2039 6EB1 52 93 CLM R22 clear for shift
2040 6EB3 80 EIB R22 shift class to 22
2041 6EB4 51 80 EIB R21 shift subclass to e
2042 6EB6 80 EIB R21 shift subclass back to 21
2043 6EB7 7E 89 ICM R76 increment line number
2044 6EB9 98 BTM back to binary
2045 6EBB
2046 6EBB 52 CB 15 B2 RMI R22,-(INPM2+8D) compute class offset
2047 6EBE 77 12 A4 LDND R67,R22 get count
2048 6EE1 89 ICM R67 inc count
2049 6EE2 A6 STBD R67,R22 save new count
2050 6EE3 CA 30 RDB R67,-30H make count an
2051 6EE5 C8 3A CMN R67,-JAH ASCII digit
2052 6EE7 FA 02 IFEQ or
2053 6EE9 CA 07 RDB R67,-7 letter

```

 * This logic assumes that the device either
 * doesn't respond at all to the SRI, or sends
 * a single byte of data and then an E10.
 * Devices which send more than one byte in
 * response to an SRI will be left waiting for
 * their second data byte to be sent back to them.
 * The next thing they'll see is a IAD1 to the
 * next device, or an UNL LMT from DONE.
 * *****

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 49
=====
>>>>>>> Auto assign loop <<<<<<<<<
2105 6F65 A Device mnemonic table A
2106 6F65 A The "B" in position 12 is the phony class A
2107 6F65 A for HP-IBs A
2107 6F65 A
2108 6F65 A
2108 6F65 A
2109 6F65 43 4D 50 44 MNEMON ASC "CMNDIEGAKUDXB"
2109 6F69 49 45 47 41
2109 6F6D 4B 55 4F 58
2109 6F71 42
2110 6F72

```

NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer
NOMAS

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RYX101 12/13/1983 11:50 AM PG 50
>>>>>>>
AUTODIOP ON/OFF
2112 6F72 A1 BYT 241
2113 6F73 98 BIN
2114 6F74 43 00 4A 84 R10TG. BIN for sure
2115 6F78 42 0A E2 LDBD R03,-R10TLP get current status
2116 6F7B F7 04 P0BD R02,-R12 get user input
2117 6F7D 43 92 IFNZ on,
2118 6F7F F0 03 CLB R03 set the flag
2119 6F81 43 A8 01 ELSE off,
2120 6F84 LDB R03,-1 clear the flag
2121 6F8A B2 4A 84 ENDDIF
2122 6F87 9E STBD R03,-R10TLP save the new status
2123 6F88 RTN

```

NOT Manufacturer Support
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RYX101 12/13/1983 11:50 AM PG 52
>>>>>>>
LOCAL, LOCAL LOCKOUT, ETC
2154 6F87 A1 BYT 241
2155 6F88 50 A9 7A 70 LOCLO. LDM R20,-LLOMSG get message
2156 6F8C 6C 93 CLM R54 put null
2157 6F8E 0A E5 PUMD R54,-R12 list
2158 6F90 F0 2D JMP DOWM10 finish up
2159 6F92 BYT 241
2160 6F93 50 A9 7E 70 LOCAL. LDM R20,-NREMSG get all message
2161 6F97 52 A9 82 70 LDM R22,-GILMSG get spec message
2162 6F9B F0 14 JMP WICH1? go process
2163 6F9D BYT 241
2164 6F9E LDM R20,-REHMSG get all message
2165 6F9F 50 A9 8F 70 RENOT. LDM R22,-RPHMSG get spec message
2166 6F9D 52 A9 97 70 LDM R22,-RPHMSG go process
2167 6F9D F0 09 JMP WICH1?
2168 6F9E BYT 241
2169 6F9F LDM R20,-GETHMSG get all message
2170 6F9D 50 A9 A4 70 TRIGR. LDM R22,-TRIGMSG get spec message
2171 6F9D 52 A9 A8 70
2172 6F9D F0 09
2173 6F9E WICH1? JSB -DWR12? any param?
2174 6F9E CE FF FF IFEN yes,
2175 6F9F F9 05 LDM R20,R22 switch to spec message
2176 6F9F 50 12 A1 ELSE no,
2177 6F9F F0 04 CLM R54 put null
2178 6F9F 6C 93 PUMD R54,-R12 list
2179 6F9D 0A E5 ENDDIF
2180 6F9F JSB -NOVMSG move the message
2181 6F9F CE 65 70 REN error out
2182 6F9F F8 B4 LDM R54,-"TL" get suppress command
2183 6F9F 6C A9 2C 54
2184 6F9F 4C 28 PUMD R54,-R22 add to message
2185 6F9F 93 CLM R54 get a null
2186 6F9D 0A E5 PUMD R54,-R12 data on stack
2187 6F9F CE D1 6A JSB -SNDIO. do sendio
2188 7002 9E RTN
2189 7003 BYT 30,55
2190 7003 18 2D SPOLL. JSB -SPOLL. get the status
2191 7005 CE 50 70 JMP DOWM1. make it number
2192 7008 F0 12
2193 700A BYT 0,55
2194 700A 00 2D PPOLL. LDM R20,-PPLMSG get message
2195 700E 50 A9 E0 70 DOW1M CLM R54 put null
2196 7010 6C 93 PUMD R54,-R12 list
2197 7012 0A E5 JSB -NOVMSG move message
2198 7014 CE 65 70 REN error out
2199 7017 F8 E9 JSB -NTIO$. do entio$
2200 7019 CE 65 67 DOWM1. REN error out
2201 701C F8 E4 JSB -NUM. convert to number
2202 701E CE FF FF RTN
2203 7021 9E
2204 7022 BYT 0,55
2205 7022 00 2D

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RYX101 12/13/1983 11:50 AM PG 51
>>>>>>>
WIO/RIO
2125 6F88 A1 BYT 241
2126 6F89 CE FF FF NID. JSB -DWEB get data
2127 6F8C DRP 176 save code
2128 6F8C 16 A3 STM A76,R26 move it
2129 6F8E CE A9 6F JSB -ERR? get and test pil reg N
2130 6F91 56 26 B6 10 STBD R26,K46,PILRO store data in reg
2131 6F95 FF RTN
2132 6F96 9E
2133 6F97 10 2D BYT 20,55
2134 6F99 CE A9 6F RIO. JSB -ERR? get and test pil reg N
2135 6F9C DRP 176 save code
2136 6F9C 93 CLM A76 clear for load
2137 6F9D 26 B4 10 FF LDBD A76,K46,PILRO get the reg contents
2138 6F9F 1E A3 STM A76,R36 move for PUINIG
2139 6FA1 CE FF FF JSB -SYSJSB put integer on stack
2140 6FA6 FF FF DEF PUINIG with system routine
2141 6FA8 9E RTN
2142 6FA9
2143 6FA9 CE FF FF ERR? JSB -DWEB get pil reg N
2144 6FAE F4 05 IFPS ok,
2145 6FAE DRP 176 save code
2146 6FAE C9 08 00 CLM A76,-88 test maximum
2147 6FB1 FA F5 RMC ok, return
2148 6FB3 ENDDIF
2149 6FB3 CE FF FF JSB -ERROR- report error
2150 6FB6 59 BYT 890 bad parameter
2151 6FB7
2152 6FB7

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RYX101 12/13/1983 11:50 AM PG 53
>>>>>>>
LOCAL, LOCAL LOCKOUT, ETC
2206 7024 50 A9 D7 70 ADORS. LDM R20,-RADMSG get message
2207 7028 CE 10 70 JSB -DOW1M go do it
2208 702B F8 F4 REN error out
2209 702D 99 BCD for number
2210 702F 78 93 CLM R70 build a
2211 7030 7D BF MCM R75 -1
2212 7032 7C A8 FF LDB R74,-FF integer
2213 7035 78 0A E5 PUMD R70,-R12 put on stack
2214 7038 CE FF FF JSB -ADORDI dec dev count
2215 703B 9E RTN
2216 703C BYT 30,56
2217 703C 18 2E LDM R45,-"SDI" get modifier
2218 7042 49
2219 7043 F0 10 JMP DOWM1? do it
2220 7045 BYT 30,56
2221 7045 18 2E LDM R45,-"SRI" get modifier
2222 7047 45 A9 53 41 DVA16. JMP DOWM1? do it
2223 704C F0 07 BYT 30,56
2224 704E LDM R45,-"SSI" get modifier
2225 704E 18 2E LDM R45,-"SSI"
2226 7050 65 A9 53 53 SPOLL. LDM R45,-"SSI"
2227 7054 54
2228 7055 50 A9 85 70 DOWM1? LDM R20,-RDYMSG get message
2229 7059 CE 65 70 JSB -NOVMSG move it
2230 705C F8 D0 REN error out
2231 705E 65 12 E7 PUMD R45,-R22 shove in modifier
2232 7061 CE 65 67 JSB -NTIO$. do entio$
2233 7064 9E RTN
2234 7065 BYT 30,56
2235 7066 6E 93 LDM R56 clear for load
2236 7068 10 A4 LDBD R56,R20 get size
2237 706A CE FF FF JSB -RSHEN- get ran
2238 706D F8 F5 REN no, leave
2239 706F 6E 0A E5 PUMD R56,-R12 size to stack
2240 7072 56 E5 PUMD R26,-R12 addr to stack
2241 7074 12 A3 STM R26,R22 move to pointer
2242 7076 CE FF FF JSB -RVBYIS move the string
2243 7079 9E RTN
2244 707A
2245 707A LLOMSG BYT 7D size
2246 707B 4C 4C 4F ASC "LLO" local lockout
2247 707E BYT 7D size
2248 707E ASC "NRE" local
2249 707E 07
2250 707F 4E 52 45
2251 7082 GILMSG BYT 16D size
2252 7082 10 ASC "UML,LDM,GIL" local 'device'
2253 7083 55 4E 4C 2C
2254 7087 4C 41 44 23
2255 7088 2C 47 54 4C

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 54
-----
>>>>>> LOCAL, LOCAL LOCKOUT, ETC <<<<<<<<
2254 708F REMMSG BYT 110 size
2255 708F 08 ASC 'UHL,REN' remote
2256 7090 55 4E 4C 2C
2256 7094 52 45 4E
2257 7097
2258 7097 10 RNOWSG BYT 16D size
2259 7098 55 4E 4C 2C ASC 'UHL,REN,LAOW' remote 'device'
2259 709C 52 45 4E 2C
2259 70A0 4C 41 44 23
2260 70A4
2261 70A4 07 GETMSG BYT 7D size
2262 70A5 47 45 54 ASC 'GET' trigger
2263 70A8
2264 70A8 10 TRGMSG BYT 16D size
2265 70A9 55 4E 4C 2C ASC 'UHL,LAOW,GET' trigger 'device'
2265 70AD 4E 41 44 23
2265 70B1 2C 47 45 54
2266 70B5
2267 70B5 12 RDNMSG BYT 18D size
2268 70B6 55 4E 4C 2C ASC 'UHL,SZ=32,TADM,' setup stuff
2268 70BA 53 5A 3D 33
2268 70BE 32 2C 54 41
2268 70C2 44 23 2C
2269 70C5
2270 70C5 11 KEYMSG BYT 17D size
2271 70C6 55 4E 4C 2C ASC 'UHL,SZ=1,TADM,SDA' read 1 char
2271 70CA 53 5A 3D 33
2271 70CE 2C 54 41 44
2271 70D2 23 2C 53 44
2271 70D6 41
2272 70D7
2273 70D7 08 RADMSG BYT 8D size
2274 70D8 41 41 55 2C ASC 'RAU,RAD1' address
2274 70DC 41 41 44 31
2275 70E0
2276 70E0 09 PPLMSG BYT 9D size
2277 70E1 49 44 3A 30 ASC '10:00,11,' ppoll
2277 70E5 30 2C 54 4C
2277 70E9 2B
2278 70EA
2279 70EA

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 56
-----
>>>>>> DEVADDR, DEVNAME< <<<<<<<
2333 7158 5E 06 E3 POMO R36,-R06 get pointer
2334 7158 50 93 CLM R20 clear for load
2335 715D LOOP scan loop
2336 715D 50 1E 84 06 LDBO R20,R36,DADDR get the entry addr
2336 7161 00
2337 7162 26 C1 CLM R20,R46 match?
2338 7164 F6 16 IFEO yes.
2339 7166 66 1E D5 03 LDMO R46,R36,DNAME get device name
2339 716A 00
2340 716B CE FF FF JSB =RESCOM get room
2341 716E 03 BYT 3D
2342 716F F8 D9 REN nope, leave
2343 7171 6E 0A E5 PUMD R56,+R12 length to stack
2344 7174 56 E5 PUMD R26,+R12 address to stack
2345 7176 65 A8 3A LDB R45,=' get color
2346 7179 16 A7 STMD R45,R26 move name
2347 717B 9E RTM
2348 717C ENOIF
2349 717C CE FF FF JSB =SKPCWK skip to next line
2350 717F F6 DC UNMZ until done
2351 7181 CE FF FF JSB =HANDI can anyone else help
2352 7184 2D VAL Y,ADDR with name
2353 7185 59 BYT BSD illegal filesep
2354 7186 9E RTM
2355 7187
2356 7187 A1 BYT 241
2357 7188 *
2358 7188 21 SEMD, BYT 41 the runtime
2359 7189 * for these
2360 7189 * tokens does
2361 7189 * nothing.
2362 7189 9C ENO, RTM Absolutely nothing.
2363 718A
2364 718A
2365 718A 24 BYT 44
2366 718B 42 A8 01 INOIA, LDB RO2,=1 get data flag
2367 718E F0 09 JMP SETIDE set it
2368 7190
2369 7190 24 BYT 44
2370 7191 42 92 INEND, CLB RO2 get end flag
2371 7193 F0 04 JMP SETIDE set it
2372 7195
2373 7195 24 BYT 44
2374 7196 42 A8 FF IMCHD, LDB RO2,=FF get end flag
2375 7199 42 A2 64 84 SETIDE STBO RO2,=INPMH3 set it
2376 719D 9E RTM
2377 719E
2378 719E 24 BYT 44
2379 719F CE FF FF INYHUM JSB =UNEB get number
2380 71A2 6F 26 A0 LDB R57,R46 do mod 256
2381 71A5
2382 71A5 42 B0 64 84 DOCEO LDBO RO2,=INPMH3 get mode flag
2383 71A9 F5 05 IFNG cnd,

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 55
-----
>>>>>> DEVADDR, DEVNAME< <<<<<<<
2281 70EA 18 2D BYT 30,55
2282 70EC 98 DEVAD, BIN for sure
2283 70ED CE FF FF JSB =NELJSB get the name
2284 70F0 FF FF DEF FLSTAK off stack
2285 70F2 F8 1B JEN INVAD error out
2286 70F4 5A A9 02 00 LDM R32,=#2 minimum size
2287 70F8 76 C8 20 ENB R66,=' more?
2288 70FB F7 0B IFNE yes.
2289 70FD 5A 89 ICM R32 inc size
2290 70FF 77 C8 20 ENB R67,=' more?
2291 7102 F7 02 IFNE yes.
2292 7104 5A 89 ICM R32 inc size
2293 7106 ENOIF
2294 7106 F0 02 ELSE no,
2295 7108 34 A1 LDM R66,R64 move name
2296 710A ENOIF
2297 710A CE FF FF JSB =GETPAD find device
2298 710D F9 04 IFEN error,
2299 710F CE FF FF INVAD JSB =ERR1+ report, leave
2300 7112 3F BYT 63D
2301 7113 ENOIF
2302 7113 5E 93 CLM R36 clear for load
2303 7115 10 A0 LDB R36,R20 get address
2304 7117 CE 06 67 JSB =HUMOUT make number, output
2305 711A 9E RTM
2306 711B
2307 711B 00 2E BYT 0,56
2308 711D CE B7 79 KEYBS, JSB =APROP+ open file
2309 7120 63 1E 05 12 LDMO R43,R36,B18 get the device name
2309 7124 00
2310 7125 40 A8 27 LDB R00,=47 set up top reg
2311 7128 6E A9 05 00 LDB R56,=45 get max length
2312 712C LOOP scan loop
2313 712E 41 C8 20 ENB R4,=' blank?
2314 712F F6 02 IFEO yes.
2315 7131 6E 8B DCM R56 dec length
2316 7133 ENOIF
2317 7133 40 8A DCM R00 dec pointer
2318 7135 E8 22 ENB R00,=42 done?
2319 7137 F6 F3 UNME no, go on
2320 7139 6E 0A E5 PUMD R56,+R12 save length
2321 713E A9 05 00 LDM R56,=45 get max back
2322 713F CE FF FF JSB =RSNEM- get memory
2323 7142 F8 D6 REN didn't, leave
2324 7144 56 0A E5 PUMD R26,+R12 push address
2325 7147 63 16 E5 PUMD R43,+R26 push data
2326 7149 9E RTM
2327 714B
2328 714B 10 2E BYT 20,56
2329 714D CE FF FF DVMMH, JSB =THRE+ get the devfile
2330 7150 F8 F6 REN error out
2331 7152 5E 06 E5 PUMD R36,-R06 save pointer
2332 7155 CE FF FF JSB =UNEB get devaddr

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 57
-----
>>>>>> DEVADDR, DEVNAME< <<<<<<<
2384 71AB CE 64 6D JSB =SMOEND send command
2385 71AF F0 0F ELSE data,
2386 71B0 F6 05 IFZR end,
2387 71B2 6E AB 40 LDB R56,=EWN get end bits
2388 71B5 F0 03 ELSE normal
2389 71B7 6E AB 00 LDB R56,=DAD get dab bits
2390 71BA ENOIF
2391 71BA 03 A2 STB R56,RO3 move for test
2392 71BC CE 91 6D JSB =SMOGEN send data
2393 71BF ENOIF
2394 71BF 9E RTM
2395 71C0
2396 71C0 24 BYT 44
2397 71C1 CE FF FF INVST, JSB =GETADR get string
2398 71C4 5A 0A E3 POMO R32,-R12 get length
2399 71C7 LOOP byte loop
2400 71C7 5A 8B DCM R32 dec length
2401 71C9 F4 F4 RRG done
2402 71CB 6F 1C E0 POMO R57,+R34 get byte
2403 71CE CE A5 71 JSB =DOCEO send it
2404 71D1 F0 F4 UNMP until done
2405 71D3
2406 71D3 21 BYT 41
2407 71D4 5C 93 EOL, CLM R34 clear for load
2408 71D6 B0 C4 82 LDBO R34,=EOLLEN get the length
2409 71D9 0A E5 PUMD R34,+R12 length to stack
2410 71DB A9 C1 82 LDM R34,=EOL get address
2411 71DE CE FF FF JSB =PUTREL relative to stack
2412 71E1 F0 DE JMP INVST send the string
2413 71E3
2414 71E3 21 BYT 41
2415 71E4 42 A8 A0 DDL, LDB RO2,=DDI get frame pattern
2416 71E7 F0 16 JMP DOCHD process params
2417 71E9
2418 71E9 21 BYT 41
2419 71EA 42 A8 C0 DDI, LDB RO2,=DDI get frame pattern
2420 71ED F0 10 JMP DOCHD process params
2421 71EF
2422 71EF 21 BYT 41
2423 71F0 42 A8 20 LISTH, LDB RO2,=LAD get frame pattern
2424 71F3 F0 0A JMP DOCHD process params
2425 71F5
2426 71F5 21 BYT 41
2427 71F6 42 A8 60 SAD, LDB RO2,=SAD get frame pattern
2428 71F9 F0 04 JMP DOCHD process params
2429 71FB
2430 71FB 21 BYT 41
2431 71FC 42 A8 40 TALK, LDB RO2,=IAD get frame pattern
2432 71FF
2433 71FF 42 B2 64 84 DOCHD STBO RO2,=INPMH3 save pattern
2434 7203 LOOP param loop
2435 7203 CE FF FF JSB =OMR12? any params left?
2436 7206 F9 B7 REZ no, done

```



```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RV&101 12/13/1983 11:50 AM PG 58
*****
>>>>>> DEVRDDR, DEVNAME<<<<<<
2437 7208 CE FF FF JSB =OWEB get the paran
2438 7208 66 CF 1F 00 AMB R46,=6001FH do mod 32
2439 720F 1E A2 STB R46,R36 move for send
2440 7211 6F B0 64 84 LDBD R57,=INPMH3 get that pattern
2441 7215 CE 61 60 JSB =SMDIDA build and send frame
2442 7218 FO E9 WIMP until done
2443 721A
2444 721A 21 BYT 41
2445 721B 42 A8 FO IDV. LDB R02,=IDYCB get control bits
2446 721E FO 04 JMP DOJDA process params
2447 7220
2448 7220 21 BYT 41
2449 7221 42 A8 RO RDY. LDB R02,=RDY get control bits
2450 7224
2451 7224 42 B2 64 84 DOJDA STB R02,=INPMH3 save control bits
2452 7228 LDBD R02,=OWEB any left?
2453 7228 CE FF FF JSB =OWEB no, done
2454 7228 F9 92 REZ JSB =OWEB get paran
2455 7220 CE FF FF LDB R46,R46 do mod 256
2456 7230 6F 26 A0 LDBD R56,=INPMH3 get control bits
2457 7233 6E 90 64 84 STB R56,R03 move for test
2458 7237 03 A2 JSB =SMDGEN send frame
2459 7239 CE 91 60 WIMP until done
2460 723C FO EA
2461 723E
2462 723E 21 BYT 41
2463 723F 6F A8 01 GTL. LDB R57,=GTL get the frame
2464 7242 FO 2E JMP DOFRME send it
2465 7244
2466 7244 21 BYT 41
2467 7245 6F A8 93 NRE. LDB R57,=NRE get the frame
2468 7248 FO 28 JMP DOFRME send it
2469 724A
2470 724A 21 BYT 41
2471 724B 6F A8 92 REM. LDB R57,=REM get the frame
2472 724E FO 22 JMP DOFRME send it
2473 7250
2474 7250 21 BYT 41
2475 7251 6F A8 11 LLO. LDB R57,=LLO get the frame
2476 7254 FO 1C JMP DOFRME send it
2477 7256
2478 7256 21 BYT 41
2479 7257 6F A8 90 IFC. LDB R57,=IFC get the frame
2480 725A FO 16 JMP DOFRME send it
2481 725C
2482 725C 21 BYT 41
2483 725D 6F A8 98 LPD. LDB R57,=LPD get the frame
2484 7260 FO 10 JMP DOFRME send it
2485 7262
2486 7262 21 BYT 41
2487 7263 6F A8 04 SDC. LDB R57,=SDC get the frame
2488 7266 FO 0A JMP DOFRME send it
2489 7268

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RV&101 12/13/1983 11:50 AM PG 60
*****
>>>>>> OUTPUT/OUTPUT USING<<<<<<
2502 7277 A1 BYT 241
2503 7278 CE FF FF OUTPT. JSB =OWR12? any params?
2504 7278 F9 1C IFEN yes,
2505 727D 42 E8 08 LMB R02,=BD number?
2506 7280 FA 08 IFCY yes,
2507 7282 CE 69 72 JSB =UML. unlisten and
2508 7285 CE 6F 72 JSB =UML. untalk loop
2509 7288 CE FO 71 JSB =LISTM. listen that addr
2510 728B FO 07 ELSE no,
2511 728D 50 A9 BA 72 LDM R20,=OUTMSG get listen msg
2512 7291 CE EF 6F JSB =DOSMID listen on
2513 7294
2514 7294 42 A8 80 LDB R02,=BOH set cleanup flag
2515 7297 FO 02 ELSE no,
2516 7299 42 92 CLB R02 clear cleanup flag
2517 729B ENDF
2518 729B CE 24 7B JSB =SETIDB save flag
2519 729E 4F 90 TSB R17 errors?
2520 72A0 F5 04 IFNC yes,
2521 72A2 44 A9 E6 76 GTO INVED4 clean up
2522 72A6 ENDF
2523 72A6 64 93 CLM R44 clear for fake
2524 72A8 83 86 83 STMD R44,=DFPAR1 printer data
2525 72AB 60 A8 03 LDB R40,=30 get our route
2526 72AE use DFPARs as fake ptrs
2527 72AE 6A A9 86 83 LDM R52,=DFPAR1
2528 72B2 88 83 DEF DFPAR2
2529 72B4 89 83 DEF (DFPAR2+1D)
2530 72B6 CE FF FF JSB =DOCOM setup for output
2531 72B9 9E RIM
2532 72BA
2533 72BA 10 OUTMSG BYT 16D length
2534 72BB 55 4E 4C 2C ASC 'UML,UNT,LADM' listen on
2535 72BF 55 4E 54 2C
2536 72C3 4C 41 44 23
2537 72C7

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RV&101 12/13/1983 11:50 AM PG 59
*****
>>>>>> DEVRDDR, DEVNAME<<<<<<
2490 7268 21 BYT 41
2491 7269 6F A8 3F UHL. LDB R57,=UHL get the frame
2492 726C FO 04 JMP DOFRME send it
2493 726E
2494 726E 21 BYT 41
2495 726F MTR. EXACTLY the same as UNT
2496 726F 6F A8 5F UNT. LDB R57,=UNT get the frame
2497 7272 98 DOFRME BIN for sure
2498 7273 CE 64 60 JSB =SMDCHD send it
2499 7276 9E RIM
2500 7277

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&101 OBJ=RV&101 12/13/1983 11:50 AM PG 61
*****
>>>>>> ENTER USING image parsing<<<<<<
2537 72C7
2538 72C7
2539 72C7
2540 72C7
2541 72C7
2542 72C7
2543 72C7
2544 72C7
2545 72C7
2546 72C7
2547 72C7
2548 72C7
2549 72C7
2550 72C7
2551 72C7
2552 72C7
2553 72C7
2554 72C7
2555 72C7
2556 72C7
2557 72C7
2558 72C7
2559 72C7
2560 72C7
2561 72C7
2562 72C7
2563 72C7
2564 72C7
2565 72C7
2566 72C7
2567 72C7
2568 72C7
2569 72C7
2570 72C7
2571 72C7
2572 72C7
2573 72C7
2574 72C7
2575 72C7
2576 72C7
2577 72C7
2578 72C7 40 1A A1 FLDEND LDM R00,R32 get IMAGE start
2579 72C8 18 C3 RDM R00,R30 add length
2580 72C8 52 10 A3 STB R22,R20 move field start
2581 72C8 43 00 A4 LDBD R03,R00 get end+1 char
2582 72C8 42 A8 29 LDB R02,=') get a '
2583 72C8 A6 STB R02,R00 put it out
2584 72C8 LOOP scan loop
2585 72C8 42 10 E0 FORB R02,=R20 get a char
2586 72C8 C8 2C LMB R02,=') is it a '

```

NOT Manufacturer Supported
 request agrees NOT to contact manufacturer
NOMAS

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYX101 OBJ=RYX101 12/13/1983 11:50 AM PG 62
***** ENTER USING image parsing *****
2587 72DB F7 13 IFNE CMB R02,")' no,
2588 72DB C8 29 CMB R02,")' as it a )?
2589 72DF F7 0F IFNE CMB R02,")' no,
2590 72E1 C8 28 CMB R02,")' as it a )?
2591 72E3 17 08 IFNE CMB R02,")' no,
2592 72E5 50 C9 01 E0 CMB R20,=E001H at end of ran?
2593 72E9 FA 05 IFY LDB R02,")' yes,
2594 72EB 42 AB 29 LDB R02,")' get a )
2595 72EE C8 29 CMB R02,")' set the flag
2596 72F0 ENDF
2597 72F0 ENDF
2598 72F0 ENDF
2599 72F0 ENDF
2600 72F0 F6 E4 UNHME until match
2601 72F2 42 CC 29 SDB R02,")' adjust end flag
2602 72F5 43 00 A6 SDB R01,R00 put back end flag char
2603 72F8 50 8B DCM R20 backup one
2604 72FA 12 C5 SDB R20,R22 compute length
2605 72FC F6 1D IFZR if null,
2606 72FE 52 8B DCM R22 back up one
2607 7300 40 E1 PDB R00,=R22 get 2 chars
2608 7302 52 8B DCM R22 restore pointer
2609 7304 1A C1 CMB R22,R32 at beginning?
2610 7306 F6 03 IFB LDB R00,")' yes,
2611 7308 40 AB 2C LDB R00,")' fake a leading .
2612 730B ENDF
2613 730B ENDF
2614 730F F7 08 CMB R00,")' null field?
2615 7311 C9 28 2C IFNE CMB R00,")' no,
2616 7314 F7 03 CMB R00,")' another form?
2617 7316 C9 2C 29 IFNE CMB R00,")' no,
2618 7319 ENDF
2619 7319 ENDF
2620 7319 F7 4D JEQ BRDING yes, error
2621 731B ENDF
2622 731B 9E RIM
2623 731C
2624 731C 98 SETUSE BIN for sure
2625 731D 78 B1 C5 83 LDB R20,=INSLEM get image stuff
2626 7321 7C 93 CLM R74 clear upper set
2627 7323 42 3A A1 LDB R02,R72 get address
2628 7326 C9 00 60 CMB R02,=96000H possibly in rom space?
2629 7329 FA 2C IFY yes,
2630 732B C9 00 80 CMB R02,=8000H is it really?
2631 732E F8 27 IFNE LDB R00,=F800H yes,
2632 7330 40 B1 51 82 LDB R00,R00,E,ROMW get the current
2633 7334 00 B5 0C 00 LDB R00,R00,E,ROMW basic rom W
2634 7338 LDB R60,= build a ROMJSB
2635 7338 70 A9 LDB R60,=
2636 733A CE FF FF JSB =ROMJSB
2637 733D 00 00 DEF (0)
2638 733F FF FF DEF CONER-
2639 7341 9E RIM

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYX101 OBJ=RYX101 12/13/1983 11:50 AM PG 64
***** ENTER USING image parsing *****
2693 73AE B2 D9 82 STBD R40,=USING? save it
2694 73B1 9E RIM
2695 73B2
2696 73B2 CE C7 72 CHKTRM JSB =FLDEND get first field
2697 73B5 F8 FA REM error, leave
2698 73B7 67 92 CLB R47 clear output
2699 73B9 42 8A DCM R02 check for ,
2700 73BB F4 F4 RMC no, no term
2701 73BD 50 06 E5 PDB R20,=R06 save length
2702 73C0 52 E5 PDB R22,=R06 and pointer
2703 73C2 LOOP scan loop
2704 73C2 66 12 E0 PDB R46,=R22 get char
2705 73C5 CE DF F3 JSB =FANCHR check it
2706 73C8 F9 0F IFNE not valid,
2707 73CA 9D CLE assume ok
2708 73CB 67 90 YSB R47 flags set?
2709 73CD F7 04 IFNZ yes,
2710 73CF CE FF FF JSB =ERR1 error out
2711 73D2 34 BYT 52D
2712 73D3 ENDF
2713 73D3 52 06 E3 TRDNE PDB R22,=R06 restore pointer
2714 73D6 50 E3 PDB R20,=R06 restore length
2715 73D8 9E RIM
2716 73D9 ENDF
2717 73D9 50 8B DCM R20 dec length
2718 73DB F6 E5 UNHME until done
2719 73DD F0 F4 JMP TRDNE clean up
2720 73DF
2721 73DF 9C TRNCHR ICE assume failure
2722 73E0 66 CC 20 SDB R46,")' make i=1, M=3, X=5
2723 73E3 F4 F3 RMC too small, leave
2724 73E5 F3 F1 REV not ours, leave
2725 73E7 C8 06 CMB R46,=6 too big?
2726 73E9 F8 ED REY yes, leave
2727 73EB 8A DCM R46 make i=0, M=2, X=4
2728 73EC F6 01 IFZR i,
2729 73EE 88 CLB R46 make it 1
2730 73EF ENDF
2731 73EF 67 26 94 ORB R47,R46 update flags
2732 73F2 9D CLE all ok
2733 73F3 9E RIM
2734 73F4
2735 73F4 CE IC 73 FJUSE JSB =SETUSE get started
2736 73F7 F8 FA REM error out
2737 73F9 CE B2 73 JSB =CHKTRM check for term
2738 73FC F8 F5 REM error out
2739 73FE 67 16 E4 PDB R47,=R26 set term byte
2740 7401 F7 11 IFNZ term found
2741 7403 52 10 C3 RDB R22,R20 skip field
2742 7406 89 ICM R22 skip,
2743 7407 42 B1 C5 83 LDB R02,=INSLEM get form length
2744 7408 C0 04 00 SDB R02,=84 reduce by one field
2745 740E B3 C5 83 STBD R02,=INSLEM save new length

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYX101 OBJ=RYX101 12/13/1983 11:50 AM PG 63
***** ENTER USING image parsing *****
2640 7342 40 33 A3 STN R00,R63 with run rom
2641 7345 70 B3 64 84 STN R60,=INPMH3 put it in rom
2642 7349 CE 64 84 JSB =INPMH3 move the string
2643 734C F8 F3 REN error out
2644 734E 7C 0A E3 PDB R74,=R12 get the location
2645 7351 38 A3 STN R74,R70 copy it
2646 7353 78 B3 C5 83 STBD R70,=INSLEM replace the original
2647 7357 ENDF
2648 7357 ENDF
2649 7357 64 B1 C5 83 LDB R44,=INSLEM get IMAGE len/addr
2650 735B 18 A3 STN R44,R30 move for later
2651 735D 26 A3 STN R44,R46 swap len
2652 735F 5A 24 A3 STN R32,R44 and addr
2653 7362 12 A3 STN R32,R22 set field addr
2654 7364 66 91 STN R46 len 0?
2655 7366 F6 04 IFZR yes,
2656 7368 CE FF FF BRDING JSB =ERR1+ error out
2657 736B 34 BYT 52D
2658 736C ENDF
2659 736C 42 93 CLM R02 clear accum
2660 736E LOOP count loop
2661 736E 40 24 E0 PDB R00,=R44 get a char
2662 7371 C8 2C CMB R00,")' ?
2663 7373 F7 06 IFNE CMB R00,")' no,
2664 7375 C8 28 CMB R00,")' ?
2665 7377 F7 02 IFNE CMB R00,")' no,
2666 7379 C8 29 CMB R00,")' ?
2667 737B ENDF
2668 737B ENDF
2669 737B F6 04 IFB yes,
2670 737D 42 89 ICM R02 inc accum
2671 737F 68 93 CLM R50 clear flags
2672 7381 ENDF
2673 7381 66 8B DCM R46 dec length
2674 7383 F6 9E UNHME until done
2675 7385 42 89 ICM R02 adjust for last
2676 7387 85 CLM R02 multiply
2677 7388 85 CLM R02 by 4
2678 7389 89 ICM R02 add first byte
2679 738A 89 ICM R02 add last byte
2680 738B 1A R3 STN R02,R24 save size
2681 738D 2E R3 STN R02,R56 set up
2682 738F F4 D7 JMG BRDING error, too big
2683 7391 5A 06 E5 PDB R32,=R06 save addr
2684 7394 CE FF FF JSB =R5MEM- get ram
2685 7397 5A 06 E3 PDB R32,=R06 get addr
2686 739A F8 A5 REN none, leave
2687 739C 64 2E A1 LDB R44,R56 get length
2688 739F 66 16 A1 LDB R46,R26 get address
2689 73A2 64 20 A3 STN R44,R40 copy
2690 73A5 60 B3 C5 83 STBD R40,=INSLEM set new values
2691 73A8 B0 D9 82 LDB R40,=USING? get using flag
2692 73AC CA 80 ADB R40,=80H indicate processed

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYX101 OBJ=RYX101 12/13/1983 11:50 AM PG 65
***** ENTER USING image parsing *****
2746 7411 B3 C9 83 STBD R02,=INSLEM and again
2747 7414 ENDF
2748 7414 CE 33 74 JSB =PRCGRP process image
2749 7417 40 1A 81 LDB R00,R32 compute end of
2750 741A 18 C3 RDB R00,R30 image
2751 741C B9 ICM R00 plus 1 for term
2752 741D 12 C1 CMB R00,R22 should be there
2753 741F F7 04 IFNE if not,
2754 7421 CE FF FF ERRING JSB =ERR1+ error
2755 7424 34 BYT 52D
2756 7425 ENDF
2757 7425 40 AB 2A LDB R00,")' get end mark
2758 7428 16 E4 PDB R00,=R26 mark end of image
2759 742A
2760 742A 42 B1 C7 83 RSTING LDB R02,=INSADR get start addr
2761 742E 89 ICM R02 inc addr
2762 742F B3 C8 83 STBD R02,=INCADR set current addr
2763 7432 9E RIM
2764 7433
2765 7433 5C 92 PRCGRP CLB R34 clear end flag
2766 7435 B2 64 84 STBD R34,=INPMH3 clear field count
2767 7438 9D CLE clear error flag
2768 7439 CE C7 72 JSB =FLDEND get the field
2769 743C F8 F4 REN error out
2770 743E 42 90 STB R02 was that a group?
2771 7440 F5 48 IFMG yes,
2772 7442 CE 74 75 JSB =GETRPL get the count
2773 7445 F8 EB REN error, leave
2774 7447 50 23 C1 CMB R20,R43 check if all used
2775 744A F6 D5 JME ERRING no, error
2776 744C 52 C3 RDB R22,R43 adjust pointer
2777 744E 89 ICM R22 skip end char
2778 744F 66 93 CLM R46 clear current count
2779 7451 25 A0 LDB R46,R45 get repeat count
2780 7453 F6 01 IFZR 0,
2781 7455 88 CLB R46 make it 1
2782 7456 ENDF
2783 7456 65 92 CLB R45 clear M fields
2784 7458 64 92 CLB R44 clear type flag
2785 745A 16 E5 PDB R44,=R26 put it out
2786 745C 42 B0 64 84 LDB R02,=INPMH3 get field count
2787 7460 06 E4 PDB R02,=R06 save it
2788 7462 CE 33 74 JSB =PRCGRP recurse for group
2789 7465 42 06 E2 PDB R02,=R06 get back count
2790 7468 40 1A 81 LDB R00,R32 compute end of
2791 746B 18 C3 RDB R00,R30 image
2792 746D 89 ICM R00 adjust for term
2793 746E 52 00 C1 CMB R22,R00 there too soon?
2794 7471 F8 AE JCY ERRING yes, gave up
2795 7473 F8 8D REN error, leave
2796 7475 64 93 CLM R44 clear for load
2797 7477 65 B0 64 84 LDB R45,=INPMH3 get new field count
2798 747B 88 CLB R45 adjust for group end

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RYZ101 12/13/1983 11:50 AM PG 66
=====
>>>>>> ENTER USING image parsing <<<<<<<
2799 747C 88 ICB R45 adjust for group begin
2800 747D 64 16 E5 PUND R44,R26 put group end out
2801 7480 42 25 C2 ADB R02,R45 add to total count
2802 7483 82 64 84 STBD R02,-INPMH3 save count
2803 7486 5C 92 CLB R34 clear ) flag
2804 7488 FO AE JNP PRELP go on
2805 748A ENDFIF
2806 748A F6 02 IFZR group end,
2807 748C 5C 88 ICB R34 set flag
2808 748E ENDFIF
2809 748E 50 8B DCM R20 adjust field len
2810 7490 F5 04 IFNG null field,
2811 7492 44 A9 25 75 GTO FLPCOM go on
2812 7496 ENDFIF
2813 7496 42 12 E0 PORD R02,R22 get a char
2814 7499 CE FF FF JSB =ALFNM uppercase it
2815 749C 42 C8 4B CMB R02,'K' free field?
2816 749F F6 0C IFEQ yes,
2817 74A1 50 91 ISM R20 at end?
2818 74A3 F6 2C JNZ IMVING no, error
2819 74A5 64 93 CLM R44 clear for load
2820 74A7 A8 02 LDB R44,-02H get free type
2821 74A9 66 88 ICB R46 set char count
2822 74AB FO 6E JNP FLPCOM finish up
2823 74AD ENDFIF
2824 74AD 50 89 ICM R20 back up one
2825 74AF 52 8B DCM R22 char in image
2826 74B1 75 93 CLM R65 clear char count
2827 74B3 60 93 CLM R55 clear type and CAP
2828 74B5 LOOP field loop
2829 74B5 CE 74 75 JSB =GETRPL get possible repl
2830 74B8 F8 B9 JEN REKIMP error, leave
2831 74BA 65 8B DCM R45 adjust for char
2832 74BC F5 01 IFNG was 0,
2833 74BE 89 ICM R45 restore it
2834 74BF ENDFIF
2835 74BF 75 25 C3 ADB R65,R45 add to char count
2836 74C2 52 23 C3 ADB R22,R43 adjust pointer
2837 74C5 50 C5 SBM R20,R43 adjust count
2838 74C7 42 12 E0 PORD R02,R22 get next char
2839 74CA CE FF FF JSB =ALFNM uppercase it
2840 74CD 50 8B DCM R20 adjust length
2841 74CF F5 04 IFNG off end,
2842 74D1 CE FF FF IMVING JSBN =ERR1+ error out
2843 74D4 34 BYT 52D
2844 74D5 ENDFIF
2845 74D5 42 C8 41 CMB R02,'A' alpha field?
2846 74D8 F6 05 IFEQ yes,
2847 74DA 6E A8 81 LDB R56,-81H get the type
2848 74DB FO 1B ELSE no,
2849 74DF C8 2F CMB R02,'/' line skip?
2850 74E1 F6 05 IFEQ yes,
2851 74E3 6E A8 40 LDB R56,-40H get type

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RYZ101 12/13/1983 11:50 AM PG 68
=====
>>>>>> ENTER USING image parsing <<<<<<<
2904 7550 43 C8 50 CMB R03,'P' is it P?
2905 7553 F6 03 IFEQ yes,
2906 7555 42 A8 80 LDB R02,-80H get flag
2907 7558 ENDFIF
2908 7558 60 94 ORG R55,R02 set CAP flag
2909 755A 43 C8 45 CMB R03,'E' is it E?
2910 755D F6 05 IFEQ yes,
2911 755F 75 C8 04 00 ADB R65,-4D,0,0 adjust char count
2912 7563 00 ENDFIF
2913 7564 43 92 CLB R03 set EQ flag
2914 7566 9E RTM
2915 7567 ENDFIF
2916 7567 FO CC IMNP until done
2917 7569
2918 7569 53 4D 45 43 NMSTBL ASE 'SNECRPD2', number table
2919 7573 FF VRL FF end mark
2920 7574
2921 7574 CE FF FF GETRPL JSB =EVIL save regs
2922 7577 10 FO DEF S20-37
2923 7579 90 CLE clear error flag
2924 757A 56 12 A1 LDM R26,R22 get pointer
2925 757D 52 10 A1 LDM R22,R20 get size
2926 7580 5E A8 03 LDB R36,-3D get max digits
2927 7583 CE 8E 69 JSB =GETNM convert to number
2928 7586 65 C9 00 01 CMM R45,-0,1,0 >= 256?
2929 758B 00
2930 758D 9C IFCY yes, error
2931 758E ICE set flag
2932 758E CE FF FF ENDFIF
2933 7591 58 23 A3 JSB =CLRERR clear possible errors
2934 7594 F6 03 STM R30,R43 save count
2935 7596 65 93 IFZR no number,
2936 7598 90 CLM R45 get 0
2937 7599 CLE clear flag
2938 7599 F9 04 ENDFIF
2939 759B CE FF FF IFEM errors,
2940 759E 3A JSBN =ERR1+ report it
2941 759F BYT 52D
2942 759F 9E ENDFIF
2943 75A0 RTM
2944 75A0 74 B9 C8 B3 GETUSE LDM R64,-INCADR get field
2945 75A4 C8 2B CMB R64,'*' end of image?
2946 75A6 F6 0D IFEQ yes,
2947 75A8 CE 2A 74 JSB =ASTING reset to begin
2948 75AB 74 B9 C8 B3 LDM R64,-INCADR get field
2949 75AF C8 2A CMB R64,'*' still end?
2950 75B1 F6 02 IFEQ yes,
2951 75B3 93 CLM R64 clear field
2952 75B4 9E RTM leave

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RYZ101 12/13/1983 11:50 AM PG 67
=====
>>>>>> ENTER USING image parsing <<<<<<<
2852 74E6 FO 11 ELSE no,
2853 74E8 C8 58 CMB R02,'X' char skip?
2854 74EA F6 05 IFEQ yes,
2855 74EC 6E A8 01 LDB R56,-01H get type
2856 74EE FO 08 ELSE no,
2857 74F1 CE 31 75 JSB =NMBFNT number?
2858 74F4 F6 0B JNE IMVING no, error
2859 74F6 6E A8 80 LDB R56,-80H get type
2860 74F9 ENDFIF
2861 74F9 ENDFIF
2862 74F9
2863 74F9 6F 90 TSB R57 test type
2864 74FB F7 04 IFNZ if set,
2865 74FD 2E C0 CMB R57,R56 match?
2866 74FF F6 00 JNE IMVING no, error
2867 7501 ENDFIF
2868 7501 2E 80 LDB R57,R56 get type
2869 7503 75 89 ICM R65 inc char count
2870 7505 50 90 TSB R20 check length
2871 7507 F6 AC WHNZ until done
2872 7509 64 93 CLM R44 clear for load
2873 750B 2F A0 LDB R44,R57 get type
2874 750D 60 25 A2 STB R55,R45 get CAP flag
2875 7510 75 C9 00 01 CMM R65,-0,1,0 check count
2876 7514 00
2877 7515 FB BA JCY IMVING error, too big
2878 7517 26 A2 STB R65,R46 get count
2879 7519 F7 0B JZR FLPCOM skip output
2880 751E 42 80 64 84 FLPCOM PUND R44,R26 put it out
2881 7522 88 LDB R02 get field count
2882 7523 82 64 84 STBD R02,-INPMH3 inc it
2883 7526 52 89 ICM R22 save it
2884 7528 -5C 90 ICB R34 skip end char
2885 752A 9D CLE clear error flag
2886 752B F6 39 ANDZ yes, done
2887 752D 44 A9 37 74 GTO PRELP around again
2888 7531
2889 7531 40 A9 69 75 NMSTBL LDM R00,-NMSTBL get table addr
2890 7535 LOOP scan loop
2891 7535 43 00 E0 PORD R03,R00 get entry
2892 7538 F4 2C RMC end, leave
2893 753A 02 C0 CMB R03,R02 match?
2894 753C F6 29 IFEQ yes,
2895 753E 42 92 CLB R02 clear for ORG
2896 7540 43 C8 52 CMB R03,'R' is it R?
2897 7543 F6 03 IFEQ yes,
2898 7545 42 A8 01 LDB R02,-01H get flag
2899 7548 ENDFIF
2900 7548 43 C8 43 CMB R03,'E' is it E?
2901 754B F6 03 IFEQ yes,
2902 754D 42 A8 02 LDB R02,-02H get flag
2903 7550 ENDFIF

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RYZ101 12/13/1983 11:50 AM PG 69
=====
>>>>>> ENTER USING image parsing <<<<<<<
2953 75B5 ENDFIF
2954 75B5 ENDFIF
2955 75B5 90 TSB R64
2956 75B6 F6 2E IFZR group?
2957 75B8 75 90 TSB R65 yes,
2958 75BA F6 0B IFZR begin?
2959 75BC 77 92 CLB R67 yes,
2960 75BE 74 B8 C8 B3 STMB R64,-INCADR clear current count
2961 75C2 CE E6 75 JSB =INCGET save it
2962 75C5 FO D9 JNP GETUSE inc pointer
2963 75C7 ENDFIF try again
2964 75C7 8A OCB R65
2965 75C8 85 CLM R65 cancel group began
2966 75C9 85 CLM R65 multiply field
2967 75CA 42 B1 C8 B3 LDB R02,-INCADR count by 4
2968 75CE 35 C5 SBM R02,R65 get pointer
2969 75D0 74 02 A5 LDB R64,R02 back up for group
2970 75D3 77 88 ICB R67 get field
2971 75D5 73 90 TSB R63 inc current count
2972 75D7 F6 02 IFZR test no update
2973 75D9 74 A7 STND R64,R02 ok,
2974 75DB ENDFIF save new value
2975 75DB 77 36 C0 CMB R67,R66
2976 75DE FB E2 JCY SKPFLO check for done
2977 75E0 42 B3 C8 B3 STND R02,-INCADR yes, move to next
2978 75E4 FO DC JNP SKPFLO save pointer
2979 75E6 ENDFIF try again
2980 75E6
2981 75E6 42 B1 C8 B3 IMCGET LDM R02,-INCADR get pointer
2982 75EA C8 04 00 ADB R02,-84 inc to next
2983 75ED B3 C8 B3 STND R02,-INCADR save pointer
2984 75F0 9E RTM
2985 75F1
2986 75F1 50 06 E5 TRMSZ= PUND R20,R06 save pointer
2987 75F4 40 B1 C8 B3 LDM R00,-INCADR get pointer
2988 75F8 73 A8 01 LDB R63,-1 set no update
2989 75FB CE A0 75 JSB =GETUSE get field
2990 75FE 40 B3 C8 B3 STND R00,-INCADR restore pointer,
2991 7602 74 90 TSB R64 K or / ?
2992 7604 F4 08 IFPS maybe,
2993 7606 F2 06 IFEV yes,
2994 7608 60 92 CLB R40 treat as normal
2995 760A 50 06 E3 PORD R20,-R06 restore pointer
2996 760D 9E RTM
2997 760E ENDFIF
2998 760E ENDFIF
2999 760F 50 36 A0 LDB R20,R66 get count
3000 7611 CE FF FF JSB =LOBLOC convert to BCD
3001 7614 50 36 A2 STB R20,R66 save lower digits
3002 7617 11 A0 LDB R20,R21 get upper digits
3003 7619 CE FF FF JSB =TOASC2 convert to ASCII
3004 761C 65 A9 53 5A LDM R45,'S2' get command
3005 7620 3D

```

NOMAS
NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KRNRA 08/31/82:-----SRC=RYZIO1 08J=RYZIO1 12/13/1983 11:50 AM PG 71
ITEM LOC OBJECT CODE SRC=RYZIO1 08J=RYZIO1 12/13/1983 11:50 AM PG 71
-----
>>>>>> ENIER support routines
3017 7638 9D GETMRE CLE clear error flag
3018 7639 40 B0 09 82 LDBD R00,=USING? USING?
3019 7630 F7 28 IFBZ yes,
3020 763F B1 F7 83 LDAD R00,=INSDR get table start
3021 7642 D5 5B 82 SDR R00,=LAVAIL compute release amount
3022 7645 F5 04 IFG something wrong,
3023 7647 CE FF FF JSB =EARI+ report error
3024 764A 0F BYT 150
3025 7640 EMDIF
3026 7648 5E 01 51 82 LDAD R36,=FWARS get the E.AREN
3027 764F 52 1E 05 04 LDAD R22,R36,E.AREN value
3027 7653 00
3028 7654 00 C5 SBN R22,R00 reduce E.AREN
3029 7656 02 A7 SIMD R22,R02 save new one
3030 7658 5E 01 5B 82 LDAD R36,=LAVAIL get LAVAIL
3031 765C 00 C3 ROM R36,R00 move it up
3032 765E 03 5B 82 SIMD R36,=LAVAIL save new one
3033 7661 60 B8 C7 83 LDBI R40,=INSDR get term flags
3034 7665 F0 07 ELSE no,
3035 7667 CE FF FF JSB =RELHEN clean up hen
3036 766A FB CB REN error out
3037 766C 60 92 ELB R40 fake term flags
3038 766E EMDIF
3039 766E 50 A9 0E 82 LDM R20,=(INPMN2+1) get pointer
3040 7672 60 C8 03 CNB R40,=O3H all disabled?
3041 7675 F6 03 IFEQ yes,
3042 7677 CE F1 75 JSB =TRANSZ= set up size
3043 767A EMDIF
3044 767A 60 06 E4 PUDB R40,=R06 test 'I, save flags
3045 767D F2 08 IFEV no 'I,
3046 767F 64 A9 54 52 LDM R44,='IR', get END term
3046 7683 21 2C
3047 7685 10 E5 PUMP R44,=R20 push it
3048 7687 EMDIF
3049 7687 60 86 LRB R40 test W,
3050 7689 F2 08 IFEV no W,
3051 768B 64 A9 54 52 LDM R44,='IR', get EOL term
3051 768F 2A 2C
3052 7691 10 E5 PUMP R44,=R20 push it
3053 7693 EMDIF
3054 7693 60 86 LRB R40 test X,
3055 7695 F2 08 IFEV no X,
3056 7697 64 A9 45 54 LDM R44,='EI', get no ETO term
3056 769B 20 2C
3057 769D 10 E5 PUMP R44,=R20 push it
3058 769F EMDIF
3059 769F 61 A9 54 4C LDM R41,='IL',SDA get rest of message
3059 76A3 2B 2F 53 44
3059 76A7 41
3060 76AB 10 E5 PUMP R41,=R20 push it
3061 76AB 50 C0 E 82 SBN R20,=(INPMN2+1) compute length
3062 76AE B2 00 82 STBD R20,=INPMN2 save it
3063 76B1 CE 0F 78 JSB =TLMI00 go get input

```

```

KRNRA 08/31/82.....SRC=RYZIOI 08J=RYZIOI 12/13/1983 11:50 AM PG 71
ITEM LOC OBJECT CODE SRC=RYZIOI 08J=RYZIOI 12/13/1983 11:50 AM PG 71
*****
>>>>>> ENIER support routines
3017 7638 9D GETMRE CLE clear error flag
3018 7639 40 B0 09 82 LDBD R00,=USING? USING?
3019 7630 F7 28 IFBZ yes,
3020 763F B1 F7 83 LDAD R00,=INSDR get table start
3021 7642 D5 5B 82 SDR R00,=LAVAIL compute release amount
3022 7645 F5 04 IFG something wrong,
3023 7647 CE FF FF JSB =EARI+ report error
3024 764A 0F BYT 150
3025 7640 EMDIF
3026 7648 5E 01 51 82 LDAD R36,=FWARS get the E.AREN
3027 764F 52 1E B5 04 LDAD R22,R36,E.AREN value
3027 7653 00
3028 7654 00 C5 SBN R22,R00 reduce E.AREN
3029 7656 02 A7 SIMD R22,R02 save new one
3030 7658 5E 01 5B 82 LDAD R36,=LAVAIL get LAVAIL
3031 765C 00 C3 ROM R36,R00 move it up
3032 765E B3 5B 82 SIMD R36,=LAVAIL save new one
3033 7661 60 B8 C7 83 LDBI R40,=INSDR get term flags
3034 7665 F0 07 ELSE no,
3035 7667 CE FF FF JSB =RELHEN clean up hen
3036 766A FB CB REN error out
3037 766C 60 92 ELB R40 fake term flags
3038 766E EMDIF
3039 766E 50 A9 0E 82 LDM R20,=(INPMN2+1) get pointer
3040 7672 60 C8 03 CNB R40,=O3H all disabled?
3041 7675 F6 03 IFEQ yes,
3042 7677 CE F1 75 JSB =TRANSZ= set up size
3043 767A EMDIF
3044 767A 60 06 E4 PUSD R40,R06 test 'I, save flags
3045 767D F2 08 IFEV no 'I,
3046 767F 64 A9 54 52 LDM R44,='IR', get END term
3046 7683 21 2C
3047 7685 10 E5 PUMP R44,R20 push it
3048 7687 EMDIF
3049 7687 60 86 LRB R40 test W,
3050 7689 F2 08 IFEV no W,
3051 768B 64 A9 54 52 LDM R44,='IR', get EOL term
3051 768F 2A 2C
3052 7691 10 E5 PUMP R44,R20 push it
3053 7693 EMDIF
3054 7693 60 86 LRB R40 test X,
3055 7695 F2 08 IFEV no X,
3056 7697 64 A9 45 54 LDM R44,='EI', get no ETO term
3056 769B 20 2C
3057 769D 10 E5 PUMP R44,R20 push it
3058 769F EMDIF
3059 769F 61 A9 54 4C LDM R41,='IL',SDA get rest of message
3059 76A3 2B 2F 53 44
3059 76A7 41
3060 76AB 10 E5 PUMP R41,R20 push it
3061 76AB 50 C0 E2 SBN R20,=(INPMN2+1) compute length
3062 76AE B2 00 82 STBD R20,=INPMN2 save it
3063 76B1 CE 0F 78 JSB =LIMIOI go get input

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 74
*****
***** ENTER stuff *****
3153 7750 E4          PUNB R66,+R06      save en
3154 775E C8 03      CNB R66,+R06M    all disabled?
3155 7760 F6 01      IFEO          yes,
3156 7762 92          CLB R66          fake a normal
3157 7763            ENDIF
3158 7763 BA C7 03    STBI R66,+INSADR    set it
3159 7766 CE 38 76    JSB -GETHRE         skip line
3160 7769 76 06 F2    PUNB R66,-R06       get back flags
3161 776A BA C7 03    STBI R66,+INSADR    restore en
3162 776F E2          PUNB R66,-R06       get counter
3163 7770 BA          DEC R66          dec counter
3164 7771 F8 CE       JEM JMWIED          error out
3165 7773 F6 E2       JMWZ              until done
3166 7775            ENDIF
3167 7775 52 93       CLM R22            kill current record
3168 7777 F0 D3       JMP ENTACHN        try again
3169 7779            ENDIF
3170 7779 72 93       CLM R62            pretend not using
3171 777B            ENDIF
3172 777B            CLE                clear error flag
3173 777B 9D          TSB R61            any param
3174 777E 71 90       RZR              no, done
3175 777E 71 9A       CNB R61,+4D        number?
3176 7780 C8 04       JEQ STOMUM        yes, go do it
3177 7782 F7 04       LMB R61,+9D        numeric array?
3178 778A C8 09       IFCV            yes,
3179 7786 FA 0A       STOMUM JSB =ENTNM+  get the number
3180 7788 CE CE 77     JEM JEMIED        error, clean up
3181 778B F8 E4       JSB =STOSV        store it
3182 778D CE FF FF     ELSE            no,
3183 7790 F0 0B       JSB =ENTSI+       get the strang
3184 7792 CE A8 77     JEM JEMIED        error, leave
3185 7795 F8 0A       JSB =STOSI        store it
3186 7797 CE FF FF     ENDIF
3187 779A            RIM
3188 779A 9E          SVEBUF TSM R22      any buffer left
3189 779B            IFNG
3190 779B 52 91       CLM R22            clear size
3191 779B F5 01       IFNE
3192 779F 93          CLM R22            clear size
3193 77A0            ENDIF
3194 77A0 B3 BA 84     SIND R22,-(FILNME+2D) save buffer size
3195 77A3 54 B3 A3 84 SIND R24,-FILNME and addr
3196 77A7 9E          RIM
3197 77AB            ENDIF
3198 77AB 9D          CLE                clear error flag
3199 77AB 40 0A B5 FE LDMD R00,R12,(0-4D) get max var length
3199 77AB FF          TSB R62            USING?
3200 77AB 72 90       IFNZ              yes,
3201 77B0 F7 0C       CNB R64,+B1H      string field?
3202 77B2 74 C8 81     IFNE            no,
3203 77B5 F7 04       JSB =ERRI+       report error
3204 77B7 CE FF FF

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 76
*****
***** ENTER stuff *****
3258 7818 FE 05       IFRM              check C,
3259 781A 50 C8 2C    CNB R20,+R1  comma?
3260 781D F7 E8       JEQ SKIPCP        yes, skip it
3261 781F            ENDIF
3262 781F            ENDIF
3263 781F 50 36 E4     PUNB R20,+R66      save char
3264 7822 40 88       DCN R00          dec count
3265 7824 F6 D5       IANZ              until done
3266 7826            ENDIF
3267 7826            ENDIF
3268 7826 90          CLE                clear errors
3269 7827 CE 33 78    JSB =ENTNM+       process it
3270 782A 52 06 E3    PUNB R22,-R06      get back length
3271 782D 54 E3       PUNB R24,-R06      get back pointer
3272 782F CE 98 77    JSB =SVEBUF        save en
3273 7832 9E          RIM
3274 7833            ENDIF
3275 7833            ENDIF
3276 7833 51 92       ENTNUM CLB R21      clear --i count
3277 7835 40 92       CLB R00          clear . flag
3278 7837            LOOP              scan loop
3279 7837 52 88       DCN R22          dec length
3280 7839 F5 05       IFNG            done,
3281 783B 60 93       CLM R40          get a 0
3282 783D 0A E5       PUNB R40,+R12     put it out
3283 783F 9E          RIM
3284 7840            ENDIF
3285 7840 50 14 E0     PUNB R20,+R24      get the char
3286 7843 CE FF FF     JSB =D161I      digit?
3287 7846 F8 20       JEM SCHNUM        yes, get the #
3288 7848 40 90       TSB R00          need one?
3289 784A F7 03       IFNZ            yes,
3290 784C 92          CLB R00          clear . flag
3291 784D 51 92       CLB R21          clear count
3292 784F            ENDIF
3293 784F 50 C8 2D    CNB R20,+R1  have a -?
3294 7852 F7 02       IFNE            no,
3295 7854 C8 28       CNB R20,+R1  maybe a +?
3296 7856            ENDIF
3297 7856 F7 06       JEQ GOT+         yes, go adjust count
3298 7858 C8 2E       CNB R20,+R1  have a .?
3299 785A F6 06       IFEQ            yes,
3300 785C 00 A2       STB R20,R00      set . flag
3301 785E 51 88       ICB R21          inc count
3302 7860 F0 04       ELSE            garbage,
3303 7862 51 92       CLB R21          clear count
3304 7864 40 92       CLD R00          clear . flag
3305 7866            ENDIF
3306 7866 F0 CF       WIMP              until done
3307 7868            ENDIF
3308 7868 51 10 A2     SCHNUM STB R21,R20 build multi-byte
3309 786B 92          CLB R21          count of +-
3310 786C 54 8B       DCN R24          adjust for digit

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 75
*****
***** ENTER stuff *****
3205 77BA 78          BYI 120B
3206 77BB            ENDIF
3207 77BB 40 36 A1     LDN R00,R66      use field size
3208 77BE            ENDIF
3209 77BE 40 12 E1     ENI R00,R22      > source?
3210 77C1 FA 01       IFCV            yes,
3211 77C3 A1          LDN R00,R22      get source length
3212 77C4            ENDIF
3213 77C4 0A E5       PUNB R00,+R12      length to stack
3214 77C6 54 E5       PUNB R24,+R12      addr to stack
3215 77C8 00 C3       RDN R24,R00      adjust addr
3216 77CA E2 C5       SBN R22,R00      adjust length
3217 77CC F0 C0       JMP SVEBUF        save for later
3218 77CE            ENDIF
3219 77CE 72 90       ENTNM+ TSB R62      USING?
3220 77D0 F7 61       IFNZ              yes,
3221 77D2 74 C8 80     CNB R64,+R0H      number?
3222 77D5 F7 04       IFNE            no,
3223 77D7 CE FF FF     JSB =ERRI+       error
3224 77DA 52          BYI 82B
3225 77DB            ENDIF
3226 77DB 42 14 A1     LDN R02,R24      get pointer
3227 77DE 36 C3       RDN R02,R66      inc pointer
3228 77E0 06 E5       PUNB R02,+R06      save it
3229 77E2 12 A1       LDN R02,R22      get length
3230 77E4 36 C5       SBN R02,R66      dec length
3231 77E6 06 E5       PUNB R02,+R06      save it
3232 77E8 F4 03       IFPS            <= source
3233 77EA 52 36 A1     LDN R22,R66      get length
3234 77ED            ENDIF
3235 77ED 75 90       TSB R65          test CRP flags
3236 77EF F7 35       IFNZ              set,
3237 77F1 40 12 A1     LDN R00,R22      get length
3238 77F4 F7 30       IFNZ
3239 77F6 42 14 A1     LDN R02,R24      get pointer
3240 77F9 76 A1       LDN R66,R24      again
3241 77FB            LOOP
3242 77FB 50 02 E0     PUNB R20,+R02      process [PR
3243 77FE 75 90       TSB R65          get char
3244 7800 F5 09       IFNG            test for P
3245 7802 50 C8 2E     CNB R20,+R1  have a period?
3246 7805 F6 04       IFEQ            yes,
3247 7807 52 8B       DCN R22          dec length
3248 7809 F0 17       JMP CPROK        carry on
3249 780B            ENDIF
3250 780B            ENDIF
3251 780B 75 90       TSB R65          test for R
3252 780D F3 09       IFOD            yes,
3253 780F 50 C8 2C     CNB R20,+R1  comma?
3254 7812 F6 02       IFEQ            yes,
3255 7814 A8 2E       LDN R20,+R1  get a period
3256 7816            ENDIF
3257 7816 F0 07       ELSE            no,

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8101 OBJ=RYX101 12/13/1983 11:50 AM PG 77
*****
***** ENTER stuff *****
3311 786E C5          SBM R24,R20      and count
3312 786F 52 89       ICM R22          adjust for digit
3313 7871 C3          RDN R22,R20      and count
3314 7872 18 A3       STM R22,R30      move length
3315 7874 14 C3       RDN R22,R24      compute end of string
3316 7876 06 E5       PUNB R22,+R06      save end
3317 7878 CE 82 78    JSB =GETVAL      get the number
3318 787B 52 06 E3    PUNB R22,-R06      get end back
3319 787E 14 C5       SBM R22,R24      compute new size
3320 7880 F0 AD       JMP SVEBUF+      save for later
3321 7882            ENDIF
3322 7882 58 14 C3     GETVAL RDN R30,R24 compute end of string
3323 7885 58 18 A4     LDND R32,R30      get char after string
3324 7888 58 A8 25     LDB R33,+X'    change to X
3325 788B 86          STDB R33,R30      to mark end
3326 788C 5C 08 A1     LDN R34,R10      save R10
3327 788F 48 14 A1     LDN R10,R24      set R10 for scan
3328 7892 5E 93       CLM R36          clear sign flag
3329 7894 CE FF FF     JSB =GCHAR      get first char
3330 7897            SCANVL LOOP        scan string
3331 7897 CE 4A 65     JSB =SSCAN      get a token
3332 789A 4C C8 2B     CNB R14,+PLUS  plus sign?
3333 789D F7 F8       WHEQ            yes, ignore
3334 789F C8 2D       CNB R14,+MINUS  minus sign?
3335 78A1 F6 04       IFEQ            yes,
3336 78A3 5E 88       ICB R36          add one sign
3337 78A5 F0 F0       JMP SCANVL        get more
3338 78A7            ENDIF
3339 78A7 C8 1A       CNB R14,+J2     integer?
3340 78A9 F7 0A       JEQ INTVAL        yes, process
3341 78AB C8 04       CNB R14,+4       floating pt number?
3342 78AD F7 0F       JEQ VALDUM        yes, number there
3343 78AF CE FF FF     JSB =ERROR      bad paran (non numeric)
3344 78B2 59          BYI 89D
3345 78B3 F0 13       JMP RESTOV        finish up
3346 78B5            ENDIF
3347 78B5 2D 24 A1     INTVAL LDN R75,R44 move for push
3348 78B8 7E A8 FF     LDB R74,+FF    get int flag
3349 78BB 78 0A E5     PUNB R70,+R12 push number
3350 78BE 5E 90       VALDUM TSB R36 odd # of signs?
3351 78C0 F3 05       IFOD            yes,
3352 78C2 99          BCO              for CHSROI
3353 78C3 CE FF FF     JSB =CHSROI    do nonadic minus
3354 78C6 98          BIM              for us
3355 78C7            ENDIF
3356 78C7 9D          CLE                clear error flag
3357 78C8 42 A8 20     RESTOV LDB R02,+R1 get a blank
3358 78CB 54 08 A1     LDN R24,R10      get the stop point
3359 78CE 8B          DCN R24          back up 1 (stop after non-blank)
3360 78CF            LOOP              blank backup
3361 78D1 54 8B       DCN R24          back up 1
3362 78D3 42 14 D8     CNB R02,R24      blank?
3363 78D4 F7 F9       WHEQ            until not

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 78
-----
>>>>>>> ENTER stuff <<<<<<<<
3364 78D6 54 89 ICM R24 forward 1
3365 78D8 5C 08 R3 SIM R34,R10 restore R10
3366 78D8 5A 18 A6 STBD R32,R30 restore char
3367 78DE 9E R1N done
3368 78DF
3369 78DF 43 80 80 82 1LM10 LDBD R03,-PLSTAT get pil stats
3370 78E3 06 E4 PURD R03,+R06 save it
3371 78E5 CF EB RNM R03,-O0EBH clear disp/print
3372 78E2 82 8D 82 STBD R03,-PLSTAT put back
3373 78EA 64 93 CLM R44 clear for
3374 78EC 0A E5 PURD R44,+R12 null device
3375 78EE 50 A9 00 82 LDM R20,-INPM2 get message
3376 78F2 CE 65 70 JSB =INPM2G send it
3377 78F5 F8 E7 REN error out
3378 78F7 CE 65 67 JSB =N100L get result
3379 78FA 42 06 E2 PURD R02,-R06 get back stats
3380 78FD B2 8D 82 STBD R02,-PLSTAT restore en
3381 7900 9E R1N
3382 7901
3383 7901 R1 BYT 241
3384 7902 CE FF FF ENTER. JSB =OWR12? device specified?
3385 7905 F8 04 IFEZ no,
3386 7907 42 92 CLB R02 clear cleanup flag
3387 7909 F0 20 ELSE yes,
3388 790B 42 C8 08 LMB R02,-8D number?
3389 790E F6 08 IFEQ yes,
3390 7910 CE 69 72 JSB =UNL unlisten
3391 7913 CE 6F 72 JSB =UNL and untalk
3392 7916 CE FC 71 JSB =TALK talk it
3393 7919 F0 18 ELSE no,
3394 791B 43 80 8D 82 LDBD R03,-PLSTAT get pil stats
3395 791F 06 E4 PURD R03,+R06 save it
3396 7921 CF EB RNM R03,-O0EBH clear disp/print
3397 7923 82 8D 82 STBD R03,-PLSTAT put back
3398 7926 50 A9 4E 79 LDM R20,-EWM2G get message
3399 792A CE EF 6F JSB =DOSM10 send it
3400 792D 42 06 E2 PURD R02,-R06 get back stats
3401 7930 82 8D 82 STBD R02,-PLSTAT restore en
3402 7933
3403 7933 42 A8 80 LDB R02,-80H set cleanup flag
3404 7936
3405 7936 CE 24 78 JSB =SETJOB save flag
3406 7939 4F 90 ISB R17 errors?
3407 793B F5 04 IFNG yes,
3408 793D 44 A9 E6 76 GTO INVEDA clean up
3409 7941
3410 7941 60 93 CLM R40 clear
3411 7943 82 09 82 STBD R40,-USING? using flag,
3412 7946 83 C5 83 STBD R40,-IMSLEN using data and
3413 7949 64 83 A9 84 STBD R44,-FILNME buf markers
3414 794D 9E R1N
3415 794E
3416 794E 10 ENTHSG BYT 16D length

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 80
-----
>>>>>>> IOSIZE processing <<<<<<<<
3420 7958 R1 BYT 241
3421 795E 98 IOSIZ. BIN for sure
3422 795D CE 8F 79 JSB =APROP+ get the file
3423 7960 F8 EB REN error out
3424 7962 5E 06 E5 PURD R36,+R06 save pointer
3425 7965 CE FF FF JSB =ONEB get number
3426 7968 5E 06 E3 PURD R36,-R06 restore pointer
3427 796B 66 91 ISN R46 number ok?
3428 796D F5 01 IFNG " make 0
3429 796F 93 CLM R46
3430 7970
3431 7970 C9 00 60 CMY R46,-86000H too big?
3432 7973 FA 04 IFCY yes,
3433 7975 CE FF FF JSBH =ERR1+ error out
3434 7978 59 BYT 85D
3435 7979
3436 7979 1E B7 10 00 STBD R46,K36,$16 set new number
3437 797D 9E R1N
3438 797E
3439 797E 00 2D BYT 0.55
3440 7980 98 IOSIZ. BIN for sure
3441 7981 CE 8F 79 JSB =APROP+ get the file
3442 7984 F8 F7 REN error out
3443 7986 5E 1E B5 10 LDM R36,K36,$16 get the size
3444 798A 00 JSB =MUMOUT output it
3445 798E 9E R1N
3446 798F
3447 798F CE FF FF APROP+ JSB =ROMJSB do a romjob
3448 7992 FF FF DEF APROP+ to our better
3449 7994 64 40 DEF BYRON2 half to open file
3450 7996 9E R1N
3451 7997
3452 7997 R1 BYT 241
3453 7998 CE FF FF KBDIS. JSB =GETADR get addr
3454 799B 5A 0F E3 PURD R32,-R12 and length
3455 799E 8B DCM R32 dec length
3456 799F F4 08 JNG KBDIFF was zero, keybd off
3457 79A1 F6 0C IFZR was 1,
3458 79A3 1C A4 LDBD R32,R34 get char
3459 79A5 C8 2A CMB R32,'A' as it a "
3460 79A7 F6 06 IFEQ yes,
3461 79A9 73 B1 FF FF KBDIFF LDM R63,-BLANKS clear dev name
3462 79AD F0 32 JMP SETKBD go set it
3463 79AF
3464 79AF
3465 79AF 4A C8 04 00 RDM R12,-84
3466 79B3 CE FF FF JSB =RELJSB restore the stack
3467 79B6 FF FF DEF FLS1AK get a device name
3468 79B8 F9 04 IFEN error,
3469 79BA CE FF FF BACKBD JSBH =ERR1+ report it
3470 79BD 3F BYT 63D
3471 79BE

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 79
-----
>>>>>>> ENTER stuff <<<<<<<<
3417 794F 55 4E 4E 2C ASE 'UNI,UNI,TADM' talk it
3418 7953 55 4E 54 2C
3419 7957 54 41 44 23
3418 7958

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RY&I01 12/13/1983 11:50 AM PG 81
-----
>>>>>>> IOSIZE processing <<<<<<<<
3472 798E 73 AB 3A LDB R63,'-' add a :
3473 79C1 06 E5 PURD R63,+R06 save the name
3474 79C3 5A A9 02 00 LDM R32,+82 assume len=2
3475 79C7 76 C8 20 CMB R66,' ' have a blank?
3476 79CA F7 08 IFNE no,
3477 79CC 5A 89 ICM R32 inc len
3478 79CE 77 C8 20 CMB R67,' ' blank?
3479 79D1 F7 02 IFNE no,
3480 79D3 5A 89 ICM R32 inc again
3481 79D5
3482 79D5 F0 02 ELSE yes,
3483 79D7 3A A1 LDM R66,R64 get the name
3484 79D9
3485 79D9 CE FF FF JSB =GETPAD get device address
3486 79DC 73 06 E3 PURD R63,-R06 restore name
3487 79DF F8 D9 JEM BACKBD error out
3488 79E1 CE 8F 79 SETKBD JSB =APROP+ get file open
3489 79E4 73 1E B7 12 STBD R63,K36,$16 save name
3489 79E8 00
3490 79E9 9E R1N
3491 79EA

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE  SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 82
>>>>>> keyboard is stuff <<<<<<<<
3493 79E9 CE FF FF DOKBD JSB =EVAL save those regs
3494 79ED 10 D0 DEF $20-77
3495 79EF CE 8F 79 JSB =BPROG get the file
3496 79F2 73 1E 85 12 LDND R63,X36,X18 get the dev name
3496 79F6 00
3497 79F7 01 FF F* CMD R63,-BLANKS enabled?
3498 79FA 27 ED REQ disabled, return
3499 79FC 43 80 8D 82 LDND R03,-PLSTAT get the loop status
3500 7A00 CF 20 ANH R03,-20H check OFF10
3501 7A02 F6 E5 RNZ OFF, leave
3502 7A04 42 92 CLD R02 clear
3503 7A06 CE 24 78 JSB =SET10B ESC flag
3504 7A09 40 81 BE 83 LDND R00,-LEEWAY get current leeway
3505 7A0D 06 E5 PUND R00,-R06 save it
3506 7A0F CD 1E 00 SBN R00,-R30 adjust for us
3507 7A12 83 BE 83 STND R00,-LEEWAY save new value
3508 7A15 81 51 82 LDND R00,-FINWARS get the current
3509 7A18 76 00 85 04 LDND R66,X00,E.RNEM E.RNEM value
3509 7A1C 00
3510 7A1D 06 E5 PUND R66,-R06 save it
3511 7A1F 93 CLN R66 clear it
3512 7A20 02 A7 SIND R66,R02 out for us
3513 7A22 80 54 84 LDND R66,-10KEN get token
3514 7A25 18 92 CNB R66,-RSGM10 doing assignio?
3515 7A27 F7 52 JEQ KBDGF+ yes, suicide
3516 7A29 CE FF FF GETKBD JSB =KEY? key already?
3517 7A2C F8 50 JEN KBDEND yes, leave
3518 7A2E CE FF FF JSB =STOP? attn or power?
3519 7A31 F8 4B JEN KBDEND yes, leave
3520 7A33 CE FF FF JSB =RELNEM release memory
3521 7A36 F8 43 JEN KBDGF+ error out
3522 7A38 CE FF FF JSB =SVECHK gave a chance
3523 7A3B CE 1D 71 JSB =KEYB8. get devname
3524 7A3E F8 3B JEN KBDGF+ error, leave
3525 7A40 50 89 CE 70 LDND R20,-KEYMSG get message
3526 7A44 CE 65 70 JSB =MOVMSG move it
3527 7A47 F8 32 JEN KBDGF+ error, leave
3528 7A49 46 C9 16 81 CNB R06,-(R6LIM1-100) stack too full?
3529 7A4D FA 06 IFCY yes,
3530 7A4F CE FF FF JSB =ERR report
3531 7A52 55 BYT 85D
3532 7A53 F0 26 JNP KBDGF+ clean up
3533 7A55
ENDIF
3534 7A55 50 89 64 84 LDND R20,-IMPW83 must protect scratch
3535 7A59 43 88 0A LDND R03,-10D get loop count
3536 7A5C LOOP move loop
3537 7A5E 60 10 E1 PUND R40,-R20 get some
3538 7A5F 06 E5 PUND R40,-R06 save it
3539 7A61 43 8A DCB R03 dec count
3540 7A63 F6 F7 MINZ until done
3541 7A65 50 E5 PUND R20,-R06 save pointer
3542 7A67 CE 65 67 JSB =H110B. get a key
3543 7A6A 50 06 E3 PUND R20,-R06 get pointer

```

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE  SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 84
>>>>>> keyboard is stuff <<<<<<<
3592 79EB 7D 1E 85 GETKBD LDND R75,R36 get line number
3593 79EE 7F 92 CLD R77 clear upper byte
3594 79F0 7C 88 FF LDND R74,-FF mark integer
3595 79F3 78 0A E5 PUND R70,-R12 on stack
3596 79F6 CE FF FF JSB =DNED get back as binary
3597 79F9 43 26 80 LDND R03,R46 move it
3598 79FC
ENDIF
3599 79FC
ENDIF
3600 79FC 42 03 80 LDND R02,R03 move it
3601 79FF CE FF FF JSB =PUKEY put it out
3602 7B02 44 89 7D 7A CTD KBDEND finish up
3603 7B06

```

NOMAS
 Not Manufacturer Supported
 Recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE  SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 83
>>>>>> keyboard is stuff <<<<<<<
3544 7A6D 43 88 0A LDND R03,-10D get counter
3545 7A70 LOOP move loop
3546 7A70 60 06 E3 PUND R40,-R06 get some
3547 7A73 10 E7 PUND R40,-R20 restore it
3548 7A75 43 8A DCB R03 dec count
3549 7A77 F6 F7 MINZ until done
3550 7A79 F9 18 IFEN error,
3551 7A7B CE 89 79 KBDGF+ JSB =KBDGF+ disable us
3552 7A7E CE FF FF KBDEND JSB =RELNEM clean up
3553 7A81 76 06 E3 PUND R66,-R06 get back old E.RNEM
3554 7A84 40 81 51 82 LDND R00,-FINWARS restore it
3555 7A88 76 00 87 04 STND R66,X00,E.RNEM so we are safe
3556 7A8C 00
3556 7A8D 06 E3 PUND R66,-R06 get old LEEWAY
3557 7A8F 83 BE 83 STND R66,-LEEWAY restore it
3558 7A92 9E RNZ
3559 7A93
ENDIF
3560 7A93 50 0A E3 PUND R20,-R12 get the address
3561 7A96 52 E3 PUND R22,-R12 get the length
3562 7A98 F7 8F JNZ GETKBD none, try again
3563 7A9A 43 10 A4 LDND R03,R20 get the char
3564 7A9D 42 80 5A 82 LDND R02,-10BYTE get ESC flag
3565 7A9F F4 11 IFPS not set,
3566 7A83 43 C8 1B CNB R03,-ESC ESC char?
3567 7A86 F6 0A IFEQ yes,
3568 7A8B 42 80 80 LDND R02,-80H get flag
3569 7A8D CE 24 78 JSB =SET10B set flag
3570 7A8E 44 89 28 7A GETKBD+ GET GETKBD get another key
3571 7A82
ENDIF
3572 7A82 F0 48 ELSE
3573 7A84 92 CLD R02 set,
3574 7A85 CE 24 78 JSB =SET10B clear ESC
3575 7A88 43 C8 1B CNB R03,-ESC was that ESC?
3576 7A8B F7 3F IFNE no,
3577 7A8D 06 E4 PUND R03,-R06 save the char
3578 7A8F 60 83 48 45 LDND R40,-KEYMAP get filename
3579 7A93 20 20
3579 7A93 CE FF FF JSB =FLOPEN get file
3580 7A9C 50 0E E2 PUND R20,-R06 restore char
3581 7A9F F8 00 JEN GETKBD+ no file, try again
3582 7A81 40 18 85 04 LDND R00,X30,DR.TYP get the type
3583 7A85 00
3583 7A86 C9 BE 54 CNB R00,-1Y1EX1 must be text
3584 7A89 F6 03 JNE GETKBD+ no, try again
3585 7A8B LOOP search loop
3586 7A8D 40 1E 84 04 LDND R00,X36,X4 get a char
3587 7A8F 00
3587 7A90 10 C0 CNB R00,R20 match key?
3588 7A92 F7 07 JEQ GETKBD+ yes, go on
3589 7A94 CE FF FF JSB =SKPEAK no, try next
3590 7A97 F6 F2 MINZ until done
3591 7A99 F0 C3 JNP GETKBD+ not found, try again

```

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE  SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 85
>>>>>> on/off stuff <<<<<<<
3605 7B06 A1 BYT 241
3606 7B07 50 0A E2 VFTG. POBD R20,-R12 get flag
3607 7B0A 82 61 84 STND R20,-LRDSTS save it
3608 7B0D 9E RNZ
3609 7B0E
3610 7B0E A1 BYT 241
3611 7B0F 98 INDTG. BIN
3612 7B10 50 0A E2 POBD R20,-R12 for sure
3613 7B13 8A DCB R20 get flag
3614 7B14 AC ICB R20 complement
3615 7B15 82 84 83 STND R20,-10OUT? it
3616 7B18 9E RNZ save it
3617 7B19
3618 7B19 A1 BYT 241
3619 7B1A 98 I/RTG. BIN
3620 7B1B 50 0A E2 POBD R20,-R12 because
3621 7B1E 8A DCB R20 get flag
3622 7B1F AC ICB R20 complement
3623 7B20 82 5A 82 STND R20,-10BYTE it
3624 7B23 9E RNZ save it
3625 7B24
3626 7B24 43 06 E4 SET10B PUND R03,-R06 save reg
3627 7B27 80 5A 82 LDND R03,-10BYTE get flags
3628 7B2A CF 7F ANH R03,-7FH clear ours
3629 7B2C 02 94 ORB R03,R02 set new value
3630 7B2E 82 5A 82 STND R03,-10BYTE put flags
3631 7B31 06 E2 POBD R03,-R06 restore reg
3632 7B33 9E RNZ
3633 7B34

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 86
>>>>>>> NOISE stuff <<<<<<<<
3635 7B34
3636 7B34 60 81 64 84 GETSON LDM R40,=INPMH3 get pointers
3637 7B38 5C 24 81 LDM R34,R44 get value ptr
3638 7B38 1E FF FF JSB =FEINUM get value
3639 7B3E 70 0A E5 PUMD R60,=R12 number to stack
3640 7B41 9D CLE clear error flag
3641 7B42 5C 20 E3 RDM R34,R40 inc value ptr
3642 7B45 24 A3 STM R34,R44 save it
3643 7B47 22 A1 LDM R34,R42 get total len
3644 7B49 20 E5 SBN R34,R40 dec at
3645 7B4B F5 04 IFNG ran out,
3646 7B4D 1E FF FF BRASOM JSBN =ERR1+ report error
3647 7B50 5B BYT 91D
3648 7B51 EMOIF
3649 7B51 22 A3 STM R34,R42 save it
3650 7B53 60 83 64 84 STMD R40,=INPMH3 save pointers
3651 7B57 4F 90 TSB R17 error check
3652 7B59 F4 F2 JMG BRASOM set E flag
3653 7B5B 9E RTN
3654 7B5C
3655 7B5C 08 03 04 MRRYSZ BYT 8D,9D,4D
3656 7B5F
3657 7B5F A1 BYT 241
3658 7B60 98 NOISE. BIN
3659 7B61 1E FF FF JSB =FEISE1 get var info
3660 7B64 5E 24 A3 STM R36,R44 move value ptr
3661 7B67 5C 1C A5 LDM R34,R42 get total len
3662 7B6A 1F FF 7F RDM R34,=R7FFFH clear option base
3663 7B6D 22 A3 STM R34,R42 move len
3664 7B6F 60 83 64 84 STMD R40,=INPMH3 save it all
3665 7B73 66 1F 30 00 RDM R46,=R0030H mask out type
3666 7B77 99 BCD for shift
3667 7B78 86 LRD R46 shift down
3668 7B79 98 BIN back to bin
3669 7B7A 26 04 5C 7B LDM R46,R46,ARRYSZ get type size
3670 7B7E 83 64 84 STMD R46,=INPMH3 save it
3671 7B81 1E 34 7B JSB =GETSON get noise type
3672 7B84 F8 05 REN error out
3673 7B86 1E FF FF JSB =DNEB get it off stack
3674 7B89 66 91 TSN R46 what is it?
3675 7B8B F7 02 IFNZ fast,
3676 7B8D 08 01 LDB R46,=1 get a 1
3677 7B8F EMOIF
3678 7B8F B2 6C 84 STMD R46,=TMP3+8 save flag
3679 7B92 1E 34 7B JSB =GETSON get repeat count
3680 7B95 F8 14 REN error, quit
3681 7B97 64 81 66 84 LDM R44,=TMP3+2 get len/ptr
3682 7B9B 83 70 84 STMD R44,=(TMPH3+12D) save it
3683 7B9E 99 BCD for the integer
3684 7B9F 1E FF FF JSB =DNE1 get repeat
3685 7BA2 98 BIN back to normal
3686 7BA3 67 90 TSN R47 check for neg
3687 7BA5 F5 02 IFNG yes,

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 88
>>>>>>> TONE. beep a note for key down length <<<<<<<
3738 7C0F 10 2D BYT 20,55
3739 7C11 98 TONE. BIN We need it
3740 7C12 1E FF FF JSB =DNE1 Get the frequency
3741 7C15 52 92 CLB R22 Set no
3742 7C17 88 1CB R22 wait flag
3743 7C18 65 91 TSM R45 0?
3744 7C1A F6 03 IFZR Yes,
3745 7C1C 89 1CN R45 make it 1
3746 7C1D 52 92 CLB R22 clear no wait flag
3747 7C1F EMOIF
3748 7C1F 67 90 TSB R47 Negative?
3749 7C21 FC 03 IFLN Yes,
3750 7C23 65 93 1CN R45 make it 0
3751 7C25 89 1CN R45 make it 1
3752 7C26 EMOIF
3753 7C26 74 A9 FF 00 LDM R64,=OFFH,00C,83C,02C Get factor
3754 7C2A 83 02
3755 7C2E 70 0A E5 PUMD R60,=R12 Numbers to
3756 7C2F 60 E5 PUMD R40,=R12 stack
3757 7C31 52 06 E4 PUMD R22,=R06 Save wait flag
3758 7C34 1E FF FF JSB =INTDIV Do the divide
3759 7C37 1E FF FF JSB =DNEB Get the result
3760 7C3A 66 CD 10 00 SBN R46,=R16 Adjust for overhead
3761 7C3E F5 01 IFNG Too small,
3762 7C41 93 1CN R46 make it 0
3763 7C41 10 A3 EMOIF
3764 7C43 62 81 81 FF LDM R42,=RTC Move for later
3765 7C47 63 83 5B 84 STMD R43,=HDBRTC Get current time
3766 7C4B 00 92 CLB R00 Save for later
3767 7C4D B2 02 FF STBD R00,=KEYSTS Disable keyboard
3768 7C50 52 06 E2 PUMD R22,=R06 interrupts
3769 7C53 4F 90 TSB R17 Get wait flag
3770 7C55 F4 2B JMG TONEDM Any errors?
3771 7C57 42 80 80 82 LDM R02,=BEEPDK Yes, give up
3772 7C5B F7 25 JZR TONEDM No, give up
3773 7C5D 40 12 A0 LDB R00,R22 Get toggle bit
3774 7C60 F6 02 IFZR If not no wait
3775 7C62 50 93 1CN R20 make it fast
3776 7C64 EMOIF
3777 7C64 LOOP Do the beep
3778 7C64 BIN consume time to make
3779 7C64 98 BIN this loop same as beep
3780 7C65 98 JSB =STOP? abort condition?
3781 7C66 1E FF FF JEN TONEDM yes, give up
3782 7C69 F8 17 LDM R02,=CMPSB get current buzzer
3783 7C6B 42 80 80 FF RRB R02,R00 toggle bit
3784 7C6F 00 96 STBD R02,=CMPSB save new buzzer
3785 7C71 B2 80 FF LDM R02,R20 get cycle count
3786 7C74 10 A1 LOOP count loop
3787 7C76 DCM RN count down
3788 7C76 8B UNCV until done
3789 7C77 FB FD

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 87
>>>>>>> NOISE stuff <<<<<<<
3688 7B87 65 93 1CN R45 make it 0
3689 7B89 EMOIF
3690 7B89 LOOP repeat loop
3691 7B89 65 83 60 84 STMD R45,=IMP3+9 save count
3692 7B8D F7 0C RZR all done
3693 7B8F LOOP noise loop
3694 7B8F 1E 34 7B JSB =GETSON get period
3695 7B87 F8 07 REN error out
3696 7B84 34 7B JSB =GETSON get duration
3697 7B87 F8 02 REN error out
3698 7B89 42 80 6C 84 LDM R02,=IMP3+8 get noise type
3699 7B8B F6 19 normal,
3700 7B87 70 0A E3 PUMD R60,=R12 get duration
3701 7B82 78 E3 PUMD R70,=R12 get freq.
3702 7B84 70 91 TSN R75 0 freq?
3703 7B86 F6 07 IFZR yes,
3704 7B88 70 E5 PUMD R60,=R12 put duration
3705 7B8A 1E FF FF JSB =WAIT. wait it
3706 7B8D F0 07 ELSE no,
3707 7B8F 78 E5 PUMD R70,=R12 put freq.
3708 7B81 70 E5 PUMD R60,=R12 put duration
3709 7B83 1E FF FF JSB =BEEP. beep it
3710 7B86 EMOIF
3711 7B86 F0 1E ELSE fast,
3712 7B88 1E FF FF JSB =DNEB get duration
3713 7B8B 60 93 1CN R40 clear for load
3714 7B8D 7E 23 A3 STM R76,R43 move it
3715 7B8E 00 06 E5 PUMD R40,=R06 save it
3716 7B8E 1E FF FF JSB =DNEB get period
3717 7B86 60 06 E3 PUMD R40,=R06 get back duration
3718 7B89 7E 10 A3 STM R76,R20 move period
3719 7B8E F6 05 IFZR wait?
3720 7B8E 1E FF FF JSB =WCKJSB yes, do it
3721 7B8F 10 03 ELSE no,
3722 7B83 1E FF FF JSB =BEEPER beep it
3723 7B86 EMOIF
3724 7B86 REN stop it
3725 7B86 F8 16 LDM R02,=TMP3+2 get length
3726 7B88 42 81 66 84 UNH2 until done
3727 7B8C F6 81 LDM R44,=(TMPH3+12D) get len/ptr
3728 7B8F 64 81 70 84 STMD R44,=IMP3+2 reset en
3729 7C02 83 66 84 LDM R45,=TMP3+9 get count
3730 7C05 65 81 6D 84 BCD for dec
3731 7C09 99 DCM R45 dec count
3732 7C0B 8B BIN back to bin
3733 7C0B 98 UNH2 until done
3734 7C0C F0 9B RTN for conditional returns
3735 7C0E 9E
3736 7C0F

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&101 OBJ=RY&101 12/13/1983 11:50 AM PG 89
>>>>>>> TONE. beep a note for key down length <<<<<<<
3790 7C79 42 81 02 FF LDM R02,=KEYSTS get the keyboard status
3791 7C7B CF 20 00 RDM R02,=R0020H test the keydown bit
3792 7C80 F6 E2 UNH2 Still there, keep going
3793 7C82
3794 7C82 42 88 02 TONEDM LDB R02,=R0020H Enable keyboard
3795 7C85 B2 02 FF STBD R02,=KEYSTS interrupts
3796 7C88 80 80 FF LDM R02,=CMPSB Get buzzer
3797 7C8B F3 02 IFOD If on,
3798 7C8D 00 96 RRB R02,R00 turn it off
3799 7C8F EMOIF
3800 7C8F B2 80 FF STBD R02,=CMPSB Save new buzzer
3801 7C92 62 81 81 FF LDM R42,=RTC Get current time
3802 7C96 63 05 5B 84 SBN R43,=HDBRTC Compute duration
3803 7C9A 99 BCD For fast shift
3804 7C9B 67 87 LRM R47 Discard some low bits
3805 7C9D 98 BIN Back to binary
3806 7C9E (Max duration ~1.5 yrs
3807 7C9E resolution 1/1024th sec)
3808 7C9E 63 91 TSN R43 Test result
3809 7C9D F6 01 IFZR If too short,
3810 7CA2 89 1CN R43 make it 1 unit
3811 7CA3 EMOIF
3812 7CA3 5E 23 A1 LDM R36,R43 Get frac secs
3813 7CA6 CF FF 03 RDM R36,=R03FFFH Mask out garbage
3814 7CA9 67 87 LRM R47 Shift garbage out
3815 7CAB 87 LRM R47 of integer secs
3816 7CAC 64 06 E5 PUMD R44,=R6 Save integer secs
3817 7CAF 1E FF FF JSB =CONBIN Convert frac sec to fp
3818 7CB2 60 28 A3 STM R40,R50 Move for divide
3819 7CB5 A9 LDM R40,=
3820 7CB6 00 00 00 00 DEC 1024EO division factor
3821 7CB8 00 00 24 10
3822 7CBE 1E FF FF JSB =DIV10 Adjust to milliseconds
3823 7CC1 1E FF FF JSB =IP5 Truncate to millisec
3824 7CC4 1E FF FF JSB =DNEB Get result
3825 7CC7 60 28 A3 STM R40,R50 Adjust back to frac sec
3826 7CCA A9 LDM R40,=
3827 7CCB 03 00 00 00 DEC 1E3 division factor
3828 7CCF 00 00 00 10
3829 7CD3 1E FF FF JSB =DIV10 Do the adjust, result on stack
3830 7CD6 64 06 E3 PUMD R44,=R6 Get back integer sec
3831 7CD9 5E 24 A1 LDM R36,R44 Get small part (max=65535 sec)
3832 7CDE CF FF 7F RDM R36,=R7FFFH Kill top bit (max=32767 sec)
3833 7CE0 CF FF 7F JSB =CONBIN Convert int sec to fp
3834 7CE2 60 0A E5 PUMD R40,=R12 Onto stack for add
3835 7CE5 1E FF FF JSB =ADDR01 Add frac and int parts
3836 7CE8 9E RTN
3837 7CE9

```



```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RYZ101 12/13/1983 11:50 AM PG 90
-----
>>>>>> CLEAR SRO
3837 7CE9 A1 BYT 241
3838 7CEA 5E 80 8C 82 CLASR. LDBO R36,+DEVCHT get number of devices
3839 7CEE F7 F8 RZR none, leave
3840 7EFO LOOP status loop
3841 7EFO 98 BIN for sure
3842 7EFA 5E 0A E4 PUBD R36,+R12 save counter
3843 7EF4 5F 92 CLB R37 clear upper
3844 7EF6 0E 06 67 JSB =MUMOUT make it a number on stack
3845 7EF9 CE 40 71 JSB =MUMM find the name
3846 7EFE F8 EA REN error out
3847 7EFE CE 50 70 JSB get its status
3848 7D01 F8 E5 REN error out
3849 7D03 64 0A E3 PORD R44,-R12 trash result
3850 7D06 5E E2 PORD R36,-R12 get counter
3851 7D08 8A DCB R36 dec it
3852 7D09 F6 E5 MMWZ until done
3853 7D0B 9E RTN
3854 7D0C

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RYZ101 12/13/1983 11:50 AM PG 92
-----
>>>>>> SET ACCESS
3906 7D75 F7 04 IFNZ yes,
3907 7D77 CE FF FF JSBN =ERR1+ report error
3908 7D7A 7F BYT 63D
3909 7D7D ENDF
3910 7D7D CE FF FF JSB =FOPEN+ get file
3911 7D7E F8 EA REN error out
3912 7D80 50 18 84 04 LDBO R20,R30,DR.TYP get file type
3913 7D85 CE FF FF JSB =RESCOM get ram space
3914 7D88 08 BYT 8D
3915 7D89 F8 DF REN error out
3916 7D8B 6E 0A E5 PUMD R56,+R12 push length
3917 7D8E 56 E5 PUMD R26,+R12 push address
3918 7D90 60 89 52 58 LDI R40,+RKLPCNT+ get default
3919 7D94 45 4C 50 43
3918 7D98 4E 54 LDB R00,+40 load reg pointer
3919 7D9A 40 88 20 LOOP bit read loop
3920 7D9D TSB R20 test bit
3921 7D9D 50 90 IFPS clear
3922 7D9F F4 03 RDB R4,+R4 make lower case
3923 7DA1 41 0A 20 ENDF
3924 7DA4 LDB R20 move bits
3925 7DA4 50 84 ICB R00 inc pointer
3926 7DA6 40 88 CNB R00,+50 done?
3927 7DA8 C8 2B MMWZ until done
3928 7DAA FA F1 PUMD R40,+R26 push result
3929 7DAC 60 16 E5 RTN
3930 7DAF 9E
3931 7D80

```

NOMAS
 recipient NOT Manufacturer Supported
 agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RYZ101 12/13/1983 11:50 AM PG 91
-----
>>>>>> SET ACCESS
3856 7D0C 50 GETRES DRP 20 get the access
3857 7D0D CE FF FF JSB =GETADM strange
3858 7D10 52 0A E3 PORD R22,-R12 get size
3859 7D13 C9 08 00 CNB R22,-88 must be 8
3860 7D16 F7 04 IFNZ yes,
3861 7D18 CE FF FF BADSAC JSBN =ERR1+ error out
3862 7D1B 59 BYT 89D
3863 7D1C ENDF
3864 7D1C 70 89 52 58 LDI R60,+RKLPCNT+ get master pattern
3865 7D20 45 4C 50 43
3866 7D24 4E 54 LDBO R50,R20 get user pattern
3867 7D26 68 10 A5 SBM R50,R60 make R=0, r=20H, etc
3868 7D29 30 E5 CLB R44 clear for bits
3869 7D2B 64 92 LDB R45,+80H get bit mask
3870 7D2D 65 88 80 LDB R00,+50 get reg pointer
3871 7D33 LOOP bit loop
3872 7D33 A1 90 TSB R4 test flag
3873 7D35 12 02 IFNZ not set,
3874 7D37 C8 20 CNB R4,+20H clear?
3875 7D39 ENDF
3876 7D39 F6 DD JME BADSAC error out
3877 7D3B 90 TSB R4 was that set?
3878 7D3E 64 25 94 IFZR yes,
3879 7D41 ORB R44,R45 set the bit
3880 7D41 65 86 ENDF
3881 7D43 40 88 ICB R00 move bit mask
3882 7D45 C8 30 CNB R00,+60 inc reg ptr
3883 7D47 FA FA MMWZ at end?
3884 7D49 9E RTN until done
3885 7D4A BYT 241
3886 7D4A A1 SETAC. BIN for sure
3887 7D4B 98 JSB =GETRES get access bits
3888 7D4C CE 0C 7D REN error out
3889 7D4F F8 F8 PUBD R44,+R06 save access
3890 7D51 64 06 E4 JSB =GETNAM get file name
3891 7D54 CE FF FF PORD R24,-R06 get access
3892 7D57 5A 06 E2 REN error out
3893 7D5A F8 ED TSB R54 device?
3894 7D5C 6C 90 JNZ MODEV yes, error
3895 7D5E F6 58 JSB =FOPEN+ open file
3896 7D60 CE FF FF REN error out
3897 7D63 F8 E4 SIND R24,X30,DR.TYP set new access
3898 7D65 54 18 86 04
3899 7D6A 9E RTN
3900 7D6B BYT 30,56
3901 7D6B 18 2E ACES$. BIN for sure
3902 7D6D 98 JSB =GETNAM get filename
3903 7D6E CE FF FF REN error out
3904 7D71 F8 F7 TSB R54 device?
3905 7D73 6C 90

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I01 OBJ=RYZ101 12/13/1983 11:50 AM PG 93
-----
>>>>>> UPDATE
3933 7D80 A1 BYT 241
3934 7D81 98 UPDTE. BIN for sure
3935 7D82 CE FF FF JSB =GETNAM get the name
3936 7D85 F8 F8 REN error out
3937 7D87 6C 90 TSB R54 device?
3938 7D89 F7 04 IFNZ yes,
3939 7D8B CE FF FF MODEV JSBN =ERR1+ can't do that
3940 7D8E 3F BYT 63D
3941 7D8F ENDF
3942 7D8F CE FF FF JSB =FOPEN+ get the file
3943 7D92 F8 EB REN error, leave
3944 7D94 CE FF FF JSB =SYSJSB get the new
3945 7D97 FF FF DEF CRINDT date
3946 7D99 64 18 B7 06 SIND R44,X30,DR.DAT set it
3947 7D9C 00
3948 7D9F RTN

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RVX101 OBJ=RVX101 12/13/1983 11:50 AM PG 94
*****
>>>>>> CREATES *****
3950 70CF 80 4A 2E BYT 200,112,56
3951 70D2 98 CRE18 BIN for sure
3952 70D3 CE FF FF JSB =DHEB get size
3953 70D6 66 91 TSB R46 -?
3954 70D8 F5 04 IFNG yes,
3955 70DA CE FF EF MODCRT JSBH =ERR1+ error out
3956 70DD 59 BYT 89D
3957 70DE ENDOIF
3958 70DE CE 0C 7D JSB =GETACS get access
3959 70E1 F8 E8 REN error out
3960 70E3 50 DRP 20 get type
3961 70E4 CE FF FF JSB =GETADN string
3962 70E7 52 0A E3 POND R22,-R12 get size
3963 70EA 88 DCM R22 must be 1
3964 70EB F6 ED JNZ BRACRT no, error
3965 70ED 65 10 A4 LDBD R45,R20 get type
3966 70F0 64 06 E5 PUND R44,R06 save type,access,size
3967 70F3 CE FF FF JSB =GETNAM get the name
3968 70F6 7C 06 E3 POND R74,-R06 restore data
3969 70F9 F8 D3 REN error out
3970 70FB 6C 90 TSB R54 device?
3971 70FD F6 DD JNZ BRACRT yes, error
3972 70FF 5A 3E A1 LDM R32,R76 get size
3973 7E02 50 3C A1 LDM R20,R74 get type,access
3974 7E05 43 A8 02 LMB R03,*02H lined bit
3975 7E08 10 C7 ANM R03,R20 test it
3976 7E0A F6 11 IFZR no,
3977 7E0C CE FF FF JSB =FERRLD create file
3978 7E0F 58 18 A5 LDBD R30,R30 get space ptr
3979 7E12 F9 07 IFEM error,
3980 7E14 9B DUCRT DCE check no room
3981 7E15 F8 B7 REN yes, leave
3982 7E17 CE FF FF JSBH =ERR1+ report duplicate
3983 7E1A 40 BYT 64D
3984 7E1B ENDOIF
3985 7E1B F0 DD ELSE yes,
3986 7E1D CE FF FF JSB =FERRML create null file
3987 7E20 F8 F2 JEM DUPERT error out
3988 7E22 58 18 A5 LDBD R30,R30 get space ptr
3989 7E25 CE FF FF JSB =ALLDC get space
3990 7E28 F8 A4 REN error out
3991 7E2A ENDOIF
3992 7E2A 5E 18 A1 LDM R36,R30 get address of space
3993 7E2D CE 06 67 JSB =NUMDHT convert to M on stack
3994 7E30 CE FF FF JSB =R0NJSB convert to hex string
3995 7E33 F1 FF DEF DTH.
3996 7E35 64 40 DEF MYRON2
3997 7E37 9E RIM
3998 7E38

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RVX101 OBJ=RVX101 12/13/1983 11:50 AM PG 95
*****
>>>>>> TYPES *****
4000 7E38 10 2E BYT 20,56
4001 7E3A 98 TYPES BIN for sure
4002 7E3B CE FF FF JSB =I0? get the tofile
4003 7E3E 5E 06 E5 PUND R36,*R06 save address
4004 7E41 CE FF FF JSB =DHEI get the number
4005 7E44 5E 06 E3 POND R36,-R06 restore address
4006 7E47 67 90 TSB R47 test number
4007 7E49 F7 04 IFNZ bad,
4008 7E4B CE FF FF JSBH =ERR1+ report error
4009 7E4E 59 BYT 89D
4010 7E4F ENDOIF
4011 7E4F 2E 25 A1 LDM R76,R45 get number
4012 7E52 CE FF FF JSB =FSEK seek to it
4013 7E55 19 04 IFEM no there,
4014 7E57 CE FF FF JSBH =ERR1+ error out
4015 7E5A 2D BYT 45D
4016 7E5B ENDOIF
4017 7E5B 50 1E B4 0E LDBD R20,R36,TYPE get the type
4018 7E5F 00
4019 7E60 F7 06 IFNZ not ours,
4020 7E62 CE FF FF FILNAM JSB =HANDI let others
4021 7E65 4D VAL V.WEHN try it
4022 7E67 9E BYT 63D
4023 7E68 ENDOIF
4024 7E68 40 B4 02 00 LDBD R00,R36,LENM get the length
4025 7E6C C8 0C CMB R00,-MYLENN ok?
4026 7E6E F6 F2 JNZ FILNAM no, let others
4027 7E70 60 B5 03 00 LDM R40,R36,NAMEM get the filename
4028 7E74 56 A1 LDM R26,R36 move to addr
4029 7E76 CE FF FF JSB =FLOPEN open file
4030 7E79 F9 04 IFEM not there
4031 7E7B CE FF FF JSBH =ERR1+ error out
4032 7E7E 3E BYT 62D
4033 7E7F ENDOIF
4034 7E7F 50 18 B4 05 LDBD R20,R30,DR.LNM get file type
4035 7E83 00
4036 7E84 C8 42 CMB R20,-TYMBAS basic?
4037 7E86 F6 39 IFEQ yes,
4038 7E88 65 16 B5 0B LDM R45,R26,DATAM get pointers
4039 7E8C 00
4040 7E8D 35 A3 STM R45,R65 move em
4041 7E8F 88 ICB R45 move up one
4042 7E90 16 B7 0B 00 SHP R45,R26,DATAM save it
4043 7E94 CE FF FF JSB =R0NJSB go call other
4044 7E97 23 67 DEF (TYPE) rom to find
4045 7E99 65 40 DEF MYRON3 it
4046 7E9B 75 16 B7 0B SIMD R65,R26,DATAM restore old
4047 7E9F 00
4048 7E9F 09 0A IFEM error,
4049 7E9F CE FF FF JSB =CLERRA clear error
4050 7E9F 65 A9 45 4F LDM R45,*'EOF' at end of file
4051 7E9F 46

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RVX101 OBJ=RVX101 12/13/1983 11:50 AM PG 96
*****
>>>>>> TYPES *****
4048 7E9A F0 38 JMP GOTIYP finish up
4049 7E9C ENDOIF
4050 7E9C 50 1E A4 LDBD R20,R36 get token
4051 7E9D *****
4052 7E9F *****
4053 7E9F *****
4054 7E9F *****
4055 7E9F *****
4056 7E9F *****
4057 7E9F *****
4058 7E9F *****
4059 7E9F *****
4060 7E9F *****
4061 7E9F C8 05 CMB R20,*05H string?
4062 7E9F F6 07 IFEQ yes,
4063 7E9F 65 A9 53 54 LDM R45,*'STR' mark it
4064 7E9F 52 *****
4065 7E9F F0 05 ELSE LDM R45,*'NUM' no,
4066 7E9F 4D *****
4067 7E9F F0 23 ENDOIF no,
4068 7E9F 7D 16 B5 0B ELSE LDM R75,R26,DATAM get pointers
4069 7E9F 90 TSB R75 move up?
4070 7E9F F7 04 IFNZ yes,
4071 7E9F 99 BCD for anc
4072 7E9A 7E B9 ICM R76 next line
4073 7E9C 98 BIN normal
4074 7E9D ENDOIF
4075 7E9D CE FF FF JSB =FSEK get line
4076 7E9D 42 1E A5 LDM R02,R36 get line number
4077 7E9D C9 99 A9 CMB R02,*ENDLIN at end?
4078 7E9D F6 07 IFEQ yes,
4079 7E9D 65 A9 45 4F LDM R45,*'EOF' mark it
4080 7E9D F0 05 ELSE no,
4081 7E9D 65 A9 53 54 LDM R45,*'STR' must be string
4082 7E9D 52 *****
4083 7E9D *****
4084 7E9D CE FF FF GOTIYP JSB =RESCON get some ran
4085 7E9F 03 BYT 3D for result
4086 7E9F F8 08 REN error out
4087 7E9A 6E 0A E5 PUND R56,*R12 len to stack
4088 7E9D 56 E5 PUND R26,*R12 addr to stack
4089 7E9F 65 16 E5 PUND R45,*R26 result out
4090 7E9F 2E 9E RIM
4091 7E9F 3

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RVX101 OBJ=RVX101 12/13/1983 11:50 AM PG 97
*****
>>>>>> INPUTS *****
4093 7E9F 18 2E BYT 30,56
4094 7E9F 98 LIMB BIN for sure
4095 7E9F 54 DRP 24 get string
4096 7E9F CE FF FF JSB =GETADN addr
4097 7E9A 52 0A E3 POND R22,-R12 and length
4098 7E9D C9 08 00 CMB R22,*88 could be special?
4099 7E9D F6 5E IFEQ yes,
4100 7E9D 60 14 A5 LDM R40,R24 get the string
4101 7E9D C9 52 6F 6F CMB R40,*'Roo-man!' ours?
4102 7E9D 21 *****
4103 7E9D F6 0F IFEQ yes,
4104 7E9D 6E A9 19 00 LDM R56,*'(WHOWER-COPYRT) get size
4105 7E9D CE FF FF JSB =RSNEM- get some ran
4106 7E9D F8 09 REN error out
4107 7E9D F0 32 LDM R20,*COPYRT get pointer
4108 7E9D C9 A4 6F 65 ELSE no,
4109 7E9D 73 21 3F 21 CMB R40,*'Joey'?'?' another one?
4110 7E9D 3F *****
4111 7E9D F6 0F IFEQ yes,
4112 7E9D 6E A9 1C 00 LDM R56,*'(ENDFUN-FORPPE) get size
4113 7E9D CE FF FF JSB =RSNEM- get some ran
4114 7E9D F8 0F REN error out
4115 7E9D 50 A9 E0 7F LDM R20,*FORPPE get pointer
4116 7E9D F0 18 ELSE no,
4117 7E9D 39 5A 61 72 CMB R40,*'Zarqoon.' last chance
4118 7E9D 71 75 6F 6E *****
4119 7E9D 2E *****
4120 7E9D F6 1C JWE WHRLIN no, go on
4121 7E9D 6E A9 20 00 LDM R56,*'(FORPPE-WHOWER) get size
4122 7E9D CE FF FF JSB =RSNEM- get some ran
4123 7E9D F8 05 REN error out
4124 7E9D 50 A9 C0 7F LDM R20,*WHOWER get pointer
4125 7E9D *****
4126 7E9D ENDOIF
4127 7E9D PUND R56,*R12 size to stack
4128 7E9D 56 E5 PUND R26,*R12 addr to stack
4129 7E9D *****
4130 7E9D 42 10 E0 LDBD R02,*R20 move loop
4131 7E9D 16 E4 PUND R02,*R26 get char
4132 7E9D 6E 88 DCM R56 dec length
4133 7E9D F6 F7 WHMZ until done
4134 7E9D 9E RIM
4135 7E9D *****
4136 7E9D ENDOIF
4137 7E9D *****
4138 7E9D *****
4139 7E9D *****
4140 7E9D *****
4141 7E9D *****
4142 7E9D *****
4143 7E9D *****
4144 7E9D *****
4145 7E9D *****
4146 7E9D *****
4147 7E9D *****
4148 7E9D *****
4149 7E9D *****
4150 7E9D *****
4151 7E9D *****
4152 7E9D *****
4153 7E9D *****
4154 7E9D *****
4155 7E9D *****
4156 7E9D *****
4157 7E9D *****
4158 7E9D *****
4159 7E9D *****
4160 7E9D *****
4161 7E9D *****
4162 7E9D *****
4163 7E9D *****
4164 7E9D *****
4165 7E9D *****
4166 7E9D *****
4167 7E9D *****
4168 7E9D *****
4169 7E9D *****
4170 7E9D *****
4171 7E9D *****
4172 7E9D *****
4173 7E9D *****
4174 7E9D *****
4175 7E9D *****
4176 7E9D *****
4177 7E9D *****
4178 7E9D *****
4179 7E9D *****
4180 7E9D *****
4181 7E9D *****
4182 7E9D *****
4183 7E9D *****
4184 7E9D *****
4185 7E9D *****
4186 7E9D *****
4187 7E9D *****
4188 7E9D *****
4189 7E9D *****
4190 7E9D *****
4191 7E9D *****
4192 7E9D *****
4193 7E9D *****
4194 7E9D *****
4195 7E9D *****
4196 7E9D *****
4197 7E9D *****
4198 7E9D *****
4199 7E9D *****
4200 7E9D *****

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RV8101 OBJ=RVZ101 12/13/1983 11:50 AM PG 98
=====
>>>>>> LINPUTS <<<<<<<<
4140 7F6F 50 A9 80 B1 LDM R20,=INPBUF get buffer pointer
4141 7F73 LOOP move loop
4142 7F73 40 8A DCL R00 dec size
4143 7F75 F4 07 JNG IMPLIN done, go on
4144 7F77 42 14 E0 P0BD R02,=R24 get char
4145 7F7A 10 E4 PUBD R02,=R20 put char
4146 7F7C F0 F5 WHNP
4147 7F7E
4148 7F7E 54 12 A0 IMPLIN LDB R24,R22 get size
4149 7F81 50 A9 80 B1 LDM R20,=INPBUF get start
4150 7F85 R3 SIM R20,R22 put cursor
4151 7F86 CE FF FF JSB =GETREP get input
4152 7F89 6E 93 CLJ R56 clear for load
4153 7F8B 14 A0 LDB R56,R24 get the length
4154 7F8D CE FF FF JSB =SYSJSB reserve the room
4155 7F90 FF FF DEF RESPUT and info to stack
4156 7F92 F8 CB REN error out
4157 7F94 40 A9 80 B1 LDM R00,=INPBUF get the pointer
4158 7F98 LOOP move loop
4159 7F98 54 8A DCL R24 dec count
4160 7F9A F4 07 JNG LINDPW quit if done
4161 7F9C 42 00 E0 P0BD R02,=R00 get character
4162 7F9F 16 E4 PUBD R02,=R26 put character
4163 7FA1 F0 F5 WHNP until done
4164 7FA3 CE FF FF LINDPW JSB =BLIMP clean up
4165 7FA6 9C RTN
4166 7FA7
4167 7FA7
4168 7FA7 43 6F 70 79 COPYAT ASC "Copyright (c) [(hp)] 1983"
4168 7FAB 72 69 67 68
4168 7FAC 74 20 28 63
4168 7FAD 29 20 58 28
4168 7FAE 68 20 29 50
4168 7FAB 20 31 39 38
4168 7F8F 33
4169 7F80 4D 54 53 3A ANSWER ASC "ANS: Raan Young, PH: Ball Wickers"
4169 7F84 20 52 61 61
4169 7F86 6E 20 59 6F
4169 7F8C 75 6E 67 2C
4169 7F90 20 50 4D 3A
4169 7F94 20 42 69 6C
4169 7F98 6C 20 57 69
4169 7F9C 63 68 65 73
4170 7F80 52 6F 6F 2D FORPPC ASC "Roo-man says 'Welcome, PPC.'"
4170 7F84 6D 61 6E 20
4170 7F88 73 61 79 73
4170 7F8C 20 27 57 65
4170 7F90 6C 63 6F 6D
4170 7F94 65 2C 20 50
4170 7F98 50 43 2E 27
4171 7F7C ENDFUN
4172 7F7C
4173 7F7C LEWEND

```

```

KARMA 08/31/82=====
SYMBOL VALUE TYPE COUNT Symbol Table 12/13/1983 11:50 AM PG100
=====
0000CH 000C EQU 1
0001FH 001F EQU 1
00020H 0020 EQU 1
00030H 0030 EQU 1
000C0H 00C0 EQU 1
003FFH 03FF EQU 1
016 0010 EQU 4
018 0012 EQU 3
02 0002 EQU 3
0256 0100 EQU 1
03 0003 EQU 1
030 001E EQU 1
04 0004 EQU 4
043A0H 43A0 EQU 1
05 0005 EQU 3
06000H 6000 EQU 2
064A0H 64A0 EQU 1
07 0007 EQU 2
077FFH 77FF EQU 2
08 0008 EQU 3
08000H 8000 EQU 2
086 0056 EQU 1
09 0009 EQU 1
09+1 003A EQU 1
094 005E EQU 1
0:-0 000A EQU 1
0A-0 0011 EQU 1
0A-: 0007 EQU 1
0C0FFH 10FF EQU 1
0E001H 0001 EQU 1
0E058H 0058 EQU 2
0FF80H 0F80 EQU 1
0FFE0H 0FE0 EQU 4
100001 628E LCL 1
ADAMS6 70D7 LCL 1
ACES4 706D LCL 1
ADDR01 FFFF EXT 2
ADDRS 7024 LCL 1
ALFAM FFFF EXT 3
ALLOC FFFF EXT 1
APROP+ 798F LCL 6
APROPV FFFF EXT 1
ARRYSZ 7B5C LCL 1
ASCLEK 6103 LCL 1
ASGNID 0092 EQU 1
ASLP4 6DEB LCL 1
ASLPDM 6E14 LCL 1
ASMLP 6DE1 LCL 2
ATOTG 6F73 LCL 1
AUTODM 62CD LCL 2
AUDLP 844A G DAD 4
BACK3 6D16 LCL 7
BACK3- 6D17 LCL 1
BACKRI 70DA LCL 2

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RV8101 OBJ=RVZ101 12/13/1983 11:50 AM PG 99
=====
>>>>>> LINPUTS <<<<<<<<
4174 7F7C FIN

```

```

KARMA 08/31/82=====
SYMBOL VALUE TYPE COUNT Symbol Table 12/13/1983 11:50 AM PG101
=====
BADDGT 698A LCL 3
BADENI 6581 LCL 1
BADING 7368 LCL 2
BROKBO 798A LCL 1
BROSAC 7018 LCL 1
BROSAT 6E66 LCL 1
BROSDM 784D LCL 1
BROIDM 6033 LCL 1
BROIKM 66B2 LCL 3
BRNO 66E3 LCL 1
BRNP 6701 LCL 1
DEEP FFFF EXT 1
DEEPER FFFF EXT 2
DEEPLM 0004 EQU 1
DEEPOK 82A0 G DAD 1
DEOR 66ED LCL 1
BIMDGI 6972 LCL 3
BIMEND 6706 LCL 3
BIMSP 6713 LCL 1
BIMSUP 670D LCL 3
BJOR 66F7 LCL 1
BLANKS FFFF EXT 2
BLIMP FFFF EXT 2
BSEADM 829E G DAD 3
CHARGE 6E0E LCL 2
CHARGE FFFF EXT 1
CHARY 6385 LCL 1
CHMSEM 6694 LCL 1
CHMIRM 73B2 LCL 1
CHSKOI FFFF EXT 1
CLEAM+ 6E22 LCL 3
CLEAM- FFFF EXT 1
CLEM+ 6E82 LCL 1
CLEM+ 6E8A LCL 1
CLEM- 6E27 LCL 1
CLRDME 633C LCL 1
CLXERR FFFF EXT 5
CLXOFF FFFF EXT 2
CLXSR 7CEA LCL 1
CLSTND 694F LCL 1
CND 7189 LCL 1
CNOREP FFFF EXT 2
CNOSCH 6C46 LCL 1
CNOSMD FFFF EXT 1
CNOTBI 690D LCL 1
CNOK 001B EQU 2
CNPSB FF80 G DAD 4
CONRA 002C G EQU 2
CONBIM FFFF EXT 3
CONCA- FFFF EXT 1
COPYRT 7FA7 LCL 2
CPROK 7822 LCL 1
CR 000D G EQU 2
CRDSTS 8461 G DAD 1

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 12/13/1983 11:50 AM PG114
if0223	710A	LCL	1	
if0224	7113	LCL	1	
if0226	7133	LCL	1	
if0230	717E	LCL	1	
if0231	71B0	LCL	1	
if0232	71BF	LCL	1	
if0233	71B7	LCL	1	
if0234	71BA	LCL	1	
if0239	7299	LCL	1	
if0240	728D	LCL	1	
if0241	7294	LCL	1	
if0242	729B	LCL	1	
if0243	7286	LCL	1	
if0246	72F0	LCL	1	
if0247	72F0	LCL	1	
if0248	72F0	LCL	1	
if0249	72F0	LCL	1	
if0250	731B	LCL	1	
if0251	730B	LCL	1	
if0252	7319	LCL	1	
if0253	7319	LCL	1	
if0254	7357	LCL	1	
if0255	7357	LCL	1	
if0258	736C	LCL	1	
if0260	737B	LCL	1	
if0261	737B	LCL	1	
if0262	73B1	LCL	1	
if0265	73D9	LCL	1	
if0266	73D3	LCL	1	
if0268	73EF	LCL	1	
if0270	7414	LCL	1	
if0271	7425	LCL	1	
if0273	74BA	LCL	1	
if0274	7456	LCL	1	
if0275	748E	LCL	1	
if0276	7496	LCL	1	
if0277	74AD	LCL	1	
if0279	74BF	LCL	1	
if0280	74D5	LCL	1	
if0281	74DF	LCL	1	
if0282	74F9	LCL	1	
if0283	74E8	LCL	1	
if0284	74F9	LCL	1	
if0285	74F1	LCL	1	
if0286	74F9	LCL	1	
if0287	7501	LCL	1	
if0290	7567	LCL	1	
if0291	754B	LCL	1	
if0292	7550	LCL	1	
if0293	7558	LCL	1	
if0294	7564	LCL	1	
if0295	758E	LCL	1	
if0296	7599	LCL	1	
if0297	759F	LCL	1	

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 12/13/1983 11:50 AM PG116
if0365	7866	LCL	1	
if0367	78A7	LCL	1	
if0368	78C7	LCL	1	
if0371	790B	LCL	1	
if0372	7936	LCL	1	
if0373	791B	LCL	1	
if0374	7933	LCL	1	
if0375	7941	LCL	1	
if0377	7970	LCL	1	
if0378	7979	LCL	1	
if0380	79AF	LCL	1	
if0381	79AF	LCL	1	
if0382	79BE	LCL	1	
if0383	79D7	LCL	1	
if0384	79D5	LCL	1	
if0385	79D9	LCL	1	
if0388	7A55	LCL	1	
if0391	7A93	LCL	1	
if0392	7ABA	LCL	1	
if0393	7AB2	LCL	1	
if0394	7AFC	LCL	1	
if0395	7AFC	LCL	1	
if0397	7B51	LCL	1	
if0399	7BBF	LCL	1	
if0401	7BA9	LCL	1	
if0404	7BD8	LCL	1	
if0405	7BCF	LCL	1	
if0406	7BD6	LCL	1	
if0407	7BF6	LCL	1	
if0408	7BF3	LCL	1	
if0409	7BF6	LCL	1	
if0412	7C1F	LCL	1	
if0413	7C26	LCL	1	
if0414	7C41	LCL	1	
if0415	7C64	LCL	1	
if0418	7C8F	LCL	1	
if0419	7CA3	LCL	1	
if0422	7D1C	LCL	1	
if0424	7D39	LCL	1	
if0425	7D41	LCL	1	
if0428	7D7B	LCL	1	
if0430	7DAA	LCL	1	
if0432	7DBF	LCL	1	
if0433	7DDE	LCL	1	
if0435	7E1D	LCL	1	
if0436	7E1B	LCL	1	
if0437	7E2A	LCL	1	
if0438	7E4F	LCL	1	
if0439	7E5B	LCL	1	
if0440	7E6B	LCL	1	
if0441	7E7F	LCL	1	
if0442	7EC1	LCL	1	
if0443	7EAC	LCL	1	
if0444	7EBA	LCL	1	

RECEIVED
FBI
FEDERAL BUREAU OF INVESTIGATION
U.S. DEPARTMENT OF JUSTICE
WASHINGTON, D.C. 20535
JAN 1 1984

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 12/13/1983 11:50 AM PG115
if0298	75B5	LCL	1	
if0299	75B5	LCL	1	
if0300	75E6	LCL	1	
if0301	75C7	LCL	1	
if0302	75D8	LCL	1	
if0303	760E	LCL	1	
if0304	760E	LCL	1	
if0305	7667	LCL	1	
if0306	764B	LCL	1	
if0307	766E	LCL	1	
if0310	767B	LCL	1	
if0311	7687	LCL	1	
if0312	7693	LCL	1	
if0313	769F	LCL	1	
if0316	76CB	LCL	1	
if0317	76CB	LCL	1	
if0318	76DC	LCL	1	
if0319	76DA	LCL	1	
if0320	76E5	LCL	1	
if0321	76E5	LCL	1	
if0322	76F0	LCL	1	
if0323	7700	LCL	1	
if0324	7700	LCL	1	
if0326	7723	LCL	1	
if0327	771E	LCL	1	
if0329	777B	LCL	1	
if0330	7743	LCL	1	
if0331	7741	LCL	1	
if0332	777B	LCL	1	
if0333	7751	LCL	1	
if0334	7779	LCL	1	
if0335	7775	LCL	1	
if0337	7763	LCL	1	
if0339	7792	LCL	1	
if0340	779A	LCL	1	
if0341	77B0	LCL	1	
if0345	77BE	LCL	1	
if0346	77BB	LCL	1	
if0347	77C4	LCL	1	
if0348	7833	LCL	1	
if0349	77DB	LCL	1	
if0350	77ED	LCL	1	
if0351	7826	LCL	1	
if0352	7826	LCL	1	
if0354	780B	LCL	1	
if0355	780B	LCL	1	
if0356	7818	LCL	1	
if0357	7816	LCL	1	
if0358	781F	LCL	1	
if0359	781F	LCL	1	
if0361	7840	LCL	1	
if0362	784F	LCL	1	
if0363	7856	LCL	1	
if0364	7862	LCL	1	

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 12/13/1983 11:50 AM PG117
if0445	7EBF	LCL	1	
if0446	7EE4	LCL	1	
if0447	7ECD	LCL	1	
if0448	7EDF	LCL	1	
if0449	7EE4	LCL	1	
if0451	7F60	LCL	1	
if0452	7F1F	LCL	1	
if0453	7F51	LCL	1	
if0454	7F39	LCL	1	
if0455	7F51	LCL	1	
if0457	7F6D	LCL	1	
if0011	630E	LCL	1	
if0015	636D	LCL	1	
if0027	6412	LCL	1	
if0029	6419	LCL	1	
if0037	646A	LCL	1	
if0040	6484	LCL	1	
if0053	65BF	LCL	1	
if0063	6672	LCL	1	
if0102	687E	LCL	1	
if0123	6993	LCL	1	
if0125	69B8	LCL	1	
if0129	6A6F	LCL	1	
if0132	6B3A	LCL	1	
if0134	6B59	LCL	1	
if0147	6BFO	LCL	1	
if0149	6C04	LCL	1	
if0150	6C3A	LCL	1	
if0164	6CCD	LCL	1	
if0175	6D44	LCL	1	
if0183	6E02	LCL	1	
if0225	712C	LCL	1	
if0229	7150	LCL	1	
if0235	71C7	LCL	1	
if0237	7203	LCL	1	
if0238	7228	LCL	1	
if0245	7206	LCL	1	
if0259	736E	LCL	1	
if0264	73C2	LCL	1	
if0278	74B5	LCL	1	
if0289	7535	LCL	1	
if0336	7757	LCL	1	
if0353	77FB	LCL	1	
if0360	7837	LCL	1	
if0366	7897	LCL	1	
if0369	78CF	LCL	1	
if0389	7A5C	LCL	1	
if0390	7A7D	LCL	1	
if0396	7ABD	LCL	1	
if0402	7BA9	LCL	1	
if0403	7BAF	LCL	1	
if0416	7C64	LCL	1	
if0417	7C76	LCL	1	
if0421	7CFO	LCL	1	

```

KARMA 08/31/82-----
SYMBOL VALUE TYPE COUNT Symbol Table 12/13/1983 11:50 AM PG118
-----
100423 7D33 LCL 1
100429 7D90 LCL 1
100456 7F56 LCL 1
100458 7F73 LCL 1
100460 7F98 LCL 1
rtn002 62D3 LCL 6
rtn014 633F LCL 1
rtn021 63B5 LCL 2
rtn059 660F LCL 2
rtn066 666E LCL 1
rtn080 6762 LCL 1
rtn121 694E LCL 1
rtn126 6989 LCL 1
rtn127 6B32 LCL 3
rtn151 6C38 LCL 1
rtn158 6E45 LCL 2
rtn176 6D1B LCL 2
rtn178 6D6D LCL 1
rtn182 6DA7 LCL 2
rtn214 6FA8 LCL 2
rtn217 7002 LCL 2
rtn218 7021 LCL 1
rtn219 703B LCL 1
rtn220 7064 LCL 1
rtn227 711A LCL 1
rtn228 714A LCL 2
rtn236 71BF LCL 3
rtn257 7341 LCL 2
rtn263 73B1 LCL 2
rtn267 73D8 LCL 3
rtn269 73F3 LCL 2
rtn272 7432 LCL 3
rtn288 7566 LCL 2
rtn308 7637 LCL 1
rtn315 76E5 LCL 1
rtn328 76F0 LCL 2
rtn338 779A LCL 1
rtn370 78DE LCL 1
rtn376 794D LCL 1
rtn379 797D LCL 1
rtn386 79E9 LCL 2
rtn398 7B5B LCL 5
rtn410 7C0E LCL 1
rtn420 7CE8 LCL 3
rtn426 7D49 LCL 3
rtn427 7D6A LCL 3
rtn431 7D8F LCL 2
rtn434 7DCE LCL 4
rtn450 7EF2 LCL 4
rtn459 7F5F LCL 1

```

```

-----
KLINK load map size size RYMI02 11:57 AM TUE., 13 DEC., 1983
file start end hex/oct decimal version message page 1
-----
RYZIO2 6000 7FFE 1FFE 8190
KXZLAK 7FFE 7FFE 0000 0

```

total size=1FFE = 8190 decimal

NOT Manufacturer Supported
 Recipient agrees NOT to contact manufacturer
NOMMAS

HEADING	Table of Contents	PAGE	PG119
exts, ents, equs, etc	1		
ROM tables	6		
HARDI call processing	11		
Parse routines	17		
bit twiddling stuff	23		
goto/gosub x	24		
shell on/off toggle	25		
EWI04 enter a string	26		
command frame table	32		
SENDIO to pil device	35		
device/command string process	36		
search through pil command tables	39		
PROSCH search for command string	40		
locate numeric suffixes to PIL commands	44		
Auto assign loop	45		
AUTOLOOP ON/OFF	50		
WIO/RIO	51		
LOCAL, LOCAL LOCKOUT, ETC	52		
DEVADDR, DEVNAME	55		
OUTPUT/OUTPUT USING	60		
ENTER USING image parsing	61		
ENTER support routines	71		
ENTER stuff	73		
IOSIZE processing	80		
keyboard is stuff	82		
on/off stuff	85		
NOISE stuff	86		
TOWE. beep a note for key down length	88		
CLEAR SRQ	90		
SET ACCESS	91		
UPDATE	93		
CREATE	94		
TYPE	95		
INPUTS	97		

RYZIO1 HAD 0 ERRORS 0 WARNINGS1179 LABELS SIZE 8188 LAST ERROR AT 0

```

-----
KLINK load map RYMI02 11:58 AM TUE., 13 DEC., 1983 page 2
-----
A.ZERO EAC7
ABORT FBED
ABSS 3A43
ABSS- 3A46
ABTHRD F8EA 6546
ACMD 35F2
ACREAT 7210
ADD15 3B0A
ADD20 3B0F
ADD9 3B04
ADDR F042
ADDR01 37D4 6A79 71B1 7D74 7D79 7D86
ADJUST 233F
BINCHK 67B0
ALARM 7430
ALBEEP 7233
ALCALL IF93
ALFA 0E02
ALFAW 0E03 6701 6725 71E0 7221 7571 76B7 79E4
ALFAM 0E03 79EC 79F4 7A02 7B94 7BCE
ALINI 2E39
ALLDC 1F9F 669A
ALMCHK 6C07
ALOSNV 27F9
ALVENT 28E8
ALWAYS E9D1
AMP 4E6A
AMN.A 440A
AMN.A- 4415
AMN.B 43E4
AMN.B- 43E9
AMN.E 43EE 6B05 6C03
AMN.E- 43F3
AMN.P 4400
AMN.P- 4405
AMN.MS 43E0
AMOTE 3607
AMVER 4B3A
AMVEN 7220
AMVEN 722A
AMDEL 6401
AMENIT 66BA
AMFLO 09EF
AMFILE 09F7
AMFNO 644E
AMFNO 6361
AMMSKE 6537
AMMSKY 653C
AMPROC 72B1
AMPT 4993
AMPTND 65E4
AMPTS 7229
AMPIIN 6552
AMROPN 64C6
AMSTAT 7300

```


KLINK load map RYH102 11:58 AM TUE., 13 DEC., 1983 page 3

NPTRK	7318				
NPTRK	6864				
NPDEL	63F2				
NPIDSP	6418				
NPERR	642C				
NPEND	644A				
NPGET	646C				
NPINS	6482				
NPTRG	7350				
NPTR	6484				
NPTR	6482				
NPTRIG	7383				
NRD0A	13ED				
NRTH	F3A9				
NRWAY	1058				
NRSCIS	6069				
NRSCAM	2E20				
NRSCMI	2E29				
NRSMIT	664C				
NRSHPR	49CF				
NRSPCK	64D7				
NRSSIM	1303				
NRSSIM	E4BF				
NRSTCD	742D				
NRTHPLT	650D				
NRTH2	74F4				
NRTH2	7501				
NRTH2	E2ED	7F79			
NRTH2	084B				
NRVADR1	5666				
NRVADR2	5668				
NRVALL	2803				
NRVALL	567E				
NRVALL2	5683				
NRVALL	09C9				
NRVALL	064C				
NRVALL	0E2D				
NRVALL	6AD8	66A0			
NRVALL	209A				
NRVALL	5245				
NRVALL	E002				
NRVALL	358C				
NRVALL	2E77				
NRVALL	438D				
NRVALL	FED4				
NRVALL	FED1				
NRVALL	706C				
NRVALL	FED7				
NRVALL	7DBB				
NRVALL	2EDA				
NRVALL	09FB	6348	648C	77BD	7A4B 7E56
NRVALL	48CA	6E0D			
NRVALL	4100	656C	66A5	72AA	7307
NRVALL	5FAE				
NRVALL	0862				

KLINK load map RYH102 11:58 AM TUE., 13 DEC., 1983 page 5

NRVALL	040A				
NRVALL	40AA				
NRVALL	04B9				
NRVALL	7448				
NRVALL	E781				
NRVALL	018A	6846	6E3A	6F0C	7060 71AB 71C2
NRVALL	7C3D				
NRVALL	7C32				
NRVALL	58D7				
NRVALL	1037				
NRVALL	51CC				
NRVALL	22E7				
NRVALL	1A43	7891			
NRVALL	35E5				
NRVALL	39D5				
NRVALL	107F				
NRVALL	F6CA	65AE			
NRVALL	F696				
NRVALL	F664				
NRVALL	F66C				
NRVALL	455D				
NRVALL	FA2C	65ED			
NRVALL	7528				
NRVALL	009D				
NRVALL	398A				
NRVALL	39D1				
NRVALL	676E				
NRVALL	EFBF				
NRVALL	06BF	63C4			
NRVALL	06C4	63CD			
NRVALL	EFB9				
NRVALL	F35A				
NRVALL	4848				
NRVALL	2C93				
NRVALL	2C9C				
NRVALL	159C				
NRVALL	60D8				
NRVALL	6027				
NRVALL	7C74				
NRVALL	7C83				
NRVALL	EC24				
NRVALL	6787				
NRVALL	700A				
NRVALL	7008				
NRVALL	6DAD				
NRVALL	6E23				
NRVALL	0A56				
NRVALL	6E0E				
NRVALL	6930				
NRVALL	6E3F				
NRVALL	32AB				
NRVALL	3E03				
NRVALL	6FFC	7838	78E4		
NRVALL	67B2				
NRVALL	7001				

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

KLINK load map RYH102 11:58 AM TUE., 13 DEC., 1983 page 4

NRVALL	7549				
NRVALL	7541				
NRVALL	069F				
NRVALL	06A1				
NRVALL	E022				
NRVALL	0889				
NRVALL	EAA8				
NRVALL	2EFF				
NRVALL	540F				
NRVALL	50EE				
NRVALL	514F				
NRVALL	0044				
NRVALL	498B				
NRVALL	7E80				
NRVALL	593B				
NRVALL	58DA				
NRVALL	590D				
NRVALL	7EE8				
NRVALL	7F62	74E6			
NRVALL	7F88				
NRVALL	58EB				
NRVALL	7F5E				
NRVALL	180B				
NRVALL	37A8				
NRVALL	5AFD				
NRVALL	5C02				
NRVALL	087A				
NRVALL	78E4				
NRVALL	5E8D				
NRVALL	42F4				
NRVALL	FC7A				
NRVALL	378D	71DB			
NRVALL	6772				
NRVALL	5E4F				
NRVALL	58F2				
NRVALL	7186				
NRVALL	2C57				
NRVALL	5E46				
NRVALL	6A93	790A			
NRVALL	509E				
NRVALL	48C2	6952	7638		
NRVALL	7805				
NRVALL	EFA4				
NRVALL	5E18				
NRVALL	603F				
NRVALL	EC1F	7985			
NRVALL	E815				
NRVALL	052C				
NRVALL	04F7				
NRVALL	04EA				
NRVALL	05D1	7F73			
NRVALL	05BF				
NRVALL	05B5				

KLINK load map RYH102 11:58 AM TUE., 13 DEC., 1983 page 6

NRVALL	2F4D				
NRVALL	2F46				
NRVALL	EFE8				
NRVALL	13E1				
NRVALL	4E8E				
NRVALL	4EC8				
NRVALL	79FB				
NRVALL	36A8				
NRVALL	38D8				
NRVALL	7D3F				
NRVALL	1C01				
NRVALL	1FB3	6623	667B		
NRVALL	21F3				
NRVALL	222D				
NRVALL	07A3				
NRVALL	7830				
NRVALL	5260				
NRVALL	0E1C				
NRVALL	1358				
NRVALL	1373				
NRVALL	E38A				
NRVALL	376E				
NRVALL	3771				
NRVALL	376B	6A87	7100		
NRVALL	3786				
NRVALL	3744				
NRVALL	136A				
NRVALL	1365				
NRVALL	5919				
NRVALL	0121				
NRVALL	E3A3				
NRVALL	58FC				
NRVALL	F01C				
NRVALL	516A				
NRVALL	58FE				
NRVALL	45BF				
NRVALL	0A4B				
NRVALL	0A4C				
NRVALL	7547				
NRVALL	7816				
NRVALL	6E7B				
NRVALL	49CA				
NRVALL	525A				
NRVALL	74AF				
NRVALL	5293				
NRVALL	4482				
NRVALL	4486				
NRVALL	1986				
NRVALL	597A				
NRVALL	EC55				
NRVALL	6A2A				
NRVALL	5204				
NRVALL	E30E	744E			
NRVALL	5038				
NRVALL	00CC				

NOTAS
 Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

NOTAS
 Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

NOTAS
 Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

NOTAS
 Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

KLINK load map RYU102 11:58 AM TUE., 13 DEC., 1983 page 11

MLFOUT	0706		
IMPUSH	570A		
I/ROFF	7CE6		
ICONS1	54CE		
ICOS.	7535		
IDYSND	EB0D		
IF	16D5		
ILE1	1644		
IMEMV?	53EE		
IMF10	39C8	6908	6B42
IMF02	3A7C		
IMF03	3A4E		
IMF04	3A51		
IMF09	3A90		
IMIPR-	0B06		
IMIPRS	0B02		
IMISIZ	675D		
IM11.	666B		
IM11GL	00F5		
IMPUR.	4373		
IMPUN.	4309		
INPUT	1557		
INPUT.	4276		
INSERI	1FFC		
INSRIM	23CE		
INTS	3A6B		
INTCOM	52D7		
INTDEV	3A6F	6CB2	73FE
INTEGR	0D16		
INTMUL	394E		
INTROL	3E16		
INTRP	00BE		
INWPOP	ED23		
INVRIN	ED04		
INWIND	EFFB		
ID.OFF	F015		
ID.OH	F00C		
ID?	E99F	693D	7FCB
IDFILE	09C7		
IDOPEN	218E		
IPS	3A68	733F	7367
ISCWA?	1075		
ISEIUP	4305		
ISIN.	752D		
ITAIL.	430F		
ITIN.	753D		
ITHEH	16D7		
IZOPEN	2199		
JAPLHW	EC54		
JAPSD8	EC44		
JSDCRT	702D		
JIRUEN	EC35		
KEY.	FC60	754B	
KEY?	0B54		
KEYOSP	419A		

recipient agrees NOT to contact manufacturer

NOVMS

KLINK load map RYU102 11:58 AM TUE., 13 DEC., 1983 page 13

MLJSB	4661	78D3	78E2	78F2	7908	7917	7928	7934
MLJSB	4661	7953	795A	7962	798F	79A5		
MEN.	FCC6							
NERCE.	5339							
NIN10	3025							
NIMDD	6FAB							
NIMHH	6FAE							
NIMMH	6FAB							
NIMMH	6FBD							
NINYY	6FA6							
NKEPRV	F6F0							
MOD10.	7705	73E0						
MODE?	64AB							
MODEKY	6472							
MOYE	2270							
NOVEIT	5EAE							
NOVENV	2C16							
NOVEUP	2218	630F						
NPY28	38E2							
NPY30	38E5							
NPYR?	7F9A							
NPYROI	393D	6A76	6F12	718F	735D			
NRGIN.	FC01							
NSGOUT	07E3							
MULT60	7473							
MVBYTS	3F43							
MVURD	30A0							
NYGETM	7418							
NYSET2	74EB							
NYSTPR	7412							
NWAE.	5253							
NEULIM	EC55	63FF						
NEXT	17B5							
NEXT.	EDD7							
NFR.	3AAA							
NOCLE?	17FC	640A	6472					
ND1.	4E61							
NDTRSH	71ED							
NULER1	0A03							
NULOLD	4108							
NUN.	FCDS							
NUNBER	0D40							
NUNCHK	0A69							
NUNOUT	F9B5							
NUNREP	10B6							
NUNVRA	0E6D							
NUNVAL	0E70							
NUNPCK	6FB5							
NWTRPT	7470							
NWTKUF	47B3							
NWTLIN	2EB7							
NWTRON	47AC							
OFALRN	7444							
OFF10.	5E11							
OFF10.	5E06							

KLINK load map RYU102 11:58 AM TUE., 13 DEC., 1983 page 12

KEYARM	4EAD	62F1	62FF
KEYS.	499F		
KEYSRV	0277		
KEYTRN	4195		
KOPY	0A5F		
KYRDDR	E457	6A23	
KYDPEH	4EA7		
LA	7DDF	7146	
LADDEV	45FE		
LADREP	EC18		
L0L4	726D		
LCDUUI	59B8		
LCDRDY	F027		
LCDASI	F037		
LEAPYR	6F9A		
LEN.	FCBE		
LENAL	E939		
LEQ.	4DC4		
LEQ.	4E0B		
LET	1649		
LETO	079A		
LIFT.	52AD		
LIMEDR	44EF		
LIMEND	FE56		
LIMEND	0C15		
LIMLEX	21E5		
LISI.	1B48		
LIT?	06B3		
LN20	362A		
LN30	366B		
LN38	3654		
LN5	3732		
LN77	3676		
LORD	F7CC		
LOCK.	79B6		
LOCK?	79C8		
LOGCOM	399A		
LOGIS	379A		
LOOKUP	0A26		
LPOFF	5E20		
LPOW	5E28		
LSTIO.	50AB		
LT.	4DD6		
LT.	4E3C		
MAKEIT	06A8		
MALLUC	74DC		
MARGH.	7F67		
MARIO	3034		
MARDAT	6AF0		
MALIM	7446		
MALIMP	7440		
MLJSB	4661	694B	6967
MLJSB	4661	749F	74AC
MLJSB	4661	7519	75DE
MLJSB	4661	76FF	7706
MLJSB	4661	76FF	7735
MLJSB	4661	76FF	773F
MLJSB	4661	76FF	7821
MLJSB	4661	76FF	7836
MLJSB	4661	76FF	7885

KLINK load map RYU102 11:58 AM TUE., 13 DEC., 1983 page 14

OFFTHR	3F50		
OFFUNL	5E09		
OF10K.	5F6C		
OH.	1796		
OH.	ECDD	640F	
OH/OFF	5FD7		
OH?	E2B3		
OH?	E2B7		
ONE7AB	7DEE		
ONE7PR	E375		
ONEB	3E8B	6ABE	6AER
ONEB	3E8B	6EF7	6F15
ONEB	3E8B	785C	78EC
ONEB	3E8B	7DF0	7F86
ONEI	3EAC	68DA	6C88
ONER	3ECD	732C	7343
ONER-	3ED0		
ONER1	3EE1		
ONERDI	3EDE	6E1F	6EEF
ONERRO	1340	6490	
ONOFF2	5FDD		
ONR12?	1F3A	6B79	6BB3
ONTR	3F57		
ONTOK.	5FF6		
OPTION	131F		
OR.	4E4B		
OTHER	014D		
OUTICH	0B1D		
OUTIC40	0B14		
OUTICR	0B0C		
OUTEOL	0B2C		
OUTERB	494A		
OUTESC	0B19	63B4	63BB
OUTILIM	022B		
OUTINUM	5F6C		
OUTISTR	07EA		
PWARRAY	E87E		
PIANC?	122C		
PACK.	64E6		
PARI	09D7		
PARSE?	0BAA		
PARSER	0B0E		
PARSIT	0BA7		
PER.	7465		
PFHOPR	2164		
PGANNH	43F8		
PITO	39F6		
PILOF?	E803		
PILOH?	EAE8		
PIPREP	45E5		
PLIST	1906		
PLIST.	1B41		
PUP.	ENIE		
POPEHV	5040		
PDS.	FD4F		

NOT MANUFACTURER SUPPORTED
recipient agrees NOT to contact manufacturer

ROOM1	1F BA						
ROOM2	1F BD						
ROFEN	24D1						
ROH10	3C88						
ROH10	5941						
ROH10J	6F8E						
RP1IMP	70F2						
RP1PWS	7105						
RSETEM	1F2D						
RSETUP	E9C2						
RSHEH-	241F	682D	69A3	69F9	6A34	6A84	6DA8
RSHEH-	241F	6E8E	6EC1	6F21	6F4E	6F88	6FDE
RSHEH-	241F	70FD	731D	7588	7EE6	7D89	7DFA
RSHEH-	2420						
RSTBUF	64D9						
RSTREG	0E51						
RTOJM	1CEA						
RUN.	50AC						
RUNH1	50E7						
RUNKEE	50B9						
RUNROM	4754						
S. OFF	EAC5						
S. DN	EAD1						
SAFE1	53D0						
SALT	1B0E						
SAVBUF	64E2						
SAVENE	4741						
SAVEMV	4FC4						
SAVREG	0E33						
SAVRG+	0E39						
SBSER	1087						
SEAH	0EA3	6496					
SEAH+	0CA0						
SEAH1	1232						
SEAHX	F193						
SEAHN	F048						
SEHST	304F						
SEHST1	54D5						
SEAREH	0A05						
SECI10	398A						
SEMIN.	E7B1						
SEP10	3CE3						
SEP15	3CDF						
SEP20	3CC8						
SEOW	17E7						
SEQND	17D8						
SET	FEA0	68A7					
SETCAL	0067						
SETCLW	522A						
SETCOM	4D96						
SETED	5308						
SETI1/R	7CE8						
SETI1	5E30						
SETI30	745D						
SETI3H	0704						

[illegible]

SETOFF	5E11		
SETPR	1F5C		
SETPSB	09CD		
SETR12	00E1		
SETRW	1F47		
SETTOP5	4D8A		
SETMOB	7E55		
SFSEAR	0C9A		
SGN5	397C		
SHELD?	7E7C	6309	
SHF 10	3789		
SHFE1	7F6F		
SHLA	7F46		
SHRA	7F58		
SHRONF	3AA4		
SIGNW	24FE		
SIGNW?	24FB	753E	
SIM10	39DD		
SISFUN	3425		
SKIP1	FEFA		
SKIPW	35E2		
SKIPD	FEFA		
SKIPEM	FEF7		
SKIP1	FEDE		
SKIP1?	FEED		
SK1PL	20DA		
SK1PR	FEFA		
SK1PS	FEF1		
SKPCHK	5E8A	690F	763E 7FDD
SKPCTON	2ECA		
SKPLM	2185		
SKPLMH	2187	6648	7F8D
SKPLDP	EE95	7F93	
SKPVR	27C2		
SM11TC	10A3		
SMDFRM	EB18		
SPY	4617		
SQR30	38A5		
SQR5	3890		
STALL	51C3		
STALL?	51C8		
STALLS	51B8		
STARLM	7400		
STARND	EA0F	7482	7516
STARND?	EA2C		78EF
STARND	EA8F		
STAR?	58C3		
STAR1	7A20		
STAR?	7A0E		
STBEP	FEFA	6BD8	6C06
STBEP?	7054		
STRATE	6811		
STEP	EE56		
STMSK	6545		
STOP	5189		

MENJSD E36F
 WRITER F913
 YEARM 659F
 YINCHW 67F7
 YIXS 3974 71ED
 YIXS 7788
 Z36.56 2A67
 ZRONT 2A79
 ZRONT 2A8L
 ZZZZZZ 0899
 File RYH102 had 2 files 1142 symbols (1011 unreferenced, 0 undefined)
 1 0000
 2 0000
 3 6000

NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer
 NOVA

ITEM	LOC	OBJECT CODE	SRC=RYH102 OBJ=RYH102 11/16/1983 10:27 AM PG 2
exts, ents, equs, etc			
58 6000	EXT	HANDI	handi call
59 6000	EXT	INF10	infinity
60 6000	EXT	INTDIV	integer divide
61 6000	EXT	IO?	get iofile
62 6000	EXT	IPS	get integer part
63 6000	EXT	KEY4	get a key
64 6000	EXT	KEYMAN	keyfile name
65 6000	EXT	KEYADR	get key from key file
66 6000	EXT	LA	do left arrow
67 6000	EXT	MELJSD	job to melrom
68 6000	EXT	MOD10	mod divide
69 6000	EXT	MOVEUP	move in file
70 6000	EXT	MPYROI	multiply
71 6000	EXT	NEXT	next parse
72 6000	EXT	NOCLC1	insure prog mode
73 6000	EXT	ON/OFF	on/off parsing
74 6000	EXT	ONEB	get one binary
75 6000	EXT	ONEI	get one integer
76 6000	EXT	ONER	get one real
77 6000	EXT	ONEROI	get one real/integer
78 6000	EXT	ONERRO	parse on statement
79 6000	EXT	ONR12?	anything on stack?
80 6000	EXT	OUTESC	output esc char
81 6000	EXT	PREFND	find line
82 6000	EXT	PUTADR	put relative addr (R34)
83 6000	EXT	PUTKEY	put a key
84 6000	EXT	PUTREL	put relative addr (drp)
85 6000	EXT	RA	do right arrow
86 6000	EXT	RDTRY	read loop address
87 6000	EXT	RENIO	do renamer
88 6000	EXT	REPLIN	replace a line
89 6000	EXT	RESCOM	reserve space
90 6000	EXT	RETURN	a non-switching return
91 6000	EXT	RONJSD	JSD to ron >IV
92 6000	EXT	RSNEN	reserve memory
93 6000	EXT	SRM	scan input
94 6000	EXT	SET	set men to value
95 6000	EXT	SHIELD?	check shield
96 6000	EXT	SIGNIF	get significant key
97 6000	EXT	SKPLWK	skip to next line (devfile)
98 6000	EXT	SKPLWP	skip loop
99 6000	EXT	SKPLWW	skip file line >RH
100 6000	EXT	STAND?	stand by set
101 6000	EXT	STBLEP	beep
102 6000	EXT	SUBSTF	do gosub stuff
103 6000	EXT	SVCSET	set service word
104 6000	EXT	SYSJSD	jsb to alt ron
105 6000	EXT	THERE?	demand devfile
106 6000	EXT	THERE?	get devfile
107 6000	EXT	TOBCD2	convert bin to bcd >MK
108 6000	EXT	TOBIN2	convert bcd to bin
109 6000	EXT	TIMGET	get timer #
110 6000	EXT	TMEFRC	1 second real

ITEM	LOC	OBJECT CODE	SRC=RYH102 OBJ=RYH102 11/16/1983 10:27 AM PG 1
exts, ents, equs, etc			
5 6000	EXT	ABTHRD	hard abort the card reader
6 6000	EXT	ABOROI	add numbers on stack
7 6000	EXT	ALLOC	get some room
8 6000	EXT	ALFAM	upper case register
9 6000	EXT	ANN.E	error annun on
10 6000	EXT	ATISGN	clear RT2, rsnen
11 6000	EXT	BASEND	endline for line files
12 6000	EXT	BLANKS	some blanks
13 6000	EXT	BLERUF	blank error buffer
14 6000	EXT	BLIMP	clear input buffer
15 6000	EXT	CRIBUF	build catalog entry
16 6000	EXT	CNSROI	change number sign
17 6000	EXT	CLRCOD	clear tape code byte
18 6000	EXT	CLERR	clear pending error
19 6000	EXT	CNDREP	send and report command
20 6000	EXT	CNPEIN	comparator entry
21 6000	EXT	CONBIN	convert to binary
22 6000	EXT	COPY	COPY token
23 6000	EXT	CRDCPY	do card copy
24 6000	EXT	CRFILE	finish card file create
25 6000	EXT	CURSE+	cursor on
26 6000	EXT	CURSE-	cursor off
27 6000	EXT	DELREP	send DOL and report
28 6000	EXT	DELETE	delete empty space >RH
29 6000	EXT	DIV2	divide two numbers
30 6000	EXT	END.OW	clean up on statement
31 6000	EXT	ERR1	error report E=1
32 6000	EXT	ERR1+	error report E=1, return
33 6000	EXT	ERROR	error report
34 6000	EXT	ERRORR	error message in reg >RH
35 6000	EXT	EVIL	save registers
36 6000	EXT	ENKXI	next runtime
37 6000	EXT	FCARLO	create and allocate
38 6000	EXT	FCANUL	create null file
39 6000	EXT	FINWSC	output error msg
40 6000	EXT	FLOREN	get file
41 6000	EXT	FLSTAK	get pil rm device
42 6000	EXT	FDREN	open file, report error
43 6000	EXT	FDREN	open file
44 6000	EXT	FOR	for parse
45 6000	EXT	FPS	get frac part
46 6000	EXT	FPURGE	purge file
47 6000	EXT	FSEEK	get file record
48 6000	EXT	GET18	get a string, push token
49 6000	EXT	GETADW	get address to drp
50 6000	EXT	GETADR	get address
51 6000	EXT	GETCHR	get input char >IV
52 6000	EXT	GETESI	get file string
53 6000	EXT	GETPAD	get pil address
54 6000	EXT	GETPAR	get parameters
55 6000	EXT	GETIEN	get input
56 6000	EXT	GETSIP	get a string

ITEM	LOC	OBJECT CODE	SRC=RYH102 OBJ=RYH102 11/16/1983 10:27 AM PG 3
exts, ents, equs, etc			
111 6000	EXT	THROPN	open timer file
112 6000	EXT	TRYIN	get 0 or 1 numbers
113 6000	EXT	UPRC*	uppercase registers
114 6000	EXT	UPCL	uppercase string
115 6000	EXT	VBSY	wait for tape
116 6000	EXT	VBYE	untalk, unlisten
117 6000	EXT	VDDL2	send DDL to device
118 6000	EXT	VDIR	point to first tape dir
119 6000	EXT	VFEOD?	end of tape dir?
120 6000	EXT	VGET	get a device
121 6000	EXT	VHI	start up tape
122 6000	EXT	VFLAD	listen device
123 6000	EXT	TRJUMP	trace a jump
124 6000	EXT	VFRFP?	purged file on tape
125 6000	EXT	VFRN?	tape device?
126 6000	EXT	VFRNDE	down one tape dir
127 6000	EXT	VFRDE	convert to roo dir
128 6000	EXT	VFRVDE	reverse tape dir
129 6000	EXT	VFRWD	do a tape read
130 6000	EXT	VFRUSK	do a tape seek
131 6000	EXT	VFRAD	talk device
132 6000	EXT	VFRACH	move some bytes
133 6000	EXT	VFRU	do a tape write
134 6000	EXT	WAITKY	wait for key
135 6000	EXT	WARN	warning message >RH
136 6000	EXT	YIXS	compute y'x
137 6000			
138 6000	EXT	APROPN	
139 6000	EXT	DIH8	
140 6000			
141 6000		4-22 EQU OFFFAH	
142 6000		4-10 EQU OFFFGH	
143 6000		4-6 EQU OFFFAH	
144 6000		81 EQU 1D	
145 6000		82 EQU 2D	
146 6000		83 EQU 3D	equates to save
147 6000		84 EQU 4D	label space
148 6000		86 EQU 6D	for the assembler
149 6000		87 EQU 7D	
150 6000		88 EQU 8D	
151 6000		89 EQU 9D	
152 6000		810 EQU 10D	
153 6000		812 EQU 12D	
154 6000		813 EQU 13D	
155 6000		815 EQU 15D	
156 6000		816 EQU 16D	
157 6000		818 EQU 18D	
158 6000		820 EQU 20D	
159 6000		823 EQU 23D	
160 6000		828 EQU 28D	
161 6000		832 EQU 32D	
162 6000		833 EQU 33D	
163 6000		849 EQU 49D	
164 6000		880 EQU 80D	

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

PRSLX DEF MYRTM      00
      DEF HQH/DF      01 led
      DEF MONOMM      02 error report
      DEF MYRTM      03 status ?
      DEF ROMIS      04 status set
      DEF TIOPRS      05 ncopy tape to tape
      DEF ROMIS      06 install private tape
      DEF OHSRQ      07 on srq <statements>
      DEF RHOIZH      08 window

```

```

*      FUNCTION
73      DEF      1005A.
74      DEF      1006C.
75      DEF      1007B.
76      DEF      FILE$.
77      DEF      LIMEN.
78      DEF      LIST$.
79      DEF      DISP$.
80      DEF      PRTN$.
81      DEF      ENDL$.
82      DEF      OFKY$.
83      DEF      DELY$.
84      DEF      HIDI?.
85      DEF      QUOT?.
86      DEF      MARK?.
87      DEF      ASEB.
88      DEF      HEKO.
89      DEF      INPL$.
90      DEF      MSRT.
91      DEF      MEMO.
92      DEF      USIZE.
93      DEF      RIRN$.
94      DEF      LIRN$.
95      DEF      ROIB.
96      DEF      REV$.
97      DEF      RP1$.

```

ASCLC	ASP				
	ASP	'LCD'			01
	ASP	'DO ERROR'			02
	ASP	'STATUSA'			03
	ASP	'STATUS'			04
	ASP	'MCOPIY'			05
	ASP	'INSTALL'			06
	ASP	'ON SRQ'			07
	ASP	'WINDOW'			08
	ASP	'USERMSG'			09
	ASP	'REASSIGN'			0A
	ASP	'OFF SRQ'			0B
	ASP	'ENABLE SRQ'			0C
	ASP	'ADJUSTA'			0D

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 8
-----
>>>>>> ROM tables <<<<<<<
341 6171 41 44 40 55 ASP 'ADJUST' 0E
341 6175 53 04 ASP 'NEXT' 0F
342 6177 4E 45 58 04 ASP 'FOR' 10
343 6178 46 4F 02 ASP 'EXIT' 11
344 617E 45 58 49 04
345 6182
346 6182 4
347 6182 54 4F 42 41 ASP 'FUNCTIONS' 12
347 6186 53 45 04 ASP 'TOBASE' 13
348 6189 54 4F 44 45 ASP 'TODEC' 14
348 618D C3 ASP 'LEA18' 15
349 618E 54 43 41 54 ASP 'FILES' 16
349 6192 04 ASP 'LIMEN' 17
350 6193 46 49 4C 45 ASP 'LISTIO' 18
350 6197 04 ASP 'DISPLAY' 19
351 6198 4C 49 4E 45 ASP 'PRINTERS' 1A
351 619C 03 ASP 'ENDLINES' 1B
352 619D 4C 49 53 54 ASP 'DEFKEYS' 1C
352 61A1 49 4F 04 ASP 'DELAY?' 1D
353 61A4 44 49 53 50 ASP 'WIDTH?' 1E
353 61A8 4C 41 59 04 ASP 'PWIDTH?' 1F
354 61AC 50 52 49 4E ASP 'MARGIN?' 20
354 61B0 54 45 52 04 ASP 'ASL' 21
355 61B4 45 4E 44 4C ASP 'HEX' 22
355 61B8 49 4E 45 04 ASP 'TEMPLATES' 23
356 61DC 44 45 46 4B ASP 'USTART?' 24
356 61E0 45 59 04 ASP 'MEND?' 25
357 61C3 44 45 4C 41 ASP 'MSIZE?' 26
357 61C7 59 0F ASP 'ATRINO' 27
358 61C9 57 49 44 54 ASP 'LFRINO' 28
358 61CD 48 0F ASP 'ROT' 29
359 61CF 50 57 49 44 ASP 'REV' 2A
359 61D3 54 48 0F
360 61D6 4D 41 52 47
360 61DA 49 4E 0F
361 61DD 41 53 43 04
362 61E1 48 45 58 04
363 61E5 54 45 4D 50
363 61E9 4C 41 54 45
363 61ED 04
364 61EE 57 53 54 41 ASP 'AP16' 2A
364 61F2 52 54 0F
365 61F5 57 45 4E 44
365 61F9 0F
366 61FA 57 53 49 5A
366 61FE 45 0F
367 6200 52 54 52 49
367 6204 4D 04
368 6206 4C 54 52 49
368 620A 4D 04
369 620C 52 4F 54 04
370 6210 52 45 56 04
371 6214 52 50 54 04

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 9
-----
>>>>>> ROM tables <<<<<<<
372 6218 41 41 4E 44 ASP 'ARMOS' 2B
372 621C 04 ASP 'ADRS' 2C
373 621D 41 4F 52 04 ASP 'AKORS' 2D
374 6221 41 58 4F 52
374 6225 04
375 6226 4B 41 4E 44 ASP 'NAND' 2E
375 622A 04
376 622B 4B 4F 52 04 ASP 'NORS' 2F
377 622F 4B 58 4F 52 ASP 'NXORS' 30
377 6233 04
378 6234 41 53 43 49 ASP 'ASC116' 31
378 6238 49 04
379 623B 53 50 41 CE ASP 'SPAN' 32
380 623E 42 52 45 41 ASP 'BREAK' 33
380 6242 CB
381 6243 4D 41 50 04 ASP 'NOPS' 34
382 6247 46 49 4C 4C ASP 'FILL' 35
382 624B 04
383 624C 4C 57 52 43 ASP 'LURC' 36
383 6250 04
384 6251 4C 41 53 54 ASP 'LASTLEN?' 37
384 6255 4C 4E 0F
385 6258 55 53 43 52 ASP 'USCR' 38
385 625C 04
386 625D 4E 53 43 52 ASP 'MSCR' 39
386 6261 04
387 6262 45 53 43 04 ASP 'ESCR' 3A
388 6266 53 4D 45 59 ASP 'SKEY' 3B
388 626A 04
389 626B 57 4B 45 59 ASP 'UKEY' 3C
389 626F 04
390 6270 46 4C 41 47 ASP 'FLAG' 3D
390 6274 04
391 6275 46 4C 41 47 ASP 'FLAG2' 3E
391 6279 0F
392 627A 42 49 04 ASP 'BIT' 3F
393 627D 42 54 14 ASP 'BID' 40
394 6280 4F 54 14 ASP 'OID' 41
395 6283 4B 54 13 ASP 'HID' 42
395 6286 44 54 42 04 ASP 'DIB' 43
397 628A 44 54 4F 04 ASP 'DIB' 44
398 628E 44 54 4B 04 ASP 'DIB' 45
399 6292 41 53 4B 46 ASP 'ASHB' 46
399 6296 04
400 6297 41 52 4F 54 ASP 'AR018' 47
400 629B 04
401 629C 48 53 4B 46 ASP 'MSHFB' 48
401 62A0 04
402 62A1 48 52 4F 54 ASP 'MR018' 49
402 62A5 04
403 62A6 43 4F 55 4E ASP 'COUNT?' 4A
403 62AA 54 0F
404 62AC 42 55 46 04 ASP 'BUF' 4B

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 10
-----
>>>>>> ROM tables <<<<<<<
405 62B0 46 49 4E C4 ASP 'FIND' 4C
406 62B4 53 55 42 04 ASP 'SUBR' 4D
407 62B8 54 49 4D 45 ASP 'TIMER?' 4E
407 62BC 52 0F
408 62BE 52 45 50 4C ASP 'REPL' 4F
408 62C2 04
409 62C3 52 49 47 48 ASP 'RIGHTS' 50
409 62C7 54 04
410 62C9 43 41 54 03 ASP 'CRAB' 51
411 62CD 4C 45 46 54 ASP 'LEFT' 52
411 62D1 04
412 62D2 4D 49 44 04 ASP 'MID' 53
413 62D6 FF BYT 00FFH end of table
414 62D7
415 62D7 61 VAL REVIDE revision number
416 62D8 02 BYT 00000010B ROM LEN access bits
417 62D9

```

NOT MANUFACTURER SUPPORTED
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 11
-----
>>>>>> NAND call processing <<<<<<<
419 62D9 40 C8 07 INILEX LMS R00.=V.EALD check low end
420 62DE FA 04 IFCY no
421 62DE 44 A9 79 63 G10 CHKHI check rest
422 62E2 ENDDIF
423 62E2
424 62E2 90 TSB R00 "V.COLD?" doing coldstart?
425 62E3 F6 04 IFZR yes
426 62E5 44 A9 98 64 GTD CRTAPR go create aprfile
427 62E9 ENDDIF
428 62E9
429 62E9 8A DEB R00 "V.WARN?" doing warnstart?
430 62EA F6 71 IFZR yes
431 62EC CE 99 64 JSB =CRTAPR make sure aprfile is there
432 62EF 60 B1 FF FF LDMD R40.=KEYHAM get name of keys file
433 62F3 CE FF FF JSB =FOPEN check if there
434 62F6 F9 05 IFEN no
435 62F8 CE 42 63 JSB =ASTCHK AUTOST must be
436 62F8 F8 44 REN there or not return
437 62FD ENDDIF
438 62FD 60 B1 FF FF LDMD R40.=KEYHAM get name of keys file
439 6301 50 A9 BE 54 LDM R20.=LYIEKT keys file is a text file
440 6305 CE FF FF JSB =SYSJSB
441 6308 FF FF DEF FCNUL find or create it
442 630A 9B DCE
443 630B 9B DCE
444 630C F9 33 REZ leave, out of memory
445 630E CE FF FF JSB =MOVEUP move to first line
446 6311 7E A9 59 01 LDM R76.=B159C r76 = 159 BCD
447 6315 CE FF FF JSB =FSECK find the line
448 6318 F9 21 IFEN if not found:
449 631A CE 42 63 JSB =ASTCHK insist on AUTOST
450 631D F8 22 REN or leave
451 631F 60 B1 4E 63 LDMD R40.=KEYDEF get some def
452 6323 B3 0D 82 STMD R40.=INPM2 put at out
453 6326 B1 55 63 LDMD R40.=ASTIME get rest of command
454 6329 B3 14 82 STMD R40.=INPM2+7D put that out
455 632C 5C A9 0D 82 LDM R34.=INPM2 r34 = source addr
456 6330 5A A9 0F 00 LDM R32.=B15 set up length
457 6334 CE FF FF JSB =SYSJSB
458 6337 FF FF DEF REPLIN put the line in
459 6339 F8 06 REN quit if no room
460 633B ENDDIF
461 633B 42 A8 9F LDB R02.=1590 get chr(159)
462 633C CE FF FF JSB =PUTKEY put this key
463 6341 9E RTM
464 6342
465 6342 60 B1 55 63 ASTCHK LDMD R40.=ASINIE get filename
466 6346 66 B1 FF FF LDMD R46.=BLANKS clear garbage
467 634A CE FF FF JSB =FOPEN find file
468 634D 9E RTM
469 634E
470 634E 20 KEYDEF ASC ' '
471 634F 06 BYT 166D

```



```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 12
=====
>>>>>> HANDI call processing
472 6350 52 55 4E 20 ASC "RUN "
472 6354 27
473 6355 41 55 54 4F ASINME ASC "AUTOSTI"
473 6359 53 54 27
474 635C RA BYI 170D
475 635D
476 635D ENDF
477 635D
478 635D 40 CC 03 SBB ROO,=3 V.RSIN? continuing?
479 6360 F5 04 IFZR yes,
480 6362 4A A9 51 74 G1D SROOK. go do it
481 6366 ENDF
482 6366
483 6366 RA DCB ROO =V.SPV? doing spy?
484 6367 F6 04 IFZR yes,
485 6369 44 A9 07 64 G1D DOSAQ go do it
486 636D ENDF
487 636D
488 636D RA DCB ROO =V.SIAL? stalling?
489 636E F6 03 IFZR yes,
490 6370 CC CC 64 JSD =APROPH get our file
491 6373 F8 D8 RBN error, gave up
492 6375 42 92 CLD ROZ clear the
493 6377 1E A6 S1BD ROZ,R36 valid arg flag
494 6379 ENDF
495 6379 9E RTN
496 637A
497 637A 40 C8 3E CHKHI CMB ROO,=V.VOLT doing volatile purge?
498 637D F6 0A IFEQ yes,
499 637F 60 01 9B 64 CMB ROO,=APROPH is it our file?
500 6383 F6 03 IFEQ yes,
501 6385 CC 1E 75 JSB =HANDLE save it
502 6388 ENDF
503 6388 9E RTN
504 6389 ENDF
505 6389
506 6389 C8 17 CMB ROO,=V.SROR doing srv request?
507 638D F6 07 IFEQ yes,
508 638D 42 A8 02 LDB ROZ,=BIM1 set spy bat to
509 6390 CC FF FF JSB =SVCSE1 get a chance
510 6393 9E RTN
511 6394 ENDF
512 6394
513 6394 C8 32 CMB ROO,=V.CARD doing a card track
514 6396 F6 04 IFEQ yes,
515 6398 44 A9 10 65 G1D RECVR go process info
516 639C ENDF
517 639C
518 639C C8 31 CMB ROO,=V.FILE doing a file thing?
519 639E F6 04 IFEQ yes,
520 639D 44 A9 4B 65 G1D RECVR go try it
521 639A ENDF
522 639A

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 14
=====
>>>>>> Parse routines
571 63F0 4C A8 8C FOR+ LDB R14,=F0RTOK get mainframe for token
572 63F3 CC FF FF JSB =SYSJSB call mainframe
573 63F6 FF FF DEF FOR for parsing
574 63F8 9E RTN
575 63F9
576 63F9 4C A8 8F NEXT+ LDB R14,=NEXTOK get mainframe next token
577 63FC CC FF FF MEK1- JSB =SYSJSB call mainframe
578 63FF FF FF DEF MEK1 next parsing
579 6401 9E RTN
580 6402
581 6402 CC FF FF ROMIS JSB =SYSJSB get a string
582 6405 FF FF DEF GET16 and put out token
583 6407 F0 28 JNP ROM1OK go clean up
584 6409
585 6409 CC FF FF NOW/OF JSB =MOCLC! can't do in calc mode
586 640C CC FF FF JSB =SYSJSB get the on/off
587 640F FF FF DEF ON/OFF word
588 6411 F0 1E JNP RONTOK finish up
589 6413
590 6413 CC FF FF ROMOIN JSB =SYSJSB go get a numeric
591 6416 FF FF DEF IRYIN parameter (maybe)
592 6418 F0 17 JNP RONTOK go finish up
593 641A
594 641A CC 7E 64 ANISIM JSB =G1STP+ get a string
595 641D 4C C8 2E CMB R14,=CONMA more?
596 6420 F7 05 IFNE no,
597 6422 5E 0A E4 PUBD R36,=R12 put token on stack
598 6425 F0 0A JNP RONTOK finish up
599 6427 ENDF
600 6427 1E A0 LDB R14,R36 get token
601 6429
602 6429 5D A8 01 LDB R35,=1 want one number
603 642E
604 642E CC FF FF GETINMB JSB =SYSJSB go get the
605 642F FF FF DEF GETIPAR parameters
606 6431
607 6431 6C A9 RONTOK LDM R54,=
608 6433 B4 VAL EROTK get the rom switch token
609 6434 64 40 DEF NYROMM and ny rom number
610 6436 00 BYI 0 and garbage
611 6437 6F 0A E2 POBD R57,=R12 get my token
612 643A 6C E5 PUND R54,=R12 put it all back
613 643C 9E RTN
614 643D
615 643D 4C 0E E4 ITOPRS PUBD R14,=R06 save our token
616 6440 CC 7E 64 JSB =G1STP+ get a string
617 6443 4C C8 9F CMB R14,=T0TK TO next?
618 6446 F7 07 IFNE no,
619 6448 42 06 E2 POBD ROZ,=R06 trash token
620 644B CC FF FF JSBN =ERR1+ report error
621 644E 53 BYI 83D missing to
622 644F ENDF
623 644F A8 DB LDB R14,=T0TK2 get the nop to

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 13
=====
>>>>>> HANDI call processing
523 639A C8 14 CMB ROO,=V.CHED a CHEDI intercept?
524 639E F6 0B RME no, leave
525 639B
526 639B 42 B0 DF 82 LDBD ROZ,=JUSTSD leave things alone?
527 639C F6 E5 RMZ yes
528 639E
529 639E
530 639E 60 C8 C0 ** the CNTL-CLR (clear tv) key
531 639F F6 0A CMB R40,=(CNTL+CLRKEY) is it CNTL-CLR
532 63A3 CC FF FF IFEQ
533 63A6 48 JSB =OUTESE put out an 'ESC'
534 63A7 CC FF FF BYI 'H' 'H' (hone up)
535 63A8 4A JSB =OUTESE put out an 'ESC'
536 63AB F0 2F BYI 'J' 'J' (clear display)
537 63AD JNP KEYNOP nop the key and leave
538 63AD ENDF
539 63AD
540 63AD C8 C8 ** underline modifier key (CNTL-I/R)
541 63BF F6 15 CMB R40,=(CNTL+IRKEY) is it CNTL-I/R ???
542 63C1 A8 8D IFEQ yes,
543 63C3 CC FF FF LDB R40,=BITM7 get UPSHIFT flag
544 63C6 CC FF FF JSB =CURSE+ turn on the cursor
545 63C9 60 02 94 JSB =GETCHR get a key
546 63CC CC FF FF ORB R40,ROZ set the top bit
547 63CF 42 A8 FF JSB =CURSE- turn off the cursor
548 63D2 B2 DF 82 LDB ROZ,=OFFH set the leave alone
549 63D5 9E S1BD ROZ,=JUSTSD flag
550 63D6 RTN let's not upshift ourselves!
551 63D6 ENDF
552 63D6
553 63D6 CC FF FF ** the "jump word" keys (TAB) and (shift TAB)
554 63D9 FF FF JSB =SYSJSB call external routine
555 63DB F9 12 DEF SHED? do we have a shield?
556 63DD 42 20 A0 IFEN no,
557 63E0 CC BE LDB ROZ,R40 get the key
558 63E2 F7 04 SBB ROZ,=TABKEY test for tab key
559 63E4 C8 20 JZR DOTAB yes, go do it
560 63E6 F6 07 CMB ROZ,=SHIFT test for shift
561 63E8 IFZR yes,
562 63E8 DRP '2 save code
563 63E8 A2 ARP '40 for us
564 63E9 CC 3B 71 S1B ROZ,R40 set tab dir flag
565 63EC 60 A8 FF JSB =TABII do the tab
566 63EF KEYNOP LDB R40,=NOPKEY no key hit
567 63EF ENDF
568 63EF 9E RTN
569 63F0

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 15
=====
>>>>>> Parse routines
624 6451 CC 7C 64 JSB =G1STP+ get a string
625 6454 5E 0A E4 PUBD R36,=R12 push TO token
626 6457 06 E2 PUBD R36,=R06 get our token
627 6459 0A E4 PUBD R36,=R12 put it out
628 645B F0 04 JNP RONTOK finish up
629 645D
630 645D CC FC 63 EXIT JSB =MEK1- parse like a next
631 6460 F0 1F JNP RONTOK finish up
632 6462
633 6462 50 92 ROM12M CLD R35 get what you can
634 6464 CC 2C 64 JSB =GETINMB go get em
635 6467 5C C8 03 CMB R34,=3 more than 2?
636 646A FA 04 IFLY yes,
637 646C CC FF FF JSBN =ERR1+ error out
638 646F 54 BYT 84D
639 6470 ENDF
640 6470 9E RTN
641 6471
642 6471 CC FF FF ROM1A+ JSB =MOCLC! not in calc mode
643 6474 4C 0A E4 ROM1A PUBD R14,=R12 put the token
644 6477 CC 31 64 JSB =RONTOK adjust output
645 647A CC 93 64 JSB =SSCAN do a post scan
646 647D 9E MYRTM RTN
647 647E
648 647E CC FF FF G1STP+ JSB =SYSJSB do a sysjsb
649 6481 FF FF DEF G1STP to getstip
650 6483 9E RTN
651 6484
652 6484 4C 0A E4 DMSRQ PUBD R14,=R12 put token on stack
653 6487 CC 31 64 JSB =RONTOK add error stuff
654 648A 4C 0A E2 PUBD R14,=R12 get back our token
655 648D CC FF FF JSB =SYSJSB put back token, scan,
656 6490 FF FF DEF OMERRD parse rest of line
657 6492 9E RTN
658 6493
659 6493 CC FF FF SSCAN JSB =SYSJSB do a sysjsb
660 6496 FF FF DEF SCAN to scan
661 6498 9E RTN
662 6499

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 16
*****
>>>>>>> Create .aprf file for varp *****
664 6499 60 89 CRTAPR LDM R40,= get the filename
665 6498 2E 69 6F 72 APRNHE ASC 'loronf1'
666 6497 61 60 66 6C
667 6493 5A 89 17 00 LDM R32,=B23 get file size
668 6492 5A 89 17 00 LDM R20,=IYVSVM get file type
669 6491 5A 89 00 00 JSB =FCRALO create it
670 6490 5E 18 85 REN error, or already there
671 6489 5E 18 85 LDM R36,=R30 get data pointer
672 6488 42 93 JSB =CLAPR clear out SRO stuff
673 6487 1E 15 CLM R02 clear IOSIZE
674 6486 63 81 FF FF LDM R43,=BLANKS clear KEYBOARD IS
675 6485 1E 15 LDM R43,=R36 stuff
676 6484 9E RTN
677 6483
678 6482 60 93 CLAPR CLM R40 clear out
679 6481 1E 15 PDM R40,=R36 the SRO space
680 6480 1E 15 PDM R40,=R36 for initial
681 6479 9E RTN
682 6478
683 6477 60 81 98 64 APRDPM LDM R40,=APRHE get the filename
684 6476 1E 15 JSB =FOPEM open the file
685 6475 09 05 IFEM not there,
686 6474 1E 15 JSB =CRTAPR create it
687 6473 1E 15 REN error out
688 6472
689 6471 5E 18 85 LDM R36,=R30 get file loc
690 6470 9E RTN
691 6469
692 6468 1E 15 JSB =EVIL save those regs
693 6467 1E 15 DEF S20-77
694 6466 1E 15 JSB =APRDPM get the file
695 6465 08 05 REN error, give up
696 6464 42 1E 81 LDM R02,=R36 save pointer
697 6463 73 11 PDM R63,=R36 get the seq stuff
698 6462 90 TSB R63 are we active?
699 6461 77 10 RZR no, quit
700 6460 60 11 PDM R40,=R36 get runfile name
701 6459 01 75 82 CHM R40,=RANHE is it us?
702 6458 1E 15 RNE no, quit
703 6457 5C 92 CLM R34 clear active
704 6456 02 86 STB R34,=R02 flag
705 6455 09 00 80 LDM R34,=R0000H get normal wub flag
706 6454 5E 08 81 LDM R36,=R10 get current R10
707 6453 1E 15 JSB =SUBSIF set up subroutine
708 6452 1E 15 REN error, give up
709 6451 74 12 83 STM R64,=R22 move new inc data
710 6450 54 03 47 82 ADMD R24,=RHF1LE absolutize R10
711 6449 08 83 STM R24,=R10 save new R10
712 6448 52 03 47 82 ADMD R22,=RHF1LE absolutize PCR
713 6447 1E 15 JSB =RJUMP trace that jump
714 6446 9E RTN
715 6445

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 17
*****
>>>>>>> recover card stuff *****
717 6511 64 06 15 RECV+ PDM R44,=R06 save scratch reg
718 6510 66 80 61 84 LDM R46,=CRDSTS get current status
719 6509 2E 25 IFMG doing recover,
720 6508 93 CLM R46 clear for load
721 6507 80 90 84 LDM R46,=FLHEAD get the track number
722 6506 1E 15 LDM R46 double (2K)
723 6505 00 80 81 ADMD R46,=INPBUF add start of inbuf
724 6504 24 83 STM R46,=R44 move for later
725 6503 01 88 84 LDM R46,=PR11ST get partial info
726 6502 1E 15 PDM R46,=R44 save in inbuf+2K
727 6501 81 80 81 LDM R46,=INPBUF get current end
728 6500 1E 15 CHM R46,=R44 new one bigger?
729 6499 1E 15 IFMG yes,
730 6498 01 81 LDM R46,=R44 move for store
731 6497 03 80 81 STMD R46,=INPBUF save new end
732 6496
733 6495 67 80 9E 84 LDM R47,=FLHEAD+1 get # of tracks
734 6494 82 0E 81 STB R47,=INP+94 save for later
735 6493 80 83 84 LDM R47,=F11TYP get file type
736 6492 1E 02 RHM R47,=TYLIN? is it recoverable?
737 6491 1E 02 IFZR no,
738 6490 1E 15 JSB =ERROR report error
739 6489 1E 15 BVI 680 not recoverable
740 6488 44 89 FF FF GTD R01HARD go kill the CR
741 6487
742 6486
743 6485 64 06 15 PDM R44,=R06 restore scratch reg
744 6484 9E RTN
745 6483
746 6482 42 80 84 84 RECV+ LDM R02,=I0KEN get the token
747 6481 08 7C LDM R02,=C0PYTK is it a copy?
748 6480 1E 15 RNE no, give up
749 6479 64 14 81 LDM R44,=R24 get the device
750 6478 1E 42 43 52 CHM R44,==BERD' is it us?
751 6477 44 RNE no, give up
752 6476 1E 15 TSM R34 is target ram?
753 6475 1E 15 RWZ no, give up
754 6474 58 88 ICB R33 is it special file?
755 6473 1E 15 RWZ yes, give up
756 6472 8A DCB R33 restore flag
757 6471 06 44 PDM R33,=R06 save us from recon
758 6470 1E 15 PDM R56,=R06 save us from recon
759 6469 1E 15 JSB =BLIMP blank inbuf
760 6468 1E 15 JSB =RESCDM get room for saving
761 6467 01 80 BVI 80 key intercept
762 6466 0E 06 15 PDM R56,=R06 restore file name
763 6465 5B 1E 15 PDM R33,=R06 restore special type
764 6464 1E 15 REN name, give up
765 6463 60 80 61 84 LDM R40,=CRDSTS get current status
766 6462 1E 15 ADMD R40,=R000H get recover bit
767 6461 82 61 84 STB R40,=CRDSTS save new status
768 6460 81 78 83 LDM R40,=KYIDLE get current intercept

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 18
*****
>>>>>>> recover card stuff *****
769 6585 16 87 STMD R40,=R26 save it
770 6584 89 LDM R40,=
771 6583 1E 15 JSB =R0MJSB get a romjsb
772 6582 1E 15 DEF KEYINT to key intercept
773 6581 64 40 DEF NYR0RM in my rom
774 6580 9E RTN and return
775 6579 60 83 83 STMD R40,=KYIDLE set up intercept
776 6578 56 06 15 PDM R26,=R06 save pointer to old intercept
777 6577 89 82 81 LDM R26,=IMP+2 get first track pointer
778 6576 83 80 81 STMD R26,=IMPBUF set last track
779 6575 88 88 FF LDM R26,=OOFFH unit last track
780 6574 82 0E 81 STB R26,=IMP+94 to not found
781 6573 64 93 CLM R44 make sure target
782 6572 1E 83 STM R44,=R34 is ran
783 6571 43 41 52 LDM R44,=CARD' make the source
784 6570 44
785 6569 14 83 STM R44,=R24 a normal card
786 6568 1E 15 JSB =CRDCPY load the cards
787 6567 1E 15 * intercept will prevent
788 6566 1E 15 * purge of file when attn hit
789 6565 56 06 15 PDM R26,=R06 get intercept pointer
790 6564 60 16 85 LDM R40,=R26 get old intercept
791 6563 83 88 83 STMD R40,=KYIDLE put it back
792 6562 80 61 84 LDM R40,=CRDSTS get current status
793 6561 1E 80 SBB R40,=R000H clear recover bit
794 6560 82 61 84 STB R40,=CRDSTS save new status
795 6559 42 80 8E 84 LDM R02,=R01FLG test abort flag
796 6558 1E 04 IFZR no abort,
797 6557 44 89 83 66 GTD RCVROK go finish up
798 6556 9E ENDMF
799 6555 81 8C 84 LDM R02,=R01SVE get back dir pointer
800 6554 1E 15 * saved by intercept
801 6553 1E 15 CHM R02,=R01RY was it attn or error?
802 6552 1E 15 RZR error, give up
803 6551 80 83 STM R02,=R60 move for CRIFLE
804 6550 58 85 LDM R30,=R60 get file pointer
805 6549 1E 15 JSB =LSTRK? get last track?
806 6548 1E 15 IFMG no,
807 6547 66 93 CLM R46 clear accum
808 6546 1E 15 LOOP multiply loop
809 6545 42 8A DCB R02 decrement track count
810 6544 04 06 JNG NEWSZE done, go on
811 6543 66 03 5F 84 ADMD R46,=FULTRK compute last track
812 6542 1E 15 WHIP go again
813 6541 66 83 81 84 NEWSZE STMD R46,=F1LSZE set new file size
814 6540 1E 15 ENDMF
815 6539 1E 15 JSB =CRIFLE clean up the directory
816 6538 5E 01 81 84 LDM R20,=F1LSZE get file size
817 6537 1E 04 IFZR all gone,
818 6536 44 89 8A 66 GTD RCVROK clean up
819 6535 9E ENDMF
820 6534 81 80 81 LDM R20,=IMPBUF get last track pointer

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 19
*****
>>>>>>> recover card stuff *****
821 6533 42 93 CLM R02 get a
822 6532 88 DCB R02 -1
823 6531 10 87 STMD R02,=R20 mark the end of table
824 6530 52 89 82 81 LDM R22,=IMP+2 get pointer to first entry
825 6529 5A 89 86 FF LDM R32,=(O-PCBLEN) get -pcblen in accum
826 6528 58 30 85 LDM R30,=R60 get start of file
827 6527 10 83 STM R30,=R20 save for later
828 6526 78 92 CLM R70 clear flow flag
829 6525 10 83 LOOP scan loop
830 6524 10 83 LOOP missing track scan
831 6523 10 83 ELE clear found flag
832 6522 5C 12 1E PDM R34,=R22 get next track info
833 6521 09 20 20 CHM R34,=R2020H track there?
834 6520 1E 04 IFEQ no,
835 6519 81 6F 84 LDM R34,=FULTRK get track size
836 6518 1E 04 ICE set found flag
837 6517 10 83 ENDMF
838 6516 5A 1E 15 ADMD R32,=R34 add to accum
839 6515 08 1E 15 WHEN while found
840 6514 1E 15 JSB =DELETE delete accum at R30
841 6513 78 90 TSB R70 test flow flag
842 6512 1E 04 IFZR first pass,
843 6511 58 08 00 ADMD R30,=PCBLEN adjust pointer for pcb
844 6510 10 83 ENDMF
845 6509 78 88 ICB R70 change flow flag
846 6508 5E 18 81 LDM R36,=R30 get lineH pointer
847 6507 03 5F 84 ADMD R16,=FULTRK point to end of track
848 6506 1E 15 SBB R36,=R34 adjust for inset
849 6505 1E 15 LOOP line scan loop
850 6504 90 ELE clear found flag
851 6503 42 89 99 89 LDM R02,=ENDLIN get endline #
852 6502 1E 09 LDM R02,=R30 at end of file?
853 6501 1E 04 IFEQ yes,
854 6500 78 92 CLM R70 change flow flag
855 6499 20 20 JMP REVEN- set 0 post delete
856 6498 1E 15 ENDMF
857 6497 58 1E 83 STM R30,=R34 save current line ptr
858 6496 1E 15 JSB =SKPLWM skip to next line,
859 6495 58 1E 15 CHM R30,=R36 past end of track?
860 6494 1E 15 IFZY yes,
861 6493 5A 12 1E PDM R32,=R22 get next track info
862 6492 5F 04 IFMG end of table,
863 6491 78 92 CLM R70 change flow flag
864 6490 1E 04 JMP REVEN- set trailing delete
865 6489 1E 15 ENDMF
866 6488 09 20 20 CHM R32,=R2020H track there?
867 6487 1E 06 IFMG yes,
868 6486 5E 03 5F 84 ADMD R36,=FULTRK get track size
869 6485 1E 11 ELSE no,
870 6484 1E 15 PDM R32,=R22 back up for other loop
871 6483 58 1E 15 RECV+ CHM R30,=R36 exact fat?
872 6482 1E 08 IFMG no,
873 6481 5A 81 LDM R32,=R36 get track start

```



```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 24
-----
set status of system switches
ENDIF
1034 677E
1035 677E
1036 677E
1037 677E
1038 677E
1039 677E 40 A8 33 LDB R00,-63 get reg pointer
1040 6781 LOOP flip sense loop
1041 6781 41 90 TSB R* test flag
1042 6783 F4 06 IFPS if specified,
1043 6785 F6 B3 IFZR if 0,
1044 6787 88 ICB R0 make it 1
1045 6788 F0 01 ELSE if not 0,
1046 678A 92 CDB R0 make it 0
1047 678B ENDMIF
1048 678B ENDMIF
1049 678B 40 88 ICB R00 inc reg pointer
1050 678D C8 3A CMB R00,-72 done?
1051 678F FA F0 SAME no, go again
1052 6791
1053 6791 73 90 TSB R63 test shell flag
1054 6793 F4 03 IFPS specified,
1055 6795 B2 59 82 STBD R63,-SHELOW set it
1056 6798 ENDMIF
1057 6798 74 90 TSB R64 test beep flag
1058 679A F4 03 IFPS specified,
1059 679C B2 00 82 STBD R64,-BEEPON set it
1060 679F ENDMIF
1061 679F 75 90 TSB R65 test default flag
1062 67A1 F4 03 IFPS specified,
1063 67A3 B2 77 83 STBD R65,-DEFAULT set it
1064 67A6 ENDMIF
1065 67A6 76 90 TSB R66 test standby flag
1066 67A8 F4 04 IFPS specified,
1067 67AA 8D ICB R66 make i=-1
1068 67AB B2 83 83 STBD R66,-STAND? set it
1069 67AE ENDMIF
1070 67AE 77 90 TSB R67 test timeout flag
1071 67B0 F4 08 IFPS specified,
1072 67B2 8A DEB R67 make i=0 (timeout on)
1073 67B3 F5 02 IFNG if was 0
1074 67B5 A8 01 LDB R67,-1 make it 1 (timeout off)
1075 67B7 ENDMIF
1076 67B7 B2 84 83 STBD R67,-THOUT? set it
1077 67BA ENDMIF
1078 67BA 78 90 TSB R70 test verify flag
1079 67BC F4 03 IFPS specified,
1080 67BE B2 61 84 STBD R70,-CRDSTS set it
1081 67C1 ENDMIF
1082 67C1 79 90 TSB R71 test degrees flag
1083 67C3 F4 03 IFPS specified,
1084 67C5 B2 81 82 STBD R71,-DRG set it
1085 67C8 ENDMIF
1086 67C8

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 26
-----
set status of system switches
1139 681B 44 RSC 'D' Default on
1140 681C 53 RSC 'S' Standby on
1141 681D 54 RSC 'T' Timeout on
1142 681E 56 FLGTB: RSC 'V' Verify on
1143 681F 44 RSC 'D' Degrees
1144 6820 54 RSC 'T' Trace flow/vars
1145 6821 4D RSC 'N' NDV
1146 6822 41 RSC 'R' RM/PM
1147 6823

```

ICOMAS
 Not Manufacturer Supported
 Recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 25
-----
set status of system switches
1087 67C8 7A 90 TSB R72 test trace flag
1088 67CA F4 1A IFPS specified,
1089 67CC F6 04 IFZR flow only,
1090 67CE A8 80 LDB R72,-OORON set negative
1091 67D0 F0 11 ELSE otherwise,
1092 67D2 C8 10 CMB R72,-8V-F was that vars?
1093 67D4 F6 04 IFEQ yes,
1094 67D6 A8 01 LDB R72,-0001H set odd
1095 67D8 F0 09 ELSE no,
1096 67DA C8 0E CMB R72,-81-F was that both?
1097 67DC F6 04 IFEQ yes,
1098 67DE A8 81 LDB R72,-0081H set negative and odd
1099 67E0 F0 01 ELSE no,
1100 67E2 92 CDB R72 set off (must be 1)
1101 67E3 ENDMIF
1102 67E3 ENDMIF
1103 67E3 ENDMIF
1104 67E3 B2 A8 82 STBD R72,-TRFLAG set the flag
1105 67E6 ENDMIF
1106 67E6 ENDMIF
1107 67E6 7B 90 TSB R73 test ndy flag
1108 67E8 F4 0A IFPS specified,
1109 67EA 7F 1F DF RHM R77,-NDTWS clear bit (ndy)
1110 67EC 7B 90 TSB R73 dry?
1111 67EE F6 03 IFZR yes,
1112 67F1 7F EA 20 RDB R77,-BITWS set bit (dry)
1113 67F4 ENDMIF
1114 67F4 ENDMIF
1115 67F4 7C 90 TSB R74 test an/pm flag
1116 67F6 F4 0A IFPS specified,
1117 67F8 7F 1F EF RHM R77,-NDTWS clear bit (an/pm)
1118 67FA 7C 90 TSB R74 24hr?
1119 67FC F6 03 IFZR yes,
1120 67FE 7F EA 10 RDB R77,-BITWS set bit (24hr)
1121 6802 ENDMIF
1122 6802 ENDMIF
1123 6802 7F B2 F6 83 STBD R77,-PSSSTAT save new appt/time status
1124 6806 9F RM
1125 6807
1126 6807 53 10 B4 16 ISIFLG LDBD R23,X20,FLGTBL get valid flag char
1127 6808 68
1128 680C 50 89 ICB R20 increment table pointer
1129 680E 52 18 2E CMB R22,-'.' (unspecified)?
1130 6811 F7 02 IFNE no,
1131 6813 13 10 CMB R22,R23 equal valid char?
1132 6815 ENDMIF
1133 6816 RM
1134 6816 41 FLGTBL RSC 'R' Alarm on
1135 6817 4C RSC 'L' Autoloop on
1136 6818 49 RSC 'I' esc-i/r on
1137 6819 53 RSC 'S' Shell on
1138 681A 42 RSC 'B' Beep on

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 27
-----
status of system switches
1149 6823 00 2E BYT 0,56
1150 6825 98 STAT4, RJM to be sure
1151 6826 6E A9 0D 00 LDM R56,-813 get string length
1152 682A 0A E5 PUMD R56,-R12 put it on stack
1153 682C CE FF FF JSB -RSHEN get some space
1154 682F 56 0A E5 PUMD R26,-R12 put address on stack
1155 6832 70 B1 16 68 LDMD R60,-FLGTBL get assumed values
1156 6836 68 B1 1E 68 LDMD R53,-FLGTBL for status
1157 683A 7F B0 F6 83 LDBD R77,-PSSSTAT get appt/time status
1158 683E 43 A8 04 LDB R03,-BITW2 mask for alarm bit
1159 6841 3F C7 RHM R03,R77 test alarm bit
1160 6843 F7 03 IFNZ alarm off,
1161 6845 70 A8 61 LDB R60,-'a' change status
1162 6848 ENDMIF
1163 6848 43 B0 A8 84 LDBD R03,-AUTOLP test autoloop
1164 684C F7 03 IFNZ autoloop off,
1165 684E 71 A8 6C LDB R61,-'1' change status
1166 6851 ENDMIF
1167 6851 43 B0 5A 82 LDBD R03,-IOBYIE test esc-i/r
1168 6855 F3 03 IFOD off,
1169 6857 72 A8 69 LDB R62,-'1' change status
1170 685A ENDMIF
1171 685A 43 B0 59 82 LDBD R03,-SHELOW get shell flag
1172 685E F6 03 IFZR shell off
1173 6860 73 A8 73 LDB R63,-'s' change status
1174 6863 ENDMIF
1175 6863 43 B0 A0 82 LDBD R03,-BEEPON get beep flag
1176 6867 F6 03 IFZR beep off,
1177 6869 74 A8 62 LDB R64,-'b' change status
1178 686C ENDMIF
1179 686C 43 B0 77 83 LDBD R03,-DEFAULT get default flag
1180 6870 F6 03 IFZR default off,
1181 6872 75 A8 64 LDB R65,-'d' change status
1182 6875 ENDMIF
1183 6875 43 B0 B3 83 LDBD R03,-STAND? get standby flag
1184 6879 F6 03 IFZR standby off,
1185 687B 76 A8 73 LDB R66,-'s' change status
1186 687E ENDMIF
1187 687E 43 B0 B4 83 LDBD R03,-THOUT? get timeout flag
1188 6882 F3 03 IFOD timeout off,
1189 6884 77 A8 74 LDB R67,-'t' change status
1190 6887 ENDMIF
1191 6887 43 B0 61 84 LDBD R03,-CRDSTS get card status
1192 688B F2 03 IFEV verify off,
1193 688D 68 A8 76 LDB R53,-'v' change status
1194 6890 ENDMIF
1195 6890 43 B0 81 82 LDBD R03,-DRG get degrees flag
1196 6894 F6 03 IFZR radians,
1197 6896 6C A8 52 LDB R54,-'R' change status
1198 6899 ENDMIF
1199 6899 43 B0 A8 82 LDBD R03,-TRFLAG get trace flag
1200 689D 6D DRP 55 save code
1201 689E F6 04 IFZR trace off,

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 28
-----
>>>>>> status of system switches <<<<<<<
1202 68A0 A8 74 LDB R55,'1' change status
1203 68A2 F0 04 ELSE trace something.
1204 68A4 F4 04 IFPS trace vars.
1205 68A6 A8 56 LDB R55,'V' change status
1206 68A8 F0 04 ELSE trace flow?
1207 68AA F2 02 IFEV yes,
1208 68AC A8 46 LDB R55,'F' change status
1209 68AE ENDIF
1210 68AF ENDIF
1211 68B0 ENDIF
1212 68B2 43 A8 20 LDB R03,'BITMS MDV mask
1213 68B4 C7 DMY R03,R77 test MDV bit
1214 68B6 F7 03 IFMZ DMY,
1215 68B8 6E A8 44 LDB R56,'D' change status
1216 68B0 ENDIF
1217 68B2 43 A8 10 LDB R03,'BITMA 12/24 mask
1218 68B4 C7 DMY R03,R77 test 12/24 bit
1219 68B6 F7 03 IFMZ 24 hour,
1220 68B8 6F A8 2A LDB R57,'A' change status
1221 68C0 ENDIF
1222 68C2 70 16 E5 PUND R60,R26 to space
1223 68C4 6B E5 PUND R53,R26 and some more
1224 68C6 9E RTN
1225 68C8

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 30
-----
>>>>>> file # = name?, line# of # <<<<<<<
1279 6930 9E RTN
1280 6931
1281 6931 67 90 GIMFLE TSB R47 check for line number
1282 6933 F7 04 IFMZ too big,
1283 6935 CE FF FF JSBN =ERR1+ report error
1284 6938 59 BYI 89D file number too big
1285 6939 ENDIF
1286 6939 7E 25 A1 LDM R76,R45 move line number
1287 693C CE FF FF JSB =10? get file
1288 693F F8 EF REN not there
1289 6941 CE FF FF JSB =FSEEX get line
1290 6944 9E RTN
1291 6945

```

NOMAS
 Not Manufacturer Supported
 Reprint agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 29
-----
>>>>>> file # = name?, line# of # <<<<<<<
1227 68C6 10 2E BYI 20,56
1228 68C8 CE FF FF FILED JSB =RESUM get some space
1229 68C0 08 BYI 8D one name
1230 68CC F8 F7 REN no room, leave
1231 68CE 54 2E A1 LDM R24,R56 move length
1232 68D1 13 A2 STB R24,R23 set strang flag
1233 68D3 F0 04 JMP DOWSIF go to it
1234 68D5
1235 68D5 10 20 BYI 20,55
1236 68D7 53 92 LIMEN CLD R23 clear strang flag
1237 68D9
1238 68D9 CE FF FF DOWSIF JSB =ONEI get line number
1239 68DC 98 BIN for sure
1240 68DD 67 C8 90 CMB R47,'90C negative number?
1241 68DE FA 0C IFCY yes,
1242 68E2 60 B1 6B 82 LDM R40,'EDNAME get editfile name
1243 68E6 60 B1 7D 82 LDM R55,'CURLNM get editfile line#
1244 68EA 6F 92 CLD R57 clear garbage
1245 68EC F0 2E ELSE no,
1246 68EE CE 31 69 JSB =GIMFLE get the 10file
1247 68F1 60 1E 05 03 LDM R40,X36,NAMEW get the name of #
1248 68F5 00
1249 68F6 F7 08 JZR NOFILE name=0 if not declared
1250 68F8 6D 05 0C 00 LDM R55,X36,LIMEN get the line# in #
1251 68FE F9 0B CMB R57 clear garbage
1252 6900 54 93 IFEN if not valid
1253 6902 53 90 CLM R24 get null string
1254 6904 F6 1A ISB R23 doing what?
1255 6906 99 JNZ MNEOUT put out name
1256 6907 CE FF FF BCD for INF 10
1257 690A 9E JSB =INF 10 put out INF
1258 690B RTN
1259 690B CE FF FF ELSE
1260 690E F9 0C JSB =FLOPEN file really there?
1261 6910 99 IFEN not found
1262 6911 6D 8F BCD make line # -
1263 6913 98 MCM R55 to flag non-
1264 6914 70 A8 80 BIN existant file
1265 6917 71 30 A1 LDM R60,'BOH get a top bit
1266 691A 60 C3 LDM R61,R60 copy it
1267 691C ADM R40,R60 make chars underlined
1268 691E ENDIF
1269 691E ENDIF
1270 691E 53 90 TSB R23 test flow flag
1271 691E F7 0A IFNZ strang,
1272 6920 54 0A MNEOUT PUND R24,'R12 put out length
1273 6923 56 E5 PUND R26,'R12 put out address
1274 6925 60 16 E5 PUND R40,'R26 put string in space
1275 6928 F0 06 ELSE number.
1276 692A 6E A8 FF LDB R54,'OFFH get integer flag
1277 692D 6B 0A E5 PUND R50,'R12 number to stack
1278 6930 ENDIF

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 31
-----
>>>>>> reassignio device name <<<<<<<
1293 6945 A1 BYI 241
1294 6946 98 REASW. BIN for sure
1295 6947 CE FF FF JSB =NELJSB get a device name
1296 694A FF FF DEF FLSTAK off the stack
1297 694C F8 F6 REN error, give up
1298 694E CE 0B 74 JSB =GETPD+ name already used?
1299 6951 CE FF FF JSB =CLRERR clear any errors
1300 6954 F8 04 IFEZ yes,
1301 6956 CE FF FF JSBN =ERR1+ error
1302 6959 40 BYI 64D duplicate name
1303 695A
1304 695A 5A C8 03 CMB R32,'3 name size?
1305 695D FA 04 IFCY >2,
1306 695F CE FF FF MANSZE JSBN =ERR1+ error
1307 6962 2A BYI 42D name too long
1308 6963 ENDIF
1309 6963 76 06 E5 PUND R66,'R06 save name
1310 6966 CE FF FF JSB =NELJSB get another device
1311 6969 FF FF DEF FLSTAK off stack
1312 696B 7E 06 E3 POND R76,'R06 get back name
1313 696E F8 D4 REN error, give up
1314 6970 CE 0B 74 JSB =GETPD+ find device
1315 6973 F9 04 IFEN not there,
1316 6975 CE FF FF JSBN =ERR1+ error
1317 6978 3F BYI 63D invalid filespec
1318 6979 ENDIF
1319 6979 5A C8 03 CMB R32,'3 name size?
1320 697C F8 E1 JCY MANSZE >2, give up
1321 697E 7E 1E 07 03 STMD R76,X36,DNAME set new name
1322 6983 9E RTN
1323 6984

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 32
=====
listio8 - return string for assignio
1325 6984 00 2E BYI 0,56
1326 6986 52 AB 01 DISPS. LDB R22,=0001H set flag for disp device
1327 6989 F0 0B JMP DOLST go do it
1328 698B
1329 698B 00 2E BYI 0,56
1330 698D 52 AB 00 PRNPS. LDB R22,=0080H set flag for print device
1331 6990 F0 04 JMP DOLST go do it
1332 6992
1333 6992 00 2E BYI 0,56
1334 6994 52 92 LSTI8. CLM R22 set flag for all
1335 6996
1336 6996 98 DOLST BIN for sure
1337 6997 54 93 CLM R24 Assume no devices
1338 6999 6E 93 CLM R56 Clear for load
1339 699B 80 0C 82 LDBD R56,=DEVCHI Number of devices
1340 699E F7 48 JZR LDPDME No devices
1341 69A0 84 LDB R56 Times 2
1342 69A1 84 LDB R56 Times 4 = bytes needed
1343 69A2 CE FF FF JSB =RSMEN- get some space
1344 69A5 F8 0C REN no room, give up
1345 69A7 43 80 8D 82 LDBD R03,=PLSIAT Get PLSIAT
1346 69AB CF 20 ANM R03,=B1IHS Get offio bit
1347 69AD F6 39 JNZ LDPDME If offio, return null
1348 69AF CE FF FF JSB =IHERE? get the devfile
1349 69B2
1350 69B2
1351 69B2 54 8B DEN R24 We know it is there since
1352 69B4 50 A9 2C 3A LDM R20,=',, we looked at DEVCHI
1353 69B8 24 A3 STM R20,R44 to -1 (for comma)
1354 69BA 16 A1 LDM R20,R26 get a ,:
1355 69BC 89 IEM R20 set up for pushes
1356 69BD LOOP move name loop
1357 69BD 66 1E 85 03 LDBD R46,X36,DNAME get the name
1358 69C1 00
1359 69C2 52 90 ISD R22 what are we doing?
1360 69C4 F7 08 JZR DNOU7 all names
1361 69C6 53 84 05 00 LDBD R23,X36,SETUP get the disp/print flag
1362 69C8 12 C7 ANM R23,R22 test it against desired
1363 69CA F7 10 IFNZ yes,
1364 69CC 64 16 E5 PUND R44,=R26 but name ,:
1365 69CE 54 C8 04 00 ADH R24,=84 add to length
1366 69D0 F6 04 CND R47,=', one char name?
1367 69D2 56 8B IFEO yes,
1368 69D4 54 8B DEN R26 urate over blank
1369 69D6 DEN R24 adjust length
1370 69D8 ENDIIF
1371 69DA CE FF FF JSB =SKPCWK go to next line
1372 69DC F6 0A WHNZ not end of file
1373 69DE 54 91 ISM R24 any found?
1374 69E0 F5 01 IFNG no,
1375 69E2 93 CLM R24 make it null string
1376 69E8 ENDIIF

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 33
=====
listio8 - return string for assignio
1377 69E8 54 0A E5 LDPDME PUND R24,=R12 length to stack
1378 69EA 50 E5 PUND R20,=R12 address to stack
1379 69ED 9E RIM
1380 69EE

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 34
=====
get endline string
1382 69EE 00 2E BYI 0,56
1383 69F0 98 ENDL8. BIN to be sure
1384 69F1 6E 93 CLM R56 clear for load
1385 69F3 80 C4 82 LDBD R56,=EOLLEN get the string length
1386 69F6 0A E5 PUND R56,=R12 length to stack
1387 69F8 CE FF FF JSB =RSMEN- get some space
1388 69FB F8 F0 - REN no room, give up
1389 69FD 56 0A E5 PUND R26,=R12 address to stack
1390 6A00 6E 91 ISM R56 anything to get?
1391 6A02 F7 0B IFNZ yes,
1392 6A04 40 AB 38 LDB R00,=70 compute the
1393 6A07 2E C4 SDB R00,R56 reg pointer
1394 6A09 41 B1 C1 82 LDBD R*,=EOL load the string
1395 6A0D 16 E5 PUND R*,=R26 put in space
1396 6A0F ENDIIF
1397 6A10 9E RIM
1398 6A10

```

NO MARS
 Not a real Mars. It's just a computer.

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 35
=====
get current key definition
1400 6A10 18 2E BYI 30,56
1401 6A12 CE FF FF DFKYS. JSB =DEIADR get char address
1402 6A15 5E 0A E3 PUND R36,=R12 get char length
1403 6A18 8B DEN R36 test for 1 char
1404 6A19 F7 04 IFNZ no,
1405 6A1B CE FF FF JSBW =ERR1+ report error
1406 6A1E 59 BYI 85D invalid data
1407 6A1F ENDIIF
1408 6A1F 60 1C E0 POND R40,=R34 get the char
1409 6A22 CE FF FF JSB =KVADDR find it in the key file
1410 6A25 F9 05 IFEM not there,
1411 6A27 6C 93 CLM R54 get null string
1412 6A29 0A E5 PUND R54,=R12 for return
1413 6A2B 9E RIM
1414 6A2C ENDIIF
1415 6A2C 65 1E E1 POND R45,=R36 get line#, length
1416 6A2F 6E 93 CLM R56 clear for load
1417 6A31 27 80 LDB R56,R47 get length
1418 6A33 CE FF FF JSB =RSMEN- get some space
1419 6A36 F8 F3 REN none, give up
1420 6A38 6E 0A E5 PUND R56,=R12 put length on stack
1421 6A3B 56 E5 PUND R26,=R12 put address on stack
1422 6A3D LOOP move string loop
1423 6A3D 67 1E E0 POND R47,=R36 get a char
1424 6A40 16 E4 PUND R47,=R26 put in string
1425 6A42 6E 8B DEN R56 dec char count
1426 6A44 F6 F7 WHNZ until done
1427 6A46 9E RIM
1428 6A47

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 36
-----
>>>>>> get current delay setting <<<<<<<
1430 6A47 00 2D BYI 0,55
1431 6A49 98 DELY? RIM for sure
1432 6A4A 73 B1 B7 82 LDND R63,=DELAY get delay
1433 6A4E
1434 6A4C 40 AB 33 BIN7RL LDB R00,=63 get register pointer
1435 6A51 LOOP move to stack
1436 6A51 50 01 R0 LDB R20,R# get next byte (low-high)
1437 6A52 CE FF FF JSB =IDBCDZ convert to bcd
1438 6A57 64 AB FF LDB R44,=OFFH get integer flag
1439 6A5A 65 10 A1 LDM R45,R20 add number
1440 6A5D 57 92 CLB R47 clear top byte
1441 6A5F 05 0A E5 PUMD R40,=R12 onto stack
1442 6A62 40 88 ICB R00 incr reg pointer
1443 6A64 C8 38 CMB R00,=70 done?
1444 6A66 FA E9 MWC no, go again
1445 6A68
1446 6A68 50 AB 04 LDB R20,=4D get loop counter
1447 6A6B LOOP multiply loop
1448 6A6B 7C A9 FF 56 LDM R74,=OFFH,56C,02C,00C get 256 integer
1449 6A71 78 0A E5 PUMD R70,=R12 onto stack
1450 6A74 99 BCD for multiply
1451 6A75 CE FF FF JSB =MPYRDI 256K
1452 6A78 CE FF FF JSB =MODRDI 256K+Y
1453 6A7B 98 BIN for decr
1454 6A7C 50 8A DCB R20 decrement count
1455 6A7E F6 EB WNZ not done
1456 6A80 60 B1 FF FF LDND R40,=INEFAC get 1 second
1457 6A84 0A E5 PUMD R40,=R12 onto stack
1458 6A86 CE FF FF JSB =DIV2 convert to seconds
1459 6A89 9E RIM
1460 6A8A

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 38
-----
>>>>>> Flag stuff <<<<<<<
1515 6A61 ENDF
1516 6A61 65 92 CLB R45 clear for shift
1517 6A63 67 87 LRM R47 divide by 8
1518 6A65 87 LRM R47 to get
1519 6A66 87 LRM R47 byte number
1520 6A67 65 86 LRB R45 move bit count
1521 6A69 99 RCD down to
1522 6A6A 86 LRB R45 start of byte
1523 6A6B CE FF FF JSB =GETADR get the flag string
1524 6A6E 5E 93 CLM R36 assume no flag
1525 6B00 5A 0A E3 PUMD R32,=R12 get flag length
1526 6B03 5C 1A C3 ADM R34,R32 compute end of
1527 6B06 66 C1 CMM R46,R32 length >= byte number?
1528 6B08 9E RIM
1529 6B09
1530 6B09 28 2D BYI 50,55
1531 6B0B CE E9 6A FLAG? JSB =FLGPRC go do some stuff
1532 6B0E FB 35 JCY NUMOUT no flag, false
1533 6B10 5C 8B DCM R34 adjust end
1534 6B12 2E C5 SDM R34,R46 compute byte pointer
1535 6B14 5E 1C A4 LDND R36,R34 get byte
1536 6B17 FMOB11 LOOP find bit loop
1537 6B17 65 8A DCB R45 dec bit count
1538 6B19 F5 06 IFNG done,
1539 6B1B 5E CF 01 00 RHM R36,=61 mask out garbage
1540 6B1F F0 24 JNP NUMOUT return number
1541 6B21 ENDF
1542 6B21 5E 86 LRB R36 shift bits
1543 6B23 F0 F2 WNP go again
1544 6B25

```

NOMAS

NOT Manufacturer Supported
Recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 37
-----
>>>>>> Flag stuff <<<<<<<
1462 6A8A 00 38 2E BYI 0,70,56
1463 6A8B CE FF FF FLAG? JSB =DNEB get flag setting
1464 6A90 66 91 ISM R46 0?
1465 6A92 F7 07 IFNZ no,
1466 6A94 8B DCM R46 1?
1467 6A95 F7 04 IFNZ no,
1468 6A97 CE FF FF FLGERR JSB =ERR1 no go
1469 6A9A 59 BYI 89D
1470 6A9B ENDF
1471 6A9B ENDF
1472 6A9B 7E 06 E4 PUMD R76,=R06 save flag for later
1473 6A9E CE E9 6A JSB =FLGPRC do some stuff
1474 6AA1 FA 05 IFCY bit past end of flags
1475 6AA3 5F 06 E2 PUMD R37,=R06 trash flag
1476 6AA6 F0 EF JNP FLGERR do the error
1477 6AAB ENDF
1478 6AAB 5F 06 E2 PUMD R37,=R06 get the flag back
1479 6AAB 5E 88 ICB R36 make a mask (was 0)
1480 6AAB 6E 1A A1 LDM R56,R32 get the size
1481 6AB0 5E 18 A3 SIM R34,R30 move pointer
1482 6AB3 CE FF FF JSB =RSMEN- get the ram
1483 6AB6 FB D1 REN nope, give up
1484 6AB8 6E 0A E5 PUMD R56,=R12 size to stack
1485 6AB8 5E E5 PUMD R26,=R12 address to stack
1486 6ABD 2E C3 ADM R26,R56 compute end of target
1487 6ABF LOOP move data loop
1488 6ABF 6E 8B DCM R56 dec char count
1489 6AC1 F4 C6 RRG done
1490 6AC3 43 18 E2 PUMD R03,=R30 get a byte
1491 6AC6 66 91 ISM R46 test byte number
1492 6AC8 F6 18 IFZR one to change,
1493 6ACA LOOP bit loop
1494 6ACA 65 90 JSB R45 test bit number
1495 6ACC F7 06 JZR BITFND found, go on
1496 6ACE 5E 84 LCB R36 shift mask
1497 6AD0 65 8A DCB R45 dec bit number
1498 6AD2 F0 F6 WNP go again
1499 6AD4 5F 90 JSB R37 flag 0?
1500 6AD6 F6 07 IFZR yes,
1501 6AD8 5E BE MCB R36 complement mask
1502 6ADA 43 1E C7 RHM R03,R36 and AMD flag
1503 6ADD F0 03 ELSE no,
1504 6ADF 43 1E 94 DRB R03,R36 or in flag
1505 6AE2 ENDF
1506 6AE2 ENDF
1507 6AE2 66 8B DCM R46 dec byte number
1508 6AE4 43 1E E6 PUMD R03,=R26 char to output
1509 6AE7 F0 D6 WNP until done
1510 6AE9
1511 6AE9 CE FF FF FLGPRC JSB =DNEB get the bit number
1512 6AE1 66 91 ISM R46 reg number?
1513 6AE5 F5 01 IFNG yes,
1514 6AF0 93 CLM R46 make it 0

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 39
-----
>>>>>> get current margin/width/pwidth <<<<<<<
1546 6B25 00 2D BYI 0,55
1547 6B27 5E B1 B0 83 MARG? LDND R36,=RNARG get margin
1548 6B2B F0 18 JNP NUMOUT output the number
1549 6B2D
1550 6B2D 00 2D BYI 0,55
1551 6B2F 5E 93 CLM R36 clear for load
1552 6B31 B0 CC 82 LDND R36,=PRNTLM get print width
1553 6B34 F7 08 JZR INFOUT 0 mean inf
1554 6B36 F0 0D JNP NUMOUT output the number
1555 6B38
1556 6B38 00 2D BYI 0,55
1557 6B3A 5E 93 CLM R36 clear for load
1558 6B3C B0 CB 82 LDND R36,=DISPLM get the display width
1559 6B3F F6 04 IFZR 0,
1560 6B41 CE FF FF INFOUT JSB =INFIO output inf
1561 6B44 9E RIM
1562 6B45 ENDF
1563 6B45 CE FF FF NUMOUT JSB =COMBIN convert to real
1564 6B48 60 0A E5 PUMD R40,=R12 put on stack
1565 6B4B 9E RIM
1566 6B4C

```



```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 44
*****
>>>>>> hex and/or xor two strings <<<<<<<
1688 6C20 2A 2E BYI 52,56
1689 6C22 52 AB FF HANDR. LDB R22,+OFFH set and flag
1690 6C25 F0 0B JMP DOHROX go do it
1691 6C27
1692 6C27 2A 2E BYI 52,56
1693 6C29 52 92 HKDRR. CIB R22 set xor flag
1694 6C2B F0 05 JMP DOHROX go do it
1695 6C2D
1696 6C2D 2A 2E BYI 52,56
1697 6C2F 52 AB 01 MORR. LDB R22,+1 set or flag
1698 6C32
1699 6C32 52 06 E4 DOHROX PORD R22,+R06 save flow flag
1700 6C35 CE 1B 6F JSB +RSCB. convert one param
1701 6C38 F8 0D JEM POPHMX error, give up
1702 6C3A 64 0A E3 PORD R44,-R12 get result
1703 6C3D 06 E5 PORD R44,+R06 save result
1704 6C3F CE 1B 6F JSB +RSCB. convert the other
1705 6C42 64 06 E3 PORD R44,-R06 get back the first
1706 6C45 0A E5 PORD R44,+R12 put it on stack
1707 6C47 52 06 E2 POPHMX PORD R22,-R06 get flow flag
1708 6C4A F8 0D JEM error, give up
1709 6C4C CE 85 6D JSB +DOHROX go do stuff
1710 6C4F CE 9D 6F JSB +HENDR. convert result
1711 6C52 9E
1712 6C53

```

Not Manufacturer Supported
 Secretant serves NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 45
*****
>>>>>> hex/ascii rotate/shift string <<<<<<<
1767 6CBA 60 0A E3 PORD R40,-R12 save count
1768 6CBB CE 1E 6E JSB +RSCB. convert to ascii
1769 6CCE F8 06 JEM ROTERR error out
1770 6CE2 60 0A E5 PORD R40,+R12 restore count
1771 6CE5
1772 6CE5 CE 1E 6E JSB +R01B. rotate that much
1773 6CE8 50 06 E3 ROTERR PORD R20,-R06 restore R20/21
1774 6CEB F8 0D JEM error out
1775 6CED 54 DRP 24 get the result
1776 6CEE CE FF FF JSB +GETROM address
1777 6CF1 56 0A E3 PORD R26,-R12 and length
1778 6CF4 4A 0B 04 00 RORL R12,+B4 we didn't do that
1779 6CF8 52 14 A1 LDM R22,R24 copy address
1780 6CFB 51 0B 01 CNB R21,+1D rotate?
1781 6CFE F7 1E IFNE no,
1782 6CE0 66 18 A1 LDM R46,R30 direction?
1783 6CE3 4 recover count from R01B
1784 6CE3 F6 07 IFZR complete shift?
1785 6CE5 50 90 TSB R20 really?
1786 6CE7 F6 03 IFZR yes,
1787 6CE9 66 16 A1 LDM R46,R26 get length
1788 6CEE
1789 6CEE
1790 6CEE F5 07 IFNG left,
1791 6CEE 66 8D TCM R46 positive count
1792 6CF0 52 16 C3 RDM R22,R26 find start of
1793 6CF3 26 E5 SSM R22,R46 filler
1794 6CF5
1795 6CF5
1796 6CF5 66 8B DCM R46 fill loop
1797 6CF7 F4 05 JNG SRDME? dec count
1798 6CF9 51 12 E4 PORD R21,+R22 no more
1799 6CFE F0 F7 UHMP put out filler
1800 6CFE
1801 6CFE 52 16 A1 SRDME? LDM R22,R26 until done
1802 6D01 8B DCM R22 get length
1803 6D02 90 CLE adjust for add
1804 6D03 50 90 TSB R20 set right flag
1805 6D05 F7 53 JZR SRDOME any more bits?
1806 6D07 F5 04 IFNG no, all done
1807 6D09 9C ICE left,
1808 6D0A 8C TCB R20 set left flag
1809 6D0B 52 93 CLM R22 positive count
1810 6D0D
1811 6D0D 52 14 C3 RDM R22,R24 cancel preset
1812 6D10 51 0B 01 CNB R21,+1D compute rotate and
1813 6D13 F6 02 IFEQ rotate?
1814 6D15 12 A4 LDBD R21,R22 yes,
1815 6D17
1816 6D17 F9 20 IFEN get end byte
1817 6D19
1818 6D19 53 14 A4 LOOP left,
1819 6D1C 52 04 01 00 LDBD R23,R24 bit sr loop
LDBD R22,R24 get leading byte
LDBD R22,R24,B1 get next byte

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 45
*****
>>>>>> hex/ascii rotate/shift string <<<<<<<
1714 6C53 00 3B 2E BYI 0,70,56
1715 6C56 50 AB 01 MSHF. LDB R20,+1 set hex flag
1716 6C59 F0 15 JMP MSHF go do it
1717 6C5B
1718 6C5B 2B 2E BYI 50,56
1719 6C5D 50 AB 01 HROT. LDB R20,+1 set hex flag
1720 6C60 F0 04 JMP HROT go do it
1721 6C62
1722 6C62 2B 2E BYI 50,56
1723 6C64 50 92 ARD1. CIB R20 clear hex flag
1724 6C66 51 AB 01 HROT LDB R21,+1D set rotate flag
1725 6C69 F0 16 JMP DOHRS go do it
1726 6C6B
1727 6C6B 00 3B 2E BYI 0,70,56
1728 6C6E 50 92 MSHF. CIB R20 clear hex flag
1729 6C70 CE FF FF MSHF JSB +OME6 get the fill bit
1730 6C73 66 91 TSM R46 =0?
1731 6C75 F7 0B IFNZ no,
1732 6C77 8D DCM R46 <1?
1733 6C78 F7 04 IFNZ no,
1734 6C7A CE FF FF JSB +ERR1+ error out
1735 6C7D 59 BYI 89D
1736 6C7E
1737 6C7E 8B DCM R46 make it all ones
1738 6C7F
1739 6C7F 11 A2 EMOIF SIB R46,R21 save filler
1740 6C81
1741 6C81 60 0A E3 DOHRS PORD R40,-R12 get bit count
1742 6C84 E5 PORD R40,+R12 put it back
1743 6C85 E5 PORD R40,+R12 and a copy
1744 6C86 E5 PORD R40,+R12 and a copy
1745 6C87 CE FF FF JSB +OME1 get it as int
1746 6C8A 98 BEM for sure
1747 6C8B 65 91 TSM R45 0?
1748 6C8D F6 0C IFZR yes,
1749 6C8F 60 0A E3 PORD R40,-R12 trash number
1750 6C92 E3 PORD R40,-R12 trash number
1751 6C93 50 90 TSB R20 doing hex?
1752 6C95 F7 0B RZR no, all done
1753 6C97 CE FF FF JSB +UPC6. uppercase it
1754 6C9A 9E RIM done
1755 6C9B
1756 6C9B CE 64 6D JSB +EN18 get an 8
1757 6C9E CE FF FF JSB +PEL18B compute the
1758 6CA1 FF FF DEF REN10. extra bit count
1759 6CA3 CE FF FF JSB +OME6 get the result
1760 6CA6 50 B2 64 84 SIBD R20,-1MPM3 save hex flag
1761 6CAB 26 A0 LDB R20,R46 save it (0-7)
1762 6CAC 06 E5 PORD R20,+R06 save R20/21
1763 6CAE CE 64 6D JSB +EN18 get another 8
1764 6CB1 CE FF FF JSB +INIDIV compute the byte count
1765 6CB4 42 80 64 84 LDBD R02,-1MPM3 get hex flag
1766 6CB8 F7 0B IFNZ set,

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 47
*****
>>>>>> hex/ascii rotate/shift string <<<<<<<
1820 6D20 56 8B DCM R26 dec count
1821 6D22 F4 36 JNG SRDOME done
1822 6D24 F6 03 IFZR last byte
1823 6D26 52 11 A0 LDB R22,R21 get filler/end
1824 6D29
1825 6D29 40 10 A0 LDB R00,R20 get bit count
1826 6D2C LOOP bit loop
1827 6D2E 52 85 LLM R22 shift left
1828 6D2E 40 8A DCB R00 dec count
1829 6D30 F6 FA UHMZ until done
1830 6D32 53 14 E4 PORD R23,+R24 put back, move on
1831 6D35 F0 E2 UHMP until done
1832 6D37 F0 21 ELSE right,
1833 6D39 54 16 C3 RDM R24,R26 compute end of string
1834 6D3C LOOP bit sr loop
1835 6D3E 52 14 E2 PORD R22,-R24 get trailing byte
1836 6D3F 53 B4 FF FF LDBD R23,R24,(0-10) get previous byte
1837 6D43 56 8B DCM R26 dec count
1838 6D45 F4 13 JNG SRDOME done
1839 6D47 F6 03 IFZR last byte,
1840 6D49 53 11 A0 LDB R23,R21 get filler/end
1841 6D4C
1842 6D4C 40 10 A0 LDB R00,R20 get bit count
1843 6D4F LOOP bit loop
1844 6D51 53 87 LRM R23 shift right
1845 6D51 40 8A DCB R00 dec count
1846 6D53 F6 FA UHMZ until done
1847 6D55 52 14 A6 SIBD R22,R24 put back, move on
1848 6D58 F0 E2 UHMP until done
1849 6D5A
1850 6D5A 42 80 64 84 SDOHME LDBD R02,-1MPM3 get hex flag
1851 6D5E F7 03 IFNZ set,
1852 6D60 CE 9D 6F JSB +HEX6. convert to hex
1853 6D63
1854 6D63 9E RIM
1855 6D64

```

```

KARMA 08/31/82:-----
ITEM LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 48
-----
>>>>>> get some constants <<<<<<<
1857 6064 42 AR 08 INTR LDB R02,+8C for base 8
1858 6067 98 B1M B1M for callers
1859 6068 64 AR FF LDB R44,+FF integer flag
1860 6068 65 93 CLT R45 clear
1861 606D 02 R0 LDB R45,R02 and load
1862 606F 60 OR E5 PUSD R40,+R12 push on stack
1863 6072 9E R1M R1M
1864 6073
1865 6073

```

```

KARMA 08/31/82:-----
ITEM LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 50
-----
>>>>>> special and/or xor two strings <<<<<<<
1920 60DE F0 0A ELSE what?
1921 60DE F6 05 JZR R2R xor,
1922 60E0 43 02 96 XRB R03,R02 xor then
1923 60E3 F0 03 ELSE or,
1924 60E5 43 02 94 ORB R03,R02 or then
1925 60E8 ENDDIF
1926 60E8 ENDDIF
1927 60E8 16 E4 PUSD R03,+R26 result out
1928 60EA F0 03 WHIP go on
1929 60EC

```

WOMAS
 Resident NBI Manufacturer Supported
 Resident series NOT to contact manufacturer

```

KARMA 08/31/82:-----
ITEM LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 49
-----
>>>>>> special and/or xor two strings <<<<<<<
1867 6073 2A 2E B1M B1M 52,56
1868 6075 52 AR FF RANDL LDB R22,-OFFH set and flag
1869 6078 F0 0B JMP DORADN go do it
1870 607A
1871 607A 2A 2E B1M B1M 52,56
1872 607C 52 92 AXORB CLB R22 set xor flag
1873 607E F0 05 JMP DORADN go do it
1874 6080
1875 6080 2A 2E B1M B1M 52,56
1876 6082 52 AR 01 AXORB LDB R22,+1 set or flag
1877 6085
1878 6085 54 DORADN DRP 24 get a string
1879 6086 CE FF FF JSB -GETRADN address
1880 6089 56 OR E3 POND R26,-R12 get the length
1881 608C CE FF FF JSB -GETRADN get another address
1882 608F 6E OR E3 POND R56,-R12 and length
1883 6092 16 C1 CHM R56,R26 compare the two
1884 6094 F8 06 IFMC if smaller
1885 6096 1E R3 SIN R56,R36 save small size
1886 6098 16 R1 LDM R56,R26 get large size
1887 609A F0 0A ELSE if larger
1888 609C 5C 1E R3 SIN R34,R36 sub address so
1889 609F 14 R1 LDM R34,R24 r24 has long
1890 60A1 54 1E R1 LDM R24,R36 and r34 has short
1891 60A4 56 R3 SIN R26,R36 move small size
1892 60A6
1893 60A6 5C 06 E5 POND R34,+R06 save us from RSMEN-
1894 60A9 CE FF FF JSB -RSMEN- get some room
1895 60AC 5C 06 E3 POND R34,-R06 get back that addr
1896 60AF F8 E1 REM none, leave
1897 60B1 52 90 TSB R22 test finu flag
1898 60B3 F5 05 IFMG and,
1899 60B5 5E OR E5 POND R36,+R12 short len to stack
1900 60B8 F0 03 ELSE or/xor,
1901 60BA 6E OR E5 POND R55,+R12 long len to stack
1902 60BD ENDDIF
1903 60BD 56 E5 POND R26,+R12 addr to stack
1904 60BF LOOP or loop
1905 60BF 6E 8D DCM R56 der count
1906 60C1 F4 AF RRG done
1907 60C3 43 1E E0 POND R03,+R34 get short char
1908 60C6 5E 8D DCM R36 der count
1909 60C8 F5 08 IFMG none left,
1910 60CA 43 9C CLB R03 clear char
1911 60CC 52 90 TSB R22 xor?
1912 60CE F6 02 JZR R2R yes,
1913 60D0 43 8E MCB R03 make mask ff
1914 60D2 ENDDIF
1915 60D2 ENDDIF
1916 60D2 42 14 E0 POND R0C,+R24 get long char
1917 60D5 52 90 TSB R22 what are we doing?
1918 60D7 F5 05 IFMG and,
1919 60D9 43 02 C7 RMB R03,R02 and then

```

```

KARMA 08/31/82:-----
ITEM LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 51
-----
>>>>>> trim trailing blanks <<<<<<<
1931 60EC 2A 2E B1M B1M 52,56
1932 60EE CE FF FF RTRMB JSB -GETRADN get string address
1933 60F1 5E OR E3 POND R36,-R12 get string length
1934 60F4 54 1C R1 LDM R24,R34 copy address
1935 60F7 1E C3 RDM R24,R36 point to end
1936 60F9 56 DRP R26 get trim char
1937 60FA CE FF FF JSB -GETRADN address
1938 60FD 52 OR E3 POND R22,-R12 and length
1939 6E00 F7 13 JZR TRADNE none, quit
1940 6E02 LOOP count loop
1941 6E02 43 14 E2 POND R03,-R24 get previous char
1942 6E05 50 16 R1 LDM R20,R26 get get trim pointer
1943 6E08 40 12 R1 LDM R00,R22 get trim length
1944 6E0B CE 66 70 JSB -SCHSIR search table
1945 6E0E F9 05 JEZ TRADNE not found
1946 6E10 5E 8B DCM R36 dec length
1947 6E12 F5 EE UMPS while not done
1948 6E14 89 ICM R36 make length 0
1949 6E15 5E OR E5 TRADNE POND R36,+R12 push length
1950 6E18 CE FF FF JSB -PUTREL put address back
1951 6E1B 9E R1M
1952 6E1C

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 52
-----
rotate string
1954 6E1C 28 2E BYI 50,56
1955 6E1E CE FF FF ROT8 JSB =ONERO1 get the count
1956 6E21 54 DRP 24 get the string
1957 6E22 CE FF FF JSB =GETADW address off stack
1958 6E25 54 12 A3 STM R24,R22 save for later
1959 6E28 6E 0A E1 POND R56,-R12 get the string len
1960 6E2B 1E R3 SIM R56,R36 save for combin
1961 6E2D CE FF FF JSB =RSNEN- get some room
1962 6E30 F8 E9 REN none, give up
1963 6E32 6E 0A E5 PUND R56,+R12 length to stack
1964 6E35 56 E5 PUND R26,+R12 address to stack
1965 6E37 60 E5 PUND R40,+R12 count back to stack
1966 6E39 CE FF FF JSB =CONBIN convert len
1967 6E3C 60 0A E5 PUND R40,+R12 len to stack
1968 6E3F 6E 06 E5 PUND R56,-R06 save us from ren10
1969 6E42 CE FF FF JSB =ELLISB do a mod(count,len)
1970 6E45 FF FF DEF REN10, to reduce the rotation
1971 6E47 CE FF FF JSB =ONEB get result
1972 6E4A 6E 06 E3 POND R56,-R06 get back length
1973 6E4D 90 CLE clear for caller
1974 6E4E 66 18 A3 SIM R46,R30 rotate left((-)/right(+))
1975 6E51 a save count for RRO18
1976 6E51 F5 11 IFNG left,
1977 6E53 8D TCM R46 complement count
1978 6E54 54 26 C3 ADM R24,R46 skip n chars
1979 6E57 CE 6F 6E JSB =NVPCE2 move rest
1980 6E5A 54 12 A1 LDM R24,R22 get start
1981 6E5D 40 26 A1 LDM R00,R46 get n
1982 6E60 CE 78 6E JSB =NVSIR move n
1983 6E63 9E RIM done
1984 6E64 ELSE
1985 6E64 54 2E C3 ADM R24,R56 go to end
1986 6E67 26 E5 SBM R24,R46 back up n chars
1987 6E69 CE 5D 6E JSB =NVPCE1 move then
1988 6E6C 54 12 A1 LDM R24,R22 get start
1989 6E6F 40 2E A1 LDM R00,R56 get len
1990 6E72 26 E5 SBM R00,R46 compute len-n
1991 6E74 CE 78 6E JSB =NVSIR move those
1992 6E77 9E RIM done
1993 6E78 ENDF
1994 6E78
1995 6E78 NVSIR LOOP move loop
1996 6E78 40 8B DCM R00 dec count
1997 6E7A F4 F8 RNG done
1998 6E7C 42 14 E0 PODB R02,+R24 get a char
1999 6E7F 16 E4 PUSB R02,+R26 put a char
2000 6E81 F0 F5 WHNP go on
2001 6E83

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 54
-----
lowercase a string
2022 6E87 18 2E BYI 30,56
2023 6E89 50 A8 20 LWRCE, LDB R20,+20H lowercase flag
2024 6E8C F0 00 JMP DOUL8 go do it
2025 6E8E
2026 6E8E 18 2E BYI 30,56
2027 6E90 50 A8 80 USCR8, LDB R20,+80H underscore flag
2028 6E93 F0 04 JMP DOUL8 go do it
2029 6E95
2030 6E95 18 2E BYI 30,56
2031 6E97 50 92 NSCR8, CLB R20 no underscore flag
2032 6E99
2033 6E99 54 DOUL8 DRP 24 get string addr
2034 6E9A CE FF FF JSB =GETADW off stack
2035 6E9D 6E 0A E3 POND R56,-R12 and length
2036 6E9F CE FF FF JSB =RSNEN- get some room
2037 6E9F F8 82 REN none, give up
2038 6E9F 6E 0A E5 PUND R56,+R12 length to stack
2039 6E9F 56 E5 PUND R26,+R12 addr to stack
2040 6E9A LOOP lower loop
2041 6E9A 6E 8B DCM R56 dec char count
2042 6E9C F4 A9 RNG done
2043 6E9C 43 14 E0 PODB R03,+R24 get a char
2044 6E9D 50 90 TSB R20 doing underscore?
2045 6E9D F4 10 JNG LHWGIT yes, go do it
2046 6E9D F6 05 IFZR if no underscore
2047 6E9D 43 CF 7F RNM R03,+7FH clear bit
2048 6E9A F0 0C ELSE no, must be lwr
2049 6E9D 43 CF 5B LNB R03,+BZ+1 <= Z?
2050 6E9D F8 07 IFMC yes,
2051 6E9E 18 41 LNB R03,+R' >= A?
2052 6E9E F8 03 IFCY yes,
2053 6E9E 43 10 94 CHNGIT ORB R03,R20 change it
2054 6E9E ENDF
2055 6E9E ENDF
2056 6E9E ENDF
2057 6E9E 16 E4 PUSD R03,+R26 char to output
2058 6E9E F0 DE WHNP go on
2059 6E9E

```

```

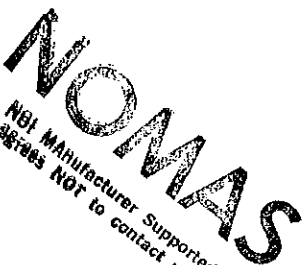
KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 53
-----
escape string
2003 6E83 18 2E BYI 30,56
2004 6E85 5E ESC8 JSB =GETADW get the string
2005 6E86 CE FF FF JSB =GETADW address
2006 6E89 6E 0A E3 POND R56,-R12 and length
2007 6E8C 85 LLM R56 double length
2008 6E8D CE FF FF JSB =RSNEN- get some room
2009 6E90 F8 E5 REN none, give up
2010 6E92 6E 0A E5 PUND R56,+R12 length out
2011 6E95 56 E5 PUND R26,+R12 address out
2012 6E97 42 A8 18 LDB R02,+1BH get an ecc
2013 6E9A LOOP do that move
2014 6E9A 6E 8B DCM R56 dec that
2015 6E9C 8B DCM R56 count
2016 6E9D F4 D8 RNG all done
2017 6E9F 43 1E E0 PODB R03,+R36 get a char
2018 6EA2 42 16 E5 POND R02,+R26 out esc/char
2019 6EA5 F0 F3 WHNP go again
2020 6EA7

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYZ102 11/16/1983 10:27 AM PG 55
-----
repeat a string n times
2061 6EE8 28 2E BYI 50,56
2062 6EE8 CE FF FF RPI8, JSB =ONERO1 get the count
2063 6EF1 60 06 E5 PUND R40,+R06 save number from ONEB
2064 6EF4 0A E5 PUND R40,+R12 back to stack
2065 6EF6 CE FF FF JSB =ONEB get as binary
2066 6EF9 66 10 A3 SIM R46,R20 save for later
2067 6EFF 60 06 E3 POND R40,-R06 get back number
2068 6EFF 54 DRP 24 get string address
2069 6F00 CE FF FF JSB =GETADW off stack
2070 6F03 5E 0A E3 POND R36,-R12 get string length
2071 6F06 12 A3 STM R36,R22 save for later
2072 6F08 60 0A E5 PUND R40,+R12 put count on stack
2073 6F0B CE FF FF JSB =CONBIN convert length
2074 6F0E 60 0A E5 PUND R40,+R12 put on stack
2075 6F11 CE FF FF JSB =NVPYRI multiply
2076 6F14 CE FF FF JSB =ONEB get total length
2077 6F17 F5 04 IFNG neg len
2078 6F19 CE FF FF JSBH =ERR1+ error report
2079 6F1C 10 BYI 16D arg out of rang
2080 6F1D ENDF
2081 6F1D 6E 26 A1 LDM R56,R46 get length
2082 6F20 CE FF FF JSB =RSNEN- get some room
2083 6F23 F8 5A REN none, give up
2084 6F25 6E 0A E5 PUND R56,+R12 length to stack
2085 6F28 56 E5 PUND R26,+R12 address to stack
2086 6F2A 54 1C A3 SIM R24,R34 save string addr
2087 6F2D LOOP copy loop
2088 6F2D 50 8B DCM R20 dec count
2089 6F2F F4 4E RNG done
2090 6F31 54 1C A1 LDM R24,R34 point to string
2091 6F34 40 12 A1 LDM R00,R22 get string len
2092 6F37 CE 78 6E JSB =NVSIR move it
2093 6F3A F0 F1 WHNP go on
2094 6F3C

```



 Recipient NBT Manufacturer Supported

 agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 56
=====
>>>>>>> left tran string <<<<<<<<<
2096 6F3C 2A 2E BY1 52,56
2097 6F3E CE 46 4F LTRMB, JSB =REV8, reverse the string
2098 6F41 CE EE 6D JSB =RTRN9, trim it
2099 6F44 JSB fall into rev8
2100 6F44

```

NOMAS
 NOT Manufacturer Supported
 Supporter NOT to contact Manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 58
=====
>>>>>>> hex/bin routines for address and data <<<<<<<<<
2119 6F65 54 A8 10 HEXBIN LDB R24,=160 set base for error check
2120 6F68 5E 10 A0 LDB R36,R20 get the high character
2121 6F68 CE DF 71 JSB =BINDBG1 convert to binary
2122 6F6E F8 0F REN error, give up
2123 6F70 5E 84 LDB R36 shift to
2124 6F72 B4 LDB R36 high nibble
2125 6F73 B4 LDB R36 for add
2126 6F74 B4 LDB R36 of low nibble
2127 6F75 10 A2 STB R36,R20 save it
2128 6F77 11 A0 LDB R36,R21 get the low character
2129 6F79 CE DF 71 JSB =BINDBG1 convert to binary
2130 6F7C 50 1E C2 ADB R20,R36 add to high nibble
2131 6F7F 9E RIM
2132 6F80
2133 6F80 50 11 A2 BINHEX STB R20,R21 move for low char
2134 6F83 CF F0 0F AMB R20,=FOFH,00FH mask high nibble, low nibble
2135 6F86 86 LRB R20 shift high
2136 6F87 86 LRB R20 nibble
2137 6F88 86 LRB R20 right for
2138 6F89 86 LRB R20 conversion
2139 6F8A 3E A2 STB R20,R76 get the high nibble
2140 6F8C CE CB 73 JSB =CHRDG1 convert to char
2141 6F8F 7E 10 A2 STB R76,R20 set high char
2142 6F92 11 A0 LDB R76,R21 get the low nibble
2143 6F94 CE CB 73 JSB =CHRDG1 convert to char
2144 6F97 51 3E A0 LDB R21,R76 set the low char
2145 6F9A 9E RIM
2146 6F9B

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 57
=====
>>>>>>> reverse string <<<<<<<<<
2102 6F44 18 2E BY1 30,56
2103 6F46 5E DRP 36 get the string
2104 6F47 CE FF FF JSB =GETADR from the stack
2105 6F4A 6E 0A E3 POMB R56,-R12 and the length
2106 6F4D CE FF FF JSB =RSNEM- get some room
2107 6F50 F8 2D REN none, give up
2108 6F52 6E 0A E5 PUMB R56,+R12 length on stack
2109 6F55 56 E5 PUMB R26,+R12 address on stack
2110 6F57 5E 2E C3 ADM R36,R56 end of string
2111 6F5A LOOP move loop
2112 6F5A 6E 8D DCM R56 dec count
2113 6F5C F4 21 RRG done
2114 6F5E 42 1E E2 POBD R02,-R36 get a char
2115 6F61 16 E4 PUBD R02,+R26 put a char
2116 6F63 F0 F5 WHMP go on
2117 6F65

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 59
=====
>>>>>>> convert ascii string to hex or other way <<<<<<<<<
2148 6F9B 18 2E BY1 30,56
2149 6F9D 5E HEXL DRP 36 get
2150 6F9E CE FF FF JSB =GETADR address
2151 6FA1 6E 0A E3 POMB R56,-R12 get length
2152 6FA4 12 A3 STN R56,R22 move for later
2153 6FA6 85 LLM R56 double length
2154 6FA7 CE FF FF JSB =RSNEM- get some space
2155 6FAA F8 EE REN error, give up
2156 6FAC 6E 0A E5 PUMB R56,+R12 put length on stack
2157 6FAF 56 E5 PUMB R26,+R12 put address on stack
2158 6FB1 2E C3 ADM R26,R56 point to new end
2159 6FB3 52 1E C3 ADM R22,R36 point to old end
2160 6FB6 LOOP conversion loop
2161 6FB6 6E 8B DCM R56 dec char count
2162 6FB8 8B DCM R56 by two
2163 6FB9 F4 DF RRG done
2164 6FBB 50 12 E2 POBD R20,-R22 get end char
2165 6FBE CE 80 6F JSB =BINHEX convert to hex
2166 6FC1 50 16 E7 PUMB R20,-R26 put it at new end
2167 6FC4 F0 F0 WHMP go again
2168 6FC6
2169 6FC6 18 2E BY1 30,56
2170 6FC8 CE FF FF ASCL JSB =GETADR get string address
2171 6FCB 7F 92 CLB R77 clear odd flag
2172 6FCD 6E 0A E3 POMB R56,-R12 get string length
2173 6FDD F3 04 IFOD length odd?
2174 6FE2 89 ICM R56 make it even
2175 6FD3 7F A8 30 LDB R77,'0' get leading zero
2176 6FD6 ENDF
2177 6FD6 6F 87 LRM R57 half length
2178 6FD8 6E E5 PUMB R56,+R12 new length to stack
2179 6FDA 5C 06 E5 PUMB R34,R06 save us from RSNEM-
2180 6FDD CE FF FF JSB =RSNEM- get space
2181 6FE0 56 0A E5 PUMB R26,+R12 new address to stack
2182 6FE3 5C 06 E3 POMB R34,-R06 get back old address
2183 6FE6 LOOP conversion loop
2184 6FE6 6E 8B DCM R56 dec char count
2185 6FE8 F4 B0 RRG done
2186 6FEA 7F 90 JSB R77 test odd flag
2187 6FEC F7 08 IFNZ yes,
2188 6FEE 10 A2 STB R77,R20 add leading zero
2189 6FF0 92 CLB R77 clear odd flag
2190 6FF1 51 1E E0 POBD R21,+R34 get the other char
2191 6FF4 F0 03 ELSE no,
2192 6FF6 50 1E E1 POMB R20,+R34 get hex number
2193 6FF9 ENDF
2194 6FF9 CE 65 6F JSB =HEXBIN convert to binary
2195 6FFC F8 9C REN error, give up
2196 6FFE 50 16 E4 PUMB R20,+R26 put in string
2197 7001 F0 E3 WHMP go again
2198 7003

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 60
=====
>>>>>> span/break string <<<<<<<<
2200 7003 00 3A 2D BYT 0,72,55
2201 7006 42 A8 FF BREAK LDB R02,-OFFH set break flag
2202 7009 F0 05 JMP D0SPBR go do it
2203 700B
2204 700B 00 3A 2D BYT 0,72,55
2205 700E 42 92 SPAN CLD R02 set span flag
2206 7010
2207 7010 42 06 E4 D0SPBR PUBD R02,-R06 save flag
2208 7013 CE FF FF JSB -ONEB get the pointer
2209 7016 58 06 E2 POBD R30,-R06 get flag
2210 7019 66 91 ISN R46 test pointer
2211 701B F4 03 JNG DFRPNT negative, default
2212 701D 8B DCM R46 dec for 0 test
2213 701E F5 02 IFNG yes,
2214 7020 66 93 DFRPNT CLM R46 set to zero
2215 7022 ENDF
2216 7022 89 ICM R46 restore after test
2217 7023 CE FF FF JSB -GETADR get target addr
2218 7026 5E 0A E3 POBD R36,-R12 get target len
2219 7029 2E A3 STM R36,R56 save length
2220 702B 26 C5 SBM R36,R46 adjust length for
2221 702D 89 ICM R36 pointer and base
2222 702E 5E 12 A3 STM R34,R22 save string start
2223 7031 26 C3 ADM R34,R46 adjust start for
2224 7033 8D DCM R34 pointer and base
2225 7034 54 DRP 24 get the pattern
2226 7035 CE FF FF JSB -GETADR string addr
2227 7038 56 0A E3 POBD R26,-R12 and length
2228 703B
2229 703B 5E 8B SENSIN DCM R36 dec length
2230 703D F5 04 IFNG done,
2231 703F 5C 93 CLM R34 get a zero
2232 7041 F0 19 JMP POSOUT and put out number
2233 7043 ENDF
2234 7043 43 1C E0 POBD R03,-R34 get char, inc pointer
2235 7046 40 16 A1 LDM R00,R26 get pattern len
2236 7049 50 14 A1 LDM R20,R24 get pattern start
2237 704E CE 66 70 JSB -SENSIN search for char
2238 704F 58 90 ISB R30 what were we doing?
2239 7051 F5 04 IFNG Break,
2240 7053 F9 E6 JEZ SENSIN not found, go on
2241 7055 F0 02 ELSE span,
2242 7057 F8 E2 JEM SENSIN found, go on
2243 7059 ENDF
2244 7059 5C 12 C5 SBM R34,R22 compute position
2245 705E 5C 1E A3 POSOUT STM R34,R36 move for combin
2246 705F CE FF FF JSB -COMBIN convert to integer
2247 7062 60 0A E5 POBD R40,-R12 number to stack
2248 7065 9E RTH
2249 7066
2250 7066 9D SENSIN CLE clear found flag
2251 7067 LOOP search loop
2252 7067 40 8B DCM R00 dec char count

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 62
=====
>>>>>> map string to string <<<<<<<
2261 7074 80 3A 2E BYT 200,72,56
2262 7077 54 MAPB. DRP 24 get the target
2263 7078 CE FF FF JSB -GETADR string address
2264 707B 6E 0A E3 POBD R56,-R12 get the length
2265 707E CE FF FF JSB -RSMEN- get some room
2266 7081 F8 F0 REM none, give up
2267 7083 56 2C A3 STM R26,R54 save the output pnt
2268 7086 2A A3 STM R26,R52 save it again
2269 708B 2E A1 LDM R26,R56 get target length
2270 708A 6E 93 CLM R56 clear output len
2271 708C CE FF FF JSB -GETADR get the to addr
2272 708F 5E 0A E3 POBD R36,-R12 and length
2273 7092 66 DRP 46 get the from
2274 7093 CE FF FF JSB -GETADR addr and
2275 7096 66 24 A3 STM R46,R44 move it
2276 7099 0A E3 POBD R46,-R12 get the length
2277 709B
2278 709B MAPSEN LOOP scan loop
2279 709B 56 8B DCM R26 dec char count
2280 709D F4 25 JNG MAPDNE done
2281 709F 43 14 E0 POBD R03,-R24 get a char
2282 70A2 40 26 A1 LDM R00,R46 get from len
2283 70A5 50 24 A1 LDM R20,R44 get from start
2284 70A8 CE 66 70 JSB -SENSIN search from
2285 70AB F8 05 IFEZ not found,
2286 70AD 43 2C E4 PUBD R03,-R54 put char out
2287 70B0 F0 0E ELSE found,
2288 70B2 50 24 C5 SBM R20,R44 compute position
2289 70B5 8B DCM R20 bias for table
2290 70B6 1E E1 CMM R20,R36 matching to?
2291 70B8 FB E1 JCY MAPSEN no, go on
2292 70BA 1C E3 RDM R20,R34 compute to pointer
2293 70BC 10 A4 LOBD R20,R20 get to char
2294 70BE 2C E4 PUBD R20,-R54 push to char out
2295 70C0 ENDF
2296 70C0 6E 89 ICM R56 inc out len
2297 70C2 F0 D7 MHAP
2298 70C4
2299 70C4 6E 0A E5 MAPDNE PUBD R56,-R12 length to stack
2300 70C7 2A A1 LDM R56,R52 get addr
2301 70C9 0A E5 PUBD R56,-R12 addr to stack
2302 70CB 9E RTH
2303 70CC

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 61
=====
>>>>>> span/break string <<<<<<<
2253 7069 F4 FA RNG done
2254 706B 42 10 E0 POBD R02,-R20 get char
2255 706E 03 10 CNB R02,R03 match?
2256 7070 F6 F5 MWNE no, go on
2257 7072 9E ICE set found flag
2258 7073 9E RTH
2259 7074

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG 63
=====
>>>>>> get ascii char string <<<<<<<
2305 70CC 2A 2E BYT 52,56
2306 70CE CE FF FF RSL13. JSB -GETADR get end char addr
2307 70D1 55 1C E0 POBD R25,-R34 get end char
2308 70D4 5E 0A E3 POBD R36,-R12 get length
2309 70D7 F6 03 IFZR null,
2310 70D9 55 A8 FF LDB R25,-OFFH assume very end
2311 70DE ENDF
2312 70DE CE FF FF JSB -GETADR get start char addr
2313 70DF 54 1C E0 POBD R24,-R34 get start char
2314 70E2 5E 0A E3 POBD R36,-R12 get length
2315 70E5 F6 02 IFZR null,
2316 70E7 54 92 CLB R24 assume very start
2317 70E9 ENDF
2318 70E9 6E 93 CLM R56 clear for load
2319 70EB 54 15 C0 CNB R24,R25 which way?
2320 70EE FA 07 IFCY reverse,
2321 70F0 6E 14 A0 LDB R56,R24 get last char
2322 70F3 15 C4 SBB R56,R25 compute size
2323 70F5 F0 04 ELSE forward,
2324 70F7 6E A0 LDB R56,R25 get last char
2325 70F9 14 C4 SBB R56,R24 compute size
2326 70FB ENDF
2327 70FB 89 ICM R56 adjust for subtract
2328 70FC CE FF FF JSB -RSMEN- get some room
2329 70FF F8 EA REM none, give up
2330 7101 6E 0A E5 PUBD R56,-R12 length to stack
2331 7104 56 E5 PUBD R26,-R12 addr to stack
2332 7106 LOOP char gen loop
2333 7106 54 16 E4 PUBD R24,-R26 put out char
2334 7109 15 C0 CNB R24,R25 which way?
2335 710B FA 03 IFCY down,
2336 710D 8A DCM R24 dec char
2337 710E F0 01 ELSE up,
2338 7110 8B ICB R24 anc char
2339 7111 ENDF
2340 7111 6E 8B DCM R56 dec char count
2341 7113 F6 F1 MWNZ while not done
2342 7115 9E RTH
2343 7116

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RV&102 OBJ=RVX102 11/16/1983 10:27 AM PG 64
*****
>>>>>> TAB key processing <<<<<<<<
2345 7116      *
2345 7116      *
2346 7116      *
2347 7116      *
2347 7116      *
2347 7116      *
2348 7116      *
2349 7116 9D    WHITE? CLE      clear white flag
2350 7117 C8 20  CMB RW,="  test for blank
2351 7119 F7 0C  JEQ SEIE1  yes, its white
2352 711D C8 2C  CMB RW,="  test for ,
2353 711D F7 08  JEQ SEIE1  yes, its white
2354 711F C8 3B  CMB RW,="  test for ;
2355 7121 F7 04  JEQ SEIE1  yes, its white
2356 7123 C8 2E  CMB RW,="  test for .
2357 7125 F6 EE  RNE      no, that's all
2358 7127      SETIE1      set E=1 and return
2359 7127 9C     BLACK ICE  set white flag
2360 7128 9E     RTN
2361 7129
2362 7129
2363 7129
2364 7129
2364 7129
2364 7129
2365 7129
2366 7129 42 B1 60 83 WHBLK? LDM R02,=IMPIR  get input pointer
2367 712D 8B      DCM R02      look back one and
2368 712E 02 A5   LDM R02,R02  get last and next characters
2369 7130 CE 16 71 JSB =WHITE?  test these characters
2370 7133 F9 F2  JEZ BLACK     E=0 if non-white
2371 7135 43      DRP 3        test next char
2372 7136 F0 DE  JMP WHITE?    go do it
2373 7138
2374 7138      TABIT LOOP      tab loop
2375 713B 60 90   JSB R40      test tab flag
2376 713A F6 07   IFZR      tab right,
2377 713C CE FF FF JSB =SYSJSB  RA will right arrow one
2378 713F FF FF   DEF RA      char and set E=0 if EOL
2379 7141 F0 05   ELSE      tab left,
2380 7143 CE FF FF JSB =SYSJSB  LA will left arrow one
2381 7146 FF FF   DEF LA      char and set E=0 if EOL
2382 7148
2383 7148 F9 0E   ENOIF      return if done
2384 714A CE 29 71 JSB =WHBLK?  will set E=0 if white
2385 714D      *              character precedes black
2386 714D FB E9   WHEN      no, go again
2387 714F 9E     RTN
2388 7150

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RV&102 OBJ=RVX102 11/16/1983 10:27 AM PG 66
*****
>>>>>> convert string to decimal (string base given) <<<<<<<<
2402 715A 28 2D  BY1 50,55
2403 715C CE F7 71 TONEC. JSB =GETBSE  get the base
2404 715F 98      BIN      binary for TOBIN2
2405 7160 50 3D A0  LDB R20,R75  convert base
2406 7163 CE FF FF JSB =TOBIN2  to binary
2407 7166 50 14 A2  SIB R20,R24  for compare
2408 7169 52      DRP 22   get the number
2409 716A CE FF FF JSB =GETADM  string address
2410 716D 50 0A E3  PDM R20,-R12  and length
2411 7170 88      DCM R20   adjust length for sign
2412 7171 56 93  CLM R20   clear "." flag
2413 7173 70 93  CLM R60   put a 0
2414 7175 E5      PUMD R60,+R12  on the stack
2415 7176
2416 7176 55 92  CLB R25      assume positive
2417 7178 5E 12 E0 PDM R36,+R22  get first char
2418 717B C8 2D  CMB R36,="-"  is it -
2419 717D F6 04  IFED      yes,
2420 717F 55 88  ICB R25      sign is negative
2421 7181 F0 08  ELSE      no,
2422 7183 C8 2B  CMB R36,="+"  is it +
2423 7185 F7 04  IFME      no,
2424 7187 50 89  ICM R20      restore length
2425 7189 5E E6  PUBD R36,-R22  replace character
2426 718B
2427 718B      ENOIF
2428 718B
2429 718B      LOOP      conversion loop
2430 718B 78 0A E5  PUMD R20,+R12  put base on stack
2431 718E CE FF FF JSB =MPIR01  multiply
2432 7191 98      BIN      back to binary
2433 7192 5E 93  CLM R36      clear for load
2434 7194 12 E0  PDM R36,+R22  get the next char
2435 7196 C8 2E  CMB R36,="."  radix point?
2436 7198 F6 0B  IFEQ      yes,
2437 719A 56 91  TSM R26      already have one?
2438 719C F6 6D  JNZ BRDDGT  yes, give up
2439 719E A1      LDM R26,R22  save location for later
2440 719F 50 8B  DCM R20      adjust length
2441 71A1 F7 15  JZR TDFIN  done, finish up
2442 71A3 F0 EC  JMP WKTCR   get next character
2443 71A5
2444 71A5 CE DF 71 JSB =BINDGT  convert digit to binary
2445 71A8 F8 AF  REN      give up if error
2446 71AA CE FF FF JSB =COMBIN  convert to real
2447 71AD 60 0A E5  PUMD R40,+R12  put on stack
2448 71B0 CE FF FF JSB =ADDROL  add to accum
2449 71B3 98      BIN      back to binary
2450 71B4 50 8B  DCM R20      dec char count
2451 71B6 F6 D3  WMWZ      not done, go on
2452 71B8
2453 71B8 56 91  TDFIN TSM R26  radix point found?
2454 71BA F7 19  IFNZ      yes,

```

```

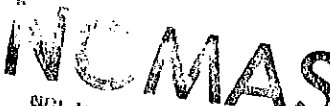
KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RV&102 OBJ=RVX102 11/16/1983 10:27 AM PG 65
*****
>>>>>> LCD ON/OFF disable lcd until restored <<<<<<<<
2390 7150
2390 7150
2391 7150      *
2392 7150      * Disable display until reenabled by user or keyboard *
2393 7150      * usage. *
2394 7150
2395 7150
2396 7150 A1      BY1 241
2397 7151 98      LID. BIN      for the complement
2398 7152 42 0A E2 PDM R02,-R12  get the on/off flag
2399 7155 8A      DCB R02      complement it
2400 7156 02 E0 B2 STBD R02,=LIDOFF  set it
2401 7159 9E      RTN
2402 715A

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RV&102 OBJ=RVX102 11/16/1983 10:27 AM PG 67
*****
>>>>>> convert string to decimal (string base given) <<<<<<<<
2455 71BC 52 16 C5  SBN R22,R26  compute position
2456 71BF 1E A3  STM R22,R36  move for CONBIN
2457 71C1 CE FF FF JSB =CONBIN  convert to real
2458 71C4 55 06 E4  PUBD R25,+R06  save us from VIX5
2459 71C7 78 0A E5  PUMD R70,+R12  put base on stack
2460 71CA 60 E5  PUMD R40,+R12  put power on stack
2461 71CC CE FF FF JSB =VIX5   compute divisor
2462 71CF CE FF FF JSB =DIV2   shift decimal point
2463 71D2 55 06 E2  PDM R25,-R06  recover sign flag
2464 71D5
2465 71D5 55 90  TSB R25      check original sign
2466 71D7 F7 04  IFNZ      -,
2467 71D9 99      BCD      to get sign right
2468 71DA CE FF FF JSB =CHSR01  change sign of number
2469 71DB
2470 71DB 9D      CLE      clear error flag
2471 71DE 9E      RTN
2472 71DF
2473 71DF CE FF FF BINDGT JSB =ALFAM  uppercase it
2474 71E2 5E CE 30  SBB R36,="0"  shift to 0
2475 71E5 F4 24  JNG BRDDGT  bad character
2476 71E7 C8 0A  CMB R36,=":"  greater than 9?
2477 71E9 FA 06  IFCY      yes,
2478 71EB C8 11  CMB R36,="A-0  less than A?
2479 71ED FA 1C  JNC BRDDGT  bad character
2480 71EF CC 07  SBB R36,="A-1  shift to 10
2481 71F1
2482 71F1 14 C0  CMB R36,R24  greater than base?
2483 71F3 FB 16  JCY BRDDGT  bad character
2484 71F5 9D      CLE      clear error flag
2485 71F6 9E      RTN
2486 71F7
2487 71F7 CE FF FF GETBSE JSB =ONEI  get the base
2488 71FA 60 38 A3  STM R40,R70  move it
2489 71FB 65 C9 37 00 CMI R45,+37L,0,0  check for range [0-36]
2490 7201 00
2491 7202 F8 04  JCY POPRTH  no good
2492 7204 C8 02  CMB R45,="2"  ok end
2493 7206 F8 EE  RCY      trash return
2494 7208 42 06 E3  PDM R02,-R06  report error
2495 720B CE FF FF BRDDGT JSBN =ERR1+  bad digit
2496 720F

```



 NOT Manufacturer Supported

 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY6102 OBJ=RYX102 11/16/1983 10:27 AM PG 68
-----
template line input (protected fields)
2498 7201 2A 2E      BTI 52,56
2499 7216 4E FF FF  TEMPLB. JSB =GETADR get filler addr
2500 7214 5E 0A E3   POND R36,-R12 get filler length
2501 7217 8B        DCN R36 test for 1
2502 7218 F7 04      IFNZ no,
2503 721A CE FF FF  BADTIF JSBN =ERR1+ report error
2504 721D 59        BYT 89D bad filler
2505 721E          ENDIF
2506 721E 1C E0      POND R36,-R34 get filler char
2507 7220 CE FF FF  JSB =ALLFAN uppercase it
2508 7223 5E CC 55   SBB R36,-'U' 0 = unprotected filler
2509 7226 F7 05      IFNZ no,
2510 7228 0A 05      ADJ R36,-#U-P adjust to 0 if 'P'
2511 722A F6 EE      JNZ BADTIF no, give up
2512 722C 88        ICB R36 adjust to 1
2513 722D          ENDIF
2514 722D 3F R2      STB R36,R77 save filler flag
2515 722F CE FF FF  JSB =GETADR get the template addr
2516 7232 5E 0A E3   POND R36,-R12 and length
2517 7235 F6 04      IFZR null,
2518 7237 7F 90      ISB R77 unprotected?
2519 7239 F6 06      JNZ BADTIF no, give up
2520 723B          ENDIF
2521 723B 5E C9 60 00 CHN R36,-#96 check length
2522 723F 0A 04      IFCY too long,
2523 7241 CE FF FF  BADTIF JSBN =ERR1+ report error
2524 7244 7A        BYT 122D bad template
2525 7245          ENDIF
2526 7245 5C 06 E5   POND R34,-R06 save us from rescon
2527 7248 CE FF FF  JSB =RESCON get some room
2528 724B 5F        BYT 95D for output
2529 724C 5C 06 E3   POND R34,-R06 get back templ addr
2530 724F F8 A5      REN error, give up
2531 7251 5E 05      POND R26,-R06 save output ptr
2532 7253 A9 64 84   LDM R26,-INPMN3 get scratch area
2533 7256 5C 16 E5   POND R34,-R26 save templ addr
2534 7259 5E 05      POND R36,-R26 and length
2535 725B 7E 92      CLB R76 clear unprtc found flag
2536 725D 56 89      ICM R26 save byte for shield len
2537 725F 50 92      CLB R20 clear shield count
2538 7261          LOOP
2539 7261 42 AB 08   LOB R02,-#80 shield build loop
2540 7264 40 92      CLB R00 bit count
2541 7266          LODP clear accum
2542 7266 5E 0B      DCN R36 bit loop
2543 7268 F4 18      JNG FINBYT dec templ len
2544 726A 0A 84      LDB R00 done, go on
2545 726C 43 1C E0   POND R03,-R34 left shift accum
2546 726F F5 04      IFNG get next char
2547 7271 A8 01      LDB R03,-1 if protected,
2548 7273 F0 01      ELSE get a 1
2549 7275 92        CLB R03 if not,
2550 7276          ENDIF get a 0

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY6102 OBJ=RYX102 11/16/1983 10:27 AM PG 70
-----
template line input (protected fields)
2604 7207          EMDIF
2605 7207 56 26 A3   SIM R26,R46 move shield pointer
2606 720A 89 FF FF  LDM R26,-RETURN get addr of return
2607 720D 24 A3      SIM R26,R44 move it
2608 720F CE FF FF  JSB =GETTEM disp template, get input
2609 7212 F7 C5      WHZR while clirkey hit
2610 7214 56        DRP 26 save code
2611 7215 F5 03      IFNG if not return,
2612 7217 93        CLM R26 null string
2613 7218 F0 03      ELSE if return,
2614 721A 93        CLM R26 clear for load
2615 721B 14 A0      LOB R26,R24 get string len
2616 721D          EMDIF
2617 721D 0A E5      POND R26,-R12 length on stack
2618 721F 12 A3      STM R26,R22 save length
2619 7221 06 E3      POND R26,-R06 get output pointer
2620 7223 0A E5      POND R26,-R12 addr on stack
2621 7225 50 A9 80 81 LDM R20,-INPMN3 get input pointer
2622 7229 52 91      ISN R22 null string?
2623 722B F7 09      IFNZ no,
2624 722D          LOOP
2625 722D 42 10 E0   POND R02,-R20 move output loop
2626 7230 16 E4      POND R02,-R26 get next char
2627 7232 52 8B      DCN R22 move to output
2628 7234 F6 F7      WHMZ dec char count
2629 7236          EMDIF while not done
2630 7236 CE FF FF  JSB =BLIMP blank input buffer
2631 7239 9E        RIM
2632 723A          EMDIF
2633 723A 0A 16 E4   CKTINPL POND R00,-R26 save shield byte
2634 723D 88        ICB R00 test for all protected
2635 723E F7 02      IFNZ no,
2636 723F 7E 88      ICB R76 set unprotected flag
2637 7241          EMDIF
2638 7241 50 88      ICB R20 inc byte count
2639 7243 9E        RIM
2640 7245

```

NOMAS

Not Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY6102 OBJ=RYX102 11/16/1983 10:27 AM PG 69
-----
template line input (protected fields)
2551 7276 40 03 C2   ADJ R00,R03 add to accum
2552 7279 42 BA      DCB R02 dec bit count
2553 727B F6 E9      WHMZ while not done
2554 727D CE 0A 73   JSB =CKTINPL check for unprotected bit
2555 7280 F0 DF      WHMP go again
2556 7282          EMDIF
2557 7282 42 C8 08   FINBYT CNB R02,-#80 even byte boundary?
2558 7285 F7 08      IFNE no,
2559 7287          LOOP
2560 7287 40 84      LLB R00 fill up byte loop
2561 7289 3F C2      ADJ R00,R77 left shift accum
2562 728B 42 BA      DCB R02 add in filler
2563 728D F6 F8      WHMZ dec bit count
2564 728F CE 0A 73   JSB =CKTINPL while not done
2565 7292          EMDIF check last byte
2566 7292          EMDIF
2567 7292 7E 90      ISB R76 any unprotected bits?
2568 7294 F6 09      IFZR no,
2569 7296 7F 90      ISB R77 fallier unprotected?
2570 7298 F7 05      IFNZ no,
2571 729A 56 06 E3   POND R26,-R06 trash output pointer
2572 729D F0 A2      JMP BADTIF give up
2573 729F          EMDIF
2574 729F          EMDIF
2575 729F 7F 90      ISB R77 what was filler?
2576 72A1 F7 02      IFNZ protected,
2577 72A3 50 8E      MCB R20 set fill shield flag
2578 72A5          EMDIF
2579 72A5 50 B2 84 84 STBD R20,-INPM3+4 save the size/fill flag
2580 72A9          LOOP
2581 72A9          LOOP
2582 72A9 CE FF FF  JSB =BLIMP user input loop
2583 72AC 5A A9 80 81 LDM R12,-INPMN3 blank the input buffer
2584 72B0 10 A3      STM R32,R20 get input pointer
2585 72B2 12 A3      STM R32,R22 save for later
2586 72B4 56 A9 64 84 LDM R26,-INPMN3 set default cursor position
2587 72B8 5C 16 E1   POND R34,-R26 get pointer to saved data
2588 72BB 5E 01      POND R36,-R26 get template addr
2589 72BD 14 A2      STB R36,R24 get length
2590 72BF F7 16      IFNZ get input size
2591 72C1 52 93      CLM R22 not null,
2592 72C3          LOOP clear cursor position
2593 72C3 43 1C E0   POND R03,-R34 template move loop
2594 72C6 F4 06      IFPS get next char
2595 72C8 52 91      ISN R22 if not protected,
2596 72CA F6 02      IFZR found one already?
2597 72CC 1A A1      LDM R22,R32 no,
2598 72CE          EMDIF save its position
2599 72CE          EMDIF
2600 72CE 43 1F 7F   ANN R03,-#7FH clear top bit
2601 72D1 1A E4      POND R03,-R32 put in input buf
2602 72D3 5E 8B      DCN R36 dec char count
2603 72D5 F6 EC      WHMZ while not done

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY6102 OBJ=RYX102 11/16/1983 10:27 AM PG 71
-----
convert decimal number to string in base given
2642 7315 20 2E      BTI 40,56
2643 7317 98        TOBS$. BIN binary mode
2644 7318 6E A9 00 01 LDM R56,-#256 room to move
2645 731C CE FF FF  JSB =RSMEM- get it
2646 731F F8 F3      REN none, give up
2647 7321 50 93      CLM R20 clear char count
2648 7323 56 CB 00 01 ADM R26,-#256 point to end of ran
2649 7327 CE FF 71   JSB =GETBASE get base
2650 732A 98        BIN for ONER
2651 732B CE FF FF  JSB =ONER get number
2652 732E 61 86      LRB R41 shift mantissa sign to E
2653 7330 55 92      CLB R25 clear for load
2654 7332 80        ELB R25 move sign into R25
2655 7333 71 86      LRB R61 shift out sign
2656 7335 84        LLB R61 shift in zero
2657 7336 54 92      CLB R24 clear frac part flag
2658 7338 78 0A E5   POND R70,-R12 save us from IP5
2659 733B 70 E5      POND R60,-R12 number to stack
2660 733D E5          POND R60,-R12 (twice)
2661 733E CE FF FF  JSB =IP5 get integer part
2662 7341 98        BIN (for ONER)
2663 7342 CE FF FF  JSB =ONER off stack as real
2664 7345 70 0A E3   POND R60,-R12 get original number
2665 7348 78 E3      POND R70,-R12 get back the base
2666 734A 70 20 C1   CHN R60,R40 equal?
2667 734D F7 15      IFNE no,
2668 734F 54 88      ICB R24 set frac part flag
2669 7351 60 0A E5   POND R40,-R12 int part on stack
2670 7354 70 E5      POND R60,-R12 number on stack
2671 7356 CE FF FF  JSB =FP5 get fraction part
2672 7359          LODP frac shift loop
2673 7359 78 0A E5   POND R70,-R12 base to stack
2674 735C CE FF FF  JSB =MPYR01 mult times frac part
2675 735F 60 0A E5   POND R40,-R12 copy to stack
2676 7362 78 0A E5   POND R70,-R12 save base from IP5
2677 7364 60 E5      POND R40,-R12 copy to stack
2678 7366 CE FF FF  JSB =IP5 get int part
2679 7369 60 0A E3   POND R40,-R12 of product,
2680 736C 93        BIN for lcb/dcb
2681 736D 65 91      ISN R45 -zero (leading zero?)
2682 736F F6 05      IFZR yes,
2683 7371 54 88      ICB R24 inc zero count
2684 7373 F6 01      IFZR too many,
2685 7375 8A        DCB R24 set max
2686 7376          EMDIF
2687 7376          EMDIF
2688 7376 78 E3      POND R70,-R12 recover base
2689 7378 CE FF FF  JSB =FP5 get new frac part
2690 737B 70 0A E3   POND R60,-R12 of product
2691 737E F6 09      WHMZ not zero, go again
2692 7380 60 E3      POND R40,-R12 get shifted frac part
2693 7382 70 E3      POND R60,-R12 get int part
2694 7384          EMDIF

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 72
=====
>>>>>> convert decimal number to string in base given <<<<<<<
2695 7384          ISB R24          test frac part flag
2696 7384 54 90      IFNZ          yes,
2697 7384 57 80      PUMD R60,+R06 save int part
2698 7388 70 06 E5   LDM R60,R40 move for conv
2699 7388 20 A1      JSB =TBCONV convert frac part
2700 738D CE D2 73   BIN          back to binary
2701 7390 98         IFEZ          if ok so far,
2702 7391 F8 15      LOOP          leading zero loop
2703 7393          DEB R24          dec leading zero count
2704 7393 54 8A      JZR OUT'.' done, go on
2705 7395 F7 11      LDB R60,'.' get a '0'
2706 7397 70 A8 30   PUBD R60,-R26 put it out
2707 739A 16 E6      ICM R20          inc char count
2708 739C 50 89      CMI R20,-#254 too big?
2709 739E C9 FE 00   IFCY          yes,
2710 73A1 FA 03      JSB =WARN42 warn user, set flag
2711 73A3 CE F7 73   ENDIF
2712 73A6          UNHEZ          while ok, loop
2713 73A6 F9 E8      ENDIF
2714 73A8          ICM R20          adjust length
2715 73A8 50 89      LDB R60,'.' get .
2716 73AA 70 A8 2E   PUBD R60,-R26 put in output
2717 73AD 16 E6      POMO R60,-R06 get int part back
2718 73AF 06 E3      JEN 18F1M error, finish up
2719 73B1 F8 03      ENDIF
2720 73B3          JSB =TBCONV convert int part
2721 73B3 CE D2 73   TBFIN BIN          back to binary
2722 73B6          ISB R25          test sign
2723 73B6 98         IFNZ          neg,
2724 73B7 55 90      LDB R76,'-' add sign
2725 73B9 F7 07      PUBD R76,-R26 to output
2726 73BB 7E A8 2D   ICM R20          inc char count
2727 73BE 16 E6      ENDIF
2728 73C0 50 89      PUMD R20,+R12 put length
2729 73C2          PUMD R26,+R12 and address on stack
2730 73C2 50 0A E5   RTN
2731 73C5 56 E5
2732 73C7 9E
2733 73C8          CHRG1 RDB R76,'0' shift to '0'
2734 73C8 7E CA 30   CMB R76,'.' greater than '9'
2735 73C8 C8 3A      IFCY          yes,
2736 73CD FA 02      RDB R76,-#A- shift to 'A'
2737 73CF CA 07      ENDIF
2738 73D1          RTN
2739 73D1 9E
2740 73D2          TBCONV LOOP conversion loop
2741 73D2          PUMD R70,+R12 base to stack
2742 73D2 78 0A E5   PUMD R60,+R12 number to stack
2743 73D5 70 E5      PUMD R70,+R12 base to stack
2744 73D7 78 E5      PUMD R60,+R12 number to stack
2745 73D9 70 E5      PUMD R70,+R12 base to stack
2746 73DB 78 E5      JSB =HELJSB do mod number,base
2747 73DD CE FF FF

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 74
=====
>>>>>> some helpful subroutines <<<<<<<
2772 7408 5A A9 02 00 GETPD+ LDM R32,+#2 set minimum size
2773 740F 76 C8 20   CMB R66,'.' more?
2774 7412 F7 12      IFNE          yes,
2775 7414 5A 89      ICM R32          inc char count
2776 7416 77 C8 20   CMB R67,'.' more?
2777 7419 F7 02      IFNE          yes,
2778 741B 5A 89      ICM R32          inc char count
2779 741D          ENDIF
2780 741D 5C A9 0D 82 LDM R34,+IMPMM2 get pointer to ran
2781 7421 74 1C A7   SIND R64,R34 move name
2782 7424 F0 02      ELSE          no,
2783 7426 34 A1      LDM R66,R64 get name
2784 7428          ENDIF
2785 7428 CE FF FF   JSB =GETPAD get the pil address of dev
2786 742B 9E        RTN
2787 742C

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 73
=====
>>>>>> convert decimal number to string in base given <<<<<<<
2748 73E0 FF FF      DEB MOD10,
2749 73E2 CE FF FF   JSB =ONEB get result (do bin)
2750 73E5 CE E8 73   JSB =CHRG1 convert bin to char
2751 73E8 7E 16 E6   PUBD R76,-R26 put out character
2752 73EB 50 89      ICM R20          inc char count
2753 73ED C9 FE 00   CMI R20,-#254 string too long
2754 73F0 FA 0B      IFCY          yes,
2755 73F2 70 0A E3   POMO R60,-R12 clean up
2756 73F5 E3        POMO R60,-R12 R12
2757 73F6 E3        POMO R60,-R12 stack
2758 73F7 CE FF FF   JSB =WARN4 report warning
2759 73FA 2A        BYI 420
2760 73FB 9C          ICE          set error flag
2761 73FC 9E        RTN
2762 73FD          ENDIF
2763 73FD CE FF FF   JSB =INTDIV intdiv number,base
2764 7400 70 0A E3   PUMD R60,-R12 get result
2765 7403 78 E3      PUMD R70,-R12 get back base
2766 7405 75 91      ISM R65          zero?
2767 7407 F6 C9      UNHZ          not done, go on
2768 7409 9D          CLE          clear error flag
2769 740A 9E        RTN
2770 740B

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 75
=====
>>>>>> set up srq statement <<<<<<<
2789 742E E1        BYI 341
2790 742D 98        OMSRD, BIN
2791 742E CE C6 64   JSB =APROPN for sure
2792 7431 F8 F8      REN          get the file
2793 7433 66 B1 4D 82 LDM R46,+PER error, give up
2794 7437 05 47 82   SIND R46,+RNF1L get ptr of line
2795 743A 24 A3      SIM R46,R44 relativize it
2796 743C 08 A1      LDM R46,R10 move for later
2797 743E 05 47 82   SIND R46,+RNF1L get R10 of start
2798 7441 63 A8 01   LDB R43,+1 relativize it
2799 7444 1E E5      PUMD R43,+R36 set active flag
2800 7446 60 B1 75 82 LDM R40,+RNAME save it all
2801 744A A7          SIND R40,R36 get our name
2802 744B CE FF FF   JSB =SYSJSB save it
2803 744E FF FF      DEF END.ON go finish up,
2804 7450 9E        RTN          move to next line
2805 7451
2806 7451 A1        BYI 241
2807 7452 CE C6 64   JSB =APROPN get our file
2808 7455 F8 F9      REN          error, give up
2809 7457 42 1E A1   LDM R02,R36 get the pointer
2810 745A 63 D2 E1   POMO R43,+R02 get past first info
2811 745D 60 E1      POMO R40,+R02 get our name
2812 745F D1 75 82   CMB R40,-RNAME is it us?
2813 7462 F6 E1      RME          no, quit
2814 7464 42 A8 01   LDB R02,+1 set the
2815 7467 1E A6      SIND R02,R36 valid srq flag
2816 7469 9E        RTN
2817 746A
2818 746A A1        BYI 241
2819 746B 98        OFSRD, BIN
2820 746C CE C6 64   JSB =APROPN for sure
2821 746F F8 F8      REN          get the file
2822 7471 CE C0 64   JSB =CLRAPP error, give up
2823 7474 9E        RTN          clear info
2824 7475
2825 7475

```



```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 76
*****
>>>>>> CR18 of cassette <<<<<<<
2827 7475 28 2E BYT 50,56
2828 7477 98 YEAT8. BIN for sure
2829 7478 42 80 B3 83 LDBD R02,=STAND? get current standby
2830 747C 06 'E4 PUSD R02,=R06 save it
2831 747E 43 98 FF LDB R03,=OFFH get standby on
2832 7481 CE FF FF JSB =STAND? set it
2833 7484 CE FF FF JSB =DWB? get user number
2834 7487 CE FF FF JSB =MELJSB get the device
2835 748A FF FF DEF FLSHAK name
2836 748C F9 06 IFEN error, give up
2837 748E CE FF FF JSB =ERR1 report it
2838 7491 3F BYT 63D invalid filespec
2839 7492 F0 7A JMP TCEND clean up
2840 7494 ENDI?
2841 7494 7E 06 E5 PUSD R76,=R06 save file count
2842 7497 CE FF FF JSB =MELJSB
2843 749A FF FF DEF VFH? talk to device
2844 749C F8 05 JEN TCERR? error, give up
2845 749E CE FF FF JSB =MELJSB
2846 74A1 FF FF DEF VDIR? find start of dir
2847 74A3 7E 06 E3 TCERR? PUSD R76,=R06 get file count
2848 74A6 F8 1E JEN TCERR error, give up
2849 74A8 8B DCM R76 adjust for loop
2850 74A9 F4 5C JNG TCNULL too small, stop
2851 74AB LOOP
2852 74AB JSB =MELJSB
2853 74AB CE FF FF DEF VFEDD? end of dir?
2854 74AE FF FF JZR TCNULL not there, stop
2855 74B0 F7 55 JSB =MELJSB
2856 74B2 CE FF FF DEF VFENP? purged file?
2857 74B5 FF FF IFM? no,
2858 74B7 F7 04 DCM R76 decrement count
2859 74B9 7E 8B JNG TCFIN there?, finish up
2860 74BB F4 0D ENDI?
2861 74BD JSB =MELJSB
2862 74BD CE FF FF DEF VFENXDE get next directory
2863 74C0 FF FF UMEZ
2864 74C2 F9 E7
2865 74C4
2866 74C4 CE FF FF TEERR JSB =ERR1 error, give up
2867 74C7 39 BYT 570 bad transmission
2868 74C8 F0 44 JMP TCEND clean up
2869 74CA
2870 74CA CE FF FF TCFIN JSB =MELJSB
2871 74CD FF FF DEF VFVDE? reverse the dir
2872 74CF CE FF FF JSB =MELJSB
2873 74D2 FF FF DEF VFDE? translate to rco
2874 74D4 58 1E A1 LDM R30,R36 compute pointer
2875 74D7 CD 58 00 ROM R30,=VF.RDE to rco dir
2876 74DA 56 A9 0D 82 LDM R26,=IMPHN2 temporary space
2877 74DE 5E 06 E5 PUSD R36,=R06 save us from
2878 74E1 50 E5 PUSD R20,=R06 catbuf
2879 74E3 CE FF FF JSB =MELJSB go do a

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 78
*****
>>>>>> cause an error number <<<<<<<
2907 7524 R1 BYT 241
2908 7525 CE FF FF ERROR. JSB =DWR12? anything there?
2909 7528 F9 08 IFEN JSB =DWB? yes,
2910 752A CE FF FF JSB =DWB? get error number
2911 752D 50 26 A0 LDB R20,R46 get the number
2912 7530 F0 04 ELSE no,
2913 7532 50 80 7A 83 LDBD R20,=EAMUNH get the last error
2914 7536 ENDI?
2915 7536 CE FF FF JSB =ERRORR and report it
2916 7539 9C RIM
2917 753A

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 77
*****
>>>>>> CR18 of cassette <<<<<<<
2880 74E6 FF FF DEF CRIBUF catalog
2881 74E8 50 06 E3 PUSD R20,=R06 retrieve saved
2882 74EB 5E 2E A3 SIM R36,R56 save length
2883 74EE 06 E3 PUSD R36,=R06 registers
2884 74F0 66 1E B5 20 LDMD A46,R36,(VF.CDE+8D) get rest of
2885 74F4 00
2886 74F5 16 B7 08 00 SHND A46,R26,B8 name and put it out
2887 74F9 62 B5 0E 00 LDMD A42,R26,B12 get size
2888 74FD 67 10 A0 LDB R47,R20 and add on the
2889 7500 62 16 B7 0C SHND A42,R26,B12 multiplier
2890 7504 00
2891 7505 F0 02 JMP TCOUT
2892 7507
2893 7507 6E 93 TCNULL CLN R56 output a null string
2894 7509
2895 7509 6E 0A E5 TCOUT PUSD R56,=R12 push length
2896 750C 56 E5 PUSD R26,=R12 and address
2897 750E 42 80 B3 83 TCEND LDBD R02,=STAND? get current standby
2898 7512 43 06 E2 PDBD R03,=R06 get back old standby status
2899 7515 CE FF FF JSB =STAND? and restore it
2900 7518 CE FF FF JSB =MELJSB go do
2901 751B FF FF DEF VFBYE shut up everyone
2902 751D 9E RIM
2903 751E
2904 751E 5E 92 HANDLE CLB R36
2905 7520 B2 75 83 SHND R36,=HANDLD signal that we handled
2906 7523 9E RIM
2907 7524

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 79
*****
>>>>>> signif and wait key <<<<<<<
2919 753A 00 2E BYT 0,56
2920 753C 98 SKEY8. BIN for sure
2921 753D CE FF FF JSB =SIGNIF get a signif key
2922 7540 F0 06 JMP GOTKEY go process it
2923 7542
2924 7542 00 2E BYT 0,56
2925 7544 98 MKEY8. BIN for sure
2926 7545 CE FF FF JSB =WAITKY wait for the next key
2927 7548 CE FF FF GOTKEY JSB =SVSJSB go process
2928 754B FF FF DEF KEY8. it onto stack
2929 754D 9E RIM
2930 754E

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 80
*****
***** Clone tape on multi-tapes *****
2932 754E 18 20
2933 7550 56
2934 7551 CE FF FF
2935 7554 SE 0A E3
2936 7557 F7 38
2937 7559 59 A8 08
2938 755C CE FF FF
2939 755F F8 08
2940 7561 50 90
2941 7563 F7 08
2942 7565 C8 20
2943 7567 F7 04
2944 7569 CE FF FF
2945 756C 3F
2946 756D
2947 756D
2948 756D 60 1C A5
2949 7570 CE FF FF
2950 7573 F8 05
2951 7575 63 C8 2E
2952 7578 F6 EF
2953 757A
2954 757A CE FF FF
2955 757D F9 06
2956 757F CE FF FF
2957 7582 4D
2958 7583 3E
2959 7584 9E
2960 7585
2961 7585 56 01 45 82
2962 7589 5E 18 A5
2963 758C 83 45 82
2964 758F F0 04
2965 7591 56 01 45 82
2966 7595
2967 7595 7E A9 39 A9
2968 7599 CE FF FF
2969 759C 56 03 45 82
2970 759D 65 14 A5
2971 75A3 64 A8 FF
2972 75A6 67 92
2973 75A8 65 C9 39 A9
2974 75AC 00
2975 75AD F6 01
2976 75AF 93
2977 75B0
2978 75B3 9E
2979 75B4

BYT 30.55
LISTEN. DRP 26
JSB =GETIOW
POMD R36,-R12
IFNZ
LDB R31,+8D
JSB =GEIFST
JEM INVLIN
ISB R20
IFNZ
CMB R20,="
IFNE
JSDN =ERR1+
BYT 63D
ENDIF
ENDIF
LDND R40,R34
JSB =ALFAM
IFEZ
CMB R40,="
JNE INVLIN
ENDIF
JSB =FOPEN
IFEN
JSB =HAMD1
VAL V.NEWM
BYT 62D
RTN
ENDIF
LDND R26,=EDFILE
LDND R36,R30
STND R36,=EDFILE
ELSE
LDND R26,=EDFILE
ENDIF
LDND R76,=ENDLIN
JSB =PREFND
STND R26,=EDFILE
LDND R45,R24
LDB R44,=OFFH
CLB R47
CMB R45,=099H,0A9H,000H last line?
IFEO
CLM R45
ENDIF
PUMD R40,+R12
RTN

get the filename
off the stack
and the length
not null,
set max length
get a file name
bad, give up
ended in?
something,
use it blank?
no,
error
bad filespec
get the name
first char alpha?
no,
is it .?
no, error
go get the file
not there,
give others a
chance
or error
save current edfile
get our start
take edfile
null,
get edfile
get last line
find line in front
restore edfile
get the line number
set integer flag
clear garbage
R45,="099H,0A9H,000H last line?
yes,
change it to 0
number to stack

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 82
*****
***** Clone tape on multi-tapes *****
3032 7615 76 A9 20 20
3033 7619 CE FF FF
3034 761C FF FF
3035 761E F8 0E
3036 7620 74 12 E5
3037 7623 7F 88
3038 7625 CE FF FF
3039 7628 FF FF
3040 762A F8 68
3041 762C F0 0F
3042 762E 42 80 7B 83
3043 7632 C8 38
3044 7634 F7 04
3045 7636 C8 5C
3046 7638 F6 5A
3047 763A
3048 763A CE FF FF
3049 763D
3050 763D CE FF FF
3051 7640 F6 CE
3052 7642 7F 90
3053 7644 F7 08
3054 7646
3055 7646
3056 7646
3057 7646 F0 32
3058 7648
3059 7648 SE 88
3060 764A F7 02
3061 764C F5 06
3062 764E CE FF FF
3063 7651 3F
3064 7652 F0 40
3065 7654
3066 7654 42 16 E0
3067 7657 C8 3A
3068 7659 F6 F3
3069 765B 59 A8 04
3070 765E CE FF FF
3071 7661 F8 EB
3072 7663 50 90
3073 7665 F7 04
3074 7667 C8 2C
3075 7669 F6 E3
3076 766B
3077 766B 74 1C A5
3078 766E 12 E5
3079 7670 7F 88
3080 7672 C8 1F
3081 7674 F8 08
3082 7676 50 90
3083 7678 F6 CE
3084 767A

JEMCE9
JEMCE8
ELSE
LDB R02,=ERRORS
CMB R02,=56D
IFNE
CMB R02,=92D
JNE JEMCE8
ENDIF
JSB =CLRERR
ENDIF
JSB =SKPCNK
WHNZ
TSB R77
JZR BADIAR

** process the list
PRDVC LOOP
DCH R36
JZR BADIAR
IFAG
JSB =ERR1
BYT 63D
JMP JEMCE8
ENDIF
POBD R02,+R26
CMB R02,="
JNE BADIAR
LDB R31,=4
JSB =GEIFST
JEM BADIAR
TSB R20
IFNZ
CMB R20,="
JNZ BADIAR
ENDIF
LDND R64,R34
PUMD R64,+R22
ICB R77
CMB R77,=31D
JCY BADIAR
TSB R20
WHNZ
ENDIF

get blank fill
go find out
if its a NH dev
yes,
save name
inc count
shut up the
device
no, (E=1)
what was the error
no loop response?
no, (wasn't multimeter)
no, mass memory?
no, give up and report
clear the error
next device
while not done
any NH dev?
no, give up
parse list loop
dec for ":"
error, no more
if not there,
report error
bad filespec
clean up
get ':'
is it?
no, give up
get max length
get a device
error, give up
end char?
not last,
better be ,
no, give up
get the name
put name in table
inc table count
too many?
yes, error out
last char?
no, go again

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 81
*****
***** Clone tape on multi-tapes *****
2981 750A R1
2982 7505 98
2983 7506
2984 7506
2985 7506
2986 7506 6E A9 20 01
2987 750A CE FF FF
2988 750D F8 F4
2989 750F 60 93
2990 7511 16 E5
2991 7513 E5
2992 7514 E5
2993 7515 E5
2994 7516 56 B3 68 84
2995 751A CE FF FF
2996 751D 7A
2997 751E F8 E3
2998 7520
2999 7520
3000 7520
3001 7520 42 80 B3 83
3002 7524 B2 64 84
3003 7527 43 B8 FF
3004 752A CE FF FF
3005 752B CE FF FF
3006 752D FF FF
3007 752E
3008 752E
3009 752E
3010 752E 7F 92
3011 752A 16 E4
3012 7526 56 B3 65 84
3013 752B F8 3E
3014 752C 14 A3
3015 752E 89
3016 752F 12 A3
3017 7531 CE FF FF
3018 7534 5E 0A E3
3019 7537 F9 03 00
3020 753A F6 4C
3021 753C 75 16 A5
3022 753F 40 A9 35 FD
3023 7543 CE FF FF
3024 7546 75 C9 41 4E
3025 754A 4C
3026 754A
3027 754A
3028 754A
3029 754D CE FF FF
3030 754D
3031 754D 74 1E 85 03
3032 754D 00

MEOPY. BIN
BYT 241
for sure

** get the scratch ram we need
LDB R56,=8288
JSB =RSNEN-
RTN
CLM R40
PUMD R40,+R26
PUMD R40,+R26
PUMD R40,+R26
PUMD R40,+R26
STND R26,=INP3+4
JSB =RESCOM
BYT 122D
RTN
one record size + dir
get space for
none, give up
clear out dir
entry
for end
of directory
(used later)
save pointer
get some scratch
space for table
none, give up

** turn standby on and clean up loop
LDND R02,=STAND?
STND R02,=INPM3
LDB R03,=OFFH
JSB =STAND?
JSB =NELJSB
DEF VFVBE
get current standby
save it
get standby on
set it
make sure all is
quiet on loop

** parse the target list
CLB R77
POBD R77,+R26
STND R26,=INPM3+1
JEM JEMCE9
STN R26,R24
ICB R26
STN R26,R22
JSB =GETIOW
POMD R36,-R12
CMB R36,+83
IFEO
LDND R65,R26
LDB R00,=65.375
JSB =UPRC4
CMB R65,="ALL"
JME PRDVC
clear list counter
clear success flag
save pointer
error, give up
move table pointer
save space for count
move pointer for later
get target string addr
and length
right length for ALL?
yes,
get the string
set up for uppercasing
uppercase it
is it ALL?
no, try to parse string

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 83
*****
***** Clone tape on multi-tapes *****
3085 767A
3086 767A
3087 767A
3088 767A 7F 14 A6
3089 767D 50 E0
3090 767F
3091 767F 74 14 E1
3092 7682 50 06 E4
3093 7685 CE FF FF
3094 7688 FF FF
3095 768A 50 06 E2
3096 768D F8 17
3097 768F CE FF FF
3098 7692 FF FF
3099 7694 F8 6D
3100 7696 50 8A
3101 7698 F6 E5
3102 769A
3103 769A
3104 769A
3105 769A 56
3106 769B CE FF FF
3107 769E 5E 0A E3
3108 76A1 8B
3109 76A2 F7 02
3110 76A4 F5 06
3111 76A6 CE FF FF
3112 76A9 3F
3113 76AA F8 57
3114 76AC
3115 76AC 42 92
3116 76AE B2 67 84
3117 76B1 B2 70 84
3118 76B4 16 E0
3119 76B6 CE FF FF
3120 76B9 42 C8 2E
3121 76B6 F6 05
3122 76B8 B2 70 84
3123 76C1 F0 23
3124 76C3 C8 3A
3125 76C5 F7 1F
3126 76C7 C1 4E
3127 76C9 F7 04
3128 76CB C8 02
3129 76CD F6 D7
3130 76CF
3131 76CF B2 67 84
3132 76D0 5E 88
3133 76D4 F7 00
3134 76D6 42 16 E0
3135 76D9 C8 2E
3136 76DB F6 05
3137 76DD B2 70 84

** establish devfile entry for each target
STND R77,R24
POBD R20,+R24
LOOP
POMD R64,+R24
POBD R20,+R06
JSB =NELJSB
DEF VFGET
to the device
POBD R20,-R06
JEM BRDSCR
JSB =NELJSB
DEF VFVBE
device
JEM JEMCE7
error, give up
DDB R20
dec counter
while not done

** parse the source string
DRP 26
JSB =GETIOW
POMD R36,-R12
DCH R36
JZR BRDSCR
IFAG
JSB =ERR1
BYT 63D
JEM JEMCE7
clean up
ENDIF
CLB R02
STND R02,=INP3+3
STND R02,=(INPM3+12D) and noint flag
POBD R02,+R26
JSB =ALFAM
CMB R02,="
IFEQ
yes,
STND R02,=(INPM3+12D) set noint flag
ELSE
no,
CMB R02,="
is it :?
IFNE
SOB R02,="M"
is it normal?
no,
IFNZ
CMB R02,=BP-M
is it private?
no, give up
ENDIF
STND R02,=INP3+3
save private flag
DCH R36
adjust string length
JZR BRDSCR
only 2 chars, not good
POBD R02,+R26
get next char
CMB R02,="
check for noint
yes,
STND R02,=(INPM3+12D) set flag

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 84
***** Clone tape on multi-tapes *****
3138 76E0 F0 04 ELSE no,
3139 76E2 C8 3A CMB R02,=1 better be ;
3140 76E4 F6 C0 JMC BRDSRC no, give up
3141 76E6 ENDDIF
3142 76E6 ENDDIF
3143 76E6 ENDDIF
3144 76E6 59 A8 04 LDB R31,=40 get max length
3145 76E9 C8 FF FF JSB =GEIFSI get device name
3146 76E8 F8 88 JEM BRDSRC bad, give up
3147 76E6 50 90 JSB R20 should be end
3148 76F0 F6 B4 JNZ BRDSRC no, give up
3149 76F2 74 1C A5 LDM R64,R34 get the name
3150 76F5
3151 76F5 ** establish source in devfile
3152 76F5 JSB =MELJSB how do a hi
3153 76F8 FF FF DEF VFIH to it
3154 76F8 FF FF JEM BRDSRC error, give up
3155 76F8 F8 A8 CLM R45 record 0
3156 76F6 65 93 JSB =MELJSB seek to its
3157 76F6 C8 FF FF DEF VFIHUSK 0th record
3158 7701 FF FF JEMCE7 JEM JEMCE6 error, give up
3159 7703 F8 37 JSB =MELJSB untunl the
3160 7705 C8 FF FF DEF VFIH source device
3161 7708 FF FF JEM JEMCE6 error, give up
3162 770A F8 30
3163 770C
3164 770C ** reestablish targets in case something moved
3165 770C
3166 770C 54 B1 65 B4 LDM R24,=TNP3+1 get back table pointer
3167 7710 50 93 CLM R20 clear counters
3168 7712 14 E0 PDBD R20,=R24 get dev counter
3169 7714 54 12 R3 STM R24,R22 move pointer
3170 7717 5E 16 R3 STM R36,R26 save source dev ptr
3171 771A 74 14 E1 PDM R64,=R24 get first name
3172 771D F0 08 JMP IF dive into loop
3173 771F LOOP 2nd hi loop
3174 771F 74 14 E1 PDM R64,=R24 get the name
3175 7722 5E 16 C1 CMB R36,R26 previous dev*arc?
3176 7725 F7 05 IFNE no,
3177 7727 5E 12 E5 1H PDM R36,=R22 save previous dev ptr
3178 772A F0 02 ELSE yes,
3179 772C 51 88 ICB R21 inc dup dev count
3180 772E ENDDIF
3181 772E 50 8A DCR R20 dec counter
3182 7730 F4 15 JMC MCPHMT when done
3183 7732 06 E5 PDM R20,=R06 save counters
3184 7734 C8 FF FF JSB =MELJSB hi again in case
3185 7737 FF FF DEF VFIH it moved (clear COB)
3186 7739 50 06 E3 PDM R20,=R06 get counters back
3187 773C F8 1A JEMCE6 JEM JEMCE5 error, give up
3188 773E C8 FF FF JSB =MELJSB shut up the
3189 7741 FF FF DEF VFIH device
3190 7743 F8 13 JEM JEMCE5 error, give up

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 85
***** Clone tape on multi-tapes *****
3191 7745 F0 D8 WHMP go again
3192 7747
3193 7747 ** compute new target count
3194 7747
3195 7747 54 B1 65 B4 MCPHMT LDM R24,=TNP3+1 get table ptr
3196 7748 50 14 A4 LDM R20,R24 get count
3197 774E 11 C4 SBB R20,R21 adjust for duplicates
3198 7750 14 A6 STBD R20,R24 save new count
3199 7752 F6 06 IFZR no targets, give up
3200 7754 C8 FF FF JSB =ERR1 report
3201 7757 3F BY1 63D bad filespec
3202 7758 F0 60 JEMCE5 JMP JEMCE4 clean up
3203 775A ENDDIF
3204 775A E6 PDBD R20,=R24 set success flag
3205 775B
3206 775B ** process volume for directory size
3207 775B
3208 775B JSB =GEIREC read part of volume
3209 775B C8 3F 79 ADM R26,=R20 point to dir size
3210 775E 56 C8 14 00 PDBD R02,=R26 get low byte
3211 7762 42 16 E2 PDBD R03,=R26 get high byte
3212 7765 43 E2 STMD R02,=TNP3+6 save directory size
3213 7767 42 B3 6A B4 JSB =GEIREC skip rest of volume
3214 7768 C8 3F 79 LDM R30,=TNP3+6 get dir size (sectors)
3215 776E A8 B1 6A B4
3216 7772
3217 7772 ** scan directory for private files
3218 7772
3219 7772 C8 3F 79 SCHLP JSB =GEIREC get dir record
3220 7775 56 C8 0A 00 ADM R26,=R10 point to type
3221 7779 50 A8 08 LDB R20,=R0 dir counter
3222 777C
3223 777C ** scan record of 8 entries
3224 777C
3225 777C LOOP dir scan loop
3226 777C 5B 16 A4 LDM R33,R26 get filetype high
3227 777F 5A B4 01 00 LDM R32,R26,81 and low bytes
3228 7783 91 TSN R32 check type
3229 7784 F7 63 IFNZ if not purged,
3230 7786 89 ICM R32 test for dir end
3231 7787 F7 3E JZR DIRDWN yes, go on
3232 7789 8B DCR R32 restore type
3233 778A
3234 778A ** check for standard (titan) files
3235 778A
3236 778A C9 D8 E0 CMB R32,=E0DBH in standard range?
3237 778D F0 A0 IFCY maybe,
3238 778F C9 00 E4 CMB R32,=E000H definitely?
3239 7792 F8 2D JCY PRVERR no, error out
3240 7794 C8 02 00 RNM R32,=R2 test for private file
3241 7797 F6 28 JNZ PRVERR yes, give up
3242 7799 F0 4E ELSE maybe some other,
3243 779B 8B DCR R32 LFI ?

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 86
***** Clone tape on multi-tapes *****
3244 779C F7 4B IFNZ no,
3245 779E 89 ICM R32 restore type
3246 779F 5B C8 E0 CMB R33,=E00H known range?
3247 77A2 F6 10 JMC PRVERR no, give up
3248 77A4
3249 77A4 ** check for 75C files
3250 77A4
3251 77A4 5A C8 51 CMB R32,=51H 75C file?
3252 77A7 FA 04 IFCY maybe,
3253 77A8 C8 5C CMB R32,=5CH definitely?
3254 77AB FA 08 JMC CHK75C yes, go check passwor
3255 77AD ENDDIF
3256 77AD C8 88 CMB R32,=88H more 75C?
3257 77AF FA 1C IFCY maybe,
3258 77B1 C8 8C CMB R32,=8CH definitely?
3259 77B3 F8 18 IFNC yes,
3260 77B5 64 16 B5 12 CHK75C password?
3261 77B8 F7 0D IFNZ maybe,
3262 77BC 01 FF FF CMB R44,=BLANKS is it?
3263 77BF F7 08 IFNE yes, error out
3264 77C1 C8 FF FF PRVERR JSB =ERR1 report error
3265 77C4 41 BY1 65D and give up
3266 77C5 F0 69 JEMCE4 JMP JEMCE3 clean up
3267 77C7 F0 41 JMP DIREND going down
3268 77C9 ENDDIF
3269 77C9 ENDDIF
3270 77C9 F0 1E JMP NMTDIR go on
3271 77CB F0 A5 JMP SCHLP going up
3272 77CD ENDDIF
3273 77CD ENDDIF
3274 77CD
3275 77CD ** check for 41C files
3276 77CD
3277 77CD 62 A9 10 80 LDM R42,=10H,80H,000H,40H,50H,60H get 41C
3278 77D0 00 40 50 60
3279 77D0 00 40 21 LDB R00,=41 get reg pointer
3280 77D0 40 88 LOOP 41C test loop
3281 77D0 C8 28 ICB R00 next reg
3282 77D0 F8 E3 CMB R00,=50 done?
3283 77D0 5A 01 C0 JCY PRVERR yes, unknown
3284 77E1 F7 F5 CMB R32,R4 valid 41C?
3285 77E3 16 B4 14 00 WHCO no, go on
3286 77E7 F2 D8 LDM R32,X26,620 get private bit
3287 77E9 JOD PRVERR set, give up
3288 77E9 ENDDIF
3289 77E9 ENDDIF
3290 77E9 56 C8 20 00 NMTDIR ADM R26,=R32 point to next dir
3291 77ED 50 8A DCR R20 dec dir count
3292 77EF F6 8B WHZ until done
3293 77F1
3294 77F1 ** at end of record move last entry to before

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 87
***** Clone tape on multi-tapes *****
3295 77F1 ** first so we can back up into it if the next
3296 77F1 ** record starts with a logical end of directory
3297 77F1
3298 77F1 16 A1 LDM R20,R26 get pointer
3299 77F3 C8 0A 00 SBR R20,=R10 back up to last dir
3300 77F6 54 B1 68 B4 LDM R24,=TNP3+4 get pointer to ran
3301 77FA 40 A8 04 LDB R00,=40 get loop count
3302 77FD LOOP move dir loop
3303 77FD 60 10 E3 PDM R40,=R20 get dir part
3304 7800 14 E7 PDM R40,=R24 move dir part
3305 7802 40 8A DCR R00 dec count
3306 7804 F6 F7 WHZ until done
3307 7806 58 8B DCR R30 dec dir rec count
3308 7808 F6 C1 JNZ LOOPUP until done
3309 780A
3310 780A ** at end of directory, back up to last valid file
3311 780A ** to get the end of data pointer for tape
3312 780A
3313 780A 56 C8 1C 00 DIREND SBN R26,=R28 back up to last start
3314 780E 55 16 E0 PDBD R25,=R26 get high byte
3315 7811 54 E0 PDBD R24,=R26 get low byte
3316 7813 50 E1 PDM R20,=R26 skip to size
3317 7815 51 E0 PDBD R21,=R26 get high byte
3318 7817 50 E0 PDBD R20,=R26 get low byte
3319 7819 14 C3 ADM R20,R24 compute end sector
3320 781B 83 6E 84 STMD R20,=(TNP3+10D) save for later
3321 781E 55 93 CLM R45 record 0
3322 7820 C8 FF FF JSB =MELJSB get us back
3323 7823 FF FF DEF VFIHUSK to the beginning
3324 7825 F8 09 JEM JEMCE3 error out
3325 7827
3326 7827 ** if needed, initialize targets
3327 7827
3328 7827 54 B0 70 84 LDM R24,=(TNP3+120) get nount flag
3329 782B F6 14 IFZR if not set,
3330 782D C8 61 79 JSB =LISTEN listen all
3331 7830 F8 0D JEMCE3 JEM JEMCE2 error out
3332 7832 6F A8 05 LDB R57,=5D get init ddl
3333 7835 C8 FF FF JSB =MELJSB send ddl to
3334 7838 FF FF DEF DDLREP all targets
3335 783A F8 03 JEM JEMCE2 error, give up
3336 783C C8 96 79 JSB =BSVALL wait for done
3337 783F F8 64 JEMCE2 JEM JEMCE1 error out
3338 7841 ENDDIF
3339 7841
3340 7841 ** rewind, and move volume header
3341 7841
3342 7841 C8 7E 79 JSB =RSTALL reset ALL
3343 7844 C8 3F 79 JSB =GEIREC get vol part
3344 7847 C8 11 79 JSB =WSTALL send it out
3345 784A C8 3F 79 JSB =GEIREC get vol rest
3346 784D C8 11 79 JSB =WSTALL send it out
3347 7850

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 88
***** Clone tape on multi-tapes *****
3348 7850      ** move directory, make private if needed
3349 7850
3350 7850 40 B1 6A 84      LDND R00,=INP3+6      get dir size
3351 7854      LOOP      scan dir loop
3352 7854 40 B3 74 84      SIMD R00,=(INP3+16D) save size
3353 7858 CE 3F 79      JSB =GETREC      get a dir rec
3354 785B 50 B0 67 04      LDND R20,=INP3+3      test private flag
3355 785F F7 37      IFNZ      yes,
3356 7861 A8 08      LDB R20,=8D      dir counter
3357 7863 56 CB 0A 00      ADN R26,=810      point to type
3358 7867      LOOP      scan dir loop
3359 7867 60 16 B5 F6      LDND R40,R26,8-10      get the filename
3360 786B FF
3361 786E 5A B4 01 00      LDND R33,R26      get the type high
3362 7872 C9 B8 E0      LDND R32,R26,81      and low bytes
3363 7875 F7 11      CMV R32,=8E088H      is it basic?
3364 7877 C9 B9 E0      JEQ MAKPRV      yes, make it private
3365 787A F6 14      CMV R32,=8E089H      is it lex?
3366 787E 60 C9 49 4E      IFEQ      yes,
3367 7880 53 54 41 4C      CMV R40,="INSTALL" is it our lex file?
3368 7884 4C 20      IFNE      no, do the change
3369 7888 CE B2 79      JSB =PRVPSW      get password
3370 7890      SIMD R44,R26,818      save the password
3371 7890      ENDIF
3372 7890 56 CB 20 00      ENDIF
3373 7894 50 B8      ADN R26,=832      point to next type
3374 7896 F6 CF      DCB R20      dec counter
3375 789B      UNHZ      until done
3376 789B CE 11 79      ENDIF
3377 789B 40 B1 74 84      JSB =WRITEALL      send out dir rec
3378 789F 8B      LDND R00,=(INP3+16D) get counter
3379 78A0 F6 82      DCM R00      end of directory?
3380 78A2      UNHZ      go again if not
3381 78A2      ** set up to move data
3382 78A2
3383 78A2 CE 61 79      JSB =LISTEN      listen the others
3384 78A5 F8 30      JEMCEI JEN JEMCEO      error, give up
3385 78A7 54 B1 65 B4      LDND R24,=INP3+1      get table pointer
3386 78AB 5E 14 E0      POBD R36,=R24      skip counter
3387 78AE E1      POMB R36,=R24      get the source device
3388 78AF CE FF FF      JSB =VFTRD      tell source
3389 78B2 F8 23      JEM JEMCEO      error, give up
3390 78B4 CE FF FF      JSB =HELJSB      tell listeners
3391 78B7 FF FF      DEF VFDDL2      to write
3392 78B9 F8 1C      JEM JEMCEO      error out
3393 78BB
3394 78BB      ** compute amount to move
3395 78BB
3396 78BB 50 B1 6E B4      LDND R20,=(INP3+10D) get max size

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 90
***** write to all targets *****
3446 7911      *****
3447 7911      *****
3448 7911      * This routine writes a 256 byte record from
3449 7911      * raw to all target devices
3450 7911      *****
3451 7911 CE 61 79      WRITEALL JSB =LISTEN      listen em
3452 7914 F8 24      JEM WRITEALL      error, give up
3453 7916 CE FF FF      JSB =HELJSB      tell em all
3454 7919 FF FF      DEF VFDDL2      to write
3455 791B F8 10      JEM WRITEALL      error, give up
3456 791D 56 A9 00 01      LDND R26,=8256      get record size
3457 7921 24 A3      SIM R26,R44      set up write
3458 7923 66 B1 68 84      LDND R46,=INP3+4      get record loc
3459 7927 CE FF FF      JSB =HELJSB      send the record to
3460 792A FF FF      DEF VFUR      the listeners
3461 792C F8 0C      JEM WRITEALL      error, give up
3462 792E CE 96 79      JSB =BSYALL      check for errors
3463 7931 F8 07      JEM WRITEALL      error out
3464 7933 CE FF FF      JSB =HELJSB      tell then to
3465 7936 FF FF      DEF VFBBYE      shut up
3466 7938 F9 D6      REZ      ok, return
3467 793A
3468 793A 42 06 E3      WRITEALL POMB R02,=R06      pop return
3469 793B      JMP MCPEND      error out

```

NOMAS
 Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 89
***** Clone tape on multi-tapes *****
3397 78DF F7 18      JZR RTFILE      no files, skip it
3398 78E1 8B      DCM R20      adjust for vol
3399 78E2 8B      DCM R20      header
3400 78E3 D5 6A 84      SIMD R20,=INP3+6      adjust for directory
3401 78E6 65 93      CMV R45      clear accum
3402 78E8      LOOP      multiply loop
3403 78E8 65 CB 00 01      ADN R45,=0,1H,0      add 256D
3404 78EC 00
3405 78EC 50 B8      DCM R20      dec counter
3406 78F F6 F7      UNHZ      until done
3407 78D1 8A      DCB R20      make r20 = -1
3408 78D2 CE FF FF      JSB =HELJSB      and go watch the
3409 78D5 FF FF      DEF VFURCH      bytes go by
3410 78D7 F8 0D      JEMCEO JEN MCPEND      error, give up
3411 78D9      ** rewind everybody
3412 78D9
3413 78D9 CE FF FF      RTFILE JSB =VFTRD      listen the source
3414 78DC F8 0B      JEM MCPEND      error, give up
3415 78DE 6F AB 07      LDB R57,=7D      get a rewind
3416 78E1 CE FF FF      JSB =HELJSB      send it out
3417 78E4 FF FF      DEF DDLEAP      to listeners
3418 78E6
3419 78E6      ** restore user's STANDBY state, kill loop
3420 78E6
3421 78E6 42 B0 83 83      MCPEND LDND R02,=STAND?      get current standby
3422 78E8 43 B0 64 B4      LDND R03,=INP3+1      get back old standby status
3423 78EC CE FF FF      JSB =STAND?      and restore it
3424 78F CE FF FF      JSB =HELJSB      tell everybody
3425 78F4 FF FF      DEF VFBBYE      to shut up
3426 78F6 54 B1 65 B4      LDND R24,=INP3+1      get table pointer
3427 78F8 5D 14 E2      POBD R35,=R24      get success flag
3428 78FD F7 11      RZR      not set, done
3429 78FF E1      POMB R35,=R24      restore pointer
3430 7900 E0      POBD R35,=R24      get counter
3431 7901 F7 0D      RZR      none, leave
3432 7903 88      ICB R35      include source
3433 7904
3434 7904      ** kill valid data flag so mainframe will not use
3435 7904      ** old tape info
3436 7904
3437 7904      LOOP
3438 7904 5E 14 E1      POMB R36,=R24      clear code loop
3439 7907 CE FF FF      JSB =HELJSB      get device
3440 7909 FF FF      DEF CLRCOD      clear the
3441 790C 5D 8A      DCM R35      code byte
3442 790E F6 F4      UNHZ      dec counter
3443 7910 9E      RIN      until done
3444 7911

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RY&I02 11/16/1983 10:27 AM PG 91
***** get record from master *****
3471 793F      *****
3472 793F      *****
3473 793F      * This routine gets a 256 byte record from
3474 793F      * the master device.
3475 793F      *****
3476 793F      *****
3477 793F 56 A9 00 01      GETREC LDND R26,=8256      get record size
3478 7943 24 A3      SIM R26,R44      set up for VFURD
3479 7945 B1 68 84      LDND R26,=INP3+4      get store pointer
3480 7948 26 A3      SIM R26,R46      set up
3481 794A 54 B1 65 B4      LDND R24,=INP3+1      get table pointer
3482 794E 5E 14 E0      POBD R36,=R24      skip counter
3483 7951 E1      POMB R36,=R24      get source
3484 7952 CE FF FF      JSB =HELJSB      get the record
3485 7955 FF FF      DEF VFURD      in from tape
3486 7957 F8 E1      JEM WRITEALL      error, give up
3487 7959 CE FF FF      JSB =HELJSB      tell caller
3488 795C FF FF      DEF VFBBYE      to shut up
3489 795E F8 0A      JEM WRITEALL      error, give up
3490 7960 9E      RIN
3491 7961

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 92
>>>>>>> listen all target devices <<<<<<<<
3491 7961 *
3491 7961 *
3492 7961 * This routine turns off the loop
3493 7961 * and then does a listen to all target
3494 7961 * devices.
3494 7961 *
3495 7961 *
3496 7961 CE FF FF LISTEN JSB =MELJSB shut up
3497 7961 FF FF DEF VFBYE all
3498 7966 F8 F8 REN error, give up
3499 7968 54 B1 65 B4 LDM R24,=IMP3+1 get table pointer
3500 7961 50 14 E0 PDBD R20,=R24 get counter
3501 796F 5E E1 PDM R36,=R24 skip source
3502 7971 LOOP listen loop
3503 7971 5E 14 E1 PDM R36,=R24 get device
3504 7974 CE FF FF JSB =VFLAD listen device
3505 7977 F8 E7 REN error, give up
3506 7979 50 B4 DCR R20 dec counter
3507 797B F6 F4 MHNZ until done
3508 797D 9E RIM
3509 797E

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 94
>>>>>>> wait for all target devices <<<<<<<<
3528 7996 *
3528 7996 *
3529 7996 * This routine waits until all target devices
3530 7996 * have a not busy status. It also reports
3531 7996 * any device errors.
3531 7996 *
3532 7996 *
3533 7996 54 B1 65 B4 BSYALL LDM R24,=IMP3+1 get back table ptr
3534 799A 50 14 E0 PDBD R20,=R24 get counter
3535 799D 88 ICB R20 include source
3536 799E LOOP wait loop
3537 799E 5E 14 E1 PDM R36,=R24 get device
3538 79A1 50 06 E4 PDBD R20,=R06 save counter
3539 79A4 CE FF FF JSB =MELJSB wait until device
3540 79A7 FF FF DEF VFBYV is done
3541 79A9 50 06 E2 PDBD R20,=R06 get counter
3542 79AC F8 E7 REN error, give up
3543 79AE 8A DCR R20 dec counter
3544 79AF F6 E0 MHNZ until all done
3545 79B1 9E RIM
3546 79B2

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 93
>>>>>>> reset all target devices <<<<<<<<
3511 797E *
3511 797E *
3512 797E * This routine send a device clear to
3513 797E * all target devices to reset then to
3514 797E * beginning of tape.
3514 797E *
3515 797E *
3516 797E CE 64 79 RSTALL JSB =LISTEN listen en
3517 7981 6F A8 04 LDB R57,=SDC' get device clear
3518 7984 CE FF FF JSB =CNDREP send it
3519 7987 F8 B1 JEM WRITER error out
3520 7989 CE 96 79 JSB =BSYALL wait until done
3521 798C F8 AC JEM WRITER error out
3522 798E CE FF FF JSB =MELJSB shut en all
3523 7991 FF FF DEF VFBYE up again
3524 7993 F8 A5 JEM WRITER error out
3525 7995 9E RIM
3526 7996

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I02 OBJ=RYX102 11/16/1983 10:27 AM PG 95
>>>>>>> compute password from filename <<<<<<<<
3548 7982 *
3548 7982 *
3549 7982 * INPUT: R40/47 filename
3550 7982 * OUTPUT: R44/47 password
3551 7982 *
3552 7982 * This routine derives a password from the given
3553 7982 * filename. The password is guaranteed to be
3554 7982 * a valid 75C password, with the 41C privacy bit
3555 7982 * set. This is the same routine as is used by
3556 7982 * INSTALL.
3557 7982 *
3558 7982 PRVPSM LDM R74,R40 save bottom half
3559 7985 64 A3 STM R44,R40 move top half
3560 7987 3C A1 LDM R44,R74 get bottom half
3561 7989 00 06 E5 PDM R00,=R06 save pointer
3562 798C A8 20 LDB R00,=40 get start reg
3563 798E 42 A9 41 05 LDM R02,='A',5 get start values
3564 79C2 LOOP blank rplc loop
3565 79C2 41 C8 20 CNB R4,' ' char=' '
3566 79C5 F6 11 IFEQ yes,
3567 79C7 40 C8 24 CNB R00,=44 name part?
3568 79CA FA 07 IFCY yes,
3569 79CC 42 01 A2 SIB R02,R4 get char
3570 79CF CA 03 RDB R02,=3 compute next
3571 79D1 F0 05 ELSE no,
3572 79D3 43 01 A2 SIB R03,R4 get char inc
3573 79D6 CA 05 RDB R03,=5 compute next
3574 79D8 ENDTF
3575 79D8 ENDTF
3576 79D8 40 B8 ICB R00 next reg
3577 79DA C8 28 CNB R00,=50 done?
3578 79DC FA E4 WHMC no, go on
3579 79DE 06 E3 PDM R00,=R06 restore pointer
3580 79E0 LOOP scramble loop
3581 79E0 64 20 E2 RDB R44,R40 inc char
3582 79E3 CE FF FF JSB =ALFAM valid?
3583 79E6 F9 F8 WHEZ no, go again
3584 79E8 LOOP scramble loop
3585 79E8 65 21 E2 RDB R45,R41 inc char
3586 79EB CE FF FF JSB =ALFAM valid?
3587 79EE F9 F8 WHEZ no, go again
3588 79F0 LOOP scramble loop
3589 79F0 67 23 E2 RDB R47,R43 inc char
3590 79F3 CE FF FF JSB =ALFAM valid?
3591 79F6 F9 F8 WHEZ no, go again
3592 79F8 LOOP scramble loop
3593 79F8 66 22 E2 RDB R46,R42 inc char
3594 79FB 42 A8 01 LDB R02,=01H private mask
3595 79FE 66 02 94 ORB R46,R02 set private bit
3596 79A1 CE FF FF JSB =ALFAM valid?
3597 79A4 F9 F2 WHEZ no, go again

```

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KRMH 08/31/82=====
JTEN  LOC OBJECT CODE SRC=R15I02 OBJ=RXZJ02 11/16/1983 10:27 AM PG.98
=====
>>>>>>> Load a private file from tape
3654 7A5F 40 A8 D4 LDB R00,=4 max dev length
3655 7A62 LOOP get dev loop
3656 7A62 58 89 ICM R30 inc counter
3657 7A64 42 E0 PUMB R02,=R20 get a char
3658 7A67 14 E4 PUMB R02,=R24 add to source
3659 7A69 40 8A DCB R00 dec max len
3660 7A6B F4 A5 JMG BRDSPC too long, error
3661 7A6D 52 8B DCM R22 dec length
3662 7A6F F6 F1 WMMZ until gone
3663 7A71 42 A8 2F LDB R02,='/' get psu delin
3664 7A74 E4 PUMB R02,=R24 put out to source
3665 7A75 58 89 ICM R30 adjust counter
3666 7A77
3667 7A77 ** add the password to source string
3668 7A77
3669 7A77 CE B2 79 JSB =PRVPSU get password
3670 7A7A 64 1E PUMB R44,=R24 put out to source
3671 7A7D 58 CB 04 00 ROM R30,=R4 adjust counter
3672 7A81 6A 0A E3 PUMB R52,=R12 get stuff off stack
3673 7A84 58 E5 PUMB R30,=R12 add source length
3674 7A86 6A E5 PUMB R52,=R12 put stuff back
3675 7A88
3676 7A88 ** fake out the system and do a call to COPY
3677 7A88
3678 7A88 42 A8 7C LDB R02,=COPYIK get the copy token
3679 7A8B 02 54 B4 STDB R02,=TOKEN and fake out the tape
3680 7A8E CE FF FF JSB =SYSJSB do a syscall to COPY
3681 7A91 FF FF DEF COPY. so we don't get lost
3682 7A93 42 80 7B 83 LDBD R02,=ERRORS any errors?
3683 7A97 F6 10 RMZ yes, give up
3684 7A99
3685 7A99 ** get the file and make the type private
3686 7A99
3687 7A99 60 B1 64 84 LDRD R40,=TMPNM3 get filename
3688 7A9D CE FF FF JSB =FOPEN' get the file
3689 7AA0 F8 07 REN error, gave up
3690 7AA2 40 CF CB ANM R15,=11001011B make it private
3691 7AA5 18 B6 04 00 STBD R15,X30,DR.TYP have new type
3692 7AA9 9E
3693 7AAA

```

```

KRNH 08/31/82
TIEH LOC OBJECT CODE SRC=RYA102 OBJ=RYZ102 11/16/1983 10:27 AM PG 97
>>>>>> Load a private file from tape <<<<<<<<
3601 ZAO7 A1 BYT 241
3602 ZAO8 98 INSTL B1M for sure
3603 ZAO9
3604 ZAO9 ** get the filename
3605 ZAO9
3606 ZAO9 50 DRP 20 get the
3607 ZAOA CE FF FF JSB =GETAON filename:dev addr
3608 ZAOD 52 0A E3 POMB R22,-R12 and length
3609 ZAIO F6 04 IFZR null,
3610 ZAIZ CE FF FF BNDSPC JSB =ERR1+ error out
3611 ZAI5 3F BYT 63D
3612 ZAI6 ENDIF
3613 ZAI6
3614 ZAI6 ** get scratch ran space
3615 ZAI6
3616 ZAI6 CE FF FF JSB =RESCON get room for source
3617 ZAI9 12 BYT 18D name:dev/psu
3618 ZAIR F8 EA REN nope, give up
3619 ZAIIC 56 0A E5 PUMB R26,-R12 addr to stack
3620 ZAIIF 14 A3 STM R26,R24 save for later
3621 ZAI2I CE FF FF JSB =RESCON get room for target
3622 ZAI2A 09 BYT 9D name:
3623 ZAI2S F8 DF REN nope, give up
3624 ZAI27
3625 ZAI27 ** move the name to source and target strings
3626 ZAI27
3627 ZAI27 56 0A E5 PUMB R26,-R12 addr to stack
3628 ZAI2A 40 08 1F LDB R00,=37 get reg pointer
3629 ZAI2B 58 91 EAM R30 clear counter
3630 ZAI2F LDDP get name loop
3631 ZAI2F 58 89 ICN R30 inc counter
3632 ZAI31 40 08 ICB R00 inc pointer
3633 ZAI33 C8 29 CMB R00,=51 check name overflow
3634 ZAI35 F8 0B JCY BNDSPC too long, error
3635 ZAI37 41 01 EO POMB R*,-R20 get a char
3636 ZAI39 14 E4 PUMB R*,-R24 add to source
3637 ZAI3C 16 E4 PUMB R*,-R24 add to target
3638 ZAI3E 52 8D DCN R22 dec length
3639 ZAI40 F7 0D JZR BNDSPC all gone, error
3640 ZAI42 41 C8 3A CMB R*,=1 dev delin?
3641 ZAI45 F6 E8 LWME no, keep going
3642 ZAI47 81 FF FF LDM R*,=BLANKS blank fill name (over :)
3643 ZAI4A 40 89 20 F8 LDM R00,=40,370 set up call to uppercase
3644 ZAI4E CE FF FF JSB =UPRA1 uppercase the name
3645 ZAI51 60 B3 64 M4 STM R40,=IMPMN3 save name for later
3646 ZAI55 6E 0A E3 POMB R56,-R12 get target addr
3647 ZAI58 58 8D DCN R30 remove : from length
3648 ZAI5A E5 PUMB R30,-R12 save target length
3649 ZAI5B 6E E5 PUMB R56,-R12 put back target addr
3650 ZAI5D 58 89 ICN R30 restore : to length
3651 ZAI5F
3652 ZAI5F ** move device to source string
3653 ZAI5F

```

KMAMA	08/31/82				SRC=RVX102	OBJ=RVZ102	11/16/1983 10:27 AM	Pg 99
ITEM	LOC	OBJECT CODE						
>>>>>>		bit/base conversions						<<<<<<<
3695	ZARR	18	20		BVI	30,55		
3696	ZARC	52	A9 10 00 BID.		LDM	R22,=#16	get max length	
3697	ZABO	54	A9 02 01		LDM	R24,=20,1D	get base, shift count	
3698	ZABA	F0	16		JMP	BSTODC	go do it	
3699	ZABE							
3700	ZABG	18	20		BVI	30,55		
3701	ZABB	52	A9 06 00 DID.		LDM	R22,=#6	get max length	
3702	ZABC	54	A9 08 03		LDM	R24,=80,3D	get base, shift count	
3703	ZAE0	F0	0A		JMP	BSTODC	go do it	
3704	ZAC?							
3705	ZAC2	18	20		BVI	30,55		
3706	ZAC4	52	A9 04 00 H1D.		LDM	R22,=#4	get max length	
3707	ZACB	54	A9 10 04		LDM	R24,=160,4D	get base, shift count	
3708	ZACC							
3709	ZALL	52	89	BSTODC	ICM	R22	adjust length	
3710	ZACE	CE	FF FF		JSB	=GETADR	get address	
3711	ZAD1	5A	0A E3		POMD	R32,-R12	get length	
3712	ZAD4	12	C1		CMH	R32,R22	check for size	
3713	ZAD6	FA	04		IFLY		too big,	
3714	ZADB	CE	FF FF	BSCERR	JSBM	=ERR1+	error out	
3715	ZADB	59			BVI	89D		
3716	ZADC			ENDIF				
3717	ZADC	58	93	CLM	R30		clear accum	
3718	ZADE			LOOP			conv loop	
3719	ZADE	5A	8D	DCLM	R32		dec length	
3720	ZAE0	F4	1B	JNG	BSCEND		done	
3721	ZAE2	42	15 A0	LDB	R02,R25		get shift count	
3722	ZAE5			LOOP			shift loop	
3723	ZAE5	58	85	LLM	R30		shift left (2K)	
3724	ZAE7	F8	E7	JCV	BSCERR		overflow	
3725	ZAE9	42	8A	DEB	R02		dec count	
3726	ZAEB	F6	F8	WHMZ			until done	
3727	ZAE0	5E	93	CLM	R36		clear for later	
3728	ZAF1	1C	0E	POMB	R36,+R34		get char	
3729	ZAF1	CE	DF 71	JSB	=BINDDGT		convert to digit	
3730	ZAF4	F8	B3	REW			error out	
3731	ZAF6	58	1E C3	ADM	R30,R36		add to accum	
3732	ZAF9	F8	DD	JCV	BSCERR		overflow	
3733	ZAFB	F0	E1	WMHP			go on	
3734	ZAFD	5E	18 A1	BSCEND	LDM	R36,R30	get result	
3735	ZB00	CE	45 6B	JSB	=NUMOUT		convert and push	
3736	ZB03	9E		RIM				
3737	ZB04							
3738	ZB04	10	2E	BVI	20,56			
3739	ZB06	54	A9 01 07 D1B%	LDM	R24,=10,7D		get shift counts	
3740	ZB0R	F0	0E	JMF	DC10BS		do convert	
3741	ZB0I							
3742	ZB0I	10	2E	BVI	20,56			
3743	ZB0E	54	A9 03 05 D10%.	LDM	R24,=3D,5D		get shift counts	
3744	ZB12	F0	06	JMP	DC10BS		do convert	
3745	ZB14							
3746	ZB14	10	2E	BVI	20,56			
3747	ZB16	54	A9 04 04 D1H%.	LDM	R24,=4D,4D		get shift counts	

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG100
*****
>>>>>> bit/base conversions <<<<<<<<
3748 781A
3749 781A 98 DCTOBS BIN for sure
3750 781B CE FF FF JSB =RESCOM get room for
3751 781E 10 BYT 160 result
3752 781F F8 E2 REN none, give up
3753 7821 56 2E E3 ROM R26,R56 compute end
3754 7824 12 A3 STM R26,R22 save it
3755 7826 CE 4E 78 JSB =ONEB+ get number
3756 7829 LOOP conv loop
3757 7829 42 14 A0 LDB R02,R24 get mod count
3758 782C LOOP mod shift
3759 782E 67 87 LRM R47 shift down
3760 782E 42 8A DCM R02 dec count
3761 7830 F6 FA WMWZ until done
3762 7832 15 A0 LDB R02,R25 get pos count
3763 7834 LOOP pos shift
3764 7834 65 86 LRB R45 shift down
3765 7836 42 8A DCM R02 dec count
3766 7838 F6 FA WMWZ until done
3767 783A 7E 25 A0 LDB R76,R45 move for convert
3768 783C CE C8 73 JSB =CHRDG1 convert to char
3769 7840 7E 16 E6 PUMD R76,-R26 put char out
3770 7843 66 91 TSM R46 done?
3771 7845 F6 E2 WMWZ no, go on
3772 7847 52 C5 STM R22,R26 compute length
3773 7849 0A E5 PUMD R22,+R12 length to stack
3774 784B 56 E5 PUMD R26,+R12 address to stack
3775 784D 9E RIM
3776 784E
3777 784E CE FF FF ONEB+ JSB =ONEB get number
3778 7851 F9 05 ITEM overflow,
3779 7853 66 91 TSM R46 reg?
3780 7855 F5 01 IFNZ yes,
3781 7857 88 DCM R46 adjust to 8000H
3782 7858 ENDF
3783 7858 ENDF
3784 7858 9E RIM
3785 7859
3786 7859 20 2D BYT 40,55
3787 785B CE FF FF BIT. JSB =ONEB get bit number
3788 785E 66 C9 10 00 CMB R46,+R16 check size
3789 7862 FA 04 IFCY too big,
3790 7864 CE FF FF JSBN =ERR1+ error out
3791 7867 59 BYT 89D
3792 7866 ENDF
3793 7868 06 F4 PUMD R46,+R06 save it
3794 786A 1E 4E 78 JSB =ONEB+ get another number
3795 786D 65 06 E2 PUMD R45,-R06 get bit number
3796 7870 C8 08 CMB R45,-R0 which byte
3797 7872 FA 07 IFCY high,
3798 7874 C8 08 SDB R45,-R0 adjust bit count
3799 7876 5E 27 A0 LDB R36,R47 get it
3800 7879 F0 03 ELSE low,

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG102
*****
>>>>>> BUF% runtime stuff <<<<<<<<
3807 7812 18 2E BYT 30,56
3808 7814 CE FF FF BUF% JSB =GETADR get the parameter
3809 7817 SE 0A E3 PUMD R36,-R12 and the length
3810 781A 88 DCM R36 must be 1 char
3811 781B F7 04 IFNZ nope,
3812 781D CE FF FF BRDIL JSBN =ERR1+ error out
3813 781E 59 BYT 89D
3814 781F
3815 781F 1C A4 LDBD R36,R34 get the character
3816 781F CE FF FF JSB =ALFAM uppercase it
3817 781F SE C8 49 CMB R36,="I" input buffer?
3818 781F F6 05 IFEQ yes,
3819 781F A9 80 81 LDM R36,=INPBUF get the address
3820 781F F0 07 ELSE no,
3821 781F C8 4C CMB R36,="L" LCD buffer?
3822 781F F6 E9 JME BRDIL no, error
3823 781F A9 E6 82 LDM R36,=LCDBUF get the address
3824 781F ENDF
3825 781F CE FF FF JSB =RESCOM get some ran
3826 781F 60 VAL 496
3827 781F F8 04 REN nope, give up
3828 781F 6E 0A E5 PUMD R56,+R12 length to stack
3829 781F 56 E5 PUMD R26,+R12 address to stack
3830 781F LOOP move string
3831 781F 42 1E E0 PUMD R02,+R36 from buffer
3832 781F 16 E4 PUMD R02,+R26 to ram
3833 781F E6 88 DCM R56 for this many chars
3834 781F F6 F7 WMWZ until done
3835 781F 9E RIM
3836 781F

```

NOMAS

Not Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG101
*****
>>>>>> bit/base conversions <<<<<<<<
3801 787B 5E 26 A0 LDB R36,R46 get it
3802 787E ENDF
3803 787E 1E 17 6D JSB =FNDBIT test the bit
3804 7881 9E RIM
3805 7882

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG103
*****
>>>>>> LENGTH% return char count of dasp/print <<<<<<<<
3838 78BC 18 20 BYT 30,55
3839 78BE CE FF FF CCMY? JSB =GETADR get param address
3840 78C1 SE 0A E3 PUMD R36,-R12 get param length
3841 78C4 88 DCM R36 test for 1 char
3842 78C5 F7 04 IFNZ no,
3843 78C7 CE FF FF BRDLPM JSBN =ERR1+ error out
3844 78C8 59 BYT 89D
3845 78CB ENDF
3846 78CB 1C A4 LDBD R36,R34 get the char
3847 78CD CE FF FF JSB =ALFAM uppercase it
3848 78D0 SE CC 44 SDB R36,="D" test for 'D'
3849 78D3 F7 04 IFNZ no,
3850 78D5 C8 0C CMB R36,="P-D" is it 'P'
3851 78D7 F6 EE JNZ BRDLPM no, give up
3852 78D9 ENDF
3853 78D9 90 TSB R36 what was that?
3854 78D9 F6 05 IFZA display,
3855 78D9 01 D5 82 LDM R36,=DISPTR get length
3856 78D9 F0 03 ELSE printer,
3857 78E1 01 D7 82 LDM R36,=PRIPTR get length
3858 78E4 ENDF
3859 78E4 CE 45 6B JSB =MUNOUT finish up
3860 78E7 9E RIM
3861 78E8

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG104
-----
>>>>>>> FIND
3863 70E8 80 4A 2D BYI 200,112,55
3864 70E8 CE FF FF FIMD. JSB =ONEB get occur number
3865 70E8 66 91 ISM R46 test it
3866 70F0 F5 01 IFNG neg,
3867 70F2 93 CLM R46 make 0
3868 70F3 EMDIF
3869 70F3 F6 01 IFZR 0,
3870 70F5 B9 ICM R46 make 1
3871 70F6 EMDIF
3872 70F6 CE FF FF JSB =GETADR get wild card addr
3873 70F9 5E 0A E3 POND R36,-R12 and length
3874 70FC 88 DCM R36 test length
3875 70FD F6 04 IFZR 1,
3876 70FF 1C R4 LDWD R36,R34 get char
3877 7C01 F0 07 ELSE no,
3878 7C03 F4 04 IFPS >1,
3879 7C05 CE FF FF JSBN =ERR1 error
3880 7C08 59 BYI 890
3881 7C09 ELSE 0,
3882 7C09 92 CLB R36 no card
3883 7C0A EMDIF
3884 7C0A EMDIF
3885 7C0A 50 DRP 20 get target
3886 7C0B CE FF FF JSB =GETADR address
3887 7C0E 52 0A E3 POND R22,-R12 and length
3888 7C11 54 DRP 24 get pattern
3889 7C12 CE FF FF JSB =GETADR address
3890 7C15 56 0A E3 POND R26,-R12 and length
3891 7C18 50 2C A3 STM R20,R54 save start of target
3892 7C1B 52 10 C3 ADM R22,R20 compute end of string
3893 7C1E
3894 7C1E 50 2E A3 PIRMSI SIM R20,R56 save current pos
3895 7C21 54 36 A3 SIM R24,R66 copy pattern addr
3896 7C24 56 00 A3 SIM R26,R00 and length
3897 7C27 50 12 C1 CHARIC EM R20,R22 check end of target
3898 7C2A FA 08 IFCY done,
3899 7C2C 40 91 ISM R00 pattern done?
3900 7C2E F7 2F IZR PIRMSI yes, found last match
3901 7C30 5E 93 CLM R36 pos=0 (not found)
3902 7C32 F0 31 JMP FNDONE finish up
3903 7C34 EMDIF
3904 7C34 40 8B DCM R00 dec pattern len
3905 7C36 F4 1D IFPS not done,
3906 7C38 43 10 E0 POND R03,-R20 get next target
3907 7C3B 42 36 E0 POND R02,-R66 get next pattern
3908 7C3E 5E 90 ISB R36 test for wild
3909 7C40 F7 05 IFNZ yes,
3910 7C42 42 1E E0 CMB R02,R36 is this wild?
3911 7C45 F7 E0 JED CHARIC yes, auto match
3912 7C47 EMDIF
3913 7C47 42 03 E0 CMB R02,R03 target=pattern?
3914 7C4A F7 0B JEQ CHARIC yes, match
3915 7C4E 50 8B DCM R20 no, back up

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG106
-----
>>>>>>> TIMER?
3937 7C69 10 2D BYI 20,55
3938 7C6A CE FF FF TIMR? JSB =TIMER get the timer #
3939 7C6E F9 F8 REZ error, exit
3940 7C70 69 93 CLM R51 assume not there
3941 7C72 CE FF FF JSB =TMDPH get timer file
3942 7C75 F8 0A IFEZ if there,
3943 7C77 CE FF FF JSB =FSEK get timer
3944 7C7A F8 05 IFEZ if there,
3945 7C7C 69 1E 85 03 LDWD R51,R36,R3 get interval
3946 7C80 00
3947 7C81 EMDIF
3948 7C81 73 29 A1 LDM R63,R51 get lower part
3949 7C84 CE 4E 6A JSB =BIN7RL convert to seconds
3950 7C87 9C RTM
3951 7C88
3952 7C88

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG105
-----
>>>>>>> FIND
3916 7C4E 2E C1 EM R20,R56 same char?
3917 7C50 F6 01 JFED yes,
3918 7C52 89 ICM R20 don't try again
3919 7C53 EMDIF
3920 7C53 F0 E9 JMP PIRMSI restart pattern test
3921 7C55 EMDIF
3922 7C55 66 8B DCM R46 dec occur count
3923 7C57 F7 06 IFNZ not done,
3924 7C59 50 2E A1 LDM R20,R56 get start of match
3925 7C5C 89 ICM R20 start with next char
3926 7C5D F0 BF JMP PIRMSI try again
3927 7C5F EMDIF
3928 7C5F
3929 7C5F 6E 2C E5 PIRMSI SB R56,R54 compute position
3930 7C62 89 ICM R56 adjust for count
3931 7C63 1E A3 STM R56,R36 move for output
3932 7C65
3933 7C65 CE 45 6B FNDONE JSB =NUNOUT convert and output
3934 7C68 9E RTM
3935 7C69

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8102 OBJ=RYX102 11/16/1983 10:27 AM PG107
-----
>>>>>>> REPLE
3954 7C8A 80 5A 2E BYI 240,132,56
3955 7C8B 60 0A E3 REPLE. POND R40,-R12 get occurrence count
3956 7C8E 50 DRP 20 get wild char addr
3957 7C8F CE FF FF JSB =GETADR addr
3958 7C92 52 0A E3 POND R22,-R12 and length
3959 7C95 54 DRP 24 get target
3960 7C96 CE FF FF JSB =GETADR addr
3961 7C99 56 0A E3 POND R26,-R12 length
3962 7C9C 58 DRP 30 get to
3963 7C9D CE FF FF JSB =GETADR -addr
3964 7C9F 5A 0A E3 POND R32,-R12 length
3965 7CA3 CE FF FF JSB =GETADR get from addr
3966 7CA6 5E 0A E3 POND R36,-R12 length
3967 7CA9 68 18 A1 LDM R50,R30 get values
3968 7CAB 06 E5 PUND R50,-R06 save for later
3969 7CAC 54 E5 PUND R24,-R06 save for later
3970 7CB0 56 E5 PUND R26,-R06 save for later
3971 7CB2 5E 0A E5 PUND R36,-R12 put from length
3972 7CB5 51 DRP 34 and addr
3973 7CB6 CE FF FF JSB =PUTADR back
3974 7CB9 56 0A E5 PUND R26,-R12 put target length
3975 7CBC 54 DRP 24 and addr
3976 7CBD CE FF FF JSB =PUTADR back
3977 7CE0 52 0A E5 PUND R22,-R12 put wild length
3978 7CE3 50 DRP 20 and addr
3979 7CE4 CE FF FF JSB =PUTADR back
3980 7CE7 60 0A E5 PUND R40,-R12 put count back
3981 7CEA CE EB 7B JSB =FIND go find the from string
3982 7CED CE FF FF JSB =ONEB get its location
3983 7CD0 74 06 E3 POND R64,-R06 get target addr/len
3984 7CD3 78 E3 POND R70,-R06 get to/from addr/len
3985 7CD5 66 8B DCM R46 adjust for count
3986 7CD7 F5 05 IFNG not found,
3987 7CD9 93 CLM R46 make position
3988 7CDA 3A A3 STM R46,R72 to length
3989 7CDE 3E A3 STM R46,R76 and from length 0
3990 7CDE EMDIF
3991 7CDE 6E 36 A1 LDM R56,R66 new length = target len
3992 7CE1 3A C3 ADM R56,R72 + to length
3993 7CE3 3E C5 SB R56,R76 - from length
3994 7CE5 CE FF FF JSB =RSHEN get the ran
3995 7CE8 F8 9D RTM nope, give up
3996 7CEA 6E 0A E5 PUND R56,-R12 length to stack
3997 7CEB 56 E5 PUND R26,-R12 addr to stack
3998 7CEB 50 16 A1 LDM R20,R26 to start at new
3999 7CF2 14 A3 STM R20,R24 (save for later)
4000 7CF4 26 C3 ADM R20,R46 + from position
4001 7CF6 52 10 A1 LDM R22,R20 to ends at to start
4002 7CF9 3A C3 ADM R22,R72 + to length
4003 7CFB 54 2E C3 ADM R24,R56 new ends at new + new length
4004 7CFE 58 34 A1 LDM R30,R64 rest starts at target
4005 7D01 26 C3 ADM R30,R46 + from position
4006 7D03 3E C3 ADM R30,R76 + from length

```



```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&102 OBJ=RYX102 11/16/1983 10:27 AM PG108
-----
>>>>>> REPLIC <<<<<<<
4007 7D05 LOOP move target loop
4008 7D05 56 10 C1 CLM R26,R20 at from position?
4009 7D08 F8 07 JCY D010 yes, go do to
4010 7D0A 42 34 E0 P0BD R02,R64 get target char
4011 7D0D 16 E4 P0BD R02,R26 put target char
4012 7D0F F0 F4 UNHP until done
4013 7D11 DOTW LOOP move to loop
4014 7D11 56 12 C1 CLM R26,R22 at rest position?
4015 7D14 F8 07 JCY D0E5T yes, do rest
4016 7D16 42 38 E0 P0BD R02,R70 get to char
4017 7D19 16 E4 P0BD R02,R26 put to char
4018 7D1B F0 F4 UNHP until done
4019 7D1D D0E5T LOOP move rest loop
4020 7D1D 56 14 C1 CLM R26,R24 at end?
4021 7D20 F8 07 JCY yes, all done
4022 7D22 42 18 E0 P0BD R02,R30 get rest char
4023 7D25 16 E4 P0BD R02,R26 put rest char
4024 7D27 F0 F4 UNHP until done
4025 7D29 RIM for conditional returns
4026 7D29 9C
4027 7D2A

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&102 OBJ=RYX102 11/16/1983 10:27 AM PG110
-----
>>>>>> LEF16 <<<<<<<
4050 7D4F 28 2E BYT 50,56
4051 7D51 68 0A E3 LFTB. P0MD R50,-R12 get the end number
4052 7D54 60 93 CLM R40 build the start
4053 7D56 64 A8 FF LDB R44,-00FFH number
4054 7D59 60 E5 P0MD R40,R12 onto stack
4055 7D5B 68 E5 P0MD R50,R12 put end back
4056 7D5D F0 2E JMP SUB8. go do the string
4057 7D5F

```

NOMAS

NOT MANUFACTURER SUPPORTED
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&102 OBJ=RYX102 11/16/1983 10:27 AM PG109
-----
>>>>>> RIGHT16 <<<<<<<
4029 7D2A 28 2E BYT 50,56
4030 7D2C CE FF FF RIGHT16. JSB =DNEB get count number
4031 7D2F 66 91 ISM R46 test it
4032 7D31 F5 01 IFNG -.
4033 7D33 93 CLM R46 make 0
4034 7D34 ENDIF
4035 7D34 50 0A E3 P0MD R20,-R12 get address and
4036 7D37 5E E3 P0MD R36,-R12 length off
4037 7D39 E5 P0MD R36,R12 and put back
4038 7D3A 50 E5 P0MD R20,R12 for later
4039 7D3C 5E 06 E5 P0MD R36,R06 save for later
4040 7D3F 26 C5 SBN R36,R46 compute position
4041 7D41 F4 01 IFPS positive.
4042 7D43 89 ICM R36 adjust for count
4043 7D44 ENDIF
4044 7D44 CE 45 6B JSB =MUMOUT start to address
4045 7D47 5E 06 E3 P0MD R36,-R06 len back
4046 7D4A CE 45 6B JSB =MUMOUT1 put len out as end
4047 7D4D F0 3E JMP SUB8. go do the string
4048 7D4F

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&102 OBJ=RYX102 11/16/1983 10:27 AM PG111
-----
>>>>>> MID6 <<<<<<<
4059 7D5F 00 38 2E BYT 0,70,56
4060 7D62 68 0A E3 MID6. P0MD R50,-R12 get count
4061 7D65 60 E3 P0MD R40,-R12 get start number
4062 7D67 E5 P0MD R40,R12 put start back
4063 7D68 68 E5 P0MD R50,R12 put count back
4064 7D6A 60 E5 P0MD R40,R12 put start again
4065 7D6C E5 P0MD R40,R12 put start again
4066 7D6D CE FF FF JSB =DNEB get start back
4067 7D70 99 BCD for address
4068 7D71 F5 05 IFNG neg.
4069 7D73 CE FF FF JSB =ADDR01 start = count = end
4070 7D76 F0 10 ELSE pos.
4071 7D78 CE FF FF JSB =ADDR01 start = count = end
4072 7D7B 99 BCD for number build
4073 7D7C 65 93 CLM R45 build an integer
4074 7D7E 8B DCM R45 - 1
4075 7D7F 64 A8 FF LDB R44,-FF mark as integer
4076 7D82 60 0A E5 P0MD R40,R12 put on stack
4077 7D85 CE FF FF JSB =ADDR01 add to end
4078 7D88 ENDIF
4079 7D8B F0 03 JMP SUB8. go do it
4080 7D8A

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV8102 OBJ=RV2102 11/16/1983 10:27 AM PG112
-----
>-----> SUB#
4082 ZDBA 00 38 2E BYT 0,70,56
4083 ZDBB 00 FF FF SUB# JSB -ONEB get end number
4084 ZDBB 66 91 TSM R46 test it
4085 ZDBB 66 91 IFNG neg?
4086 ZDBB 93 CLM R46 yes, make it 0
4087 ZDBB 93 ENDIF
4088 ZDBB 06 E5 PUMD R46,+R06 save for later
4089 ZDBB 06 FF FF JSB -ONEB get start position
4090 ZDBB 50 93 CLM R20 clear pad count
4091 ZDBB 66 8D CLM R46 adjust for inc end
4092 ZDBB 66 8D IFNG neg?
4093 ZDBB 10 A3 TSM R46,R20 save count
4094 ZDBB 93 CLM R46 make 0
4095 ZDBB 50 89 TSM R20 restore count
4096 ZDBB 8D TSM R20 make positive
4097 ZDBB 66 06 E3 ENDIF
4098 ZDBB 66 06 E3 PUMD R56,-R06 get end back
4099 ZDBB 58 DRP 30 get address
4100 ZDBB 06 FF FF JSB -GETADM of string
4101 ZDBB 58 0A E3 PUMD R36,-R12 and length
4102 ZDBB 66 26 C5 SBN R56,R46 compute length
4103 ZDBB 66 26 C5 IFNG neg,
4104 ZDBB 93 CLM R56 make it 0
4105 ZDBB 66 06 E3 ENDIF
4106 ZDBB 10 C3 ADM R56,R20 add in padding
4107 ZDBB 06 FF FF JSB -RSNEM- get some ram
4108 ZDBB 66 2E REM no go
4109 ZDBB 66 0A E5 PUMD R56,+R12 length to stack
4110 ZDBB 58 0A E5 PUMD R26,+R12 address to stack
4111 ZDBB 58 18 C3 ADM R36,R30 compute end of string
4112 ZDBB 58 26 C3 ADM R30,R46 start of substring
4113 ZDBB 58 26 C3 LOOP pad loop
4114 ZDBB 58 26 C3 CLM R20 dec count
4115 ZDBB 58 26 C3 JNG SUB# done, go on
4116 ZDBB 58 26 C3 CLM R56 count out padding
4117 ZDBB 58 26 C3 LDB R03,=" get a blank
4118 ZDBB 58 26 C3 PUBD R03,+R26 put it out
4119 ZDBB 58 26 C3 WHNP until done
4120 ZDBB 58 26 C3 SUB# LOOP move string loop
4121 ZDBB 58 26 C3 CLM R56 dec count
4122 ZDBB 58 26 C3 RNG all done
4123 ZDBB 58 26 C3 CLM R30,R36 past end of string?
4124 ZDBB 58 26 C3 IFCY yes,
4125 ZDBB 58 26 C3 LDB R03,=" get blank
4126 ZDBB 58 26 C3 ELSE no,
4127 ZDBB 58 26 C3 PUBD R03,+R30 get next char
4128 ZDBB 58 26 C3 ENDIF
4129 ZDBB 58 26 C3 PUBD R03,+R26 put out char
4130 ZDBB 58 26 C3 WHNP until done
4131 ZDBB 58 26 C3

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV8102 OBJ=RV2102 11/16/1983 10:27 AM PG114
-----
>-----> FILL# fill string (left/right/middle)
4137 ZDBB 80 4A 2E BYT 200,112,56
4138 ZDBB 06 FF FF FILL# JSB -ONEB get specified size
4139 ZDBB 66 91 TSM R46 ok?
4140 ZDBB 66 91 IFNG neg,
4141 ZDBB 93 CLM R46 make it 0
4142 ZDBB 93 ENDIF
4143 ZDBB 2E A3 SBN R46,R56 move for RSNEM-
4144 ZDBB 06 FF FF JSB -RSNEM- get some room
4145 ZDBB 66 8D REM none, give up
4146 ZDBB 66 8D JSB -GETADR get right string
4147 ZDBB 58 0A E3 PUMD R36,-R12 and length
4148 ZDBB 58 0A E3 DRP 20 get middle
4149 ZDBB 58 0A E3 JSB -GETADM string and
4150 ZDBB 58 0A E3 PUMD R22,-R12 length
4151 ZDBB 58 0A E3 DRP 30 get left
4152 ZDBB 58 0A E3 JSB -GETADM string and
4153 ZDBB 58 0A E3 PUMD R32,-R12 length
4154 ZDBB 58 0A E3 PUMD R56,+R12 length to stack
4155 ZDBB 58 0A E3 PUMD R26,+R12 address to stack
4156 ZDBB 58 0A E3 LDM R24,R56 get new length
4157 ZDBB 58 0A E3 SBN R24,R22 compute over/under flow
4158 ZDBB 58 0A E3 IFNG overflow,
4159 ZDBB 58 0A E3 CLM R24 make extra 0
4160 ZDBB 58 0A E3 ENDIF
4161 ZDBB 58 0A E3 ADM R56,R26 compute end address
4162 ZDBB 58 0A E3 JSIFY? CLM R66 assume left justify
4163 ZDBB 58 0A E3 TSM R32 left justify?
4164 ZDBB 58 0A E3 IFNZ no,
4165 ZDBB 58 0A E3 LDM R66,R24 get space size
4166 ZDBB 58 0A E3 TSM R36 right justify?
4167 ZDBB 58 0A E3 IFNZ no,
4168 ZDBB 58 0A E3 LRM R66 divide by two (sides)
4169 ZDBB 58 0A E3 ENDIF
4170 ZDBB 58 0A E3 SBN R24,R66 adjust remainder
4171 ZDBB 58 0A E3 ELSE yes,
4172 ZDBB 58 0A E3 TSM R36 right justify?
4173 ZDBB 58 0A E3 IFZR yes,
4174 ZDBB 58 0A E3 LDM R30,R20 copy middle addr
4175 ZDBB 58 0A E3 LDM R32,R22 copy middle size
4176 ZDBB 58 0A E3 CLM R34 clear blank count
4177 ZDBB 58 0A E3 LOOP count blank loop
4178 ZDBB 58 0A E3 CLM R32 dec size
4179 ZDBB 58 0A E3 JNG -WIR? done, go on
4180 ZDBB 58 0A E3 PUBD R02,+R30 get char
4181 ZDBB 58 0A E3 CLM R02,=" is it blank?
4182 ZDBB 58 0A E3 IFEQ yes,
4183 ZDBB 58 0A E3 CLM R34 anc count
4184 ZDBB 58 0A E3 ENDIF
4185 ZDBB 58 0A E3 WHNP go on
4186 ZDBB 58 0A E3 TSM R34 blanks found?
4187 ZDBB 58 0A E3 JNZ FILL# yes,
4188 ZDBB 58 0A E3
4189 ZDBB 58 0A E3 FILL# LDM R30,=BLANKS get left addr

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV8102 OBJ=RV2102 11/16/1983 10:27 AM PG113
-----
>-----> dummy for next routine
4133 ZDBB 01 BYT 241
4134 ZDBB 01 FROMX1 R1M
4135 ZDBB 01

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV8102 OBJ=RV2102 11/16/1983 10:27 AM PG115
-----
>-----> FILL# fill string (left/right/middle)
4190 ZDBB 1C A3 SBN R30,R34 copy to right
4191 ZDBB 1C A3 CLM R32 build a
4192 ZDBB 1C A3 CLM R32 size of 1
4193 ZDBB 1C A3 SBN R32,R36 left and right
4194 ZDBB 1C A3 JNZ JSIFY? try again
4195 ZDBB 1C A3 ENDIF
4196 ZDBB 1C A3 ENDIF
4197 ZDBB 1C A3 LOOP
4198 ZDBB 1C A3 LDM R76,R30 copy left loop
4199 ZDBB 1C A3 LDM R00,R32 get left addr
4200 ZDBB 1C A3 LDM R00,R32 get left size
4201 ZDBB 1C A3 LOOP move left loop
4202 ZDBB 1C A3 CLM R66 dec space size
4203 ZDBB 1C A3 JNG DON1DL done, go on
4204 ZDBB 1C A3 PUBD R02,+R76 get left char
4205 ZDBB 1C A3 PUBD R02,+R26 put left char
4206 ZDBB 1C A3 CLM R00 dec char count
4207 ZDBB 1C A3 WHNZ until all gone
4208 ZDBB 1C A3 WHNP start next copy
4209 ZDBB 1C A3
4210 ZDBB 1C A3 DON1DL LOOP
4211 ZDBB 1C A3 CLM R22 move middle loop
4212 ZDBB 1C A3 JNG DON1TE dec char count
4213 ZDBB 1C A3 PUBD R02,+R20 done, go on
4214 ZDBB 1C A3 PUBD R02,+R26 get middle char
4215 ZDBB 1C A3 CLM R56,R26 put middle char
4216 ZDBB 1C A3 RWC past end?
4217 ZDBB 1C A3 WHNP yes, all done
4218 ZDBB 1C A3
4219 ZDBB 1C A3 DON1TE LDM R02,R24 until all gone
4220 ZDBB 1C A3 LOOP divide loop
4221 ZDBB 1C A3 SBN R02,R36 subtract right size
4222 ZDBB 1C A3 JZR FILL# exact fit, leave
4223 ZDBB 1C A3 WHNP until all gone
4224 ZDBB 1C A3 CLM R02 complement left over
4225 ZDBB 1C A3 LDM R76,R34 get right addr
4226 ZDBB 1C A3 RDM R76,R02 add offset
4227 ZDBB 1C A3 LDM R00,R36 get right size
4228 ZDBB 1C A3 SBN R00,R02 adjust for offset
4229 ZDBB 1C A3 JNZ DON1E go start it
4230 ZDBB 1C A3
4231 ZDBB 1C A3 FILL# LOOP
4232 ZDBB 1C A3 LDM R76,R34 copy right loop
4233 ZDBB 1C A3 LDM R00,R36 get right addr
4234 ZDBB 1C A3 LOOP get right size
4235 ZDBB 1C A3 CLM R24 move right loop
4236 ZDBB 1C A3 RWC dec space size
4237 ZDBB 1C A3 JNG done
4238 ZDBB 1C A3 PUBD R02,+R76 get right char
4239 ZDBB 1C A3 PUBD R02,+R26 put right char
4240 ZDBB 1C A3 CLM R00 dec char count
4241 ZDBB 1C A3 WHNZ until all gone
4242 ZDBB 1C A3 WHNP start next copy

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&102 OBJ=RY&102 11/16/1983 10:27 AM PG116
=====
>>>>>> FILLB fill string (left/right/middle)
RTM for the conditional returns
4243 7EB1 9E
4244 7EB2
4245 7EB2
4246 7EB2 SE 93 FILLB CLM R36 clear divide accum
4247 7EB4 LOOP divide loop
4248 7EB4 SE 89 JCM R36 jnc accum
4249 7EB6 SE 1E C9 SBM R24,R34 subtract blank count
4250 7EB9 F5 F9 GAPS until done
4251 7EBB C3 RCM R24,R34 add last subtract
4252 7EBE SE 8B DCM R36 adjust accum
4253 7EBE SE 14 E5 SBM R34,R24 compute center count
4254 7EC1 SD 87 LRM R35 divide by 2
4255 7EC3 LOOP fall loop
4256 7EC3 SE 8B DCM R22 dec char count
4257 7EC5 F4 EA RMC all done
4258 7EC7 42 10 E0 PDBD R02,R20 get char
4259 7ECA 16 E4 PUMB R02,R26 put char
4260 7ECB C8 20 CMB R02,0 was it blank?
4261 7ECE F6 17 IFB0 yes,
4262 7ED0 40 1E A1 LDM R00,R36 get blank count
4263 7ED3 LOOP blank put loop
4264 7ED3 40 8B DCM R00 dec count
4265 7ED5 F4 05 JNG XTRABL done, go on
4266 7ED7 42 16 E4 PUMB R02,R26 put blank
4267 7ED8 F0 F7 MHP until done
4268 7EDC SE 8B XTRABL DCM R34 dec center count
4269 7EDE F5 07 IFNG all gone,
4270 7EE0 SE 8B DCM R24 dec leftover count
4271 7EE2 F4 03 IFPS none left,
4272 7EE4 42 16 E4 PUMB R02,R26 put one out
4273 7EE7 ENDF
4274 7EE7 ENDF
4275 7EE7 ENDF
4276 7EE7 F0 DA MHP until all done
4277 7EE9

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&102 OBJ=RY&102 11/16/1983 10:27 AM PG118
=====
>>>>>> ADJUST stuff
4332 7F5D F6 E0 JNZ BDFLG no, error
4333 7F5F 43 C8 08 RDB R03,=0008H set flag
4334 7F62 ENDF
4335 7F62 43 B2 F6 B3 STBD R03,=PSS1AT save new inc/dec flag
4336 7F66 61 1C A5 LDM R41,R34 get the string
4337 7F69 F7 04 IFNZ if not zero
4338 7F6B E2 91 TSM R42 must be >>100H
4339 7F6D F7 D0 JZR BDFLG not, error
4340 7F6E ENDF
4341 7F6F 60 A8 2A LDB R40,=CT.ADJ add clock flag
4342 7F72 CE FF FF JSB =CRPEMT set comparator
4343 7F75 SE RTM
4344 7F76

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&102 OBJ=RY&102 11/16/1983 10:27 AM PG117
=====
>>>>>> ADJUST stuff
4279 7EF9 00 ZE BYI 0,56
4280 7EEB 98 ADJST. BIN for sure
4281 7EEC CE FF FF JSB =RESCON get room for
4282 7EEF 07 BYI 7D the string
4283 7EFO F8 BF REN give up
4284 7EF2 6E 0A E5 PUMB R56,R12 length to stack
4285 7EF5 79 B1 00 B5 LDM R71,=THEIME get time interval
4286 7EF9 16 A7 STD R71,R26 put out as string
4287 7EFB 56 0A E5 PUMB R26,R12 address to stack
4288 7EFE CE 9D 6F JSB =HEX8 convert to hex
4289 7F01 43 B0 F6 B3 LDB R03,=PSS1AT get inc/dec flag
4290 7F05 F8 08 RCM R03,=08H test it
4291 7F07 F6 04 IFZR inc,
4292 7F09 A8 20 LDB R03,=0 get inc char
4293 7F0B F0 02 ELSE dec,
4294 7F0D A8 20 LDB R03,=0 get dec char
4295 7F0F ENDF
4296 7F0F 56 0A E3 PUMB R26,-R12 get string addr
4297 7F12 54 E3 PUMB R24,-R12 get string len
4298 7F14 B9 JCM R24 add char
4299 7F15 E5 PUMB R24,R12 len back
4300 7F16 B8 DCM R24 restore length
4301 7F17 56 E5 PUMB R26,R12 put addr back
4302 7F19 14 C3 RCM R26,R24 point to end
4303 7F1B 43 16 A6 STBD R03,R26 put char there
4304 7F1E CE 46 6F JSB =REV8 reverse number
4305 7F21 9E RTM
4306 7F22
4307 7F22 A1 BYI 241
4308 7F23 CE FF FF ADJST. JSB =GETADR get string addr
4309 7F26 SE 0A E3 PDM R36,-R12 and length
4310 7F29 B8 DCM R36 dec length
4311 7F2A E5 PUMB R36,R12 put length back
4312 7F2B 58 1C E0 PDBD R30,R34 get first char
4313 7F2E CE FF FF JSB =PUTREL put back new address
4314 7F31 58 06 E4 PUMB R30,R06 save char
4315 7F34 CE 46 6F JSB =REV8 unreverse number
4316 7F37 F8 06 JEN ADJERR error out
4317 7F39 CE C8 6F JSB =ASC8 to convert to binary
4318 7F3C CE FF FF JSB =GETADR get the string addr
4319 7F3F 58 06 E2 ADJERR PDBD R30,-R06 get back char
4320 7F42 F8 D0 REN error out
4321 7F44 SE 0A E3 PDM R36,-R12 get string length
4322 7F47 F9 07 00 LCM R36,=87 proper len?
4323 7F4A F7 04 IFNE no,
4324 7F4C CE FF FF JSB =ERR1+ error out
4325 7F4F 59 BYI B30
4326 7F50 ENDF
4327 7F50 43 B0 F6 B3 LDB R03,=PSS1AT get inc/dec flag
4328 7F54 F7 F7 RCM R03,=00F7H clear flag
4329 7F56 58 C8 20 CMB R30,=0 char = inc
4330 7F59 F7 07 IFNZ no,
4331 7F5B C8 20 CMB R30,=0 char = dec

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&102 OBJ=RY&102 11/16/1983 10:27 AM PG119
=====
>>>>>> EXIT for statement
4346 7F76 A1 BYI 241
4347 7F77 98 EXIT. BIN for the good of all
4348 7F78 CE FF FF JSB =R1SIGN trash stack, rsmen
4349 7F7B 58 08 B5 FA LDM R30,X10,8-6 get index rel addr
4349 7F7F FF
4350 7F80 06 E5 PUMB R30,R06 save for later
4351 7F82 54 08 A1 LDM R24,R10 copy PC
4352 7F85 76 B1 4D B2 LDM R66,=PCR save PCR
4353 7F89 66 36 A1 LDM R46,R66 copy PCR
4354 7F8C CE FF FF JSB =SKPLN skip to next line
4355 7F8F 65 26 A5 LDM R45,R46 get lineW, count
4356 7F92 CE FF FF JSB =SKPLOP find matching NEXT
4357 7F95 98 BIN back to binary
4358 7F96 52 06 E3 PDM R22,-R06 get back index
4359 7F99 4F 90 TSB R17 any errors?
4360 7F9B F4 D8 RMC yes, give up
4361 7F9D LOOP pop for loop
4362 7F9D SE B1 5B B2 LDM R36,=LAVRIL get for stack pointer
4363 7F9F 54 1E B5 EA LDM R24,X36,8-22 get last index
4363 7FA5 FF
4364 7FA6 12 E1 CCM R24,R22 was it us?
4365 7FA8 F7 C0 REQ yes, all done
4366 7FAA CE FF FF JSB =EXNXT no, pop it off
4367 7FAD F6 EE UNHZ until done
4368 7FAF CE FF FF JSB =ERR1+ error out
4369 7FB2 2F BYI 47D no matching for
4370 7FB3

```

KARMA 08/31/82-----
 ITEM LOC OBJECT CODE SRC=RV8102 OBJ=RVX102 11/16/1983 10:27 AM PG120

 CATM, a function
 4372 7FB3 10 2D BY1 20,55
 4373 7FB5 1E FF FF CATM JSB =ONE0 get the N
 4374 7FB8 65 92 CLB R45 clear next flag
 4375 7FB8 66 91 ISM R46 test the number
 4376 7FBC 15 64 IFNG ~, make it 0
 4377 7FDE 93 CLM R46
 4378 7FBF ENDF
 4379 7FDF 16 03 IFZR 0,
 4380 7FC1 B9 ICM R46 make it 1
 4381 7FC2 65 88 ICB R45 set next flag
 4382 7FC4 ENDF
 4383 7FC4 65 06 E5 PUND R45, R06 save for later
 4384 7FC7 1E FF FF JSB =IO? open the logfile
 4385 7FCA 65 06 E3 POND R45, -R06 get number back
 4386 7FCD 18 A6 REN ZOFFLE not there
 4387 7FCE LOOP find line loop
 4388 7FCE 1E FF FF JSB =SKPCNK skip to next line
 4389 7FDF 16 07 IFZR end of file,
 4390 7FDF 65 90 ISB R45 check next flag
 4391 7FDF 66 0C JNZ LNMEND yes, finish up
 4392 7FDF 93 CLM R45 return 0
 4393 7FDF 10 1C JMP CATM finish up
 4394 7FDF ENDF
 4395 7FDF 65 90 ISB R45 test next flag
 4396 7FDF 17 0F IFNZ yes,
 4397 7FDF 66 1E D9 CMND R46, R36 check line#
 4398 7FDF 17 05 IFNE no match,
 4399 7FDF 66 25 A3 LNMEND SJM R46, R45 move for output
 4400 7FDF 10 0C JMP FNDLHM finish up
 4401 7FDF ENDF
 4402 7FDF 99 BCD for inc
 4403 7FDF 89 ICM R46 try next N
 4404 7FDF 98 BIN back to normal
 4405 7FDF 10 0C ELSE no,
 4406 7FDF 66 88 DCM R46 dec line count
 4407 7FDF ENDF
 4408 7FDF 16 0D UNHZ until not there
 4409 7FDF 65 1E A5 LDM R45, R36 get the line number
 4410 7FDF 67 92 FNDLHM CLB R47 clear garbage
 4411 7FDF 64 08 FF CATM LDB R44, =OOFH get integer flag
 4412 7FDF 60 0A E5 PUND R40, R12 number to stack
 4413 7FDF 9E RIN
 4414 7FDF
 4415 7FDF LEXEND
 4416 7FDF FIN

KARMA 08/31/82-----
 SYMBOL VALUE TYPE COUNT Symbol Table 11/16/1983 10:27 AM PG121

 8-10 FFF6 EQU 1
 8-22 FFER EQU 1
 8-6 FFER EQU 1
 81 0001 EQU 4
 810 000A EQU 3
 812 000C EQU 2
 813 000D EQU 1
 815 000F EQU 1
 8159C 0159 EQU 1
 816 0010 EQU 2
 818 0012 EQU 2
 82 0002 EQU 2
 820 0014 EQU 2
 82020H 2020 EQU 3
 823 0017 EQU 1
 8254 00FE EQU 2
 8256 0100 EQU 4
 826 001C EQU 1
 8288 0120 EQU 1
 83 0003 EQU 2
 832 0020 EQU 2
 833 0021 EQU 2
 84 0004 EQU 4
 849 0031 EQU 1
 86 0006 EQU 1
 87 0007 EQU 1
 88 0008 EQU 1
 880 0050 EQU 1
 88000H 8000 EQU 1
 881 0051 EQU 2
 896 0060 EQU 2
 8:-0 000A EQU 1
 8A-0 0011 EQU 1
 8A:- 0007 EQU 2
 8E088H E088 EQU 1
 8E089H E089 EQU 1
 8E08BH E08B EQU 1
 8E402H E400 EQU 1
 8P-D 000C EQU 1
 8P-M 0002 EQU 1
 8P-F 000E EQU 1
 8P-P 0005 EQU 1
 8V-F 0010 EQU 1
 82+1 005B EQU 1
 8A0001 7727 LCL 1
 8A0001 6075 LCL 1
 8A0001 848E G DAD 1
 8A0001 FFFF EXT 1
 8A0001 FFFF EXT 5
 8A0001 7F3F LCL 1
 8A0001 7FEB LCL 1
 8A0001 7F23 LCL 1
 8A0001 FFFF EXT 12
 8A0001 FFFF EXT 1

KARMA 08/31/82-----
 SYMBOL VALUE TYPE COUNT Symbol Table 11/16/1983 10:27 AM PG122

 ANN.E+ FFFF EXT 2
 AOR8. 6082 LCL 1
 APRNME 6490 LCL 2
 APRNPH 64C6 EXT 6
 AR018. 6C64 LCL 1
 ASE8. 6FC8 LCL 5
 ASE18. 70CE LCL 1
 ASELEX 611A LCL 1
 ASMF8. 6L6E LCL 1
 ASTCHK 6342 LCL 2
 ASINME 6355 LCL 2
 ATSIGN FFFF EXT 1
 AUTOLP 844A G DAD 2
 AKOR8. 6D7C LCL 1
 BADOGT 720B LCL 4
 BACF1L 721A LCL 1
 BADFLG 7F4C LCL 2
 BADIL 788D LCL 1
 BADLPM 78C7 LCL 1
 BADSPC 7812 LCL 3
 BADSRC 76A6 LCL 8
 BADTAR 76AE LCL 6
 BADIEM 7241 LCL 2
 BASEND FFFF EXT 1
 BEEPK 82A0 G DAD 2
 BIN7RL 6A4E LCL 1
 BINUGT 71DF LCL 4
 BINHEX 6F80 LCL 1
 BIN1 0002 G EQU 1
 BIN2 0004 G EQU 2
 BIN4 0010 G EQU 2
 BIN5 0020 G EQU 3
 BIN7 0080 G EQU 1
 BIT. 785B LCL 1
 BITFND 6A04 LCL 1
 BLACK 7127 LCL 1
 BLANKS FFFF EXT 5
 BLEBU FFFF EXT 1
 BLMP FFFF EXT 4
 BREAK. 7006 LCL 1
 BSCEND 7AFD LCL 1
 BSCERR 7A0B LCL 2
 BSTDCC 7AC7 LCL 2
 BSYALL 7996 LCL 3
 BTD. 7A0C LCL 1
 BUFA. 7834 LCL 1
 CATM. 7F85 LCL 1
 CRIBUF FFFF EXT 1
 CATME 7F77 LCL 1
 CCHI? 78BE LCL 1
 CHK75C 77B5 LCL 1
 CHKH1 637A LCL 1
 CHNG11 6EES LCL 1
 CHRG1 73C8 LCL 4

Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

NOMAS

KARMA 08/31/82-----
 SYMBOL VALUE TYPE COUNT Symbol Table 11/16/1983 10:27 AM PG123

 CHRRIC 7C27 LCL 2
 CHSRDI FFFF EXT 1
 CRIMPL 730A LCL 2
 CLAPR 64C0 LCL 2
 CLACOD FFFF EXT 1
 CLARR FFFF EXT 2
 CLAREY 008B G EQU 1
 CMREP FFFF EXT 1
 CMPEH FFFF EXT 1
 CNL 0040 G EQU 2
 CNTR? 7E50 LCL 1
 CONMA 002C G EQU 1
 CONBJH FFFF EXT 6
 CONVRT 6B5C LCL 1
 COPY. FFFF EXT 1
 COPYTK 007C G EQU 2
 CRMPY FFFF EXT 1
 CROSIS 8461 G DAD 7
 CRAPR 6499 LCL 3
 CRIFLE FFFF EXT 1
 CT,ADJ 002A EQU 1
 CURLNW 827D G DAD 1
 CURSE+ FFFF EXT 1
 CURSE- FFFF EXT 1
 DC10BS 781A LCL 2
 DLREP FFFF EXT 2
 DEFVAL 8377 G DAD 2
 DELAY 8287 G DAD 1
 DELETE FFFF EXT 2
 DELY? 6A49 LCL 1
 DEVEHT 828C G DAD 1
 DFAPH1 7020 LCL 1
 DFKYB. 6A12 LCL 1
 DIRDWH 77C7 LCL 1
 DIREND 780A LCL 1
 DISP8. 6986 LCL 1
 DISPLN 82CB G DAD 1
 DISPR 82D5 G DAD 1
 DIV2 FFFF EXT 2
 DNAME 0003 G EQU 3
 DNOU1 69CE LCL 1
 DOWSTF 68D9 LCL 1
 DORAX 6185 LCL 3
 DORASR 6CB1 LCL 1
 DORAX 6C32 LCL 2
 POLCT 6996 LCL 2
 OODJHL 7E76 LCL 1
 OORCT 7F10 LCL 1
 OORIE 7E85 LCL 1
 OORIE+ 7E87 LCL 1
 OOSPB 7010 LCL 1
 OOSRQ 6408 LCL 1
 OORAR 63E8 LCL 1
 OOR 7011 LCL 1

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 11/16/1983 10:27 AM PG124
DOULB	6EB9	LCL	2	
DR.NAM	000A	G DAD	1	
DR.TYP	0004	G DAD	1	
DRC	8281	G DAD	2	
DIBL	7006	LCL	1	
DINH	7816	EXT	2	
DIOB	780E	LCL	1	
EOFILE	8245	G DAD	4	
EQNAME	8268	G DAD	1	
END.OM	FFFF	EXT	1	
ENDLS	69F0	LCL	1	
ENDLIN	8999	G DAD	2	
ENDSIZ	0005	G EQU	1	
EOL	82C1	G DAD	1	
COLLEN	82C4	G DAD	1	
ERNUMM	837A	G DAD	2	
ERDUMK	00B4	G EQU	1	
ERR1	FFFF	EXT	6	
ERR1+	FFFF	EXT	25	
ERRLEX	60F0	LCL	1	
ERRDR	FFFF	EXT	1	
ERRDR.	7525	LCL	1	
ERRDRR	FFFF	EXT	1	
ERRORS	837B	G DAD	2	
ESC6	6E85	LCL	1	
EVIL	FFFF	EXT	1	
EXIT	6450	LCL	1	
EXIT.	7F77	LCL	1	
EXXKI	FFFF	EXT	1	
FCRABO	FFFF	EXT	1	
FCRABU	FFFF	EXT	1	
FF	00FF	G EQU	3	
FILE9	68C8	LCL	1	
FILLS	70EF	LCL	1	
FILLT	7EB2	LCL	1	
FILLSZ	84A1	G DAD	2	
FILLYP	84A3	G DAD	1	
FINDV1	72B2	LCL	1	
FINDV	78E8	LCL	2	
FINDSG	FFFF	EXT	1	
FLAG6	6ABD	LCL	1	
FLAG7	6B0B	LCL	1	
FLGCHK	66F8	LCL	1	
FLGERR	6A97	LCL	1	
FLGPAC	6AE9	LCL	2	
FLGID+	681E	LCL	1	
FLGIDL	6816	LCL	2	
FLHEAD	849D	G DAD	2	
FLOPEN	FFFF	EXT	1	
FLSTAK	FFFF	EXT	3	
FMDBIT	6817	LCL	1	
FMDVME	7C65	LCL	1	
FMDLNM	7F15	LCL	1	
FOPEN	FFFF	EXT	4	

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 11/16/1983 10:27 AM PG126
IRKEY	0088	G EQU	1	
JENCE0	78D7	LCL	3	
JENCE1	78A5	LCL	1	
JENCE2	783F	LCL	2	
JENCE3	7830	LCL	1	
JENCE4	77C5	LCL	1	
JENCE5	7758	LCL	2	
JENCE6	773C	LCL	2	
JENCE7	7703	LCL	2	
JENCE8	7694	LCL	3	
JENCE9	762A	LCL	1	
JSHV?	7E27	LCL	1	
JUSTSO	82DF	G DAD	2	
KEY8	FFFF	EXT	1	
KEYDEF	634E	LCL	1	
KEYINT	66C2	LCL	1	
KEYNAM	FFFF	EXT	2	
KEYNOP	63EC	LCL	1	
KYRDRR	FFFF	EXT	1	
KYRILE	83F8	G DAD	3	
LA	FFFF	EXT	1	
LAVRIL	825B	G DAD	1	
LCD	7151	LCL	1	
LCDBUF	82E6	G DAD	4	
LCDOFF	82E0	G DAD	1	
LCDP1R	82E2	G DAD	1	
LCDVIM	82E4	G DAD	1	
LDPDNE	69E8	LCL	2	
LEXBGN	6016	LCL	2	
LEXEND	7FFE	LCL	1	
LF18	7D51	LCL	1	
LINEM	000C	G EQU	1	
LINEM.	68D7	LCL	1	
LISTEN	7961	LCL	4	
LNNEHD	7FE4	LCL	1	
LOOPUP	77CB	LCL	1	
LS118	6994	LCL	1	
LS1LM	7550	LCL	1	
LSTRK?	66B5	LCL	2	
LTRMS	6F3E	LCL	1	
LMRL8	6E49	LCL	1	
MAKPRV	7888	LCL	1	
MAR6	7077	LCL	1	
MARPDNE	70C4	LCL	1	
MARPCM	709B	LCL	1	
MARG?	6B27	LCL	1	
MARNAX	8451	G DAD	2	
MCOPIV	75B5	LCL	1	
MCPEND	78E6	LCL	3	
MCPNXT	7747	LCL	1	
MELJSH	FFFF	EXT	40	
MID8	7D62	LCL	1	
MJMHIM	8450	G DAD	2	
MDDIO	FFFF	EXT	1	

NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer
NOMIAS

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 11/16/1983 10:27 AM PG125
FOPEN+	FFFF	EXT	1	
FOR	FFFF	EXT	1	
FOR+	63F0	LCL	1	
FDEWY1	7DEB	LCL	2	
FORTOK	008C	EQU	1	
FPS	FFFF	EXT	2	
FPUAGE	FFFF	EXT	1	
FSEEK	FFFF	EXT	3	
FTRITE	7E9C	LCL	1	
FULIAR	845F	G DAD	4	
GET18	FFFF	EXT	1	
GETADW	FFFF	EXT	26	
GETADR	FFFF	EXT	20	
GETOSE	71F7	LCL	2	
GETIAR	FFFF	EXT	1	
GETISI	FFFF	EXT	3	
GETIMB	642C	LCL	1	
GETPRB	FFFF	EXT	1	
GETPRW	FFFF	EXT	1	
GETPU+	740B	LCL	2	
GETREC	793F	LCL	6	
GETISTP	FFFF	EXT	1	
GETIEM	FFFF	EXT	1	
GOVKEY	7548	LCL	1	
GIMFLE	693F	LCL	1	
GTSIP+	647E	LCL	3	
HARH8	6C22	LCL	1	
HARH7	FFFF	EXT	1	
HARH0	8375	G DAD	1	
HARHLE	751E	LCL	2	
HARH7	6C66	LCL	1	
HARH7	6C70	LCL	1	
HEN8	6F9D	LCL	4	
HENBIM	6F65	LCL	1	
HDR8	6C2F	LCL	1	
HROT8	6C50	LCL	1	
HSHV8	6C56	LCL	1	
HID	78C4	LCL	1	
HIDR8	6C29	LCL	1	
INTID	FFFF	EXT	2	
INTOUT	6841	LCL	1	
INTILEM	62D9	LCL	1	
IMP+2	8182	G DAD	3	
IMP+94	818E	G DAD	3	
IMPBUF	8180	G DAD	10	
IMPTR	8360	G DAD	1	
INSTL	7A08	LCL	1	
INT8	6964	LCL	2	
INTDTV	FFFF	EXT	2	
IMVFLG	66F4	LCL	5	
IMVFLW	75B9	LCL	2	
ID?	FFFF	EXT	2	
IOBYTE	825A	G DAD	2	
IPS	FFFF	EXT	2	

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 11/16/1983 10:27 AM PG127
MOVEUP	FFFF	EXT	1	
MPYROI	FFFF	EXT	4	
MIFILE	78D9	LCL	1	
MVPC41	6E5D	LCL	1	
MVPC42	6E6F	LCL	1	
MVSIR	6E78	LCL	4	
MYRONW	4064	EQU	3	
MYRIM	647D	LCL	4	
NAREW	0003	G EQU	1	
NARSZE	695F	LCL	1	
NEUSZE	65E8	LCL	1	
NEXT	FFFF	EXT	1	
NEXT+	63F9	LCL	1	
NEXT-	63FF	LCL	1	
NHECUI	6920	LCL	1	
NOCLL?	FFFF	EXT	2	
NOFILE	6900	LCL	1	
NOM/OF	6409	LCL	1	
NOPKEY	00FF	G EQU	1	
NOIM2	00FB	G EQU	1	
NOIM4	00EF	G EQU	1	
NOIM5	000F	G EQU	1	
NSLR8	6EB7	LCL	1	
NURDUI	6845	LCL	11	
NKICHR	7191	LCL	1	
NKIH1R	77E9	LCL	1	
NK11OK	608F	G EQU	1	
OFRO	746B	LCL	1	
ON/OFF	FFFF	EXT	1	
ONEB	FFFF	EXT	23	
ONEB+	784E	LCL	2	
ONLI	FFFF	EXT	3	
ONER	FFFF	EXT	2	
ONEROI	FFFF	EXT	2	
ONERRO	FFFF	EXT	1	
ONR12?	FFFF	EXT	4	
ONSRO	6484	LCL	1	
ONSPQ	742D	LCL	1	
OIO	7A8B	LCL	1	
OUFE3C	FFFF	EXT	2	
OUT	73A8	LCL	1	
PLBLEH	000A	G DAD	3	
PCR	824D	G DAD	2	
PLSIRI	828D	G DAD	1	
POPHUX	6C47	LCL	1	
POPRIM	720E	LCL	1	
POSOU1	705C	LCL	1	
PREIND	FFFF	EXT	1	
PRM18	698D	LCL	1	
PRM1LW	82CC	G DAD	1	
PRSHVC	7648	LCL	1	
PRSLEX	60CA	LCL	1	
PR11S1	8488	G DAD	1	
PRIPTR	82D7	G DAD	1	

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 11/16/1983 10:27 AM PG132
1F0063	676A	LCL	1	
1F0064	676A	LCL	1	
1F0065	6771	LCL	1	
1F0066	677E	LCL	1	
1F0067	677A	LCL	1	
1F0069	678B	LCL	1	
1F0070	678A	LCL	1	
1F0071	678B	LCL	1	
1F0072	6798	LCL	1	
1F0073	679F	LCL	1	
1F0074	67A6	LCL	1	
1F0075	67AE	LCL	1	
1F0076	67BA	LCL	1	
1F0077	67B7	LCL	1	
1F0078	67C1	LCL	1	
1F0079	67C8	LCL	1	
1F0080	67E6	LCL	1	
1F0081	67D2	LCL	1	
1F0082	67E3	LCL	1	
1F0083	67D4	LCL	1	
1F0084	67E3	LCL	1	
1F0085	67E2	LCL	1	
1F0086	67E3	LCL	1	
1F0087	67F4	LCL	1	
1F0088	67F4	LCL	1	
1F0089	6802	LCL	1	
1F0090	6802	LCL	1	
1F0091	6815	LCL	1	
1F0092	6848	LCL	1	
1F0093	6851	LCL	1	
1F0094	685A	LCL	1	
1F0095	6863	LCL	1	
1F0096	686C	LCL	1	
1F0097	6875	LCL	1	
1F0098	687E	LCL	1	
1F0099	6887	LCL	1	
1F0100	6890	LCL	1	
1F0101	6899	LCL	1	
1F0102	68A4	LCL	1	
1F0103	68AE	LCL	1	
1F0104	68AA	LCL	1	
1F0105	68AE	LCL	1	
1F0106	68AE	LCL	1	
1F0107	68B7	LCL	1	
1F0108	68C0	LCL	1	
1F0110	68EE	LCL	1	
1F0111	691C	LCL	1	
1F0112	690B	LCL	1	
1F0113	691C	LCL	1	
1F0114	692A	LCL	1	
1F0115	6930	LCL	1	
1F0116	6939	LCL	1	
1F0119	695A	LCL	1	
1F0120	6963	LCL	1	

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 11/16/1983 10:27 AM PG134
1F0211	6FD6	LCL	1	
1F0213	6FF6	LCL	1	
1F0214	6FF9	LCL	1	
1F0215	7022	LCL	1	
1F0216	7043	LCL	1	
1F0217	7057	LCL	1	
1F0218	7059	LCL	1	
1F0223	7082	LCL	1	
1F0224	70C0	LCL	1	
1F0225	70DC	LCL	1	
1F0226	70E9	LCL	1	
1F0227	70F7	LCL	1	
1F0228	70FB	LCL	1	
1F0231	7110	LCL	1	
1F0232	7111	LCL	1	
1F0235	7143	LCL	1	
1F0236	7148	LCL	1	
1F0238	7183	LCL	1	
1F0239	7188	LCL	1	
1F0240	7188	LCL	1	
1F0242	71A5	LCL	1	
1F0244	71D5	LCL	1	
1F0245	71D0	LCL	1	
1F0246	71F1	LCL	1	
1F0248	721E	LCL	1	
1F0249	7220	LCL	1	
1F0250	7238	LCL	1	
1F0251	7245	LCL	1	
1F0254	7275	LCL	1	
1F0255	7276	LCL	1	
1F0256	7292	LCL	1	
1F0258	729F	LCL	1	
1F0259	729F	LCL	1	
1F0260	72A5	LCL	1	
1F0262	72D7	LCL	1	
1F0264	72EE	LCL	1	
1F0265	72EE	LCL	1	
1F0266	72EA	LCL	1	
1F0267	72ED	LCL	1	
1F0268	7306	LCL	1	
1F0270	7312	LCL	1	
1F0272	7384	LCL	1	
1F0274	7376	LCL	1	
1F0275	7376	LCL	1	
1F0276	7383	LCL	1	
1F0277	73A8	LCL	1	
1F0279	73A6	LCL	1	
1F0280	73C2	LCL	1	
1F0281	73D1	LCL	1	
1F0283	73FD	LCL	1	
1F0284	7426	LCL	1	
1F0285	7410	LCL	1	
1F0286	7428	LCL	1	
1F0290	7494	LCL	1	

NO. 11/16/1983 Supported
Program Agreed to be Contact Manufacture

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 11/16/1983 10:27 AM PG133
1F0121	6979	LCL	1	
1F0124	69D1	LCL	1	
1F0125	69DE	LCL	1	
1F0126	69E8	LCL	1	
1F0128	6A0F	LCL	1	
1F0129	6A1F	LCL	1	
1F0130	6A2C	LCL	1	
1F0135	6A98	LCL	1	
1F0136	6A98	LCL	1	
1F0137	6A98	LCL	1	
1F0140	6AE2	LCL	1	
1F0142	6A0F	LCL	1	
1F0143	6AE2	LCL	1	
1F0144	6AF1	LCL	1	
1F0146	6B21	LCL	1	
1F0147	6B45	LCL	1	
1F0148	6B92	LCL	1	
1F0150	6BBF	LCL	1	
1F0152	6BC0	LCL	1	
1F0153	6BEA	LCL	1	
1F0154	6BEC	LCL	1	
1F0155	6BFC	LCL	1	
1F0156	6C0C	LCL	1	
1F0158	6C7F	LCL	1	
1F0159	6C7E	LCL	1	
1F0160	6C9B	LCL	1	
1F0162	6CC5	LCL	1	
1F0164	6C7E	LCL	1	
1F0165	6C7E	LCL	1	
1F0166	6C7E	LCL	1	
1F0167	6C75	LCL	1	
1F0169	6B0D	LCL	1	
1F0170	6B17	LCL	1	
1F0171	6B39	LCL	1	
1F0173	6D29	LCL	1	
1F0175	6D5A	LCL	1	
1F0179	6D4E	LCL	1	
1F0181	6D63	LCL	1	
1F0182	6D9C	LCL	1	
1F0183	6D46	LCL	1	
1F0185	6D8A	LCL	1	
1F0186	6D8D	LCL	1	
1F0188	6D4C	LCL	1	
1F0189	6D4C	LCL	1	
1F0190	6D4E	LCL	1	
1F0191	6D4E	LCL	1	
1F0192	6D4E	LCL	1	
1F0193	6D4E	LCL	1	
1F0196	6E64	LCL	1	
1F0201	6E4C	LCL	1	
1F0202	6EE8	LCL	1	
1F0203	6EE8	LCL	1	
1F0204	6EE8	LCL	1	
1F0205	6F10	LCL	1	

KARMA 08/31/82-----				
SYMBOL	VALUE	TYPE	COUNT	Symbol Table 11/16/1983 10:27 AM PG135
1F0292	748D	LCL	1	
1F0294	7532	LCL	1	
1F0295	7536	LCL	1	
1F0296	7591	LCL	1	
1F0297	756D	LCL	1	
1F0298	756D	LCL	1	
1F0299	757A	LCL	1	
1F0300	7585	LCL	1	
1F0301	7595	LCL	1	
1F0302	75D0	LCL	1	
1F0304	7648	LCL	1	
1F0306	762C	LCL	1	
1F0307	763D	LCL	1	
1F0308	763A	LCL	1	
1F0309	767A	LCL	1	
1F0311	7654	LCL	1	
1F0312	7668	LCL	1	
1F0314	76AC	LCL	1	
1F0316	76C3	LCL	1	
1F0318	76E6	LCL	1	
1F0319	76E6	LCL	1	
1F0320	76CF	LCL	1	
1F0321	76E2	LCL	1	
1F0323	76E6	LCL	1	
1F0325	772C	LCL	1	
1F0326	772C	LCL	1	
1F0327	775A	LCL	1	
1F0329	77E9	LCL	1	
1F0330	7798	LCL	1	
1F0331	77E9	LCL	1	
1F0332	77E9	LCL	1	
1F0333	77AD	LCL	1	
1F0334	7710	LCL	1	
1F0335	7710	LCL	1	
1F0336	77C9	LCL	1	
1F0337	77C9	LCL	1	
1F0342	7841	LCL	1	
1F0345	7898	LCL	1	
1F0347	7890	LCL	1	
1F0348	7890	LCL	1	
1F0359	79D8	LCL	1	
1F0360	79D3	LCL	1	
1F0361	79D8	LCL	1	
1F0366	7A16	LCL	1	
1F0371	7AD4	LCL	1	
1F0378	7B58	LCL	1	
1F0379	7B58	LCL	1	
1F0380	7B68	LCL	1	
1F0381	7B7B	LCL	1	
1F0382	7B7E	LCL	1	
1F0383	7B91	LCL	1	
1F0384	7B9D	LCL	1	
1F0385	7BA7	LCL	1	
1F0388	7BC8	LCL	1	

KARMA 08/31/82=====					Symbol Table 11/16/1983 10:27 AM PG136				
SYMBOL	VALUE	TYPE	COUNT						
1f0389	78D9	LCL	1						
1f0390	78E1	LCL	1						
1f0391	78E4	LCL	1						
1f0392	78F3	LCL	1						
1f0393	78F6	LCL	1						
1f0394	7C03	LCL	1						
1f0395	7C0A	LCL	1						
1f0396	7C09	LCL	1						
1f0397	7C34	LCL	1						
1f0398	7C55	LCL	1						
1f0399	7C47	LCL	1						
1f0400	7C53	LCL	1						
1f0401	7C5F	LCL	1						
1f0403	7C81	LCL	1						
1f0404	7C81	LCL	1						
1f0405	7CDE	LCL	1						
1f0411	7D34	LCL	1						
1f0412	7D44	LCL	1						
1f0413	7D78	LCL	1						
1f0414	7D88	LCL	1						
1f0415	7D95	LCL	1						
1f0416	7D96	LCL	1						
1f0417	7D96	LCL	1						
1f0421	7DE3	LCL	1						
1f0422	7DE6	LCL	1						
1f0423	7DF7	LCL	1						
1f0424	7E1E	LCL	1						
1f0425	7E35	LCL	1						
1f0426	7E30	LCL	1						
1f0427	7E61	LCL	1						
1f0428	7E61	LCL	1						
1f0430	7E4E	LCL	1						
1f0440	7EE7	LCL	1						
1f0442	7EE7	LCL	1						
1f0443	7EE7	LCL	1						
1f0444	7F0D	LCL	1						
1f0445	7F0F	LCL	1						
1f0447	7F50	LCL	1						
1f0448	7F62	LCL	1						
1f0449	7F6F	LCL	1						
1f0452	7F8F	LCL	1						
1f0453	7FC4	LCL	1						
1f0455	7FD8	LCL	1						
1f0456	7FE1	LCL	1						
1f0457	7FE9	LCL	1						
1f0458	7FF0	LCL	1						
1g0037	650E	LCL	1						
1g0040	6610	LCL	1						
1g0041	6610	LCL	1						
1g0044	6637	LCL	1						
1g0053	6689	LCL	1						
1g0056	66E7	LCL	1						
1g0057	66FD	LCL	1						
1g0068	6781	LCL	1						

KARMA 08/31/82=====					Symbol Table 11/16/1983 10:27 AM PG137				
SYMBOL	VALUE	TYPE	COUNT						
1g0123	69B0	LCL	1						
1g0132	6A30	LCL	1						
1g0133	6A51	LCL	1						
1g0134	6A68	LCL	1						
1g0139	6ABF	LCL	1						
1g0141	6AC8	LCL	1						
1g0145	6B17	LCL	1						
1g0168	6CFS	LCL	1						
1g0172	6C19	LCL	1						
1g0174	6C2C	LCL	1						
1g0176	6C3C	LCL	1						
1g0180	6C4F	LCL	1						
1g0187	6C8F	LCL	1						
1g0194	6E92	LCL	1						
1g0197	6E78	LCL	1						
1g0199	6E9A	LCL	1						
1g0200	6E6A	LCL	1						
1g0207	6F2D	LCL	1						
1g0208	6F5A	LCL	1						
1g0210	6F86	LCL	1						
1g0212	6FE6	LCL	1						
1g0219	7067	LCL	1						
1g0222	7098	LCL	1						
1g0230	7106	LCL	1						
1g0234	7138	LCL	1						
1g0241	7188	LCL	1						
1g0252	7261	LCL	1						
1g0253	7266	LCL	1						
1g0257	7287	LCL	1						
1g0261	7289	LCL	1						
1g0263	72C3	LCL	1						
1g0269	72F0	LCL	1						
1g0273	7359	LCL	1						
1g0278	7393	LCL	1						
1g0282	73D2	LCL	1						
1g0291	74AB	LCL	1						
1g0305	7610	LCL	1						
1g0310	7648	LCL	1						
1g0313	767F	LCL	1						
1g0324	771F	LCL	1						
1g0328	777C	LCL	1						
1g0338	77D8	LCL	1						
1g0339	77F0	LCL	1						
1g0343	7854	LCL	1						
1g0346	7867	LCL	1						
1g0351	78C8	LCL	1						
1g0353	7904	LCL	1						
1g0355	7971	LCL	1						
1g0356	799E	LCL	1						
1g0358	79C2	LCL	1						
1g0362	79E0	LCL	1						
1g0363	79E8	LCL	1						
1g0364	79F0	LCL	1						
1g0365	79F8	LCL	1						

KARMA 08/31/82=====					Symbol Table 11/16/1983 10:27 AM PG138				
SYMBOL	VALUE	TYPE	COUNT						
1g0368	7A2F	LCL	1						
1g0369	7A62	LCL	1						
1g0372	7ADE	LCL	1						
1g0373	7AE5	LCL	1						
1g0375	7B29	LCL	1						
1g0376	7B2C	LCL	1						
1g0377	7B34	LCL	1						
1g0387	7BB2	LCL	1						
1g0407	7D05	LCL	1						
1g0408	7D11	LCL	1						
1g0409	7D10	LCL	1						
1g0419	7DC8	LCL	1						
1g0420	7DD5	LCL	1						
1g0429	7E41	LCL	1						
1g0431	7E61	LCL	1						
1g0432	7E67	LCL	1						
1g0433	7E76	LCL	1						
1g0435	7E88	LCL	1						
1g0436	7E9C	LCL	1						
1g0437	7EAC	LCL	1						
1g0438	7EB4	LCL	1						
1g0439	7EC3	LCL	1						
1g0441	7ED3	LCL	1						
1g0451	7F9D	LCL	1						
1g0454	7FCF	LCL	1						
rtg005	6341	LCL	4						
rtg011	634D	LCL	1						
rtg017	6393	LCL	2						
rtg025	6498	LCL	1						
rtg027	64F5	LCL	1						
rtg028	64D7	LCL	4						
rtg033	654B	LCL	5						
rtg035	658F	LCL	1						
rtg055	66C1	LCL	2						
rtg109	68C5	LCL	1						
rtg117	6930	LCL	1						
rtg118	6944	LCL	2						
rtg122	6983	LCL	1						
rtg127	69E0	LCL	1						
rtg131	6A28	LCL	1						
rtg138	6A89	LCL	2						
rtg149	6B4B	LCL	2						
rtg157	6C1F	LCL	1						
rtg161	6C52	LCL	1						
rtg163	6C9A	LCL	1						
rtg184	6D72	LCL	2						
rtg195	6E18	LCL	1						
rtg198	6E77	LCL	5						
rtg206	6F7F	LCL	5						
rtg209	6F9A	LCL	4						
rtg220	7065	LCL	1						
rtg221	7073	LCL	1						
rtg229	70EB	LCL	1						
rtg233	7115	LCL	1						

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

KARMA 08/31/82=====					Symbol Table 11/16/1983 10:27 AM PG139				
SYMBOL	VALUE	TYPE	COUNT						
rtg237	7128	LCL	1						
rtg243	7159	LCL	1						
rtg247	71F6	LCL	2						
rtg271	7314	LCL	1						
rtg287	742B	LCL	1						
rtg288	7450	LCL	2						
rtg289	7469	LCL	1						
rtg303	7583	LCL	2						
rtg352	7910	LCL	3						
rtg354	7960	LCL	2						
rtg357	7995	LCL	1						
rtg367	7A06	LCL	2						
rtg370	7AA9	LCL	3						
rtg374	7B03	LCL	1						
rtg386	7B81	LCL	1						
rtg402	7C68	LCL	1						
rtg406	7C87	LCL	1						
rtg410	7D29	LCL	1						
rtg418	7DEB	LCL	3						
rtg434	7EB1	LCL	4						
rtg446	7F21	LCL	1						
rtg450	7F75	LCL	3						

HEADING	Table of Contents	PAGE	PG140
exits, ents, equs, etc	1		
ROM tables	5		
RAMBI call processing	11		
Parse routines	14		
create aprt/ldg for mmh	16		
recover card stuff	17		
set status of system switches	22		
status of system switches	27		
file # = name? line# of #	29		
reassign device name	31		
listio# - return string for assigned	32		
get endline string	34		
get current key definition	35		
get current delay setting	36		
flag stuff	37		
get current margin/width/pwidth	39		
Keywords returning window parameters	40		
keyword WINDOW	41		
user message	43		
hex and/or xor two strings	44		
hex/ascii rotate/shift string	45		
get some constants	48		
ascii and/or xor two strings	49		
Trim trailing blanks	51		
rotate string	52		
escape string	53		
lowercase a string	54		
repeat a string n times	55		
left trim string	56		
reverse string	57		
hex/bin routines for address and data	58		
convert ascii string to hex or other way	59		
span/break string	60		
map string to string	62		
get ascii char string	63		
TAB key processing	64		
LCD ON/OFF disable led until restored	65		
convert string to decimal (string base given)	66		
template line input (protected fields)	68		
convert decimal number to string in base given	71		
some helpful subroutines	74		
set up sq statement	75		
CR18 of cassette	76		
cause an error number	78		
signal and wait key	79		
LASTLN?	80		
Clone tape on multi-tapes	81		
write to all targets	90		
get record from master	91		
listen all target devices	92		
reset all target devices	93		
wait for all target devices	94		
compute password from filename	95		
Load a private file from tape	97		

HEADING	Table of Contents	PAGE	PG141
bit/base conversions	99		
BUGS runtime stuff	102		
LENGTH? return char count of disp/print	103		
FIND?	104		
TIMER?	106		
REPLC?	107		
RIGHT?	109		
LEFT?	110		
MID?	111		
SUB?	112		
dummy for next routine	113		
FILL? full string (left/right/middle)	114		
ADJUST stuff	117		
EXIT for statement	119		
CRIM, a function	120		

RYE102 HAD 0 ERRORS 0 WARNINGST060 LABELS SIZE 8190 LAST ERROR AT 0

KLINK load map size size RYWI03 12:01 PM TUE., 13 DEC., 1983
file start and hex/oct decimal version message page 1
RYLI03 6000 7FFA 1FFA 8186
RYLI11 7FFA 7FFA 0000 0

Total size=1FFA = 8186 decimal

NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer
NOMAS

KLINK load map	RYWI03 12:02 PM TUE., 13 DEC., 1983 page 2
A.ZERO	ERE7
ABORT	F8C0
ABSS	3A43
ABSS-	3A46
ABTHRD	F8EA
ACRND	35F2
ACREAI	7210
ADD15	380A
ADD20	380F
ADD9	3804
ADDR	F042
ADDR01	37D4
ADJUST	233F
ALCHK	67B0
ALARM	7430
ALBEEP	7233
ALCALL	1F93
ALFA	0E02
ALFAH	0E03
ALJMI	2E39
ALLOC	1F9F 6B6B 6B01 6C3B 6CBA 6D08 6DA4 778A
ALLOC	1F9F 7EAD 7F23 7F87
ALCHK	6CD7
ALOSNV	27F9 7BAC
ALVENI	2BE8 7157 7C5E
ALWAYS	E9D1
AND	4E6A
AND.A	440A
AND.A-	4415
AND.B	43E4
AND.B-	43E9
AND.E	43EE
AND.E-	43F3
AND.P	4400
AND.P-	4405
ANDUNS	43E0
ANDIE	36D7
ANDYER?	4B3A
ADPEN	7220
ADPEN?	722A
APDEL?	6A01
APKIL	668A
APFLO	09EF
APFILE	09F7
APFHL?	644E
APIWI 0	6361
APMSAE	6532
APMSKY	653C
APPROC	72BC
APPI	4993
APPIAD	65E4
APPIKS	7229
APPIIN	6552
APROPM	64C6
APSIAT	7300

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

KLHK Load Map	RYM103	12:02 PM	TUE., 13	DEC., 1983	page 10
GE1M1	5094				
GE1M2	509A				
GE1IN	416D				
GE114	128E				
GE1IN	1261				
GE12N	1258				
GE1AUM	1F12	6FB1	738B	7394	74A7 7922
GE1A14	1F14				
GE1A1R	1F11	6D8D	6D89	720B	
GE1ALM	00E6				
GE1CHR	0750				
GE1CLK	745B				
GE1CN2	106A	63B9			
GE1CMA	1066				
GE1F1	EDCC				
GE1F51	4AFB				
GE1JMT	2E6F	7102	710A	711B	7D8D
GE1L1M	15D0				
GE1L1M	0619				
GE1L1M	06E9				
GE1LNX	64C7				
GE1HAM	4A77	795E	79DE		
GE1HM	3133	6230			
GE1NXT	2E85				
GE1PA2	125D				
GE1PAD	5070				
GE1PAR	1264				
GE1REP	4158				
GE1R1D	24F8				
GE1RM2	475E				
GE1ROM	4761				
GE1RUM	18AA				
GE1S1P	129E	655F	6569		
GE1TD	746C				
GE1TEH	4110				
GE1TIER	412E				
GF2LNM	188B				
GOCARD	F868				
GOPEN	24CC				
GOPEN1	2497				
GORTIN	568E				
GOTOFR	17BB				
GOTOSU	17BE				
GR1	4DE6				
GR1	4E25				
GRANGE	18D8				
GTXKT1	2E85				
GXX3	12E9				
HALT2	18E9				
HAND1	5419	6997	7A1E		
HAND10	544C				
HAND1A	581C				
HANG	016A				
H1FL1M	07F1				
H1FLUT	0746				

KLHK Load Map	RYM103	12:02 PM	TUE., 13	DEC., 1983	page 10
GE1M1	5094				
GE1M2	509A				
GE1IN	416D				
GE114	128E				
GE11N	1261				
GE12N	1258				
GE1A0M	1F12	6FB1	738B	7394	74A7 7922
GE1A1M	1F14				
GE1A1M	1F11	6D8D	6D89	720B	
GE1A1M	00E6				
GE1CHR	0750				
GE1CLK	745B				
GE1CN	106A	53B9			
GE1CMA	1066				
GE1F1	EDCC				
GE1F51	4AFB				
GE1JMT	2E6F	7102	710A	711B	7D8D
GE1L1M	15D0				
GE1L1M	0619				
GE1L1M	06E9				
GE1LNX	64C7				
GE1HAM	4A77	795E	79DE		
GE1HM	3133	6230			
GE1NXT	2E85				
GE1PA	125D				
GE1PAD	5070				
GE1PAR	1264				
GE1REP	4158				
GE1R1D	24F8				
GE1RM2	475E				
GE1ROM	4761				
GE1RUM	18AA				
GE1S1P	129E	655F	6569		
GE1TD	746C				
GE1TE1M	4110				
GE1TIER	412E				
GF2LNM	188B				
GOCARD	F868				
GOPEN	24CC				
GOPEN1	2497				
GORTIN	568E				
GOTOFR	17BB				
GOTOSU	17BE				
GR1	4DE6				
GR1	4E25				
GRANGE	18D8				
GTXKT1	2E85				
GXXJ	12E9				
HALT1	18E9				
HAND1	5419	6997	7A1E		
HAND10	544C				
HAND1A	581C				
HANG	016A				
H1FL1M	07F1				
H1FLUT	07H6				

```

IMPUSH 570A
I/ROFF 7CE6
ICOMST 54CE
ICOS. 7535
IDY:ND E00D
IF 16D5
ILE1 1644
IENHV? 53EE
INR10 39CB
INR2 3A7C
INR3 3A4E
INR4 3A51
INR9 3A90
INIPR- 0806
INIPRS 0802
INISIZ 6750
INIT. 666B
INITGL 00F5
IMPUS. 4373
IMPUN. 4309
INPUT. 1557
INPUT. 4276
INSERT 1FFC
INSRIN 23CE
INTS 3A68
INTCOM 57D7
INTDIV 3A6F
INTEGR 0D16
INTMUL 394E 712E 71D3 7672 76CD
INTORL 3F16
INTRP 008E
INWPOP E023
INVRTN E004
INWIND EFB
IO. OFF F015
IO. ON F00C
IO? E99F 693E 6A6C 7E97 7F6A 7FD2
IOFILE 09E7
IOPLEN 218E
IPS 3AEB
ISCAN 1075
ISCTUP 4305
ISM. 752D
ITAIL. 43DF
ITAM. 753D
ITHEW 16DF
IZOFEN 2139
JAPLHN ECS4
JAPSUB EC44
JSBER1 702D
JTRUW EC35
KEYS. FL60
KEY? 0854
KEYLSP 419A
KEYNAM 4EAD

```

```

MINHM 6FRE
MINHM 6F88
MINHM 6FB0
MINVY 6F86
MKEPRV F6F0
MOV10. 7705
MODE? 648B
MODEKY E472
MOVE 2270
MOVEIT 5E86
MOVENV 2E16
MOVEUP 2218
MPY28 38E2
MPY30 38E5
MPYR? 7F9A
MPYRDI 393D
MAGIN. FC01
NSGOUT 07E3
MULT60 7473
MVBVIS 3F43
MSURD 30A0 622D
MYCE1N 7418
MYSET2 74E8
MYSIPR 7412
NARE. 5253
NEWLIN EC55
NEXT 1785
NEXT. E0D7
NFR 38AA
NMLE? 17FC 64DA 66A3
NOT. 4E61
NOTASH 71ED
NULP1 0A03
NULOLO 4108
NUN. FLD5
NUMBER 0D40
NUNCHK 0A69
NUNOUT F985
NUNREP 1086
NUNVR+ OE6D 6281 630D 63DE
NUNVAL OE70 62E2 631E
NUNPCK 6F85
NXTAP1 747D
NXTWUF 4783
NXTLIN 2E82
NXTROM 47AL
OFALRM 7444
OFF10+ 5E11
OFF10. 5E06
OFFTHR 3F50
OFFUHL 5E09
OFFOK. 5FFC
ON 1796
ON. ECD0
ON/OFF 5FD7 63C5

```

NOMAS
 Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KEYS. 499F
KEYSRV 0277
KEYTRN 4195
KOPY 0A5F
KYADDR E452
KYOPEN 4ER7
LA 7DDF
LADDEV 45FE
LAURFP EL18
LBI4 726D
LCDDU1 598B
LCDDRY F027
LCDDSI F037
LEMPYR 6F94
LEN. FCB8
LEMCAL E939
LEOS. 4DCA
LEQ. 4E0B
LE1 1649
LE160 079A
LIFT. 524D
LINEDR 44EF
LINEHD FES6
LINEHD 0C15
LINLEN 21E5
LIST. 1B46
LIT? 0683
IN20 362A
IN30 3668
IN3R 3654
IMS 173C
IN72 3676
LDPA F27C
LDL. 7986
LDL? 79CB
LDGCOM 399A
LDG15 379A
LDGUP 0A26
LPWF 5E29
LPW 5E28
LS110. 568B
LT. 4046
LT. 4E3C
MALLIT 0686
MALLDE 7406
MARCN. 7F67
MARVO 3F54
MARZAI 6A8B
MBLIN 7446
MBLINP 744D 79E9 79F0 79F6
MELTD 4661
MEW. FLC6
MERGE. 5319
MINVO 3D25
MINVQ 6FAD

```

```

ON?I E2B3
ON?R E2B7
ONE74B 7DEE
ONE7PR E375
ONEB 3E8B 6A90 6EF9 6F65 6FB6 76AB
ONEI 3EAC 6861 68F3 6A19 6F68 7E68
OVER 3EC0
OVER- 3E0D
OVERI 3EE1
ONEROI 3EDE
ONERRO 134D
ONOFF? 5FDD
ONR12? 1F3A 6873 6A88 7E47
ONLAR 3F57
ONLON. 5FF6
OPTION 131F
OR. 4E4D
OTHER 014D
OUTICH 081D
OUTIC40 0814
OUTICR 080C
OUTEOL 082C
OUTERB 494A
OUTESC 0819
OUTLJM 022B
OUTNUM 5F6C
OUTSTR 07ER
PWARRY E87E
PIAK? 122C
PAK. 64E6
PAR1 09D7
PARSE? 0BAA
PARSER 0B0E
PARSI1 0BA7
PIA. 7465
PFNDPR 2164
PGHANN 43FB
PFO 39F6
PILOF? E603
PILOH? ERE8
PIUREP 45E5
PLIS1 1906
PLIS1. 1B41
POP. EDIE
POPEW 5040 797E
POS. F04F
POGRUN FDS7 6E14 6E66
POOLO 3D16
PRMWH 737
PRMVAL E781
PR. EOL 4596
PRETHD 216A
PRIW. E54C
PRIWS. 5887
PPJNT 188E

```


STRICH	FC70								
STR7A7	5899								
STRCH	1193	66AC							
STRCH	1299								
STRCH	10FA								
STRCHP	10FA								
STRIND	1E78								
STRKET	1156								
STRME	7901								
STRK -	129C								
SUBRO1	37CA								
SUBSCR	339A								
SUBS11	57E2								
SUBG12	5821								
SUBSTF	EC64								
SUM11	FA16								
SUM11	FA19								
SVADCK	580A								
SVADR	54AA								
SVADR	54A9								
SVCCMK	462B								
SVCCLR	063F								
SVCSST	063A								
SVVAL3	54AA								
SVCLAL	7A2B								
SVCLM1	7A2B								
SVSJSB	466B	62BF	62C0	630B	631C	63B7	63B9	63C3	63C4
SVSJSB	466B	630C	655D	6567	667D	66AA	66EE	6700	6700
SVSJSB	466B	6718	7981	7996	7998	7A9A	7BCE		
TAB	58DB								
TAB	FD91								
TADREP	EC0F								
TAMIO	39ED								
TBL3	3E53								
TBL3B	3EAB								
TCK1G1	7500								
TENR11	7047								
TEX1	5E3E								
THERE1	5E7D								
THERE1	734C								
THERE2	5E74								
TICK	6844								
TICKHK	71A7								
TIDATV	6AEE								
TIME	7C6D								
TIME	7CFA								
TIMEHD	673A								
TIMEASK	65AA								
TIMECK	67F7								
TIMECT	3FDC								
TIMEHD	757A								
TIMEFAC	3F3B								
IMESET	76B2								
TIMELR	409E								
TIMEHF	402D								

TRML0	3FF2	
TRML1	4071	
TRML2	4049	
TRML3	3178	
TRML4	4027	
TRML5	3FFE	
TRML6	408E	
TRML7	408A	
TD	EE34	
TD1	2B94	
TD3C2	0281	
TD3C2	0A9E	
TD3C8	0A8B	
TD3C7	16B0	
TD3H2	0A11	
TD3H8	0A0B	
TD3C6	702E	
TD3PROM	1606	
TRAPTX	1A61	
TRC01	3C91	
TRC7	3C95	
TRCFF	98E5	
TRCHE0	9957	
TRC1L	8800	
TRCVBL	489C	
TRFL0	4887	
TRFLH	7312	
TRJUMP	4926	
TRKEY	0286	
TRMSCL	7056	
TRR0F	488F	
TRRVR	4890	
TRVIN	12FF	
TS1LH0	2297	
TS1LH1	2825	709E
TS1H1	2519	
TS1H11	6907	
TD0B	3E8E	76A7
TD0C	310C	
TD0F01	3E19	
TDYEM1	6723	
TDYEM2	2478	
TDYEM3	1317	
UL1H10	0113	
UM1H10	0113	
UMH05	410E	
UMH01	410A	
UMHREF	18F6	
UMHREF	18F6	
UMH1M0	18F5	
UMH1M1	121C	
UMH1M2	471C	
UMH1E	76C1	
UMH1M1	3204	
UMH1E1	1C0B	

UNIT SKD	EB12	
UNITUAL	7026	
UPC#	7DEC	
UPDISP	67FF	
UPEN1	5913	
UPER#	1835	70C6
UPRTSE	7989	
UPSET	005C	
UPROM1	1F85	
USING	CF39	
UTZ10	300B	
VAL#	7C85	
VAL	7C85	
VECALL	5101	
VECRST	41FC	
VER	7BFA	
VERB	6B47	
VERIFY	F829	
VERNM0	069C	
VERNM1	358A	
VERNM2	5A17	
VERNM3	7201	
VERNM4	F5C5	
VERNM5	6F44	
VERSUB	FC20	
VF1102	6FD8	
VFDCL2	6CE8	
VFDODR	6B80	
VFD5Y	8006	
VFD5Y#	67C1	
VFD5E	60A1	79F8
VFD046	6D8C	
VFD0C0	6D97	
VFD0EP	6D86	
VFLCH	6CED	
VFD0L2	6CBE	
VFDCL2	6D0F	
VFD1R	6D8C	
VFD1R#	67AD	
VFD0DE	6EEE	
VFD0D7	60F7	
VFERR	6D48	
VFERM#	6E7A	
VFGET	6A44	
VFGLOC	6E19	
VFIH	6A0F	79E9
VFIJ1	67AB	
VFLAD	7C18	
VFLAD#	67AB	
VFL0D7	6E09	
VFL1F7	6EB3	
VFL1B9	6CFF	
VFLTY#	67CB	
VFNFP7	6E25	
VFNFP7	6A90	

VFMDFE	6E38	
VFMDSG	6ED9	
VFMHSD	6E08	
VFMHRE	6E0C	
VFMKAE	6E7E	
VFMPEL	6D0F	
VFMREX	6D76	
VFMREU	6F68	
VFMREH	6E2B	
VFMRLT	6EBA	
VFMROD	6E90	
VFMRECE	6C80	
VFMRVNE	6EC3	
VFMROW	67D0	
VFMRWK	679E	
VFMRRP	6BF3	
VFMRSO	698C	
VFMRSB	68E6	
VFMRSK	688E	
VFMRSW	6D7F	
VFMRRW	6E3E	
VFMSECT	6FE3	
VFMSTFL	6EE3	
VFMSTAT	6D0F	
VFMTRD	EE0F	
VFMTRP	67D5	
VFMTRM	6D9E	
VFMTRN	6F4E	
VFMTRN	6F0F	
VFMU1L	67D8	
VFMU2C	6E3E	
VFMU2K	6C27	
VFMU3D	6E84	
VFMUOP	6C8A	
VFMUR	6C78	
VFMURB	6CC3	
VFMURL	6CF8	
VFMURJ	6D0F	
VFMURJ	6DCE	
VFMURDE	6DCE	
VFMURCE	6CDB	
VFSPECH	27B0	70B8
VSTOR	27E2	7BE0
VHJ1L	E36E	
VHJ1LY	7E64	
VHJ1LY	0755	
VHJCEUP	7666	
VHJRN	4B03	
VHJRW, R	4F34	
VHJJSB	E36F	
VHJ1CR	F913	
VHJRTM	659F	
VHJLHK	6717	
Y1K5	3974	
Y1K5*	728E	

136.56 2867
180M1- 2879
180M1 286C
222727 0899

File RYWI03 had 2 files 1145 symbols (1059 unreferenced, 0 undefined)

NOTES
recipient agrees NOT to contact manufacturer
NOT Manufacturer Supported

entries and externals

52 6000 EXT FLOPEN
53 6000 EXT FNDLPR
54 6000 EXT FPURGE
55 6000 EXT FOPAC?
56 6000 EXT FOPEN
57 6000 EXT FOPEN1
58 6000 EXT FARMAN
59 6000 EXT FARMAN
60 6000 EXT FSEEK
61 6000 EXT GCHAR
62 6000 EXT GETADM
63 6000 EXT GETADR
64 6000 EXT GETCM?
65 6000 EXT GETINT
66 6000 EXT GETMKT
67 6000 EXT GETNM
68 6000 EXT GETSTP
69 6000 EXT GETNAM
70 6000 EXT GOPEN
71 6000 EXT G012N
72 6000 EXT G10R2N
73 6000 EXT HANDIO
74 6000 EXT HANDI
75 6000 EXT INSERT
76 6000 EXT INIMUL
77 6000 EXT IO?
78 6000 EXT LIST.
79 6000 EXT MELJSB
80 6000 EXT MVARD
81 6000 EXT MDCLC?
82 6000 EXT NUMVA
83 6000 EXT NUMVAL
84 6000 EXT NXLIN
85 6000 EXT POPENV
86 6000 EXT POSRUN
87 6000 EXT PRINT
88 6000 EXT PRIN1
89 6000 EXT PTALD?
90 6000 EXT PUIMTG
91 6000 EXT PUSH45
92 6000 EXT PUTADR
93 6000 EXT PUTREL
94 6000 EXT ON/OFF
95 6000 EXT OWEB
96 6000 EXT ONEI
97 6000 EXT OMR12?
98 6000 EXT RESCON
99 6000 EXT RESTEN
100 6000 EXT RMLST?
101 6000 EXT ROMCLR
102 6000 EXT ROMVPR
103 6000 EXT ROMCPY
104 6000 EXT ROMJSB

entries and externals

2 0000
3 0000
4 0000
5 6000
6 6000
7 6000
8 6000
9 6000
10 6000
11 6000
12 6000
13 6000
14 6000
15 6000
16 6000
17 6000
18 6000
19 6000
20 6000
21 6000
22 6000
23 6000
24 6000
25 6000
26 6000
27 6000
28 6000
29 6000
30 6000
31 6000
32 6000
33 6000
34 6000
35 6000
36 6000
37 6000
38 6000
39 6000
40 6000
41 6000
42 6000
43 6000
44 6000
45 6000
46 6000
47 6000
48 6000
49 6000
50 6000
51 6000

ABC
LOC 24K

A This file contains all of the code which Gary
A Cutler provided to the ID rom. This code is
A not up to usual standards of documentation, but it
A works and I have not messed with it. DO NOT trust
A any comments as they are frequently WRONG.
A
A Raan Young
A
A The ENI TYPENT is used by IORDM1 via an EQUate
A do not move it without changing the EQUate.
A The address ZEBON points to a 5H in IORDM1, do not
A move it without change the address
A

EXT ALERAI
EXT ALIMJ
EXT ALLOC
EXT ALBSW
EXT ALVENI
EXT ASEMAN
EXT AVERALL
EXT BININT
EXT BYAD.R
EXT CLPMAN
EXT CLARRR
EXT CUMDIN
EXT COPY
EXT COPY.
EXT CREDL?
EXT DALLOT
EXT DELETE
EXT DEWANE
EXT DINITI
EXT DNDIC-
EXT ENTISI2
EXT ERRI
EXT ERRI
EXT ERROR
EXT ERROR
EXT EXTRJM
EXT EVIL
EXT FERERT
EXT FUFINO

entries and externals

105 6000 EXT ROOM1
106 6000 EXT RSMA-
107 6000 EXT RUN
108 6000 EXT SAFRUM
109 6000 EXT SAFSTR
110 6000 EXT SETCLW
111 6000 EXT SETED
112 6000 EXT SETPR
113 6000 EXT SETR12
114 6000 EXT SFSICAN
115 6000 EXT SKIPO
116 6000 EXT SKPCON
117 6000 EXT SKPLN
118 6000 EXT SKPVR
119 6000 EXT SMLINT
120 6000 EXT STOSV
121 6000 EXT STRCH
122 6000 EXT STREXP
123 6000 EXT SYSJSB
124 6000 EXT TSIAM
125 6000 EXT TSIEND
126 6000 EXT TUOB
127 6000 EXT UPRE
128 6000 EXT VFBYE
129 6000 EXT VFI
130 6000 EXT VSRCH
131 6000 EXT VSIOR
132 6000 EXT WARM
133 6000
134 6000
135 6000
136 6000
137 6000 E3 IC
138 6002
139 6002 16 60
140 6004 E4 1f
141 6006 80
142 6007 4C
143 6008 20 7D C4 9D
144 600C 69 6F 72 6F
144 6010 6D 33 20 20
145 6014
146 6016
147 6016
148 6016
149 6016
150 6016
151 6016
152 6016
153 6016
154 6016
155 6016
156 6016

DEF RMHEAD this is necessary
DEF LEXBGN the first file
DEF [LEXEND]-[LEXBGN] the size
BYI 10001101B the access types
VAL TYNLEX the type of file
BYI 020H,070H,0C4H,09DH creation date
ASC 'loron3' name
BSZ 2 delimiter
EQU 20
EQU 30
EQU 40
EQU 60
EQU 80
EQU 100
EQU 120
EQU 130
EQU 150

NAME	08/31/82	LOC	OBJFILE	CODE	SRC=RV8103	OBJ=RV2103	12/13/1983	17:38 AM	Pg 5
entries and externals									
210	6016	PARVAL	EQU	420	parameter token lower limit				
211	6016	PENDIX	EQU	540	print N and token				
212	6016	PRINTOK	EQU	130	print N token				
213	6016	PRITOK	EQU	96H	prime str tk				
214	6016	QUOTA	EQU	05H	quoted str tk				
215	6016	REALTK	EQU	040	real con tk				
216	6016	REPTK	EQU	040	replace N tk				
217	6016	RPMTE	EQU	180	replace num tk				
218	6016	RPSIA	EQU	150	replace str tk				
219	6016	SARFTE	EQU	450	str array ref tk				
220	6016	SEARTE	EQU	050	search N tk				
221	6016	SEEKTK	EQU	060	seek N tk				
222	6016	SEKATK	EQU	200	seek abs tk				
223	6016	SEKRTK	EQU	210	seek rel tk				
224	6016	SEPVRL	EQU	46H	value vs reference separator				
225	6016	SERATK	EQU	160	search str tk				
226	6016	SERWTK	EQU	190	search num tk				
227	6016	SIZER	EQU	80650					
228	6016	SREFTK	EQU	430	str ref tk				
229	6016	STRATE	EQU	310	1-dim str array				
230	6016	STRATK	EQU	320	2-dim str array				
231	6016	STROIF	EQU	08H	str difference for N				
232	6016	SVRLTK	EQU	470	str val tk				
233	6016	TWELVE	EQU	0CH	this is twelve				
234	6016	TYPTK	EQU	300	typ tk				
235	6016	ZUSTIK	EQU	530	the famous ZUIS token				
236	6016								
237	6016								
238	6016	65 40	LEXBGN	DEF	MYRBNM	non number			
239	6016	22 60		DEF	RUNLCK	runtime tab			
240	6016	14 60		DEF	ASCLEX	ascii tab			
241	6016	90 60		DEF	PRSLXK	parse tab			
242	6016	51 61		DEF	ERRLEX	erroring tab			
243	6020	0H 61		DEF	INILXK	init routine			
244	6022								
245	6022	50 62		RUNLCK	DEF	MYRIN	00 dummy entry		
246	6024	68 6A			DEF	CLASL	01 clear jofile		
247	6026	87 6A			DEF	DELM	02 DELESEM data		
248	6028	50 62			DEF	MYRIN	03 return INSERTN		
249	602A	50 62			DEF	MYRIN	04 return REPLACN		
250	602E	50 62			DEF	MYRIN	05 return SEARCHN		
251	602E	50 62			DEF	MYRIN	06 return SEEKX		
252	6030	50 62			DEF	MYRIN	07 return ADVANCEN		
253	6032	00 78			DEF	IMATHN	08 chain		
254	6034	34 77			DEF	COMN	09 common		
255	6036	04 78			DEF	CMPUAN	0A clear common		
256	6038	50 62			DEF	MYRIN	0B call		
257	603A	2E 7A			DEF	MSUBL	0C subroutine		
258	603C	3D 7E			DEF	PRMSL	0D print N using		
259									

THEN	LOC	OBJECT CODE	SAC=RV2103	OBJ=RV2103	12/13/1983 11:38 AM	PG 7
***** entries and externals *****						
316	60M 0F 64	DEF	CLEAR	0A	clear common	
317	60M 66 65	DEF	MCALL	0B	call	
318	60M A2 66	DEF	MSUB	0C	subroutine	
319	60M FF 66	DEF	PMUS	0D	print # using	
320	60M 5C 62	DEF	MYRIN	0E		
321	60M 5C 62	DEF	MYRIN	0F		
322	60B0 5C 62	DEF	MYRIN	10		
323	60B2 5C 62	DEF	MYRIN	11		
324	60B4 5C 62	DEF	MYRIN	12		
325	60B6 5C 62	DEF	MYRIN	13		
326	60B8 5C 62	DEF	MYRIN	14		
327	60BA 5C 62	DEF	MYRIN	15		
328	60BC 5C 62	DEF	MYRIN	16		
329	60BE 5C 62	DEF	MYRIN	17		
330	60C0 5C 62	DEF	MYRIN	18		
331	60C2					
332	60C2					
333	60C2 FF FF	DEF	RELHAR		relocation marker	
334	60C4					
335	60C4					
336	60C4 43 4C 45 41 ASKEX ASP	'CLEAR ASSIGN'	01	clean iofile		
336	60E8 52 20 41 53					
336	60C1 53 49 47 4E					
336	60D0 A3					
337	60D1 44 45 4C 45	ASP	'DELETE W'	02	delete next data	
337	60D5 54 45 20 A3					
338	60D9 49 4E 53 45	ASP	'INSERT W'	03	insert data item	
338	60DD 52 54 20 A3					
339	60E1 52 45 50 4C	ASP	'REPLACE W'	04	replace data item	
339	60E5 41 43 45 20					
339	60F9 A3					
340	60E8 53 45 41 52	ASP	'SEARCH W'	05	search for data num	
340	60EE 43 48 20 A3					
341	60F2 53 45 45 48	ASP	'SEEK W'	06	seek to loc relative	
341	60F6 20 A3					
342	60F8 41 44 56 41	ASP	'ADVANCE W'	07	advance # data pointer	
342	60FC 4E 43 45 20					
342	6100 A3					
343	6101 43 48 41 49	ASP	'CHAIN'	08	chain the file	
343	6105 CE					
344	6106 47 4F 4D 4D	ASP	'COMMON'	09	common var declaration	
344	610A 4F CE					
345	610C 43 4C 45 41	ASP	'CLEAR COMMON'	0A	clear the common file	
345	6110 52 20 43 4F					
345	6114 4D 4D 4F CE					
346	6118 43 41 4C CE	ASP	'CALL'	0B	call with parameters	
347	611C 53 55 CE	ASP	'SUB'	0C	subroutine	
348	611F 50 52 49 4E	ASP	'PRINT W'	0D	print # using	
348	6123 54 20 A3					
349	6126 80	VAL	DUMMY	0E	insert str data	
350	6127 80	VAL	DUMMY	0F	replace str data	
351	6128 80	VAL	DUMMY	10	search str data	
352	6129 80	VAL	DUMMY	11	extract num data	


```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RY2103 12/13/1983 11:38 AM PG 8
=====
>>>>>> entries and externals <<<<<<<
353 6128 80 VAL DUMMY 12 replace num data
354 6128 80 VAL DUMMY 13 search num data
355 612C 80 VAL DUMMY 14 seek abs
356 612D 80 VAL DUMMY 15 seek rel
357 612E 80 VAL DUMMY 16 inv advanceM
358 612F 80 VAL DUMMY 17
359 6130 80 VAL DUMMY 18
360 6131 * functions
361 6131
362 6131
363 6131 49 4E 44 45 ASP 'INDEXM' 19 absolute pointer
364 6135 58 A3
365 6137 49 54 45 40 ASP 'ITEMM' 1A relative pointer
366 6138 A3
367 613C 80 VAL DUMMY 1B
368 613D 80 VAL DUMMY 1C
369 613E 4C 49 4E 45 ASP 'LJMLENM' 1D # of items in line
370 6142 4C 45 4E A3
371 6146 80 VAL DUMMY 1E
372 6147 80 VAL DUMMY 1F 1-din R-value pal
373 6148 80 VAL DUMMY 20 2-din R-value pal
374 6149 80 VAL DUMMY 21 1-din L-value pal
375 614A 80 VAL DUMMY 22 2-din L-value pal
376 614B 80 VAL DUMMY 23 1-din L-value rtn
377 614C 80 VAL DUMMY 24 2-din L-value rtn
378 614D 80 VAL DUMMY 25 1-din R-value rtn
379 614E 80 VAL DUMMY 26 2-din R-value rtn
380 614F 80 VAL DUMMY 27 1-din formal common
381 6150 80 VAL DUMMY 28 2-din formal common
382 6151 80 VAL DUMMY 29
383 6152 80 VAL DUMMY 2A num ref token
384 6153 80 VAL DUMMY 2B str ref token
385 6154 80 VAL DUMMY 2C num array ref tk
386 6155 80 VAL DUMMY 2D str array ref tk
387 6156 80 VAL DUMMY 2E num val token
388 6157 80 VAL DUMMY 2F str val token
389 6158 FF VAL FF 30 end of table
390 6159
391 6159 A8 ASP '[' 31 ( for paran list
392 6159 A9 ASP ')' 32 ) for paran list
393 615B A0 ASP '*' 33 (R) val token
394 615C A0 ASP '$' 34 (R) val token
395 615D A0 ASP '$' 35 the famous ZUIS token
396 615E 80 VAL DUMMY 36 print # end .
397 615F 7D ERRLEX BYT 125D error message table
398 6160 64 61 74 61 ASP 'data not found' 125
399 6164 20 6E 6F 74
400 6168 20 66 6F 75
401 616C 6E E4
402 616E 74 79 70 65 ASP 'type mismatch' 126
403 6172 20 6D 69 73
404 6176 6D 61 74 63

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RY2103 12/13/1983 11:38 AM PG 10
=====
>>>>>> entries and externals <<<<<<<
441 61E7 * lddb r2,=route
442 61E7 * lcb r2
443 61E7 * rnz
444 61E7 * jsb =HANDLE
445 61E7
446 61E7 44 A9 C9 7F gto VCHAR.
447 61EB endif
448 61EB C8 09 CMB RO,=V.DALO deallocation?
449 61ED F6 12 IFEQ yes.
450 61EF CE 75 62 jsb =ERT
451 61F2 F6 BE rne
452 61F4 53 C8 0C CMB R23,=NSUBTK
453 61F7 F6 04 IFEQ
454 61F9 44 A9 17 76 GTO DALSUB
455 61FD endif
456 61FD 44 A9 11 76 GTO VDALO1 determine if ours
457 6201 endif
458 6201 C8 16 CMB RO,=V.STRA is this the parser calling?
459 6203 F7 78 JEQ PARSE yes, go try it
460 6205 C8 0B CMB RO,=V.DIN is this DIME calling?
461 6207 F6 04 IFEQ yes.
462 6209 44 A9 40 63 GTO DINEN go try DIN stuff
463 620D endif
464 620D C8 39 CMB r0,=v.endl
465 620F F6 09 ifeq
466 6211 BGIFIX
467 6211 42 B0 CD 82 lddb r2,=route
468 6215 88 lcb r2
469 6216
470 6216 * ADD TO SAVE ROOM FOR BUG FIX
471 6216 F6 9A RNZ
472 6218 F0 3B JAP HANDLE
473 621A
474 621A * REMOVE TO SAVE ROOM
475 621A * jzr handle
476 621A * rln
477 621A endif
478 621A C8 0A CMB RO,=V.DEC maybe the decompiler
479 621C F6 94 RNE here if the decompiler
480 621E CE 75 62 jsb =ERT
481 6221 F6 8F rne
482 6223 53 C8 0C CMB R23,=NSUBTK
483 6226 F6 07 IFEQ
484 6228 CE 55 62 jsb =HANDLE
485 622B 44 A9 FF FF GTO MVARD
486 622F endif
487 622F CE FF FF DECOMP JSB =GETHN otherwise get the var name
488 6232 53 06 E4 PUBD R23,=R6 save token for formal array
489 6235 E2 A9 0E 03 LHM R22,=OEN,3 load up stackmarker and str tok
490 6239 0A E5 PUMD R22,=R12 put then on the stack
491 623B 5E E5 PUMD R36,=R12 push on the var name
492 623D 53 06 E2 PUBD R23,=R6 get the token
493 6240 C8 27 CMB R23,=FORMAL compare to formal limit

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RY2103 12/13/1983 11:38 AM PG 9
=====
>>>>>> entries and externals <<<<<<<
397 617A E8
398 617B 62 61 64 20 ASP 'bad paran value' 127
399 617F 70 61 72 61
400 6183 6D 20 76 61
401 6187 6C 75 E5
402 618A 69 6E 76 61 ASP 'invalid subname' 128
403 618E 6C 69 64 20
404 6192 73 75 62 6E
405 6196 61 6D E5
406 6199 62 61 64 20 ASP 'bad paran type' 129
407 619D 70 61 72 61
408 61A1 6D 20 74 79
409 61A5 70 E5
410 61A7 FF VAL FF
411 61AB
412 61AB ASC 'a' revision 'a'
413 61AB 61 BYT 00000010B this is for FALSE
414 61A9 02
415 61AA
416 61AA 40 C8 31 INILEX CMB RO,=V.FILE is this pointer allocation?
417 61AF F6 04 IFEQ
418 61AF CE 4C 7A jsb =GEICAL
419 61B2 9F rln
420 61B3 endif
421 61B3 C8 0B CMB RO,=V.ALLO is this pointer allocation?
422 61B5 F6 12 IFEQ yes.
423 61B7 CE 75 62 jsb =ERT
424 61BA F6 F6 rne
425 61BC 53 C8 0C CMB R23,=NSUBTK
426 61BF F6 04 IFEQ
427 61C1 44 A9 3A 7B GTO MPTRLO
428 61C5 endif
429 61C5 44 A9 46 70 GTO STRLOC go determine if we handle
430 61C9 endif
431 61C9 C8 07 CMB RO,=V.EALO environmental allocation?
432 61CB F6 11 IFEQ
433 61CD
434 61CD * ADD TWO LINES TO FIX P4(1) BUG
435 61CD 6C 1C E3 PUMD R54,=R34
436 61DD E1 PUMD R54,=R34
437 61D1
438 61D1 6D C8 40 CMB R55,=O1000000B
439 61D4 FA 04 IFEQ
440 61D6 44 A9 41 72 GTO SUBEMV
441 61DA endif
442 61DA 44 A9 F1 71 GTO STREMV determine if its ours
443 61DE endif
444 61DE C8 38 CMB r0,=v.char
445 61E0 F6 09 ifeq
446 61E2 * ADD TO SAVE ROOM FOR BUG FIX
447 61E2 CE 11 6C JSB =BGIFIX
448 61E5 F6 C8 RNZ
449 61E7
450 61E7 * REMOVE TO SAVE ROOM FOR BUG FIX

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RY2103 12/13/1983 11:38 AM PG 11
=====
>>>>>> entries and externals <<<<<<<
494 6242 FA 11 JMC HANDLE set handled and return if not
495 6244 5E A8 28 LDB R36,=(' load up the (
496 6247 0A E4 PUBD R36,=R12 push
497 6249 A9 2C 29 LHM R36,=',)' load ,)
498 624C 53 90 JSB R23 test for 1 or 2 din
499 624E F2 03 IFEQ 2 din.
500 6250 5E DRP 36 set for ,)
501 6251 F0 01 ELSE 1 din.
502 6253 5F DRP 37 set for )
503 6254
504 6254 E5 PUMD R4,=R12 push
505 6255 42 92 CMB R2 clear flag
506 6257 82 75 83 STDB R2,=handld signal that we handled
507 625A 9E RIN
508 625B
509 625B *here for parsing stuff
510 625B
511 625B A1 BYT 241
512 625C 9E MYRIN RIN necessary for DECOM
513 625D
514 625D 2F BYT 57 pseudo for str arrays
515 625E 98 ASTRA, bin
516 625F 40 81 51 82 ldmw r0,=fuvars enter with R20
517 6263 00 85 01 00 ldmw r0,x0,e.ronM
518 6267 F7 05 ifnz
519 6269 CE 83 75 jsb =r10n
520 626C F0 03 else
521 626E 50 08 E1 PUMD R20,=R10 get the vpa addr
522 6271
523 6271 50 0A E5 PUMD R20,=R12 put on the stack
524 6274 9E RIN
525 6275
526 6275 42 14 A1 ERT ldm r2,r24
527 6278 88 dcm r2
528 6279 4E 02 E3 pumw r14,=r2
529 627C C9 65 40 CMB r14,=INVROMM
530 627F 9E rln
531 6280
532 6280
533 6280 4C C8 28 PARSE CMB R14,='[' do we have a string array?
534 6283 F6 FA RNE nope so give up
535 6285 C1 55 62 PARSE- JSB =HANDLE
536 6288 65 0A E3 PUMD R45,=R12 yes, so clear up the stack
537 628B C8 13 CMB R45,=R23 L-value?
538 628D F6 05 IFEQ
539 628F 4E A8 21 LDB R14,=R33D if so give them our L-value
540 6292 F0 04 ELSE
541 6294 9C CMB R45
542 6295 4C A8 1F LDB R14,=R31D else give R-value
543 6298
544 6298 65 06 E4 PUBD R45,=r6
545 629B 60 A9 LHM R55,=
546 629D B4 VAL ERONTK load up with
ext ron token

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 12
-----
>>>>>> entries and externals <<<<<<<
547 629E 65 40 DEF MYROM          rom number
548 629A 0A E5 PUMD R55,+R12      put those guys on the stack
549 629B 4A 06 E5 PUMD r12,-r6
550 629C 4C 0A E4 PUMD R14,+R12      put our token on
551 629D 92 CLB R14          clear for d1n stuff
552 629E 06 E4 PUMD r14,-r6
553 629B 66 0A E5 PUMD R46,+R12
554 629F CE FF FF JSB =SYSJSB      go get a numeric parameter
555 62B1 FF FF DEF NUVMN+
556 62B3 F9 0F JEZ 3F          no numeric so error
557 62B5 CE 06 63 JSB =COMMI      get a comma
558 62B8 F9 0C JEZ 1F          nope only one param?
559 62BA 42 06 E2 PUMD r2,-r6
560 62B0 88 tcb r2
561 62BE E4 PUMD r2,-r6
562 62BF CE FF FF JSB =SYSJSB      get the next numeric
563 62C2 FF FF DEF NUVMN+
564 62C4 F9 21 3H JEZ ERR3      error not there
565 62C6 4C 08 29 1H CMB R14,'1' get the 1
566 62C9 F6 0C JNE ERR3
567 62CB CE FF FF JSB =SFSCAN
568 62CE 6C A9 LDM R54,+          load up with
569 62D0 84 VAL EROMTK          ext rom
570 62D1 65 40 DEF MYROM          rom #
571 62D3 00 BYI 0              place holder
572 62D4 42 06 E2 PUMD r2,-r6
573 62D7 58 E3 PUMD r30,-r6      R30 ptr to token
574 62D9 6F E2 PUMD r57,-r6      val or addr
575 62DB F6 12 JFZR          value
576 62DD 42 90 tsb r2          1 or 2 d1n
577 62DF F6 05 IFZR          1 d1n
578 62E1 6F AB 25 LDB R57,+37D
579 62E4 F0 07 ELSE          2 d1n
580 62E6 6F 18 A4 LDBD R57,R30
581 62E9 88 tcb R57
582 62EA A6 STBD R57,R30
583 62EB A8 26 LDB R57,+38D
584 62ED ENMT          addr
585 62ED F0 10 ELSE          1 or 2 d1n
586 62EF 42 90 tsb r2          1 d1n
587 62F1 F6 05 IFZR          1 d1n
588 62F3 6F AB 23 LDB R57,+35D
589 62F6 F0 07 ELSE          2 d1n
590 62F8 6F 18 A4 LDBD R57,R30
591 62FB 88 tcb R57
592 62FC A6 STBD R57,R30
593 62FD A8 24 LDB R57,+36D
594 62FF ENMT
595 62FF ENMT
596 62FF 6C 0A E5 PUMD R54,+R12      push to stack
597 6302 4C 08 DE CMB R14,-LBRAK  substring?
598 6305 F6 2F RNE          nope we're done
599 6307 92 CLB R14

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 14
-----
>>>>>> entries and externals <<<<<<<
653 6372 CE BC 63 JSB =SINI      get the next int
654 6375 F9 C6 JEZ ERR          wasn't there so error
655 6377 4C 08 29 1H CMB R14,'1' get the 1
656 637A F6 C1 JNEERR JNE ERR
657 637C CE FF FF JSB =SFSCAN
658 637F 6C A9 LDM R54,+          load up
659 6381 B4 VAL EROMTK          ext rom
660 6382 65 40 DEF MYROM          rom#
661 6384 00 BYI 0              place holder
662 6385 52 81 0D B2 LDM R22,+THPMN2
663 6389 6F B0 B2 B3 LDBD R57,+ORIGIN
664 638D F6 04 IFZR          1-d1n
665 638F A8 23 LDB R57,+35D
666 6391 F0 06 ELSE          2-d1n
667 6393 12 A4 LDBD R57,R22
668 6395 88 tcb R57
669 6396 A6 STBD R57,R22
670 6397 A8 24 LDB R57,+36D
671 6399 ENMT
672 6399 6C 0A E5 PUMD R54,+R12      push to stack
673 639C 4C 08 DE CMB R14,-LBRAK  max length
674 639F F6 95 RNE          nope default len
675 63A1 CE FF FF JSB =SFSCAN      get the next token
676 63A4 CE BC 63 JSB =SINI      get the integer
677 63A7 F9 BC JEZ JEZERR        not there so error
678 63A9 4C 08 DF CMB R14,-RBRAC1 right bracket
679 63AC F6 CC JNE JMEERR        nope so error
680 63AE A8 1D LDB R14,+1DH      else load 1-d1n substr
681 63B0 0A E4 PUMD R14,+R12      push to stack
682 63B2 CE FF FF JSB =SFSCAN      final scan for the
683 63B5 9E RIM
684 63B6 COMMI JSB =SYSJSB
685 63B9 FF FF DEF GETCN?
686 63BB 9E RIM
687 63BD COMMI JSB =SYSJSB
688 63BF FF FF DEF SHLINI
689 63C1 9E RIM
690 63C1 9E RIM
691 63C1 9E RIM
692 63C2

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 13
-----
>>>>>> entries and externals <<<<<<<
600 6308 06 E4 PUMD r14,-r6
601 630A CE FF FF JSB =SYSJSB      get the substring value
602 630C FF FF DEF NUVMN+
603 630F F9 29 JEZ ERR2          not there error
604 6311 CE 06 63 JSB =COMMI      2 d1n substr?
605 6314 F9 0C JEZ 2F          nope just one
606 6316 42 06 E2 PUMD r2,-r6
607 6319 88 tcb r2
608 631A E4 PUMD r2,-r6
609 631B CE FF FF JSB =SYSJSB      yes doing two
610 631E FF FF DEF NUVMN+
611 6320 F9 18 JEZ ERR2          value not there
612 6322 4C 08 DF 2H CMB R14,-RBRAC1
613 6325 F6 13 JNE ERR2          nope then error
614 6327 06 E2 PUMD r14,-r6
615 6329 F6 04 IFZR
616 632B A8 1D LDB R14,+1DH
617 632D F0 02 ELSE
618 632F A8 1E LDB R14,+1EH
619 6331 ENMT
620 6331 0A E4 PUMD R14,+R12
621 6333 CE FF FF JSB =SFSCAN
622 6336 9E RIM
623 6337
624 6337 7D 06 E3 ERR3 PUMD r75,-r6
625 633A 7D 06 E2 ERR2 PUMD r75,-r6
626 633C CE FF FF ERR JSB =ERRADR+
627 6340 4E BYI 78D
628 6341
629 6341
630 6341
631 6341 4C 08 28 DINEM CMB R14,'1' string array?
632 6344 F6 0A RNE          nope don't handle
633 6346 CE 55 62 JSB =HANDLE      else set handle
634 6349 65 0A E3 PUMD R45,-R12      get tokens off stack
635 634C 6C A9 LDM R54,+          load up
636 634E B4 VAL EROMTK          ext rom
637 634F 65 40 DEF MYROM          rom #
638 6351 1F BYI 31D          token str array
639 6352 E5 PUMD R54,+R12      onto the stack
640 6353 52 A1 LDM R22,R12
641 6355 88 CMB R22
642 6356 B3 0F B2 STMD R22,+THPMN2
643 6359 92 CLD R22
644 635A B2 02 B3 STBD R22,+ORIGIN
645 635D 66 E5 PUMD R46,+R12
646 635F CE FF FF JSB =SFSCAN      get next token
647 6362 CE BC 63 JSB =SINI      get small integer
648 6365 F9 06 JEZ EPR          nada so error
649 6367 CE 06 63 JSB =COMMI      got a comma?
650 636A F9 0E JEZ 1F          nope so 1-d1n
651 636C 42 06 01 LDB R2,-ONE      otherwise
652 636F B2 02 B3 STBD R2,+ORIGIN increment the flag

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 15
-----
>>>>>> parse time <<<<<<<
694 63C2 CE FF FF ROM/OF JSB =SYSJSB      get the
695 63C5 FF FF DEF OM/OF          on/off token
696 63C7 6C A9 LDM R54,+
697 63C9 B4 VAL EROMTK
698 63CA 65 40 DEF MYROM+
699 63CC 00 BYI 0
700 63CD 6F 0A E2 PUMD R57,-R12      get token
701 63D0 6C E5 PUMD R54,+R12      put back all of it
702 63D2 9E RIM
703 63D3

```

NOMAS
 Not Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE  SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 16
-----
>>>>>>  utilities  <<<<<<<
705 63D3
706 63D3 CE FF FF  MYNUM JSB =ROMJSB      sys call
707 63D6 FF FF      DEF  SAFNUM      to numval
708 63D8 63 40      DEF  NYROM1
709 63DA 9E          RIM
710 63DB
711 63DB CE FF FF  MYNUM JSB =SVSJSB      sys call
712 63DE FF FF      DEF  NUMVA*      to numva*
713 63E0 9E          RIM
714 63E1
715 63E1 CE FF FF  MYSTR JSB =ROMJSB      sys call
716 63E4 FF FF      DEF  SAFSTR      to strexp
717 63E6 63 40      DEF  NYROM1
718 63E8 9E          RIM
719 63E9
720 63F9

```

NOMIAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE  SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 18
-----
>>>>>>  DELET, parse routine for DELETEN  <<<<<<<
748 63F8
749 63F8
749 63F8
750 63F8
751 63F8
752 63F8
753 63F8
754 63F8
755 63F8
756 63F8
757 63F8
758 63F8
759 63F8
760 63F8
761 63F8
762 63F8
762 63F8
763 63F8
764 63F8 4C 06 E4  DELET  PUBD R14,R6      save the token
765 63FB CE D8 63  JSB  =MYNUM*      get the channel #
766 63FE F9 66      JEZ  ERR78      not there, error
767 6400 4C C8 2C      cmb  r14,=CONNA
768 6403 F6 05      jne  pu-r6
769 6405 CE D8 63  jsb  =MYNUM*
770 6408 F9 5C      jez  err78
771 640A 5E 06 E2  pu-r6  PUBD R36,-R6      get back the token
772 640D F0 DC      PU-   JNP  PUSHUS      push and return
773 640F          *          mynum* did a post scan
774 640F

```

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE  SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 17
-----
>>>>>>  PUSHUS, utility to push token and other stuff  <<<<<<<
722 63E9
723 63E9
724 63E9
725 63E9
726 63E9
727 63E9
728 63E9
729 63E9
730 63E9
731 63E9
732 63E9
733 63E9
734 63E9
735 63E9
736 63E9
736 63E9
737 63E9
738 63E9 1E A2  PUSHUS  SIB  RW,R36
739 63EB 6C A9  PUSHUS  LDM  R54,-      r54 load for push
740 63ED B4      VAL  ERQNTK      external run token
741 63EE 65 40      DEF  NYROM1      my rom number
742 63F0 90      BYI  0          place holder
743 63F1 EF 1E A0  LDB  R57,R36      input token to r57
744 63F4 6E 0A E5  PUMP  R54,R12      push to R12
745 63F7 9E          RIM
746 63F8

```

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE  SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 19
-----
>>>>>>  CLEAR, parse routine for CLEAR ASSIGN W  <<<<<<<
776 640F
777 640F
777 640F
778 640F
779 640F
780 640F
781 640F
782 640F
783 640F
784 640F
785 640F
786 640F
787 640F
788 640F
789 640F
790 640F
790 640F
791 640F
792 640F 4C      CLEAR  DRP  14
793 6410 CE E9 63  CLEAR- JSB  =PUSHW      push our stuff
794 6413 CE FF FF  JSB  =SFSCAN      final scan for parser
795 6416 9E          RIM
796 6417

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG 20
-----
>>>>>> INSER, parse routine for INSERT, REPLACE and S <<<<<<<
798 6417
799 6417
799 6417
800 6417
801 6417 * INSER: parses the INSERT, REPLACE and SEARCHM command
802 6417
803 6417 * INPUT: appropriate token in R14
804 6417
805 6417 * OUTPUT: parsed expression on R12
806 6417
807 6417 * VOLATILES: R36
808 6417
809 6417 * EXT ROUTINES: MYNUM, MYSTR, MYNUM
810 6417
811 6417 * ERRORS: syntax
812 6417
812 6417
813 6417
814 6417 4D B2 06 82 INSER ST0D R15,=PARERR set not advan flag
815 6418 0C 06 E4 INSER- PUBD R14,=R6 save our token
816 641E 0D 06 63 JSB =MYNUM+ get a num exp
817 6421 F9 43 JEZ ERR78 not there, error
818 6423 4C C8 27 OLSIF CMB R14,=COMMA compare next to semi
819 6426 F6 3E JNE ERR78 not there, error
820 6428 5E 06 E2 PUBD R36,=R6 get our token
821 642B E4 PUBD R36,=R6 save it for later
822 642C 0E 10 64 JSB =CLEAR- push it and scan
823 642F 5E A8 27 LDB R36,=SEMI load up the semi token
824 6432 4C C8 04 CMB R14,=-4
825 6435 F6 09 IFEQ
826 6437 28 0A E3 PUBD R70,=R12
827 643A 5E E4 PUBD R36,=R12
828 643E 78 E5 PUBM R70,=R12
829 643F 0A 03 ELSE
830 6440 5E 0A E4 PUBD R36,=R12
831 6443
832 6443 5E 0A 08 82 LDB R36,=PARERR from advance?
833 6447 F7 03 IFNZ yes,
834 6449 06 E2 PUBD R36,=-R6 clean stack
835 644B 9E RTN return
836 644E
837 644E 0E E1 63 JSB =MYSTR get a string
838 644F F9 07 IFEN if there,
839 6451 5E 06 E2 PUBD R36,=-R6 get our token
840 6454 0A 08 RDB R36,=STRDEF offset four
841 6456 0A 08 ELSE
842 6458 0E D3 63 JSB =MYNUM get a num exp
843 645B F9 0A JEZ ERR78 not there, error
844 645D 5E 06 E2 PUBD R36,=-R6 get our token
845 6460 0A 0E RDB R36,=MUNDEF offset seven
846 6462
847 6462 F0 A9 PU+ JMP PU+ push it and post scan

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG 22
-----
>>>>>> SEEK, parse routine for the SEEK statement <<<<<<<
888 6485
889 6485
889 6485
890 6485
891 6485 * SEEK: parses the SEEK statement
892 6485
893 6485 * INPUT: SEEK token in R14
894 6485
895 6485 * OUTPUT: parsed statement on the R12 stack
896 6485
897 6485 * VOLATILES: R36
898 6485
899 6485 * EXT ROUTINES: CLEAR, MYNUM
900 6485
901 6485 * ERRORS: syntax
902 6485
902 6485
902 6485
903 6485
904 6485 0E 0F 64 SEEK JSB =CLEAR push token and scan
905 6488 0E D3 63 JSB =MYNUM get a numeric
906 648B F9 0C ERR78A JEZ ERR78 not there, error
907 648D 4C C8 2C CMB R14,=COMMA got to get a comma
908 6490 F6 A7 JNE ERR78 error not there
909 6492 0E D8 63 JSB =MYNUM+ get a numeric
910 6495 F9 A2 JEZ ERR78 not there, error
911 6497 5E A8 14 LDB R36,=SEMIK load seek abs tk
912 649A 4C C8 2C CMB R14,=COMMA check for comma
913 649C F6 08 IFEQ comma,
914 649F 0E D8 63 JSB =MYNUM+ get a numeric
915 649D F9 E7 JEZ ERR78A no then error
916 649A 5E A8 15 LDB R36,=SEMIK load seek rel tk
917 6497
918 6497 F0 89 ENDEF
919 6499 JMP PU+ push it

```



 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG 21
-----
>>>>>> INSER, parse routine for INSERT, REPLACE and S <<<<<<<
848 6464 F0 A9 CLR- JMP CLEAR for others
849 6466 5E 06 E2 IAR78 PUBD R36,=-R6 clean up stack
850 6469 0E FF FF IAR78- JSB =ERR78+ report error
851 646E 4E BYT 78D syntax
852 646D
853 646D
854 646D 4D B2 08 82 SEARCH ST0D R15,=PARERR set not advan flag
855 6471 4C 06 E4 PUBD R14,=R6 save token
856 6474 08 03 LDB R14,=THREE load counter with 3
857 6476 E4 PUBD R14,=R6 save counter
858 6477 0E D8 63 H110 JSB =MYNUM+ get a numeric
859 647A 02 06 E2 PUBD R2,=-R6 get counter
860 647D F9 E7 JEZ ERR78 no numeric so error
861 647F BA DEB R2 dec counter
862 6480 F7 A1 JZR OLSIF got enough
863 6482 4C C8 2C CMB R14,=COMMA compare to comma
864 6485 F6 9C JNE OLSIF if not must be semi
865 6487 42 E4 PUBD R2,=R6 put back counter
866 6489 F0 E4 JMP H110 get another
867 648B
868 648B 4D A8 01 R11VAN LDB R15,=ONE load adv flag
869 648E 02 08 82 ST0D R15,=PARERR set adv flag
870 6491 92 CLR R15 clear for n adds
871 6492 0E 10 64 JSB =INSER parse the beginning
872 6495 0E D3 63 JSB =MYNUM get the ship count
873 6498 F9 0F JEZ ERR78 not there error
874 649A 4C C8 2C CMB R14,=COMMA next a comma?
875 649D F6 0A JNE ERR78 no, error
876 649F 0E FF FF JSB =SFSCAN get next token
877 64A2 4C C8 01 CAR R14,=1 a variable?
878 64A5 F6 C2 JNE ERR78 no, error
879 64A7 08 11 LDB R14,=THREE load store var
880 64A9 0A E4 PUBD R14,=R12 to the stack
881 64AB 66 24 A1 LDB R16,=R44 get the var name
882 64AD 0A E5 PUBM R16,=R12 out to stack
883 64B0 5E A8 16 LDB R36,=INADV load inv adv tok
884 64B3 F0 A9 PUS+ JMP CLR push and scan
885 64B5
886 64B5

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG 23
-----
>>>>>> COMMON, parse routine for COMMON <<<<<<<
921 64D9
922 64D9
922 64D9
923 64D9
924 64D9
925 64D9
926 64D9
927 64D9
928 64D9
929 64D9
930 64D9
931 64D9
932 64D9
933 64D9
934 64D9
935 64D9
935 64D9
935 64D9
936 64D9
937 64D9
938 64D9 0E FF FF COMMON JSB =NOCLC1 push the common tk
939 64DC 42 01 55 82 LDBD R2,=STSIZE get the stsize pos
940 64E0 89 ICM R2 inc for current
941 64E1 0A 01 CMB R2,R12 compare to R12
942 64E3 F7 04 IFNE not here then
943 64E5 0E FF FF JSB =ERR78+ error extra
944 64E8 54 BYT 84D chars
945 64E9
946 64E9 4C DRP 14
947 64EA 0E E9 63 JSB =PUSHW scan for next
948 64EC 0E FF FF JSB =SFSCAN if a CR then error
949 64F0 4C C8 0D CMB R14,=CR
950 64F3 F6 0C JNE BLOF
951 64F5
952 64F5 0E FF FF ERR89 JSB =ERR78+ no, error
953 64F8 59 BYT 89D bad parameter
954 64F9
955 64F9 0E FF FF COMMON JSB =SFSCAN scan for next
956 64FC 4C C8 0D CMB R14,=CR return on cr
957 64FF F7 64 REQ
958 6501 4C C8 03 BLOF CMB R14,=THREE str or num
959 6504 FA 37 JNE NUMER1 nc, then num
960 6505 F6 E0 JNE ERR89 ne, then neither
961 6508 50 C8 28 CMB R20,=1 str array?
962 650B F6 C8 JNE PUSTR no then simple str
963 650D 0E FF FF JSB =GCHAR get next char
964 6510 5E A8 27 LDB R36,=FORITK load 1 dim
965 6513 50 C8 2C CMB R20,=1 compare to comma
966 6516 F6 0C IFEQ 2 dim,
967 6518 5E A8 28 LDB R36,=FORITK load two dim
968 651B 0E FF FF JSB =GCHAR get next char
969 651E
970 651E 50 C8 29 CMB R20,=1 a 'J'

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV8103 OBJ=RVX103 12/13/1983 11:38 AM PG 24
*****
>>>>>> COMMON, parse routine for COMMON <<<<<<<
971 6521 F6 02 JNE ERR89 no, bad param
972 6523 CE E8 63 JSB =PUSHUS push 11 to stack
973 6526 CE FF FF JSB =GCHAR get next char
974 6529 66 24 A1 PUNAME LDM R46,R44 get var name
975 652E 0A E5 PUNU R46,R12 push to stack
976 6530 50 C8 2E CMB R20,="," is this a ','
977 6531 F6 C6 JNE COMNO- hope its a cr
978 6533 CE FF FF JSB =GCHAR yes, then get next char
979 6536 F0 C1 JNP COMNO- and continue
980 6538
981 653B 4C 09 E4 PUSTW PUBD R14,*R12 push token
982 653D F0 EC JNP PUNAME go push name
983 653D
984 653D 4C C8 02 NUMER1 CMB R14,*TWD is this an array
985 6540 FA F6 JNE PUSTW no, then simple
986 6542 AB B5 LDB R14,*OB5H load formal 1 dim
987 6544 50 C8 2C -- CMB R20,="," yes, 1 or 2
988 6547 F6 06 IFEQ 2 dim,
989 6549 4C AB B6 LDB R14,*OB6H load formal 2 dim
990 654C CE FF FF JSB =GCHAR get next char
991 654F
992 654F 50 C8 29 CMB R20,=")" a ')',
993 6552 F6 A1 JNE ERR89 nope then error
994 6554 4C 0A E4 PUBD R14,*R12 push token to stack
995 6557 CE FF FF JSB =GCHAR get next char
996 655A F0 CD JNP PUNAME push name and check char
997 655C

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV8103 OBJ=RVX103 12/13/1983 11:38 AM PG 26
*****
>>>>>> PARSE CALL, entry points and externals <<<<<<<
1024 6566
1025 6566
1026 6566
1027 6566
1028 6566
1029 6566
1030 6566
1031 6566
1032 6566
1033 6566
1034 6566
1035 6566 * entry points
1036 6566
1037 6566 * externals
1038 6566
1039 6566

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV8103 OBJ=RVX103 12/13/1983 11:38 AM PG 25
*****
>>>>>> CHAIN, parse routine for chain <<<<<<<
999 655C
1000 655C *****
1000 655C *
1001 655C *
1002 655C * CHAIN: parses the chain statement
1003 655C *
1004 655C * INPUT: chain token in R14
1005 655C *
1006 655C * OUTPUT: parsed expression on the r12 stack
1007 655C *
1008 655C * VOLATILES: R36
1009 655C *
1010 655C * EXT ROUTINES: GETSTP, SYSJSB, CLEAR-
1011 655C *
1012 655C * ERRORS: none
1012 655C *****
1013 655C
1014 655C
1015 655C CE FF FF CHAIN JSB =SYSJSB get a string exp
1016 655F FF FF DEF GETSTP
1017 6561 5E DRP 36
1018 6562 CE 10 6A JSB =CLEAR- push our stuff and
1019 6565 9E R1N post scan
1020 6566
1021 6566
1022 6566

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV8103 OBJ=RVX103 12/13/1983 11:38 AM PG 27
*****
>>>>>> MCALL, parse routine for CALL <<<<<<<
1041 6566
1042 6566 *****
1042 6566 *
1043 6566 *
1044 6566 * MCALL: parses the CALL statement with parameter list
1045 6566 *
1046 6566 * INPUT: CALL token in R14
1047 6566 *
1048 6566 * OUTPUT: parsed statement
1049 6566 *
1050 6566 * VOLATILES: just about everything
1051 6566 *
1052 6566 * EXT ROUTINES: CLEAR-, ERADR+, GCHAR, GETSTP, MYNUM
1053 6566 * MYSTR, PUSHUS, SYSJSB, SFSCAN
1054 6566 *
1055 6566 * ERRORS: 89, bad parameter on parameter list
1056 6566 *
1056 6566 *****
1057 6566
1058 6566
1059 6566 CE FF FF MCALL JSB =SYSJSB get a string
1060 6569 FF FF DEF GETSTP
1061 656B F9 F8 REZ not there then return
1062 656D 42 A8 4F LDB R2,*CALLTK load up sys call
1063 6570 0A E4 PUBD R2,*R12 push to stack
1064 6572 4C C8 27 CMB R14,*SER1
1065 6575 F6 04 IFEQ
1066 6577 E4 PUBD R14,*R12
1067 6578 CE FF FF JSB =SFSCAN
1068 657B
1069 657B 4C C8 28 ENDF
1070 657E F6 E5 CMB R14,="[" parameters ?
1071 6580 RME
1072 6580 CE FF FF PARAMS JSB =SFSCAN get first param
1073 6583 4C C8 01 CMB R14,*1 num var
1074 6586 F6 12 JNE CKSTR no, check str
1075 6588 5E A8 2A LDB R36,*NREFTK load num ref
1076 658B CE 5C 66 JSB =EOLIST test for reference
1077 658E F7 31 jeq REFER go do reference
1078 6590 CE D3 63 JSB =MYNUM must be value
1079 6593 F9 28 JEZ FRR89 no numer exp, error
1080 6595 5E AB 2E HUMP LDR R36,*MYALTK load num val
1081 6598 F0 2D JNP MORE push and check for more
1082 659A
1083 659A 4C C8 03 CKSTR CMB R14,*3 str var
1084 659D F6 58 JNE CKARR no, check array
1085 659F 50 C8 28 CMB R20,="[" yes, str array
1086 65A2 F7 12 JEQ CKSTAR go do str array
1087 65A4 5E AB 28 LDB R36,*SREFTK load str ref
1088 65A7 CE 5C 66 JSB =EOLIST test for reference
1089 65AA F7 15 jeq REFER go do reference
1090 65AC CE E1 63 IMOPE JSB =MYSTR no, get str exp

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&103 OBJ=RVZ103 12/13/1983 11:38 AM PG 28
-----
>>>>>>> MCALL, parse routine for CALL
1091 65AF F9 0C JEZ FRAB9 no str exp, error
1092 65B1 5E A8 2F STRUP LDB R36,=SVALIK load str val
1093 65B4 F0 11 JNP MORE push and check for more
1094 65B6
1095 65B6 EE 4E 66 (KSTAR JSB =eol1 get next char
1096 65B9 F7 25 JEZ ARREF
1097 65BB F0 EF JNP MORE
1098 65BD
1099 65BD CE FF FF FRAB9 JSB =ERROR error
1100 65C0 59 BYI R8D bad parameter
1101 65C1
1102 65C1 4C C8 10 REFER ADB R14,=10H increase to L-val tk
1103 65C4 EE 6F 66 JSB =SFUPUSH
1104 65C7 CE E0 63 MORE JSB =PUSHUS push type token
1105 65CA 4C C8 2C CMB R14,=, compare to ','
1106 65CD F7 B1 JEZ PARANS yes
1107 65CF C8 29 CMB R14,=, is next a ','
1108 65D1 F6 E8 JNE FRAB9 no, then error
1109 65D3 5E A8 31 PAD LDB R36,=01K load up '[' token
1110 65D6 CE E0 63 JSB =PUSHUS push to stack
1111 65D9 5E A8 3C LDB R36,=01K load up '[' token
1112 65DE CE 10 64 JSB =CLEAR- push and post scan
1113 65DF 9E RTN that's all folks
1114 65E0
1115 65E0 4C 06 E4 ARREF pubd r14,=r6 1 or 2 dim array
1116 65E3 CE 75 66 JSB =ARRAY-
1117 65E6 42 06 E2 pubd r02,=-r6
1118 65E9 F8 02 JEN FRAB9
1119 65EB 5E A8 2D LDB R36,=SARFTK
1120 65EE 42 90 TSB r02
1121 65F0 F2 03 IFEV
1122 65F2 5E A8 2C LDB R36,=AREFTK
1123 65F5 ENDF
1124 65F5 F0 D0 MORE JNP MORE
1125 65F7
1126 65F7 4C C8 02 (KARA CMB R14,=2 doing array
1127 65FA F6 18 JNE VALUE no, check expressions
1128 65FC EE 5C 66 JSB =EOLIST
1129 65FF F7 DF JEZ ARREF
1130 6601 CE D3 63 JEN =HYNUM
1131 6604 F8 8F JEN NUMV go do num val stuff
1132 6606 F0 B5 JNP FRAB9
1133 6608
1134 6608 CE E1 61 VAL JSB =MYSIR check for str exp
1135 660B F8 05 IFEZ no,
1136 660D CE D3 63 JSB =HYNUM check for num exp
1137 6610 F0 F2 JNP NUMV go do num val stuff
1138 6612 ENDF
1139 6612 F0 9D JNP STRUP go do str val stuff
1140 6614
1141 6614 4C C8 28 VALUE CMB R14,=, var value?
1142 6617 F6 EF JNE VAL no the check for exp
1143 6619 48 06 E5 PUND R10,=r6

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&103 OBJ=RVZ103 12/13/1983 11:38 AM PG 30
-----
>>>>>>> MCALL, parse routine for CALL
1197 66B3 42 AB B6 LDB r2,=Ob6h
1198 66B6 5E A8 28 LDB r36,=for2tk
1199 66B9 ENDF
1200 66B9 4C 90 TSB r14
1201 66BB F2 07 IFEV
1202 66BD 02 A0 LDB r14,r2
1203 66BF CE 6F 66 JSB =SFUPUSH
1204 66C2 F0 03 ELSE
1205 66C4 CE 64 66 JSB =ARNAME
1206 66C7 ENDF
1207 66C7 9D 9C NEWMAR STE
1208 66C9 4C C8 29 CMB r14,=,
1209 66CE F6 D6 RNE
1210 66D1 CE FF FF JSB =SFSCAN
1211 66D4 9E RTN
1212 66D6
1213 66D6 CE FF FF NSUB JSB =noc1
1214 66D8 4C DRP 14 store token
1215 66DA CE E9 63 JSB =PUSHM push to stack
1216 66DB CE FF FF JSB =SYSJSB get the
1217 66DE FF FF DEF S1ACH sub name
1218 66E0 F9 F1 REZ return not there
1219 66E0 5E A8 35 LDB R36,=ZUISIK the famous ZUIS token
1220 66E3 CE E0 63 JSB =PUSHUS push the ZUIS token
1221 66E6 4C C8 28 CMB R14,=, any parameters
1222 66E9 F6 91 JNE BADPAR no, invalid
1223 66EB CE C8 66 JSB =RDNFIX go do variables
1224 66ED 4C C8 29 CMB R14,=, end with ','
1225 66F1 F6 89 JNE BADPAR no, report error
1226 66F3 CE D3 65 JSB =PAD jump to pad and finish
1227 66F6 CE FF FF JSB =DMDCR- only ' or CR are allowed
1228 66F9 9E RTN
1229 66FA
1230 66FA CE FF FF RDNFIX JSB =SFSCAN
1231 66FD 4C C8 04 CMB r14,=four
1232 66FD F8 F7 RCY
1233 66FE 90 TSB r14
1234 66FF F3 12 JEN DNMEN
1235 66FD CE 5C 66 JSB =EOLIST
1236 66D8 F6 05 IFEQ
1237 66DA CE 6F 66 JSB =SFUPUSH
1238 66DD F0 13 ELSE
1239 66DF 50 C8 28 CMB r20,=,
1240 66E2 F6 03 IFEQ
1241 66E4 CE 4E 66 JSB =eol1
1242 66E7 ENDF
1243 66E7 CE 5C 66 DNMEN JSB =EOLIST
1244 66EA F7 0D JEZ NORMAL
1245 66EC 4C DRP 14
1246 66ED CE FF FF JSB =SYSJSB
1247 66F0 FF FF DEF D1ALIE
1248 66F2 ENDF
1249 66F2 4C C8 2C CONMA? CMB r14,=,

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&103 OBJ=RVZ103 12/13/1983 11:38 AM PG 29
-----
>>>>>>> MCALL, parse routine for CALL
1144 661E 50 E4 pubd r20,=r6
1145 661E CE FF FF JSB =SFSCAN get the var
1146 6621 4C C8 01 CMB R14,=1 numeric var?
1147 6624 F7 09 IFEZ no,
1148 6626 C8 03 CMB R14,= string var?
1149 6628 F6 0D JNE che no, error
1150 662A 5E A8 34 LDB R36,=R36IK load str val tk
1151 662D F0 03 ELSE yes,
1152 662F 5E A8 33 LDB R36,=R1TK load num val tk
1153 6632 ENDF
1154 6632 50 C8 29 CMB r20,=,
1155 6635 F7 0A IFEV
1156 6637 4C C8 28 CHE LDB r14,=,
1157 663A 50 06 F2 pubd r20,=-r6
1158 663D 48 E3 PUND R10,=-r6
1159 663F F0 E7 JNP VAL
1160 6641 ELSE
1161 6641 6D 06 E3 PUND r55,=-r6
1162 6644 ENDF
1163 6644 CE 6F 66 JSB =SFUPUSH
1164 6647 CE 97 66 JSB =NEWNAME
1165 6649 F9 A9 JEZ MORE
1166 664C F0 B8 BADPAR JNP FRAB9
1167 664E
1168 664E 42 08 A4 EOLIT LDB r2,=r10
1169 6651 CE 5D 66 JSB =eol1
1170 6654 F6 8D RNE
1171 6656 CE FF FF JSB =gchar
1172 6659 42 9C CMB r2,=
1173 665B 9E RTN
1174 665E
1175 665E 50 EOLIT DRP 20
1176 665D C8 2C EOL LDB r1,=, is next a ','
1177 665F F2 FA RED set 2 and rtn
1178 6661 C8 29 CMB r1,=, is next a ','
1179 6663 9E RTN set 2 and rtn
1180 6664
1181 6664
1182 6664 58 24 A1 ARNAME LDB r30,r44
1183 6667 5E DRP 36
1184 6668 CE 10 64 JSB =CLEAR-
1185 666B 58 0A F5 PUND r30,=r12
1186 666E 9E RTN
1187 666F
1188 666F CE FF FF SFUPUSH JSB =SYSJSB
1189 6672 FF FF DEF PUSHAS
1190 6674 9E RTN
1191 6675
1192 6675 42 AB B5 ARRAY- LDB r2,=Ob5h
1193 6678 5E A8 27 LDB r36,=for1tk
1194 667B 50 C8 24 CMB r20,=,
1195 667E F7 09 IFEV
1196 6680 CE FF FF JSB =GNAR

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&103 OBJ=RVZ103 12/13/1983 11:38 AM PG 31
-----
>>>>>>> MCALL, parse routine for CALL
1250 66F5 F6 D2 RNE
1251 66F7 F0 D1 JNP RDNFIX
1252 66F9
1253 66F9 CE 75 66 NORMAL JSB =ARRAY-
1254 66FE F9 F4 JEZ CONMA?
1255 66FE 9E RTN
1256 66FF
1257 66FF

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 32
=====
>>>>>> PRINT # parsing entry points and externals <<<<<<<
1259 66FF
1260 66FF
1261 66FF
1262 66FF
1263 66FF
1264 66FF
1265 66FF

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 34
=====
>>>>>> utility group 2 entries and externals <<<<<<<
1302 6723
1303 6723
1304 6723

```

NOMAS
 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 33
=====
>>>>>> here is the Print # Using parsing code <<<<<<<
1267 66FF
1268 66FF
1269 66FF
1270 66FF
1271 66FF
1272 66FF
1273 66FF
1274 66FF
1275 66FF
1276 66FF
1277 66FF
1278 66FF
1279 66FF
1280 66FF
1281 66FF
1282 66FF
1283 66FF CE FF FF PRNUS jsb =SYSJSB get one or two
1284 6702 FF FF def GIOR2N parameters
1285 6704 5E 0A E2 pabd r36,-r12 get our token
1286 6707 4C C8 C8 cnb r14,-ustok next using?
1287 670A F7 08 ifne no,
1288 670C 5E AB 5C ldb r36,-5CH load mainframe
1289 670F E4 pabd r36,-r12 put mainframe on
1290 6710 44 A9 FF FF gto PRNIN+ printW and go
1291 6714 endif
1292 6714 CE EB 63 jsb =PUSHUS push our token
1293 6717 CE FF FF jsb =SYSJSB
1294 671A FF FF def PRINT+ finish parsing
1295 671C 5E AB 36 ldb r36,-pendth put our own
1296 671F CE 10 64 jsb =LEARR- push and post scan
1297 6722 9E rtn
1298 6723
1299 6723
1300 6723

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 35
=====
>>>>>> POSIT, position the data pointer <<<<<<<
1306 6723
1307 6723
1308 6723
1309 6723
1310 6723
1311 6723
1312 6723
1313 6723
1314 6723
1315 6723
1316 6723
1317 6723
1318 6723
1319 6723
1320 6723
1321 6723
1322 6723
1323 6723
1324 6723
1325 6723 CE 32 67 TYPENT JSB =POSIT
1326 6726 43 1E A4 LDDB R03,R36
1327 6728 CE 06 SDB R03,-06H
1328 672B FF 04 IFRZ
1329 672D 5E A9 B0 7E LDM R36,=$7EBOH address of a SN in the rom
1330 6731 ENDF that called us (IOR01) - TYPE
1331 6731 9E RMW
1332 6732
1333 6732 7E 16 B5 0C POSIT LDDB R76,X26,LINEW get the lineW
1334 6736 00
1335 6737 CE FF FF JSB =FSEEK seek to the line
1336 673A F9 1E IFEN not there,
1337 673C CE 9F 67 FINDIT JSB =DATA1W get next data line
1338 673F F9 04 IFEN none,
1339 6741 CE FF FF JSB =ERR1+ error
1340 6744 22 BVI 34D no data
1341 6745 ENDF
1342 6745 50 1E B4 04 LDDB R20,X36,FOUR get the first token
1343 6749 00
1344 674A C8 0E CMB R20,=EOL1OK end of line?
1345 674C 16 05 IFEQ yes,
1346 674E CE FF FF HELL1 JSB =SKPLW skip this line
1347 6751 10 E9 JNP FINDIT find another
1348 6753 ENDF
1349 6753 5E 3E A3 SIM R36,R76 save pointer to line W
1350 6756 1A 04 00 ROM R36,=FOUR pointer to data item
1351 6759 9E RMW
1352 675A ENDF else,
1353 675A 52 93 CLM R02 clear reg for ptr
1354 675E 16 B4 08 00 LDDB R22,X26,DATAM load count
1355 6760 5E 3E A3 SIM R36,R76 save beginning of line

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 36
*****
>>>>>> POSIT, position the data pointer *****
1354 6763 C8 03 00 ROM R36,=THREE point to first tok
1355 6766 50 1E F0 POBP R20,=R36 get the token
1356 6769 C8 86 CMB R20,=DATA1K is this a data line?
1357 676B F7 04 IFNE no,
1358 676D 7E A3 SIM R76,R36 restore pointer
1359 676F F0 CB JMP FINDIT find the next line
1360 6771 ENDIF else,
1361 6771 CE 06 69 JSB =COUNT count down to data item
1362 6774 F7 E3 RZR not in this line?
1363 6776 40 80 57 B4 LDBD R0,=(TOKEN+3)
1364 6778 C8 0E CMB R0,=INSTR
1365 677C F7 09 JEQ EOLISTF
1366 677E C8 1' CMB R0,=INSTR
1367 6780 F7 05 JEQ EOLISTF
1368 6782 7E 1E A3 grotty SIM R76,R36 restore pointer
1369 6785 F0 E7 JMP HELLI find next line
1370 6787
1371 6787 52 C9 01 00 EOLISTF CMB R22,=ONE
1372 678B F6 F5 JNE grotty
1373 678D 5E 8D DCW R36
1374 678F 9E RTN
1375 6790

```

NOMAS
 NOT Manufacturer Supported
 Recipient agrees NOT to contact Manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 38
*****
>>>>>> DATA1N, get the next data line *****
1405 679F
1406 679F
1406 679F
1407 679F
1408 679F DATA1N: gets the next valid DATA line
1409 679F
1410 679F INPUT: R36 -- pointer to line #
1411 679F
1412 679F OUTPUT: R36 -- points to next valid data line
1413 679F R56 -- line #
1414 679F EMO: at the endline
1415 679F
1416 679F VOLATILES: R20
1417 679F
1418 679F EXT ROUTINES: SKPLN
1419 679F
1420 679F ERRORS: none
1421 679F
1421 679F
1422 679F
1423 679F 9D DATA1N CLE assume success
1424 679D 6E 1E A5 LDMO R56,R36 get the line #
1425 67A3 50 B4 03 00 LDBD R20,X36,THREE get the first token
1426 67A7 C8 86 CMB R20,=DATA1K is this DATA?
1427 67A9 F7 F3 REO yes, rtn
1428 67AB 9C ICE set error cond
1429 67AC 6E C9 99 A9 CMB R56,=ENDLIN is this end of file
1430 67B0 F7 EC REO rtn with E set
1431 67B2 CE FF FF JSB =SKPLN get next line
1432 67B5 F0 E8 JMP DATA1N continue
1433 67B7

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 37
*****
>>>>>> POSIT1, position data pointer in text *****
1377 6790
1378 6790
1378 6790
1379 6790
1380 6790 POSIT1: position data pointer in text file
1381 6790
1382 6790 INPUT: R26 -- 10file pointer
1383 6790 R36 -- pointer to beginning of text file
1384 6790
1385 6790 OUTPUT: R36 -- current data line
1386 6790 R76 -- line # of search
1387 6790 R66 -- current line #
1388 6790 EMO: R76NAGG
1389 6790 Z set: at the endline
1390 6790
1391 6790 VOLATILES: none
1392 6790
1393 6790 EXT ROUTINES: FSEK
1394 6790
1395 6790 ERRORS: none
1396 6790
1396 6790
1397 6790
1398 6790 7E 16 05 0C POSIT1 LDMO R76,X26,LINE# load data line
1398 6794 00
1399 6795 CE FF FF POSIT1 JSB =FSEK seek to it
1400 6798 76 1E A5 LDMO R66,R36 get the line #
1401 679B C9 99 A9 CMB R66,=ENLIN compare for valid
1402 679E 9E RTN continue
1403 679F

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 39
*****
>>>>>> TYPER, determine the file type *****
1435 67B7
1436 67B7
1436 67B7
1437 67B7
1438 67B7 TYPER: determines the type of the channel
1439 67B7
1440 67B7 INPUT: R30 -- ptr to dir entry of channel
1441 67B7
1442 67B7 OUTPUT: E=0: basic
1443 67B7 E=1: text
1444 67B7 E=2: unknown
1445 67B7
1446 67B7 VOLATILES: R20
1447 67B7
1448 67B7 EXT ROUTINES: none
1449 67B7
1450 67B7 ERRORS: none
1451 67B7
1451 67B7
1452 67B7
1453 67B7 9D TYPER CLE assume success
1454 67B8 50 18 B4 05 LDBD R20,X30,DR,THM get the type
1454 67BC 00
1455 67BD C8 42 CMB R20,=TYNBRAS is this basic?
1456 67BF F7 DB REO yes, rtn, E=0
1457 67C1 9C ICE E=1
1458 67C2 9E RTN for unknown data type
1459 67C3

```



```

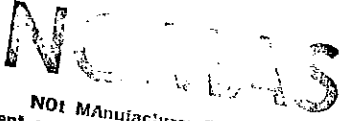
KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 40
*****
>>>>>> INEPIR, pre-increment data pointer <<<<<<<
*****
1461 67E3
1462 67E3
1463 67E3
1464 67E3
1465 67E3
1466 67E3
1467 67E3
1468 67E3
1469 67E3
1470 67E3
1471 67E3
1472 67E3
1473 67E3
1474 67E3
1475 67E3
1476 67E3
1477 67E3
1478 67E3
*****
1479 67E3 50 06 E3 INEPIR POND R20,-R6 get rtn addr
1480 67E6 65 16 B5 08 LOND R45,X26,DATAM get current ptr
1480 67E8 00
1481 67E8 88 JCB R45 incr ptr
1482 67E8 07 00 00 STAD R45,X26,DATAM store incr'd ptr
1483 67E8 8A DCB R45 dec ptr to original
1484 67D0 06 E5 PUND R45,+R6 save original
1485 67D2 50 E5 PUND R20,+R6 restore rtn addr
1486 67D4 9E RIN
1487 67D5

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 42
*****
>>>>>> CNEHLM, checks text file ptr for 9999 <<<<<<<
*****
1520 67EE
1521 67EE
1522 67EE
1523 67EE
1524 67EE
1525 67EE
1526 67EE
1527 67EE
1528 67EE
1529 67EE
1530 67EE
1531 67EE
1532 67EE
1533 67EE
1534 67EE
1535 67EE
*****
1536 67EE 9D CNEHLM CLE assume okay
1537 67F7 76 16 B5 0C LOND R66,X26,LINEM get data line #
1537 67F3 00
1538 67F4 99 BCD set BCD for compare
1539 67F5 19 99 99 CMH R66,*(9999C) compare to 9999
1540 67F8 98 BIN set BIN for others
1541 67F9 F6 F2 RAE return if okay
1542 67FB 1E FF FF JSB =ERR1+ error invalid
1543 67FE 5A BYT 90D line #
1544 67FF

```


 NOT Manufacturer Supported
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 41
*****
>>>>>> INCTEX, inc data pointer in text file <<<<<<<
*****
1489 67D5
1490 67D5
1491 67D5
1492 67D5
1493 67D5
1494 67D5
1495 67D5
1496 67D5
1497 67D5
1498 67D5
1499 67D5
1500 67D5
1501 67D5
1502 67D5
1503 67D5
1504 67D5
*****
1505 67D5 50 06 E3 INCTEX POND R20,-R6 get rtn addr
1506 67D8 65 16 B5 08 LOND R45,X26,DATAM get data line ptr
1506 67D0 00
1507 67D1 90 JSB R45 goto next line?
1508 67D1 F7 08 JFVZ yes,
1509 67D1 89 BCD bcd for inc
1510 67E1 66 89 ICB R46 increment data ptr
1511 67E3 67 0F 00 STAD R46,X26,LINEM store away
1512 67E6 88 DCB R46 decrement data ptr
1513 67E7 98 BIN bin for others
1514 67E8 ENJIF no,
1515 67E8 66 06 E5 PUND R46,+R6 save original
1516 67E8 50 E5 PUND R20,+R6 restore rtn addr
1517 67E0 9E RIN
1518 67EE

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 43
*****
>>>>>> LPR,LPR+, loop move data routines <<<<<<<
*****
1546 67FF
1547 67FF
1548 67FF
1549 67FF
1550 67FF
1551 67FF
1552 67FF
1553 67FF
1554 67FF
1555 67FF
1556 67FF
1557 67FF
1558 67FF
1559 67FF
1560 67FF
1561 67FF
1562 67FF
1563 67FF
*****
1564 67FF 50 08 96 LPR+ LDB R20,=PRITOK load prime token
1565 6802 18 E5 PUND R20,+R30 put into the file
1566 6804 LPR LOOP loop
1567 6804 5A 88 LDB R32 decr count
1568 6806 F4 E5 RCB rcb
1569 680E 50 1E 10 POND R20,+R34 get a source byte
1570 680B 18 E4 PUND R20,+R30 place it in line
1571 680D F0 F5 WHIP until none left
1572 680F

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG 44
*****
>>>>>> STRSCH, str search routine <<<<<<<<
*****
1574 680F
1575 680F
1576 680F
1577 680F
1578 680F
1579 680F
1580 680F
1581 680F
1582 680F
1583 680F
1584 680F
1585 680F
1586 680F
1587 680F
1588 680F
1589 680F
1590 680F
1591 680F
1592 680F
1593 6810 42 1E A5
1594 6813 C8 05
1595 6815 F7 D6
1596 6817 C8 06
1597 6819 F7 D2
1598 681B C8 96
1599 681D F7 CE
1600 681F 9C
1601 6820 9E
1602 6821
*****
STRSCH, str search routine
*****
STRSCH: check for string data item
*****
INPUT: R36 -- pointer to data item
R30 -- length of str data item
*****
OUTPUT: E=0: then a possible match
E=0: no chance
*****
VOLATILES: R2/3
*****
EXT ROUTINES: none
*****
ERRORS: none
*****
*****
STRSCH CLE
LDND R2,R36 assume failure
CMB R2,=QUOTK get the token
req R2,=QUOTK is this quoted
yes,
CMB R2,=BNGTK unquoted string
yes, check len
CMB R2,=PRITOK primed string
yes, check len
ace
RIN

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG 46
*****
>>>>>> CHKTOK, get next valid data item <<<<<<<
*****
1640 683D
1641 683D
1642 683D
1643 683D
1644 683D
1645 683D
1646 683D
1647 683D
1648 683D
1649 683D
1650 683D
1651 683D
1652 683D
1653 683D
1654 683D
1655 683D
1656 683D
1657 683D
1658 683D
1659 683E 50 1E A5
1660 6841 CE 4E 69
1661 6844 5E 1A C3
1662 6847 50 1E A5
1663 684A C8 0E
1664 684C F6 0F
1665 684E 60 92
1666 6850 5E 89
1667 6852 CE 9F 67
1668 6855 F8 E5
1669 6857 5E C8 04 00
1670 685B F0 02
1671 685D 60 88
1672 685F
1673 685F 9E
1674 6860
*****
CHKTOK, get next valid data item
*****
CHKTOK: gets the next valid data item
*****
INPUT: R36: ptr to current data item
*****
OUTPUT: R36: ptr to next data item
R55: data count in line
R56/57: current line N
*****
VOLATILES: R20/21
*****
EXT ROUTINES: CALC, DATALN
*****
ERRORS: none
*****
*****
CHKTOK CLE
LDND R20,R36 flag okay
JSB =CALC get token and str len
RDM R36,R32 calc data item size
LDND R20,R36 r36 ptr to next item
CMB R20,=EOLTK get the data type
IFEQ at end of line?
yes,
CLB R55 clear item counter
ICM R36 ptr to next line N
JSB =DATALN get next data line
RDM R36,=FOUR none left rtn
ELSE ptr to token
ICB R55 no,
inc item counter
ENDIF
RIN

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG 45
*****
>>>>>> NUMSCH, num search routine <<<<<<<
*****
1604 6821
1605 6821
1606 6821
1607 6821
1608 6821
1609 6821
1610 6821
1611 6821
1612 6821
1613 6821
1614 6821
1615 6821
1616 6821
1617 6821
1618 6821
1619 6821
1620 6821
1621 6821
1622 6821
1623 6821
1624 6821
1625 6821
1626 6826 C8 1A
1627 6828 F6 06
1628 682A 65 B5 01 00
1629 682E F0 08
1630 6830
1631 6830
1632 6830 C8 04
1633 6832 F6 EC
1634 6834 60 B5 01 00
1635 6838 90
1636 6839 50 C8 1A
1637 683C 9E
1638 683D
*****
NUMSCH, num search routine
*****
NUMSCH: check for numeric data item
*****
INPUT: R36 -- pointer to data item
*****
OUTPUT: E=0: numeric and R45/47: integer
or R40/47: real
Z set: integer
Z not set: real
*****
VOLATILES: R20
*****
EXT ROUTINES: none
*****
ERRORS: none
*****
*****
NUMSCH CLE
LDND R20,R36 assume failure
CMB R20,=INTK get the token
IFEQ integer?
yes,
LDND R45,X36,ONE get the value
JMP R1EHS clear flag
ENDIF
RDM R20,=REALK real?
RDM R40,X36,ONE no, rtn
CMB R20,=INTK get the real value
JMP R1EHS set success
RDM R20,=INTK set or not Z flag
RIN

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG 47
*****
>>>>>> SETCON, sets the parameters for search <<<<<<<
*****
1676 6860
1677 6860
1678 6860
1679 6860
1680 6860
1681 6860
1682 6860
1683 6860
1684 6860
1685 6860
1686 6860
1687 6860
1688 6860
1689 6860
1690 6860
1691 6860
1692 6860
1693 6860
1694 6860
1695 6863 67 90
1696 6865 F7 0A
1697 6867 4C 06 E3
1698 686A CE FF FF
1699 686D CE FF FF
1700 6870 5A
1701 6871
1702 6871 9E
1703 6872
1704 6872 CE FF FF
1705 6875 66 A9 99 A9
1706 6879 B3 38 B2
1707 687C 42 C8 18
1708 687F F6 09
1709 6881 CE 60 68
1710 6884 65 B3 3B B2
1711 6888 F0 04
1712 688A
1713 688A C8 10
1714 688C F6 0C
1715 688E CE 60 68
1716 6891 6E 25 A1
1717 6894 60 9C
1718 6896 16 A7 0B 00
1719 689A
1720 689A CE FF FF
1721 689D 90
1722 689E 9E
1723 689F
*****
SETCON, sets the parameters for search
*****
SETCON: sets start and finish for search
*****
INPUT: two, one or no numerics on the R12
stack
*****
OUTPUT: TEMP2 -- ending line
infile -- starting line and data item
*****
VOLATILES: R2/3, R46/47, R56/57
*****
EXT ROUTINES: ONEI, SETR12, ERR1, ONEI2?
*****
ERRORS: invalid line number
*****
*****
SETCON CLE
JSB =ONEI
ISB R47
IFN2
POND R02,=R06
JSB =SETR12
JSBN =ERR1
BVI 90D
ENDIF
RIN
*****
SETCON JSB =ONEI2? get N of bytes on R12
LDN R46,=ENDLIN default is endlin
SIND R46,=TEMP2 store it away
CMB R2,=24D 3 numerics?
IFEQ yes,
JSB =SETR12
SIND R45,=TEMP2 store end line in temp2
JMP H112 process the start line
ENDIF
CMB R2,=16D 2 numerics?
IFEQ yes,
JSB =SETR12 get the start line
LDN R56,R45 load up line N
CLB R55 data item is 0
SIND R55,X26,DATRN store in infile
ENDIF
JSB =SETR12 reset R12
CLF clear for no errors
RIN

```

```

KARMA 08/31/82>=====
ITEM LOC OBJECT CODE SRC=RV&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG 48
=====
>>>>>> UPPER, corrects the iofile data <<<<<<<
1725 689F *****
1726 689F A
1727 689F A
1728 689F A
1729 689F A
1730 689F A
1731 689F A
1732 689F A
1733 689F A
1734 689F A
1735 689F A
1736 689F A
1737 689F A
1738 689F A
1739 689F A
1740 689F A
1741 689F A
1742 689F 6E 3E D9 UPPER CMND R56,R76 compare loc vs info
1743 689F 17 04 IFNE not the same,
1744 689F 6D 93 CLN R55 clear item
1745 689F 6E A5 LDND R56,R76 get line N
1746 689F ENDIF else same
1747 689F 6D 16 B7 08 STND R55,X26,DATAN store in iofile line
1748 689F 00
1749 689F RTN

```

NOMAS

NOT MANUFACTURER SUPPORTED
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82>=====
ITEM LOC OBJECT CODE SRC=RV&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG 50
=====
>>>>>> CHAN, CHANI, and CHANI+ <<<<<<<
1800 68E3
1801 68E3 68 0A E3 CHANI+ POND R50,-R12 get the number
1802 68E6 06 E5 czero pumd rN,-r6
1803 68E8 9A sad
1804 68E9 CE AE 68 jsb =chan
1805 68EC 9F pad
1806 68ED 06 E3 pumd rN,-r6
1807 68EF 42 90 tab r2
1808 68F1 9E RTN
1809 68F2
1810 68F2
1811 68F2 A POND R50,-R6 save from chan
1812 68F2 A JSB =CHAN open the channel
1813 68F2 A DRP 50 set drp for common
1814 68F2 A *ZERO IFZR if all okay
1815 68F2 A POND R0,-R6 get the parameter
1816 68F2 A CLM R2 set Z for everybody
1817 68F2 A ELSE if not okay
1818 68F2 A POND R0,-R6 get the parameter
1819 68F2 A TSM R6 make sure Z not set
1820 68F2 A ENDIF
1821 68F2 A RTN return
1822 68F2

```

```

KARMA 08/31/82>=====
ITEM LOC OBJECT CODE SRC=RV&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG 49
=====
>>>>>> CHAN, CHANI, and CHANI+ <<<<<<<
1751 689E *****
1752 689E A
1753 689E A
1754 689E A
1755 689E A
1756 689E A
1757 689E A
1758 689E A
1759 689E A
1760 689E A
1761 689E A
1762 689E A
1763 689E A
1764 689E A
1765 689E A
1766 689E A
1767 689E A
1768 689E A
1769 689E A
1770 689E A
1771 689E CE 3D 69 CHAN JSB =STARTR open iofile
1772 689E FB FA REN not there
1773 689E 50 1E 04 0E lddw r20,r36,typeN get the type byte
1774 689E F6 1A not our file type
1775 689E 40 04 02 00 LDND R00,R36,LENN get the len of io line
1776 689E C8 0C CNB R00,-RYLENN compare to sys data file
1777 689E F6 12 jnz badchn not in the system
1778 689E 60 05 03 00 LDND R40,R36,NAMEW get the channel name
1779 689E 36 A1 LDM R26,R36 save iofile ptr
1780 689E CE FF FF JSB =FLOPEN open channel
1781 689E F9 04 IFNE not there
1782 689E CE FF FF JSB =ERRR+ report error
1783 689E 3C BVI 62D file not found
1784 689E ENDIF
1785 689E 42 97 CLB R2 clear for all okay
1786 689E 9E RTN
1787 689E
1788 689E 42 A8 01 badchn lddw r2,-1
1789 689E 9E rtn
1790 689E
1791 689E CE FF FF CHANI JSB =GETADR get the str loc
1792 689E DRP 134 assert R34
1793 689E B3 49 B2 STND R34,-KLUDGE store str loc in kludge
1794 689E 5A 0A E3 POND R32,-R12 get the str len
1795 689E A POND R32,-R6 save the str len
1796 689E A JSB =CHAN open the channel
1797 689E A DRP R32 set drp for common
1798 689E F0 03 JNP CZERO jump to common out
1799 689E

```

```

KARMA 08/31/82>=====
ITEM LOC OBJECT CODE SRC=RV&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG 51
=====
>>>>>> LINEN <<<<<<<
1824 68F2 *****
1825 68F2 A
1826 68F2 A
1827 68F2 A
1828 68F2 A
1829 68F2 A
1830 68F2 A
1831 68F2 A
1832 68F2 A
1833 68F2 A
1834 68F2 A
1835 68F2 A
1836 68F2 A
1837 68F2 A
1838 68F2 A
1839 68F2 A
1840 68F2 CE FF FF LINEN JSB =ONEI get the integer
1841 68F5 67 90 TSB R47 two many digits
1842 68F7 F7 03 IFNZ yes,
1843 68F9 9D 9C STB set E
1844 68FB 9E RTN return
1845 68FC ENDIF
1846 68FE 50 1E A3 STM R20,R36 restore iofile ptr
1847 68FF 7E 25 A1 LDM R76,R45 load for FSEEK
1848 6902 CE FF FF JSB =FSEEK seek to R76
1849 6905 9E RTN
1850 6906

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 52
*****
COUNT, data item counter
*****
1852 6906
1853 6906
1854 6906
1855 6906
1856 6906
1857 6906
1858 6906
1859 6906
1860 6906
1861 6906
1862 6906
1863 6906
1864 6906
1865 6906
1866 6906
1867 6906
1868 6906
1869 6906
1870 6906
1871 6906
1872 6906
1873 6906
1874 6906
1875 6906 50 1E A5 COUNT, LDND R20,R36 get the data type token
1876 6909 C8 0E CMB R20,-EOLTOX compare to end of line
1877 6908 F6 03 IFEQ yes,
1878 690D 5E 89 JCN R36 point R36 to next line
1879 690F 9E RTN
1880 6910
1881 6910 52 91 TSN R22 test the count
1882 6912 F7 F8 RZR if we entered at zero
1883 6914 88 DCM R22 derr the count
1884 6915 F7 F8 RZR we are at the right spot
1885 6917 CE 4E 69 JSB =CALC calculate length of data item
1886 691A 5E 1A C3 ADM R36,R32 point to next data item
1887 691D F0 E7 JMP COUNT, loop back
1888 691F

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 54
*****
STARTR, open the logfile
*****
1927 6930
1928 6930
1929 6930
1930 6930
1931 6930
1932 6930
1933 6930
1934 6930
1935 6930
1936 6930
1937 6930
1938 6930
1939 6930
1940 6930
1941 6930
1942 6930
1943 6930
1944 6930 CE FF FF STARTR JSB =107 open logfile
1945 6940 F8 F8 REN not there if EMO
1946 6942 50 1E A1 LDM R20,R36 save logfile ptr
1947 6945 CE F2 68 JSB =LJNEW, find the channel line
1948 6948 F9 F2 REZ found it if E=0
1949 694A CE FF FF JSB =ERR1+ error,
1950 694D 2D BYI 450 missing ASSIGMW
1951 694E

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 53
*****
COUNT, count data items in line
*****
1890 691F
1891 691F
1892 691F
1893 691F
1894 691F
1895 691F
1896 691F
1897 691F
1898 691F
1899 691F
1900 691F
1901 691F
1902 691F
1903 691F
1904 691F
1905 691F
1906 691F
1907 691F
1908 691F
1909 691F
1910 691F
1911 691F 9D COUNT, CLE
1912 6920 5E CB 04 00 ADM R36,-FOUR assume success
1913 6924 50 1E A5 LDND R20,R36 R36 to first token
1914 6927 8 0E CMB R20,-EOLTOX get the data type tk
1915 6929 F7 0F JEQ ICM36 compare to end of line
1916 6928 LOOP update pointer and rtn
1917 6928 CE 4E 69 JSB =CALC else,
1918 692E 5E 1A C3 ADM R36,R32 calculate the item size
1919 6931 52 89 JCN R22 R36 ptr to next item
1920 6933 50 1E A5 LDND R20,R36 increment count
1921 6936 CE 0E CMB R20,-EOLTOX get next type tk
1922 6938 F6 F1 JNE exit if eol
1923 693A 5E 89 ICM36 JCN R36 compare to end of line
1924 693C 9E RTN ptr to next line #
1925 693D

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 55
*****
CALC, calculates the size of the data item
*****
1953 694E
1954 694E
1955 694E
1956 694E
1957 694E
1958 694E
1959 694E
1960 694E
1961 694E
1962 694E
1963 694E
1964 694E
1965 694E
1966 694E
1967 694E
1968 694E
1969 694E
1970 694E
1971 694E 50 CB 04 CALC CMB R20,-REALIK compare to real
1972 6951 F6 05 IFEQ yes,
1973 6953 5A A9 09 00 LDM R32,-NINE size is nine
1974 6957 9E RTN
1975 6958
1976 6958 C8 1A CMB R20,-INITK compare to integer
1977 695A F6 05 IFEQ yes,
1978 695C 5A A9 04 00 LDM R32,-FOUR size is four
1979 6960 9E RTN
1980 6961
1981 6961 5A 93 CLN R32 else, string
1982 6963 11 A0 LDB R32,R21 load str size
1983 6965 89 JCN R32 incr for token
1984 6966 89 JCN R32 incr for size byte
1985 6967 9E RTN
1986 6968

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&X103 12/13/1983 11:38 AM PG 56
*****
>>>>>> POINTIR, get data pointer for search <<<<<<<<
1988 6968 *****
1989 6968 *
1989 6968 *
1990 6968 *
1991 6968 * POINTIR: returns the correct data pointer in
1992 6968 * R55/57
1993 6968 *
1994 6968 * INPUT: R26: xofile pointer
1995 6968 * R76: ptr to current line # in channel
1996 6968 *
1997 6968 * OUTPUT: R55/57: item and line # of current data
1998 6968 *
1999 6968 * VOLATILES: R55/57
2000 6968 *
2001 6968 * EXT ROUTINES: none
2002 6968 *
2003 6968 * ERRORS: none
2004 6968 *
2004 6968 *****
2005 6968
2006 6968 6D 16 B5 0B POINTIR LDND R55,X26,DATAN get the said ptr
2006 6968 00
2007 6968 6E 3E 09 CHND R56,R76 is it correct
2008 6970 F7 06 IFNE no,
2009 6972 6D 93 CLM R55 clear item count
2010 6974 6E A5 LDND R56,R76 get the line #
2011 6976 F0 02 ELSE yes,
2012 6978 6D BA DCB R55 dec item count
2013 697A
2014 697B ENDDIF
2015 697D RTN

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&X103 12/13/1983 11:38 AM PG 58
*****
>>>>>> functions; entry points and externals <<<<<<<<
2044 698D
2045 698D

```

NOMAS

Not Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&X103 12/13/1983 11:38 AM PG 57
*****
>>>>>> BEGIN, beginning for SEARCH <<<<<<<<
2017 697B *****
2018 697B *
2018 697B *
2019 697B *
2020 697B * BEGIN: opens the channel by manipulating the
2021 697B * R12 stack
2022 697B *
2023 697B * INPUT: channel # on the R12 stack
2024 697B *
2025 697B * OUTPUT: channel file open
2026 697B *
2027 697B * VOLATILES: R34/35, R40/47
2028 697B *
2029 697B * EXT ROUTINES: CHAN, SETR12
2030 697B *
2031 697B * ERRORS: none
2032 697B *
2032 697B *****
2033 697B
2034 697B SC B1 57 82 BEGIN LDND R34,=FOS get r34 to channel #
2035 697F 60 1C A5 LDND R40,R34 channel # to r40
2036 6982 0A E5 PUND R40,R12 channel # to R12 stack
2037 6984 CE AE 68 JSB =CHAN open up the channel
2038 6987 F9 03 IFNE E means trouble
2039 6989 CE FF FF JSB =SETR12 reset R12 for exit
2040 698C
2041 698E ENDDIF
2042 698D RTN

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&X103 12/13/1983 11:38 AM PG 59
*****
>>>>>> FILE#, return channel name <<<<<<<<
2047 698D *****
2048 698D *
2048 698D *
2049 698D *
2050 698D * FILE#: return the name of the channel
2051 698D *
2052 698D * INPUT: channel number on the R12 stack
2053 698D *
2054 698D * OUTPUT: channel name on the R12 stack
2055 698D *
2056 698D * VOLATILES: R54/57, R32/33, R36/37
2057 698D *
2058 698D * EXT ROUTINES: STARTR, PUTADR
2059 698D *
2060 698D * ERRORS: none
2061 698D *
2062 698D *
2063 698D * BYT 20,56
2064 698D * FILE#, JSB =STARTR open xofile
2065 698D * IFNE not there
2066 698D * CLM R54 clear for
2067 698D * PUND R54,R12 null string and push
2068 698D * JSB =CLRERR clear the error
2069 698D * ELSE there,
2070 698D * LDN R32,=EIGHT str len is eight
2071 698D * PUND R32,R12 put on the stack
2072 698D * RDN R36,=NAMEW point R36 to the name
2073 698D * JSB =PUTADR put rel to R12
2074 698D *
2075 698D * ENDDIF
2075 698D *
2075 698D * RTN
2075 698D *****
2076 698D

```

```

KARMA 08/31/82 *****
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG 60
*****
INDEX., returns the absolute loc of data ptr <-----
>>>>>>
2078 698D *****
2079 698D *****
2079 698D *****
2080 698D *****
2081 698D *****
2082 698D *****
2083 698D *****
2084 698D *****
2085 698D *****
2086 698D *****
2087 698D *****
2088 698D *****
2089 698D *****
2090 698D *****
2091 698D *****
2092 698D *****
2093 698D *****
2094 698D *****
2095 698D *****
2095 698D *****
2096 698D *****
2097 698D 10 2D BYT 20,55
2098 698F CE AE 68 INDEX. JSB =CHAM open channel
2099 6992 F8 F8 REN errors
2100 6994 F7 06 IFNZ file not is
2101 6996 CE FF FF filhan jsb =HARDJ
2102 6999 4D val V.NEWM
2103 699A 3F byt 63d
2104 699B 9C rtn
2105 699C *****
2106 699C CE 87 67 JSB =IYPER get the file type
2107 699F F8 3B JEM TEXTIT text, go
2108 69A1 54 1E A1 LDH R24,R36 save file ptr from posit
2109 69A4 CE 32 67 JSB =POSIT position the data ptr
2110 69A7 F9 04 IFEN no data
2111 69A9 CE FF FF JSB =ERROR+ report the error
2112 69AC 22 BYT 34D no data
2113 69AD *****
2114 69AD 54 1E A3 STN R24,R36 restore the file ptr
2115 69B0 CE 9F 67 JSB =DATAIN get the first data line
2116 69B3 52 93 CLN R22 clear the counter
2117 69B5 *****
2118 69B5 6E 3E 09 CMND R56,R76 loop,
2119 69B8 F7 0A JEQ LISZ2 compare to data ptr
2120 69BA CE 1F 69 JSB =COUNT+ we are here
2121 69BD F8 DC REN bad data type not here, no count the items
2122 69BF CE 9F 67 JSB =DATAIN get next data line
2123 69C2 F0 F1 *****
2124 69C4 5E 93 LISZ2 CLN R36 clear for output
2125 69C6 76 16 85 0C LDND R66,R26,LINEM get the data line N
2126 69C8 00 *****
2126 69C8 2F C1 CMN R66,R56 did we find the line?

```

```

KARMA 08/31/82 *****
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG 62
*****
ITEM., return the relative position of data pt <-----
>>>>>>
2153 6A06 *****
2154 6A06 *****
2154 6A06 *****
2155 6A06 *****
2156 6A06 *****
2157 6A06 *****
2158 6A06 *****
2159 6A06 *****
2160 6A06 *****
2161 6A06 *****
2162 6A06 *****
2163 6A06 *****
2164 6A06 *****
2165 6A06 *****
2166 6A06 *****
2167 6A06 *****
2168 6A06 *****
2168 6A06 *****
2169 6A06 *****
2170 6A06 10 2D BYT 20,55
2171 6A08 CE RE 68 ITEM. JSB =CHAM open channel
2172 6A08 F8 CE REN not there!
2173 6A0D 5E 93 CLN R36 clear for integer
2174 6A0F 16 B4 0B 00 LDND R36,R26,DATAW get the rel loc
2175 6A13 89 ICM R36 pre-inc pointer
2176 6A14 F0 C2 JMP LISZ4 output
2177 6A16 *****

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82 *****
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG 61
*****
INDEX., returns the absolute loc of data ptr <-----
>>>>>>
2127 69CD F6 05 IFEQ found the line
2128 69CF 5E 16 B4 0B LDND R36,R26,DATAW get the rel data ptr
2128 69D0 00 *****
2129 69D4 *****
2130 69D4 5E 89 ICM R36 increment for system
2131 69D6 12 C3 RDN R36,R22 add in the data count
2132 69D8 CE FF FF LISZ4 JSB =PUNITG output the integer
2133 69DB 9E RTN *****
2134 69DC 52 93 TEXTIT CLN R22 clear the counter
2135 69DE 6E 1E A5 LDND R56,R36 get the line N
2136 69E1 76 16 85 0C LDND R66,R26,LINEM get the data line N
2136 69E5 00 *****
2137 69E6 *****
2138 69E6 52 89 ICM R22 loop
2139 69E8 6E 3E C1 CMN R56,R66 increment the count
2140 69EB FB 0B JCY LISZ3 compare the line N
2141 69ED CE FF FF JSB =SKPLH we are here or past
2142 69F0 6E 1E A5 LDND R56,R36 not found, get next line
2143 69F3 C9 99 A9 CMN R56,-ENDLIN get the line N
2144 69F6 F6 EE *****
2145 69F8 5E 12 A1 LISZ3 LDH R36,R22 end of file
2146 69FB 42 16 B4 0B LDND R2,R26,DATAW no, then loop
2146 69FF 00 load in the count
2147 6A00 F7 02 IFNZ get the data ptr
2148 6A02 5E 89 ICM R36 if non-zero,
2149 6A04 ***** then increment count
2150 6A04 F0 D2 JMP LISZ4 else, count okay
2151 6A06 ***** output the count

```

```

KARMA 08/31/82 *****
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG 63
*****
IYP., the type function rtn data type <-----
>>>>>>
2179 6A16 *****
2180 6A16 *****
2180 6A16 *****
2181 6A16 *****
2182 6A16 *****
2183 6A16 *****
2184 6A16 *****
2185 6A16 *****
2186 6A16 *****
2187 6A16 *****
2188 6A16 *****
2189 6A16 *****
2190 6A16 *****
2191 6A16 *****
2192 6A16 *****
2193 6A16 *****
2194 6A16 *****
2195 6A16 *****
2196 6A16 *****
2197 6A16 *****
2198 6A16 *****
2199 6A16 *****
2200 6A16 *****
2201 6A16 *****
2202 6A16 *****
2203 6A16 *****
2204 6A16 *****
2205 6A16 *****
2206 6A16 *****
2207 6A16 *****
2208 6A16 *****
2209 6A16 *****
2210 6A16 *****
2211 6A16 *****
2212 6A16 *****
2213 6A16 *****
2214 6A16 *****
2215 6A16 *****
2216 6A16 *****
2217 6A16 *****
2218 6A16 *****
2219 6A16 *****
2220 6A16 *****
2221 6A16 *****
2222 6A16 *****
2223 6A16 *****
2224 6A16 *****
2225 6A16 *****
2226 6A16 *****
2227 6A16 *****
2228 6A16 *****
2229 6A16 *****
2230 6A16 *****
IYP. returns a numeric value that corresponds to the current data type
INPUT: channel number on the R12 stack
OUTPUT: type value on the R12 stack
1 -- numeric
2 -- string
0 -- end of file
VOLATILES: R20, R46/47, R60
EXT ROUTINES: CHAM, IYPER, INCPTR, POSIT, INCLEX, POSITIT
ERRORS: none
BYT 20,55
JSB =CHAM open channel
REN not there
IFNZ not in system
GTO FILHAN for i/o rom
ENDIF
JSB =IYPER get file type
JEM TEXTTY go do text
JSB =INCPTR increment pointer
POND R65,-R6 retrieve old ptr
JSB =POSIT position data ptr
STND R65,R26,DATAW restore old position
CLN R60 clear for output
IFEN no data
JSB =CLERR clear the errors
*MOZ1 POND R60,R12 push to output
*RTN *****
ENDIF
LDND R20,R36 get the type token
CMN R20,-IMITK integer?
IFED yes,
*MOZ2 LDH R67,-IYC load 1 for numeric
*JMP MOZ1 output
ENDIF
CMN R20,-QUOTK a string
JHC MOZ2 no, a real, output
LDH R67,-ZOC yes, load 2
JMP MOZ1 output
*TEXTTY JSB =INCLEX increment data ptr
*POND R46,-R6 retrieve old ptr
*JSB =POSIT position data ptr

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 64
=====
>>>>>> TYP., the type function rtn data type <<<<<<<
2231 6A16 * IFZR end of file *
2232 6A16 * CLM R60 clear for eof *
2233 6A16 * ELSE *
2234 6A16 * CLM R60 must be a str *
2235 6A16 * LDB R67,=20C if user in a text file *
2236 6A16 * ENDIR *
2237 6A16 * STD R46,X26,LINE# restore old position *
2238 6A16 * JNP M021 output value *
2238 6A16 *
2239 6A16 *

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 66
=====
>>>>>> LINLEN, # of data items in line <<<<<<<
2291 6A52 CE 1F 69 JSB =COUNT+ count data items
2292 6A55 SE 12 A1 LDM R36,R22 load for puntg
2293 6A58 CE FF FF OUTNUM JSB =PUNTING output count to R12
2294 6A5B 9E RTN
2295 6A5C
2296 6A5C CE 95 67 TXTLEN JSB =POSIT+ fseek to r76
2297 6A5F F8 E3 JEN CLURP not there output 0
2298 6A61 SE 1E 05 02 LDM R36,X36,100
2298 6A65 00
2299 6A66 SF 92 CLB R37
2300 6A68 FO EE JNP OUTNUM output the integer
2301 6A6A

```

NOMAS

Not Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 65
=====
>>>>>> LINLEN, # of data items in line <<<<<<<
2241 6A16 *
2242 6A16 *
2242 6A16 *
2243 6A16 *
2244 6A16 * LINLEN: returns the # of data items in the
2245 6A16 * specified line
2246 6A16 *
2247 6A16 * INPUT: channel # and line # on the R12 stack
2248 6A16 *
2249 6A16 * OUTPUT: # of data items in line #
2250 6A16 *
2251 6A16 * VOLATILES: R45/47, R76/77, R22/23, R36/37
2252 6A16 *
2253 6A16 * EXT ROUTINES: ONEI, CHAN, TYPER, FSEEK, DATALA
2254 6A16 * COUNT+, PUNTING
2255 6A16 *
2256 6A16 * ERRORS: none
2257 6A16 *
2257 6A16 *
2258 6A16 *
2258 6A16 *
2259 6A16 20 2D BYI 40,55
2260 6A18 CE FF FF LINLM JSB =ONEI get the line #
2261 6A1B 67 50 TSB R47 valid?
2262 6A1D F7 06 IFM2 no,
2263 6A1F CE FF FF JSB =ERROR report error
2264 6A22 5A BYI 90D invalid line #
2265 6A23 FO 1F JNP CLURP push out #
2266 6A25 ENDIR
2267 6A25 7E 25 A1 LDM R76,R45 load line # for fseek
2268 6A28 06 E5 puns r76,r76
2269 6A2A
2270 6A2A
2271 6A2A
2272 6A2A
2273 6A2A CE AE 68 JSB =CHAN open the channel
2274 6A2D 7E 06 E3 POND R76,-R6
2275 6A30 F8 A9 MEM not there
2276 6A32 42 90 TSB R2
2277 6A34 F7 04 IFM2 not in system
2278 6A36 44 A9 95 69 goto failhan
2279 6A3A ENDIR
2280 6A3A CE 07 67 JSB =TYPER type the file
2281 6A3D F8 10 JEN INTLEN go text
2282 6A3F CE FF FF JSB =FSEEK seek to line #
2283 6A42 F9 04 IFEN not there
2284 6A44 SE 93 CLURP CLM R36 clear for output
2285 6A46 FO 10 JNP OUTNUM output zero
2286 6A48 ENDIR
2287 6A48 CE 9F 67 JSB =DATALA check if data line
2288 6A4B 7E 2E C1 CMN R76,R56 compare input to data
2289 6A4E F6 14 JNE CLURP not the same output 0
2290 6A50 52 93 CLM R22 clear counter

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 67
=====
>>>>>> CLASH., clear the iofile <<<<<<<
2304 6A6A *
2305 6A6A *
2306 6A6A * CLASH.: clears all assignments in the iofile
2307 6A6A * by deleting all lines except line 0
2308 6A6A *
2309 6A6A * INPUT: none
2310 6A6A *
2311 6A6A * OUTPUT: empty iofse
2312 6A6A *
2313 6A6A * VOLATILES: R30/31, R32/33, R36/37, R76/77
2314 6A6A *
2315 6A6A * EXT ROUTINES: IO?, FSEEK, DELETE
2316 6A6A *
2317 6A6A * ERRORS: none
2318 6A6A *
2318 6A6A *
2319 6A6A *
2320 6A6A R1 BYI 241
2321 6A6B CE FF FF CLASH JSB =IO? open the iofile
2322 6A6E F8 EB REN not there!
2323 6A70 SE CB OF 00 ADH R36,=8000FH R36 ptr to first line
2324 6A74 * after line 0
2325 6A74 18 A3 SIM R36,R30 store for delete
2326 6A76 7E A9 99 A9 LDM R76,=ENDLIN load in endlin
2327 6A7A CE FF FF JSB =FSEEK seek to endlin
2328 6A7D 5A 1E A1 LDM R32,R36 r32 ptr to endlin
2329 6A80 18 C5 SBM R32,R30 r32 now # to delete
2330 6A82 CE FF FF JSB =DELETE delete from first line
2331 6A85 * to the endlin
2332 6A85 9E RTN
2333 6A86

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA103 OBJ=RYX103 12/13/1983 11:38 AM PG 68
*****
DELM.: delete the next data item in W *****
2335 GR86
2336 GR86
2337 GR86
2338 GR86
2339 GR86
2340 GR86
2341 GR86
2342 GR86
2343 GR86
2344 GR86
2345 GR86
2346 GR86
2347 GR86
2348 GR86
2349 GR86
2350 GR86
2351 GR86
2352 GR86
2353 GR86
2354 GR86
2355 GR87 CE FF FF
2356 GR88 A2 CE 10
2357 GR8D F6 07
2358 GR8F CE FF FF
2359 GR92 66 89
2360 GR94 F0 04
2361 GR96 66 93
2362 GR98 89
2363 GR99 89
2364 GR9A
2365 GR9B 06 E5
2366 GR9C CE RE 68
2367 GR9F 66 06 E3
2368 GR92 F8 E1
2369 GR94 42 90
2370 GR96 F7 04
2371 GR98 44 A9 95 69
2372 GR9C
2373 GR9C 66 0A E5
2374 GR9F CE B7 67
2375 GR92 F8 45
2376 GR94 CE E3 67
2377 GR97 5E 0A E5
2378 GR9A 5E 0A E3
2379 GR9D 42 E3
2380 GR9F 88
2381 GR9C F7 2F
2382 GR9C E5
2383 GR93 5E E5
2384 GR95 CE 32 67

DELM.: deletes the next data item in the channel
INPUT: channel W on the R12 stack
OUTPUT: data item deleted
VOLATILES: R20/21, R24/25, R30/31, R32/33, R45/47
EXT ROUTINES: CHAN, TYPER, INCPIR, POSIT, DELETE
INCTEX, POSITT, ERROR
ERRORS: invalid line number if in text data file
and an attempt is made to delete the internal
and line
*****
BYI 241
JOB *DNR12? open channel
cnb r2,*10h
afeq
JOB *DNR8
icm r46
else
cin r46
icm r46
icm r46
endif
pund r46,*r6
JSB *CHAN open channel
POMD R46,-R6
REN
TSB R2
IFWZ not in system
goto filhan
ENDIF
pund r46,*r12
JSB *TYPER get file type
JEN TEXTDE do text
JSB *INCPIR pre-incr pointer
pund r36,*r12
pund r36,-r12
dcw r2
jzr hll1
pund r2,*r12
pund r36,*r12
JSB *POSIT position the pointer

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA103 OBJ=RYX103 12/13/1983 11:38 AM PG 69
*****
DELM.: delete the next data item in W *****
2385 GR98 F8 27
2386 GR9A 50 IE A5
2387 GR9C CE 4E 69
2388 GR9D 58 IE A1
2389 GR93 54 93
2390 GR95 3E B4 02 00
2391 GR99 1A 15
2392 GR9B C8 02
2393 GR9D F6 09
2394 GR9F 58 3E A1
2395 GR9E 5A CB 05 00
2396 GR9E F0 04
2397 GR9E 3E B6 02 00
2398 GR9C
2399 GR9C CE FF FF
2400 GR9F F0 C9
2401 GR9F 65 06 E3
2402 GR9F 16 B7 0B 00
2403 GR9F 9E
2404 GR9F
2405 GR9F CE 05 67
2406 GR9F 42 0A E3
2407 GR9F 88
2408 GR9D F7 1C
2409 GR9D E5
2410 GR93 CE 90 67
2411 GR96 F6 06
2412 GR9B CE FF FF
2413 GR9B 5A
2414 GR9C F0 10
2415 GR9D 5A 93
2416 GR9D 1E B4 02 00
2417 GR94 CB 03 00
2418 GR97 58 A1
2419 GR99 CE FF FF
2420 GR9C F0 DE
2421 GR9F
2422 GR9F 66 06 E3
2423 GR9F 16 B7 0C 00
2424 GR9E 9E
2425 GR9E

JEN HLL1 no more data, out
LDND R20,R36 get the data type
JSB *CALC calc the data len
LDND R30,R36 position for delete
CLM R24 clear for len
LDND R24,X76,TWO get the line len
SBM R24,R32 calc new length
CNB R24,-TWO if two then null line
EFD so.
LDND R30,R36 point to line W
ADND R32,-FIVE delete five more bytes
ELSE else.
STBD R24,X76,TWO insert new len
ENDIF
JSB *DELETE delete the approp bytes
jmp contin
POMD R45,-R6 get the old pointer
STND R45,X26,DATAN post-dec pointer
RIN
LDND R30,R36 pre-inc pointer
pund r2,-r12 pre-inc pointer
dcw r2
jzr hll1
pund r2,*r12
JSB *POSIT position the pointer
IFR at the endlin?
JSB *ERROR yes, then error
BYI 90D invalid line W
ELSE no.
CLM R32
LDND R32,X36,TWO get the line len
ADND R32,-THREE add three
LDND R30,R36 position ptr
JSB *DELETE delete R32 bytes
jmp textlp
ENDIF
POMD R46,-R6 get the old pointer
STND R46,X26,LINEM post-dec pointer
RIN

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA103 OBJ=RYX103 12/13/1983 11:38 AM PG 70
*****
INSERT W runtimes, INSTR & INMUN *****
2427 GR9E
2428 GR9E
2429 GR9E
2430 GR9E
2431 GR9E
2432 GR9E
2433 GR9E
2434 GR9E
2435 GR9E
2436 GR9E
2437 GR9E
2438 GR9E
2439 GR9E
2440 GR9E
2441 GR9E
2442 GR9E
2443 GR9E
2444 GR9E
2445 GR9E 24
2446 GR9E CE D8 68
2447 GR9A F8 F9
2448 GR9C F7 04
2449 GR9C 44 A9 95 69
2450 GR9C
2451 GR9C CE B7 67
2452 GR95 F8 50
2453 GR97 CE C3 67
2454 GR9A 5A 06 E5
2455 GR9C CE 32 67
2456 GR9D F9 07
2457 GR9E 5A 06 E3
2458 GR9A F0 A5
2459 GR9D F0 D5
2460 GR9A
2461 GR9A 5A 06 E3
2462 GR9C E5
2463 GR9D 11 A2
2464 GR9F CB 02 00
2465 GR9E 52 93
2466 GR9A 3E B4 02 00
2467 GR9B 1A C3
2468 GR9A 53 90
2469 GR9C F7 09
2470 GR9E 5A 06 E3
2471 GR9C CE FF FF
2472 GR9A 43
2473 GR9E F0 1C
2474 GR97 58 IE A1
2475 GR9A CE FF FF
2476 GR9D F9 05

INSTR JSB *CHAN open channel
REN not there!
IFWZ not in system
goto filhan
ENDIF
JSB *TYPER get file type
JEN TEXTSI go if text
JSB *INCPIR pre-inc pointer
POMD R32,*R6 save string
JSB *POSIT position the pointer
IFEN no more data
POMD R32,-R6 clean up stack
JMP HLL1 reset and out
JMP HLL0
ENDIF
POMD R32,-R6 get str len
POMD R32,*R6 save the len
STB R32,R21
ADND R32,-TWO add two for token and len
CLM R22 clear for line len
LDND R22,X76,TWO get the line len
ADND R22,R32 add the inserted size
TSB R23 too big?
IFWZ yes.
POMD R32,-R6 clean up stack
JSB *ERROR report error
BYI 67D line too long
ELSE no.
LDND R30,R36 pointer for alloc
JSB *ALLOD open up space for item
IFEN not enough mem

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RYA103 OBJ=RYX103 12/13/1983 11:38 AM PG 71
*****
INSERT W runtimes, INSTR & INMUN *****
2477 GR9F 5A 06 E3
2478 GR9D F0 0F
2479 GR9A 52 3E B6 02
2480 GR9D 00
2481 GR9A 5A 06 E3
2482 GR9C 5E B1 49 82
2483 GR9C CE FF 67
2484 GR9B
2485 GR9B F0 C0
2486 GR9E F0 C0
2487 GR9B
2488 GR9B CE EE 67
2489 GR9A F8 99
2490 GR9C 5E 14 A3
2491 GR9B CE 05 67
2492 GR9C CE 00 67
2493 GR9E F8 28
2494 GR9F 65 90
2495 GR99 F7 06
2496 GR9B CE FF FF
2497 GR9E 5A
2498 GR9F F0 1C
2499 GR9A 7E 90
2500 GR9C F7 F6
2501 GR9E 99
2502 GR9E 88
2503 GR9A 98
2504 GR9B 16 B7 0C 00
2505 GR9C 5E 14 A1
2506 GR9F CE 90 67
2507 GR9B F9 E7
2508 GR9A 99
2509 GR9E 66 06 E3
2510 GR9B 88
2511 GR9B E5
2512 GR9B 98
2513 GR9B F0 02
2514 GR9B
2515 GR9B F0 25
2516 GR9B 5B 90
2517 GR9C F7 06
2518 GR9C CE FF FF
2519 GR9C 43
2520 GR9C F0 1B
2521 GR9A 5A CB 03 00
2522 GR9D 58 IE A1
2523 GR9D CE FF FF
2524 GR93 F8 90
2525 GR9D 7E 18 E5
2526 GR9B 5A CB 03 00
2527 GR9C E4
2528 GR9B 5C B1 49 82

POMD R32,-R6 clean up stack
ELSE enough mem
STBD R22,X76,TWO new line len
POMD R32,-R6 get the str size
LDND R34,*KLUDGE get the str loc
JSB *LPR put str into data file
ENDIF
ENDIF
HLL2 JMP HLL2 post-dec pointer
HLL0 JMP HLL0
TEXTSI JSB *CHENLM check for endlin
REN yes, so return
stm r36,r24
JSB *INCTEX pre-inc pointer
JSB *POSIT position pointer
IFWZ at current line?
tsb r45 test for pre-inc
ifnz yes.
JSB *ERROR report error
BYI 90D invalid line W
else no.
tsb r76
jzr BRDSTF
bcd
set bcd
dcm r76 set pre-dec
bin
store into lineW
ldu r36,r24
jsb *POSIT position here
jzr badstf no way get out
bcd
pund r46,-r6
dcm r46
pund r46,*r6
bin
jmp okstf success go do it
endif
ELSE
OKSTF TSB R33 okay,
IFWZ test str size
JSB *ERROR too big
BYI 67D report error
ELSE line too long
okay,
ADND R32,-THREE add three for line
LDND R30,R36 position for alloc
JSB *ALLOD open up space
JEN HLL0 not enough mem
POMD R76,*R30 put in line W
stm r32,-three
POMD R32,*R30 put in size
LDND R34,*KLUDGE get str size

```



```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 72
-----
>>>>>> INSERT N runtimes, INSTR & INNUM <<<<<<<
2529 6BE1 CE 04 68 JSB =LPR put in string
2530 6BE4 ENDIF
2531 6BE4 ENDIF
2532 6BE4 9D 9C STE post-dec pointer
2533 6BE6 F0 E0 JMP HLL0+
2534 6BE8 F0 99 HLL0+ jmp HLL2+
2535 6BE8

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 74
-----
>>>>>> INNUM, insert num data <<<<<<<
2587 6C32 F0 19 ELSE
2588 6C34 58 1E A1 LDM R30,R36 r30 ptr for alloc
2589 6C37 CE FF FF JSB =ALLDC open up room
2590 6C39 F8 CF JEN HLL4 not enough mem, out
2591 6C3C 52 3E 06 02 STBD R22,R76,TWO store in new size
2591 6C40 00
2592 6C41 50 18 E4 PUMB R20,+R30 put in token
2593 6C44 C8 1A CMB R20,-INTIK integer?
2594 6C46 F6 03 IFEQ yes,
2595 6C48 60 DRP S5 int in R55
2596 6C49 F0 01 ELSE no,
2597 6C4B 68 DRP S0 real in r50
2598 6C4C ENDIF
2599 6C4C A7 STND R0,R30 store in the data
2600 6C4D ENDIF
2601 6C4D 9D 9C STE set e for piggy jump
2602 6C4F F0 0A HLL5 JMP HLL4 post-dec pointer
2603 6C51

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 73
-----
>>>>>> INNUM, insert num data <<<<<<<
2537 6BE8
2538 6BE8
2538 6BE8
2539 6BE8
2540 6BE8
2541 6BE8
2542 6BE8
2543 6BE8
2544 6BE8
2545 6BE8
2546 6BE8
2547 6BE8
2548 6BE8
2549 6BE8
2550 6BE8
2551 6BE8
2552 6BE8
2553 6BE8
2554 6BE8
2555 6BE8 24
2556 6BE8 CE E3 68 INNUM JSB =CHAN+ open channel
2557 6BE8 F8 05 REN not there
2558 6BE8 F7 04 IFNZ not in system
2559 6BE2 CE 96 69 job =failhan
2560 6BE5 9E rtn
2561 6BE6 ENDIF
2562 6BE6 CE 07 67 JSB =TYPER get the file type
2563 6BE9 F9 04 IFEN not basic,
2564 6BE8 CE FF FF JSB =ERROR+ report error
2565 6BE6 44 BYT 68D anv file type
2566 6BE6 ENDIF
2567 6BE6 CE C3 67 JSB =INCPTR pre-inc pointer
2568 6C02 68 06 E5 PUMB R50,+R6 save numeric
2569 6C05 CE 32 67 JSB =POSIT position pointer
2570 6C08 68 06 E3 PUMB R50,-R6 retrieve numeric
2571 6C0B F8 DB HLL4 JEN HLL0+ no data, out
2572 6C0D 6C CB FF CMB R54,+377 integer?
2573 6C10 F6 09 IFEQ yes,
2574 6C12 50 A8 1A LDM R20,+INTIK load integer th
2575 6C15 5A A9 04 00 LDM R32,-FOUR size of four
2576 6C19 F0 07 ELSE no,
2577 6C1B 50 A8 04 LDM R20,+REALTK load real th
2578 6C1E 5A A9 09 00 LDM R32,-NINE size of nine
2579 6C22 ENDIF
2580 6C22 52 93 CLM R22 clear for size
2581 6C24 3E 04 02 00 LDBD R22,X76,TWO get the size
2582 6C28 1A C3 ADM R22,R32 add in the insert
2583 6C2A 53 90 ISB R23 too big,
2584 6C2C F7 06 IFNZ yes,
2585 6C2E CE FF FF JSB =ERROR report error
2586 6C31 43 BYT 67D line too long

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 75
-----
>>>>>> REPLACE N runtimes; RPSTR and RPNUM <<<<<<<
2605 6C51
2606 6C51
2606 6C51
2607 6C51
2608 6C51
2609 6C51
2610 6C51
2611 6C51
2612 6C51
2613 6C51
2614 6C51
2615 6C51
2616 6C51
2617 6C51
2618 6C51
2619 6C51
2620 6C51
2621 6C51
2622 6C51
2623 6C51
2624 6C51
2625 6C51 24
2626 6C52 CE 08 68 RPSTR JSB =CHAN+ open channel
2627 6C55 F8 9E REN not there
2628 6C57 F7 04 IFNZ not in system
2629 6C59 CE 96 69 job =failhan
2630 6C5C 9E rtn
2631 6C5D ENDIF
2632 6C5D 5A 0C A3 STN R32,R14 store for text
2633 6C60 CE 07 67 JSB =TYPER get file type
2634 6C63 F8 67 JEN IEXRIP go if text
2635 6C65 CE C3 67 JSB =INCPTR pre-inc pointer
2636 6C68 CE 32 67 JSB =POSIT position pointer
2637 6C6B F8 E2 JEN HLL5 no more data
2638 6C6D 50 1E A5 LDM R20,R36 get type token
2639 6C70 CE 4E 69 JSB =CALC calc data len
2640 6C73 5A 93 CLM R24 clear for line len
2641 6C75 3E 04 02 00 LDBD R24,X76,TWO get the line len
2642 6C79 1A C5 SBN R24,R32 sub data len
2643 6C7B 0C C3 ADM R24,R14 add in insert len
2644 6C7D 89 ICM R24 inc for token
2645 6C7E 89 ICM R24 inc for str len
2646 6C7F 55 90 FSB R25 test for line too long
2647 6C81 F7 06 IFNZ yes,
2648 6C83 CE FF FF JSB =ERROR report error
2649 6C86 43 BYT 67D line too long
2650 6C87 F0 3F ELSE no,
2651 6C89 50 A1 LDM R20,R14 str len to r20
2652 6C8B 89 ICM R20 inc for token
2653 6C8C 89 ICM R20 inc for str len
2654 6C8D 1A C5 SBN R20,R32 calc net gain

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 76
*****
>>>>>> REPLACE W runtimes; RPSTR and RPNUM <<<<<<<
2655 6C8F F4 0D IFPS pos,
2656 6C91 5A 06 E5 PUMD R32,R6 save len
2657 6C94 10 A1 LDM R32,R20 load in pos net
2658 6C96 CE FF FF JSB *ROOM! enough room?
2659 6C99 5A 06 E3 PUMD R32,-R6 restore stack
2660 6C9C F8 CD JEN HLL6 not enough, out
2661 6C9E EMOIF neg,
2662 6C9F 58 1E A1 LDM R30,R36 set pointer
2663 6CA1 CE FF FF JSB *DELETE delete old data item
2664 6CA4 52 93 CLN R22 clear for line len
2665 6CA6 3E B4 02 00 LDDO R22,X76,TWO get line len
2666 6CA8 1A E5 SBN R22,R32 less by old data item
2667 6CAC 5A 0C 01 LDM R32,R14 save str len
2668 6CAF C0 02 00 ROM R32,-TWO add 2 for tok and len
2669 6CB2 52 1A C3 ROM R22,R32 add to line len
2670 6CB5 3E B6 02 00 STOD R22,X76,TWO store line len
2671 6CB9 CE FF FF JSB *ALLLOC open up space for data
2672 6CDB 5A 0C A1 LDM R32,R14 restore str size
2673 6CBF 11 A2 STB R32,R21
2674 6CC1 5C B1 49 B2 LDMO R34,-KLUDGE get string loc
2675 6CC5 CE FF 67 JSB *LPR put data item in line
2676 6CC8 EMOIF
2677 6CCB 9D 9C SIE set E for to jump out
2678 6CCA F0 9F NLTB JMP HLL6 post-dec pointer
2679 6CCC IEKTRP JSB *INCPTR pre-inc pointer
2680 6CCF CE 90 67 JSB *POSIT position pointer
2681 6CD2 F6 06 IFZR at endlin,
2682 6CD4 CE FF FF JSB *ERROR report error
2683 6CD7 5A BY1 90D invalid line #
2684 6CD8 F0 3C JMP HLL7 jump out
2685 6CDA EMOIF okay,
2686 6CDB 54 93 CLN R24 clear for line len
2687 6CDE 1E B4 02 00 LDDO R24,X36,TWO get line len
2688 6CE0 0C C1 CM R24,R14 compare to replace
2689 6CE2 F5 13 IFNG smaller,
2690 6CE4 5A 01 LDM R32,R14 get replace size
2691 6CE6 14 E5 SBN R32,R24 sub current size
2692 6CE8 CE FF FF JSB *ROOM! enough room?
2693 6CEB F8 29 JEN HLL7 no, out
2694 6CEC 40 90 TSB R15 yes, test line size
2695 6CEE F7 06 IFNZ too big,
2696 6CF1 CE FF FF JSB *ERROR report error
2697 6CF4 43 BY1 67D line too long
2698 6CF5 F0 1F JMP HLL7 out
2699 6CF7 EMOIF okay
2700 6CF7 EMOIF okay
2701 6CF7 EMOIF okay
2702 6CF7 5E C0 02 00 ROM R36,-TWO pointer to line size
2703 6CFD 5A 93 CLN R32 clear for line size
2704 6CFD 1E E0 PUMD R32,R36 pop in size
2705 6CFF 5B A1 LDM R30,R36 pos r30 for delete
2706 6D01 CE FF FF JSB *DELETE delete the old line
2707 6D04 5A 0C A1 LDM R32,R14 get str len

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 78
*****
>>>>>> RPNUM, replace num data item <<<<<<<
2719 6D1E *****
2720 6D1E *****
2721 6D1E *****
2722 6D1E *****
2723 6D1E *****
2724 6D1E *****
2725 6D1E *****
2726 6D1E *****
2727 6D1E *****
2728 6D1E *****
2729 6D1E *****
2730 6D1E *****
2731 6D1E *****
2732 6D1E *****
2733 6D1E *****
2734 6D1E *****
2735 6D1E *****
2736 6D1E *****
2737 6D1E *****
2738 6D1E *****
2739 6D1E 24 BY1 44
2740 6D1F CE E3 68 RPNUM JSB *CHANN+ open channel
2741 6D22 F8 F9 REN not there
2742 6D24 F7 04 IFNZ not in system
2743 6D26 44 A9 95 69 gto filhan
2744 6D2A EMOIF
2745 6D2A CE B7 67 JSB *TYPER get file type
2746 6D2D F9 04 IFEN not basic,
2747 6D2F CE FF FF JSB *ERROR+ report error
2748 6D32 44 BY1 68D inv file type
2749 6D33 EMOIF
2750 6D33 CE C3 67 JSB *INCPTR pre-inc pointer
2751 6D36 68 06 E5 PUMD R50,R6 save numeric
2752 6D39 CE 32 67 JSB *POSIT position pointer
2753 6D3C 68 06 E3 PUMD R50,-R6 restore numeric
2754 6D3F F8 89 JEN HLL9 no data out
2755 6D41 50 1E A5 LDM R20,R36 get data type
2756 6D44 CE 4E 69 JSB *CALC calc the len
2757 6D47 5A 06 E5 PUMD R32,R6 save the len
2758 6D4A 6C C8 FF CM R54,-377 integer?
2759 6D4D F6 09 IFEQ yes,
2760 6D4F 50 A8 1A LDB R20,-INTTK load integer tk
2761 6D52 52 A9 04 00 LDM R22,-FOUR size of four
2762 6D56 F0 07 ELSE no,
2763 6D58 50 A8 04 LDB R20,-REALTK load real tk
2764 6D5B 52 A9 09 00 LDM R22,-NINE size of nine
2765 6D5F EMOIF
2766 6D5F 5A 12 C1 CM R32,R22 compare lengths
2767 6D62 F5 26 IFNG if in>out
2768 6D64 52 06 E5 PUMD R22,R6 save out size

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 77
*****
>>>>>> REPLACE W runtimes; RPSTR and RPNUM <<<<<<<
2708 6D07 CE FF FF JSB *ALLLOC alloc space for new str
2709 6D0A 5B 8D DEN R30 point to line size
2710 6D0C 5A 18 E4 PUMD R32,R30 put in the new line len
2711 6D0F 5C B1 49 B2 LDMO R34,-KLUDGE get the string loc
2712 6D13 CE 04 68 JSB *LPR put in the data item
2713 6D16 66 06 E3 HLL7 PUMD R46,-R6 get the old data pointer
2714 6D19 16 B7 0C 00 STND R46,X26,LINEM reset to old value
2715 6D1D 9E RIM
2716 6D1E
2717 6D1E

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 79
*****
>>>>>> RPNUM, replace num data item <<<<<<<
2769 6D67 1A E5 SBN R22,R32 calc net gain
2770 6D69 A3 S1M R22,R32 put in r32 for room
2771 6D6A CE FF FF JSB *ROOM! enough room?
2772 6D6D 52 06 E3 PUMD R22,-R6 restore
2773 6D70 5A E3 PUMD R32,-R6 lengths
2774 6D72 F8 C0 JEN HLL9 not enough mem,out
2775 6D74 5A 83 CLN R24 clear for stsize
2776 6D76 3E B4 02 00 LDDO R24,X76,TWO get the stsize
2777 6D78 1A C5 SBN R24,R32 sub the out size
2778 6D7C 12 C3 ROM R24,R22 add the in size
2779 6D7E 55 90 TSB R25 line too long,
2780 6D80 F7 06 IFNZ yes,
2781 6D82 CE FF FF JSB *ERR1 report error
2782 6D85 43 BY1 67D line too long
2783 6D86 F0 B7 HLL10 JMP HLL9 reset ptr and out
2784 6D88 EMOIF
2785 6D88 F0 02 ELSE no,
2786 6D8A 06 E3 PUMD R32,-R6 restore out length
2787 6D8C EMOIF
2788 6D8C 58 1E A1 LDM R30,R36 r30 ptr for delete
2789 6D8F CE FF FF JSB *DELETE delete the out data item
2790 6D92 54 93 CLN R24 clear for stsize
2791 6D94 3E B4 02 00 LDDO R24,X76,TWO get the stsize
2792 6D98 1A C5 SBN R24,R32 sub out length
2793 6D9A 12 C3 ROM R24,R22 add the in length
2794 6D9C 3E B6 02 00 STOD R24,X76,TWO reset stsize
2795 6D9E 5A 12 A1 LDM R32,R22 r32 in len for alloc
2796 6D9F CE FF FF JSB *ALLLOC open up the space
2797 6DA0 50 18 E4 PUMD R20,-R30 put in the token
2798 6DA3 C8 1A CM R20,-INTTK integer
2799 6DA6 F6 03 IFEQ yes,
2800 6DA8 6D DRP 55 int in r55
2801 6DAE F0 01 ELSE no,
2802 6DB0 68 DRP 50 real in r50
2803 6DB1 EMOIF
2804 6DB1 A7 STMD RW,R30 store the data item
2805 6DB2 9D 9C SIE set e for piggy return
2806 6DB4 F0 D0 JMP HLL10 reset pointer and out
2807 6DB6

```

NOMAS

Not Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 80
=====
SEARCH runtimes; SERNO. and SERNN.
=====
2809 60B6
2810 60B6
2810 60B6
2811 60B6
2812 60B6
2813 60B6
2814 60B6
2815 60B6
2816 60B6
2817 60B6
2818 60B6
2819 60B6
2820 60B6
2821 60B6
2822 60B6
2823 60B6
2824 60B6
2824 60B6
2824 60B6
2825 60B6
2826 60B6 24
2827 60B7 98
2828 60B8 CE FF FF
2829 60B8 SE 0A E3
2830 60B8 CE 1C A1
2831 60C1 B3 33 82
2832 60C4 CE 78 69
2833 60C7 F8 66
2834 60C9 F7 04
2835 60C8 44 A9 95 69
2836 60C7
2837 60C7 SE 06 E5
2838 60D2 CE 72 68
2839 60D5 SE 06 E3
2840 60D8 F8 55
2841 60DA CE B7 67
2842 60DD F8 60
2843 60DF CE C3 67
2844 60E2 CE 32 67
2845 60E5 F8 49
2846 60E7 CE 68 69
2847 60EA CE 0F 68
2848 60ED F9 0D
2849 60EF CE 3D 68
2850 60F2 F8 3C
2851 60F4 CE 01 38 82
2852 60F8 F8 36
2853 60FA FO EF
2854 60FC
2855 60FC SE 06 E5
2856 60FF CE E5
2857 6E01 SE E5
2858 6E03 18 A3
=====
BYT 44
SERNO. BIN
JSB =GETADR get the str addr
POMO R36,-R12 get the str len
LDI R54,R34 load for store
SIMD R54,-PROTEN store in proten
JSB =BEGIN open channel
REN not there
IFNZ not in system
gto filhan
ENDIF
POMO R36,+R6 save the file ptr
JSB =SEICON set start and fin
POMO R36,-R6 get the file ptr
REN bad line #
JSB =TYPER get the file type
JEN TEXTSH go if text
JSB =INCPTR pre-inc pointer
JSB =POSIT position pointer
JEN HLL11 no data, out
JSB =PDINTR get the data ptr
JSB =STRSCH search the string
IFEN not a match,
JSB =CHKIOK get next valid item
JEN HLL11 no more data
CMD R56,+TEMP2 compare to end cond
JCY HLL11 jump to failure
JMP BACH2 check the item
else
pumd r26,+r6 save iofile ptr
pumd r56,+r6
pumd r36,+r6 save channel ptr
str r36,r30 set r30 to string1
=====

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 82
=====
SEARCH runtimes; SERNO. and SERNN.
=====
2910 6E7D F8 08
2911 6E7F FO CE
2912 6E81
2913 6E81 SE 06 E3
2914 6E84 SE E3
2915 6E86 7E 1E A5
2916 6E89 7D 92
2917 6E8B 16 B7 0B 00
2918 6E8F 9E
2919 6E90 FO 9E
=====
JCY name-
jmp noz2 yes and out
endif check the next
found it
pumd r36,-r6 restore chan ptr
pumd r26,-r6 restore iofile ptr
ldnd r76,r36 get the line #
clb r75 clear data #
stnd r75,x26,data# store in iofile
rin that's all
HLL11+ JMP HLL11
=====

```

NOMAS

NOT MANUFACTURER SUPPORTED
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 81
=====
SEARCH runtimes; SERNO. and SERNN.
=====
2859 6E05 5B CB 02 00
2860 6E09 5A 93
2861 6E0B 03 80
2862 6E0D 74 B1 33 82
2863 6E11 1C A3
2864 6E13 CE FF FF
2865 6E16 F6 09
2866 6E18 SE 06 E3
2867 6E1B CE E3
2868 6E1D SE E3
2869 6E1F FO CE
2870 6E21
2871 6E21 SE 06 E3
2872 6E24 CE E3
2873 6E26 SE E3
2874 6E28 6D 16 B7 0B
2875 6E2C 00
2876 6E2D CE E3
2877 6E2F 9E
2878 6E30
2879 6E30 85 06 E3
2880 6E37 CE FF FF
2881 6E38 CE FF FF
2882 6E3D 7D
2883 6E3E 9E
2884 6E3F
2885 6E3F 7D 16 B5 0B
2886 6E43 00
2887 6E44 90
2888 6E45 F7 03
2889 6E47 99
2890 6E48 7E 89
2891 6E4A
2892 6E4D CE FF FF
2893 6E4F 5A 93
2894 6E51 1E B4 02 00
2895 6E55 5A 91
2896 6E57 CB 03 00
2897 6E59 SE 06 E5
2898 6E5D SE E5
2899 6E5F 74 B1 33 82
2900 6E63 1C A3
2901 6E65 CE FF FF
2902 6E68 F6 17
2903 6E6A SE 06 E3
2904 6E6D SE E3
2905 6E6F CE FF FF
2906 6E72 76 1E A5
2907 6E75 19 39 A9
2908 6E7B F7 00
2909 6E7D 01 38 82
=====
adn r30,+two pl to beginning of str
cin r32 clear for size
ldn r32,r3 size of string1
ldnd r64,proten get the second str
str r64,r34 r34 addr, r36 len
jsb =posrun POS routine
ifzr not there
pumd r36,-r6 restore r36
pumd r56,-r6
pumd r26,-r6 restore r26
jmp bachi try again
endif
pumd r36,-r6 restore r36
pumd r56,-r6
pumd r26,-r6 restore r26
stnd r55,x26,data# store location
pumd r55,-r6 clean up stack
rin that's all
ENDIF
POMO R45,-R6 post-dec pointer
SIMD R45,x26,DATAM reset in iofile
JSB =CLRERR clear errors
JSB =ERRROR report warning
BYT 125D 'data not found'
RTN
LDND R75,X26,DATAM data line #
tab r75 test data#
ifnz if increment
bcd set bcd
xcm r76 inc line #
endif
JSB =FSEK seek to line #
JMP MOZ2 check for endlin
CLR R32 clear for load
LDND R32,X36,TWO get etsize
ldn r30,r36 r30 to source line
adn r30,+three
pumd r26,+r6 save iofile ptr
pumd r36,+r6 save channel ptr
ldnd r64,proten get sub str
str r64,r34 r34 addr, r36 size
jsb =posrun go to POS routine
ifzr not found
pumd r36,-r6 restore chan ptr
pumd r26,-r6 restore iofile ptr
jsb =skipln skip to next
ldnd r66,r36 get the line #
cmi r66,endlin at the end?
jcy name- yes and out
cmd r66,+TEMP2 past the limit
=====

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 83
=====
SERNN., search for num item
=====
2921 6E92
2922 6E92
2923 6E92
2924 6E92
2925 6E92
2926 6E92
2927 6E92
2928 6E92
2929 6E92
2930 6E92
2931 6E92
2932 6E92
2933 6E92
2934 6E92
2935 6E92
2936 6E92
2937 6E92
2938 6E92 24
2939 6E93 68 0A E3
2940 6E96 B3 33 82
2941 6E99 CE 7B 69
2942 6E9C F8 F1
2943 6E9E F7 04
2944 6E9D 44 A9 95 69
2945 6E9A
2946 6E9A CE B7 67
2947 6E97 F9 04
2948 6E99 CE FF FF
2949 6E9C 44
2950 6E9D
2951 6E9D SE 06 E5
2952 6E9B CE 72 68
2953 6E93 SE 06 E3
2954 6E86 F8 D7
2955 6E88 CE C3 67
2956 6E8B CE 32 67
2957 6E8E F8 D0
2958 6E90 7D B1 33 82
2959 6E94 4C B1 38 82
2960 6E98 CE 68 69
2961 6E9B CE 21 68
2962 6E9E F9 0C
2963 6E9D CE 3D 68
2964 6E93 F8 F9
2965 6E95 6E 0C C1
2966 6E9B F8 B6
2967 6E9A FO EF
2968 6E9C
2969 6E9C F6 0C
2970 6E9E 74 E8 FF
=====
BYT 44
SERNN. POMO R50,-R12 get the matcher
SIMD R50,-PROTEN save in proten
JSB =BEGIN wave in proten
REN open channel
IFNZ not there
IFNZ not in system
gto filhan
ENDIF
JSB =TYPER get file type
IFEN not basic
JSB =ERRROR report error
BYT 68D invalid file type
ENDIF
POMO R36,+R6 save the file ptr
JSB =SEICON set start and fin cons
POMO R36,-R6 restore file ptr
REN E set for bad line #
JSB =INCPTR pre-inc pointer
JSB =POSIT position pointer
JEN HLL11 no data, out
LDND R60,-PROTEN get the matcher
LDND R14,+TEMP2 get the final line
JSB =PDINTR get the data pointer
JSB =NUMSCH search numeric
IFEN not a numeric
JSB =CHKIOK get next data item
JEN HLL12 none left
CMD R56,R14 at the end line
JCY HLL11+ yes,
JMP HANL1 check new item
ENDIF
IFZ if an integer
CNB R64,+377 search for int
=====

```

```

KARMA 08/31/82 *****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 84
*****
>>>>>> SEEK runtimes; SEEKA. and SEEKR. <<<<<<<
2971 GEE1 F6 05 IFEQ yes,
2972 GEE3 75 25 E1 CMH R65,R45 compare into
2973 GEE6 F7 07 JEQ HANL4 found match
2974 GEE8 EMOIF no,
2975 GEE8 FO E6 JMP HANL2 try next item
2976 GEEA EMOIF
2977 GEEA 70 20 E1 CMH R60,R40 compare the two reals
2978 GEE0 F6 E1 JNE HANL2 not a match continue
2979 GEEF 60 16 07 0B HANL4 STD R55,X26,DATAM found it, store old ptr
2979 GEF3 00
2980 GEF4 06 E3 GORD R55,-R6 restore stack
2981 GEF6 9E RTM
2982 GEF7
2983 GEF7

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82 *****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 86
*****
>>>>>> SEEK runtimes; SEEKA. and SEEKR. <<<<<<<
3033 GF30 TEXTSK
3034 GF30 76 93 TEXTSK CMH R66 loop,
3035 GF30 LOOP get the line N
3036 GF30 6E 1E R5 LDMD R56,R36 compare to EOF
3037 GF40 C9 99 R9 CMH R56,-ENDLIN yes,
3038 GF43 F6 0B IFEQ report error
3039 GF45 CE 37 6F JSB -ERROR1 set data ptr 1
3040 GF48 60 A8 01 LDB R55,-ONE last legal line
3041 GF48 6E 36 A1 LDM R56,R6 set eof
3042 GF4E FO 0D JMP BRANS3
3043 GF50 EMOIF
3044 GF50 66 8B DCM R46 dec abs loc
3045 GF52 F7 07 JZR BRANS2 we are here!
3046 GF54 76 A5 LDMD R66,R36 save current line N
3047 GF56 CE FF FF JSB -SKPLM not, then skip line
3048 GF59 FO E2 EMOIF
3049 GF5B 60 92 BRANS2 CLD R55 store data pointer away
3050 GF5D 60 16 07 0B BRANS3 STD R55,X26,DATAM store data pointer away
3050 GF61 00
3051 GF62 9E RTM
3052 GF63

```

```

KARMA 08/31/82 *****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 85
*****
>>>>>> SEEK runtimes; SEEKA. and SEEKR. <<<<<<<
2985 GEF7
2986 GEF7
2987 GEF7
2988 GEF7
2989 GEF7
2990 GEF7
2991 GEF7
2992 GEF7
2993 GEF7
2994 GEF7
2995 GEF7
2996 GEF7
2997 GEF7
2998 GEF7
2999 GEF7
3000 GEF7
3001 GEF7
3002 GEF7
3003 GEF7 24 BYT 44
3004 GEF8 CE FF FF SEEKR. JSB -ONEB get abs loc
3005 GEF8 66 06 E5 PUMD R46,-R6 save
3006 GEF1 CE AE 08 JSB -CHAN open channel
3007 GF01 66 06 E3 POND R46,-R6
3008 GF04 F8 FO REN not there
3009 GF06 42 90 TSB R2
3010 GF08 F7 04 IFNZ not in system
3011 GF0A 44 A9 95 69 goto filhan
3012 GF0E EMOIF
3013 GF0E CE B7 67 JSB -TYPER get file type
3014 GF11 F8 28 JEM TEXTSK go if text
3015 GF13 66 8B DCM R46 for pre-inc
3016 GF15 CE 9F 67 BRANS1 JSB -ORIGIN get data line
3017 GF18 F8 10 JEM ERROR1 all out, error
3018 GF1A 52 93 CLM R22 clear counter
3019 GF1C CE 1F 69 JSB -COUNT count data items
3020 GF1F 66 12 15 SMI R46,R22 sub from abs loc
3021 GF22 F5 09 IFMG went too far,
3022 GF24 E3 ROM R46,R22 add back on
3023 GF25 20 A2 STB R46,R55 store byte
3024 GF27 60 16 07 0B STD R55,X26,DATAM put byte and line away
3024 GF2B 00
3025 GF2C 9E RTM
3026 GF2D EMOIF
3027 GF2D 52 2D A2 STB R22,R55 else, set current line
3028 GF30 60 16 07 0B STD R55,X26,DATAM store byte
3028 GF34 00 put byte and line away
3029 GF35 FO DE JMP BRANS1 try next data line
3030 GF37 CE FF FF ERROR1 JSB -ERROR report error
3031 GF3A 22 BYT 340 out of data
3032 GF3B

```

```

KARMA 08/31/82 *****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 87
*****
>>>>>> SEEKR., seek to rel position <<<<<<<
3054 GF63
3055 GF63
3056 GF63
3057 GF63
3058 GF63
3059 GF63
3060 GF63
3061 GF63
3062 GF63
3063 GF63
3064 GF63
3065 GF63
3066 GF63
3067 GF63
3068 GF63
3069 GF63
3070 GF63
3071 GF63
3072 GF63 24 BYT 44
3073 GF64 CE FF FF SEEKR. JSB -ONEB get data N
3074 GF66 66 06 E5 PUMD R46,-R6 save
3075 GF6A CE FF FF JSB -ONE1 get line N
3076 GF6D 65 06 E5 PUMD R46,-R6 save
3077 GF70 CE AE 68 JSB -CHAN open channel
3078 GF73 75 06 E3 POND R65,-R6
3079 GF76 66 E3 POND R46,-R6
3080 GF78 F8 E8 REN not there!
3081 GF7A 42 90 TSB R2
3082 GF7C F7 04 IFNZ not in sys
3083 GF7E 44 A9 95 69 goto filhan
3084 GF82 EMOIF
3085 GF82 CE B7 67 JSB -TYPER get the file type
3086 GF85 F9 04 JEM ERROR1 not basic
3087 GF87 CE FF FF JSB -ERROR report error
3088 GF8A 44 BYT 68D inv file type
3089 GF8B EMOIF
3090 GF8B 77 90 TSB R67 test data N
3091 GF8D F7 04 IFNZ too big,
3092 GF8F CE FF FF JSB -ERROR report error
3093 GF92 5A BYT 90D inv line N
3094 GF93 EMOIF
3095 GF93 67 90 TSB R47 test data N
3096 GF95 F7 04 IFNZ too big
3097 GF97 CE FF FF JSB -ERROR report error
3098 GF9A 43 BYT 67D line too long
3099 GF9B EMOIF
3100 GF9B 66 90 TSB R46 test for 0th item
3101 GF9D F7 01 IFNZ no, then
3102 GF9F 8A DCM R46 dec for pre-inc
3103 GFA0 EMOIF yes, don't dec

```

```

KARMA 08/31/82:*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 88
*****
>>>>>> SEEKR.. seek to rel position <<<<<<<
3104 6FA0 25 A2 SIB R46,R45 store in r45
3105 6FA2 35 A1 LDM R46,R65 line N in r46/47
3106 6FA4 65 16 B7 00 SIMO R45,X26,DATAM put away in iofile
3106 6FA8 00
3107 6FA9 9E RTM
3108 6FAR

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82:*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 90
*****
>>>>>> ADVAN.. advanced the pointer <<<<<<<
3159 6FE6 66 91 TSN R46 done one already
3160 6FE8 F7 35 JZR OUT all done?, yes jump
3161 6FEA F5 08 IFNG invalid parameter
3162 6FEC CE FF FF BADPRM JSB =ERROR report error
3163 6FEF 59 BYT B9D inv parameter
3164 6FF0 66 93 CLN R46 clear for push
3165 6FF2 F0 28 JMP OUTR go output value
3166 6FF4
3167 6FF4 LOOP loop,
3168 6FF4 52 A9 02 00 LDM R22,*TWO count down one
3169 6FF8 CE 06 69 JSB =COUNTR goto next item
3170 6FFB F7 14 IFNZ sel,
3171 6FFD 52 C8 02 CMR R22,*2
3172 7000 F6 02 IFEQ
3173 7002 66 89 ICM R46
3174 7004
3175 7004 6D 92 CLB R55 clear item ptr
3176 7006 CE 9F 67 JSB =DATAM get next data line
3177 7009 F8 11 JEN OUTR no more
3178 700B 5E C8 04 00 ADM R36,*FOUR ptr to next item
3179 700F F0 02 JMP BAR store in iofile
3180 7011
3181 7011 6D 88 ICB R55 inc item ptr
3182 7013 6D 16 B7 0B BAR SIMO R55,X26,DATAM store away
3182 7017 00
3183 7018 66 8B DCM R46 decrement count
3184 701A F6 08 UNWZ done,
3185 701C CE FF FF OUTR JSB =CLRERR clear error set
3186 701F 5E 26 A1 OUT LDM R36,R46 r36 for combin
3187 7022 CE FF FF JSB =COMBIN convert to real
3188 7025 6D 0A E5 PUMD R40,*R12 put on stack
3189 7028 44 A9 FF FF G1D STOSV go store var value
3190 702E
3191 702C CE 90 67 TEXTAD JSB =POSITI position data ptr
3192 702F F7 EE JEO OUT at the endline
3193 7031 66 91 TSN R46 test skip count
3194 7033 F7 EA JZR OUT nothing to do
3195 7035 F4 85 JNG BADPRM invalid parameter
3196 7037 89 ICM R46 there is a pre-dec
3197 7038
3198 7038 40 16 B4 0B LDBD R0,X26,DATAM in the loop
3198 703C 00
3199 703D CE 3B 6F JSB =TEXTIS set pointer
3200 7040 40 16 B6 0B SIMO R0,X26,DATAM
3200 7044 00
3201 7045 F0 05 JMP OUTR finish up
3202 7047

```

```

KARMA 08/31/82:*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 89
*****
>>>>>> ADVAN.. advanced the pointer <<<<<<<
3110 6FAR
3111 6FAR
3111 6FAR
3112 6FAR
3113 6FAR
3114 6FAR
3115 6FAR
3116 6FAR
3117 6FAR
3118 6FAR
3119 6FAR
3120 6FAR
3121 6FAR
3122 6FAR
3123 6FAR
3124 6FAR
3125 6FAR
3126 6FAR
3127 6FAR
3128 6FAR
3129 6FAR
3129 6FAR
3129 6FAR
3130 6FAR
3131 6FAR 24 BYT 44
3132 6FAR 98 ADVAN. BIN set bin mode
3133 6FAR DRP 120 drp is 20
3134 6FAR 0A E3 PUMD R20,*R12 get the var name
3135 6FAR 06 E5 PUMD R20,*R6 save
3136 6FAR CE FF FF JSB =GETADM get the var addr
3137 6FAR DRP 120 drp is 20
3138 6FAR 06 E5 PUMD R20,*R6 save
3139 6FAR CE FF FF JSB =ONEB get the skip count
3140 6FAR DRP 146 drp is 46
3141 6FAR 06 E5 PUMD R46,*R6 save
3142 6FAR CE AE 68 JSB =CHAN open the channel
3143 6FAR 66 06 E3 pend r46,*r6
3144 6FAR CE E3 pend r34,*r6
3145 6FAR 50 E3 pend r20,*r6
3146 6FAR 42 90 TSB R2
3147 6FAR F7 04 IFNZ
3148 6FAR 44 A9 95 69 gto filhan
3149 6FAR
3150 6FAR CE FF FF JSB =PUTREL put the addr on R12
3151 6FAR 50 0A E5 PUMD R20,*R12 put the name on R12
3152 6FAR F8 4B JEN OUT iofile gone
3153 6FAR CE B7 67 JSB =TYPER type the file
3154 6FAR F8 53 JEN TEXTAD go do text
3155 6FAR CE 32 67 JSB =POSITI position pointer
3156 6FAR F8 3E JEN OUTR no data, finish
3157 6FAR 6D 16 B5 0B LDM R55,X26,DATAM get the data pointer
3157 6FE2 00
3158 6FE3 CE 9F 68 JSB =UPPER update the true value

```

```

KARMA 08/31/82:*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG 91
*****
>>>>>> entries and externals <<<<<<<
3204 7047
3205 7047 * entry points
3206 7047
3207 7047

```



```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 96
*****
>>>>>> STALOC in program <<<<<<<
3311 7081
3312 7081
3313 7081 7C 06 E3 ** variable not found
TRYPCN POND R74,-R6 recover R26 and R30 pointers
3314 7084 E5 POND R74,-R6
3315 7085 16 A3 STM R74,R26
3316 7087
3317 7087
3318 7087 CE FF FF ** search the PROFILE variable area
JSB =VSREN search for the variable
3319 708A F9 0A JEZ NOPE jump if not found
3320 708C
3321 708C
3322 708C 52 90 ** variable found. see if currently declaring
TSIDUP TSD R22 check declare flag
3323 708E F6 0E JNZ FNDIT if not then fine
3324 70C0
3325 70C0
3326 70C0 CE FF FF ** ERROR 35 ** declaring existing variable **
JSB =ERROR can't declare existing variable
3327 70C3 23 BVT 350 COM OR DIN EXISTING VARL
3328 70C4 F0 08 FNDI12 JNP FNDIT
3329 70C6
3330 70C6

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 98
*****
>>>>>> allocate the variable <<<<<<<
3385 7109 CE FF FF JSB =GETINT
3386 710C
3387 710C
3388 710C
3389 710C
3390 710C
3391 710C
3392 710C
3393 710C CE 76 71 JSB =STARZE calculate the array size
3394 710F 78 06 E5 1M POND R70,-R6
3395 7112 5A C8 04 00 RDM R24,-R6
3396 7116 7E 93 CLM R76
3397 7118 6E E5 POND R56,-R6
3398 711A CE FF FF JSB =GETINT
3399 711D 6E 06 E3 POND R56,-R6
3400 7120 7E 91 TSM R76
3401 7122 F6 04 IFZR
3402 7124 7E A9 20 00 2M LDM R76,-R32
3403 7128
3404 7128 89 ENDF
3405 7129 89 ICM R76
3406 712A 6E 36 A3 STM R56,R66
3407 712D CE FF FF JSB =INITIAL
3408 7130 6E 91 TSM R56
3409 7132 F7 04 IFNZ
3410 7134 CE FF FF JSB =ERROR
3411 7137 1B BVT 27D
3412 7138
3413 7138 6E 2C A1 ENDF
3414 713B 5C 3E A1 LDM R56,R54
3415 713E 1B LDM R34,R76
3416 713F 1B DCM R34
3417 7140 78 06 E3 DCM R34
3418 7143 POND R70,-R6
3419 7143 76 B1 82 B2 LDM R66,-OPTBAS test the lower bound (clone of
3420 7147 * to option base)
3421 7147 F7 04 IFNZ if clone is neg then opt=1
3422 7149 6E C8 00 80 RDM R56,-88000H set the high bit
3423 714D ENDF
3424 714D 5C 3A A3 STM R34,R72
3425 7150 40 A8 38 SUBSTA LDB R0,-70
3426 7153 6E 06 E5 POND R56,-R6 save the max size
3427 7156 CE FF FF JSB =ALVENT set up transfer
3428 7159 6E 06 E3 POND R56,-R6 save total length
3429 715C 1E E7 POND R56,-R6 store vpa entry
3430 715E 9E put back total size
3431 715F RTN
3432 715F
3433 715F CE FF FF ILLDIN JSB =ERROR+
3434 7162 24 BVT 360 ILLEGAL DIM
3435 7163

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 97
*****
>>>>>> allocate the variable <<<<<<<
3332 70C6
3333 70C6
3334 70C6
3335 70C6 CE E2 70 ** variable was NOT found so allocate
NBPE JSB =STARARR go store the variable
3336 70C9
3337 70C9
3338 70C9 6E 06 E3 ** if RUN then
POND R56,-R6 TRASH OLD R30
3339 70CC 58 E5 save new end of PROFILE var area
3340 70CE
3341 70CE
3342 70CE 58 06 E3 ** variable allocated so STORE ADDR
FNDIT POND R30,-R6 RESTORE LMA + 1 VAR
3343 70D1 56 E3 POND R26,-R6 RESTORE FMA VAR pointers
3344 70D3 5C 06 E3 FOUND POND R34,-R6 PROGRAM PTR
3345 70D6
3346 70D6 5A 05 A3 82 SBND R32,-PROFILE make relative to PROFILE
3347 70DA * note that R32 was returned
3348 70DA * in VSREN
3349 70DA
3350 70DA 4F 90 TSD R17 ERRORS?
3351 70DC
3352 70DC
3353 70DC
3354 70DC
3355 70DC
3356 70DC F4 03 ** FLA-R-A-SH! Hello Master and Missus America and
IFPS POND R32,-R34 a-a-all the s-h-hops at sea. (dit-da-da-dit-d...)
3357 70DE 5A 1C E7 ** At this v-vary point in the code the address of
POND R32,-R34 the v-variable pointer is being stored.
3358 70E1 ENDF DON'T STORE IF ERRORS
3359 70E1 9E RTN STORE ADDRESS
3360 70E2
3361 70E2
3362 70E2
3363 70E2
3364 70E2 7E B1 82 B2 STARARR LDM R76,-OPTBAS
3365 70E6 F5 05 IFNG if '1' then base is zero
3366 70E8 93 CLM R76 if not set
3367 70E9 89 ICM R76 set it!
3368 70EA A3 82 B2 STM R76,-OPTBAS
3369 70ED ENDF
3370 70ED
3371 70ED 5E 38 A3 STM R36,R70 STORE NAME
3372 70F0 7A 93 CLM R72
3373 70F2 50 A9 0A 00 LDM R20,-R10
3374 70F6 3C R2 stb R20,-R74
3375 70F8 52 90 TSD R22 are we declaring?
3376 70FA F6 67 JNZ STOR() if not go elsewhere
3377 70FC 53 F8 27 CNB R23,-F0RMAL is this a formal array
3378 70FF FB 62 JCV STOR() go do formal array
3379 7101
3380 7101
3381 7101 CE FF FF ** allocate dimensioned arrays
JSB =GETINT
3382 7104 7E 3E A3 STM R76,R74 save it
3383 7107 93 CLM R76 R76<- -1 for vector default
3384 7108 8B DCM R76 377,377

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG 99
*****
>>>>>> STOR() create array header <<<<<<<
3437 7163
3438 7163 7E A9 0A 00 STOR() LDM R76,-R10 load default dimension
3439 7167 53 90 TSD R23 test token number
3440 7169 F3 03 IFOD two dimensional arrays have
3441 716B * even token numbers
3442 716B 7E 93 CLM R76
3443 716D 8B DCM R76
3444 716E ENDF
3445 716E CE 76 71 JSB =STARZE
3446 7171 78 06 E5 POND R70,-R6
3447 7174 F0 A6 JNP 2b
3448 7176
3449 7176 76 B1 82 B2 STARZE LDM R66,-OPTBAS R66<-option base
3450 717A * (used to check the array bound)
3451 717A 8D TCM R66
3452 717B 3C C1 CCM R66,R74
3453 717D *
3454 717D F7 E0 JEQ ILLDIN
3455 717F 7E 91 TSM R76
3456 7181 F4 05 IFPS is there a second dimension?
3457 7183 76 3E C1 CCM R66,R76
3458 7186 * compare option base to second
3459 7186 F7 D7 JEQ ILLDIN dimension
3460 7188 ENDF JIF OPTBAS 1 DIM 0
3461 7188 CE 94 71 JSB =CALCL go calculate the array size
3462 718B
3463 718B 6E 91 TSM R56
3464 718D F5 04 IFNG if R56 is negative then array 1
3465 718F CE FF FF JSB =ERROR subscript error
3466 7192 1B BVT 27D
3467 7193 ENDF
3468 7193 9E RTN
3469 7194
3470 7194
3471 7194 * CALCL: this routine calculates array size
3472 7194 *
3473 7194 * INPUT: R74 - maximum row
3474 7194 * R76 - maximum column
3475 7194 * OPTBAS - indicator of option base
3476 7194 *
3477 7194 * OUTPUT: R56 - size of array
3478 7194
3479 7194 7C 30 A3 CALCL STM R74,R60
3480 7197 6E A1 LDM R54,R60
3481 7199 D3 82 B2 RDM R54,-OPTBAS
3482 719C 6E 93 CLM R56
3483 719E 6C 3A A3 STM R54,R64
3484 71A1 73 90 CALS22 TSD R63
3485 71A3 F4 10 JNG VECTOR
3486 71A5 76 32 A1 LDM R66,R62
3487 71A8 D3 82 B2 RDM R66,-OPTBAS
3488 71AB 6E 91 TSM R56
3489 71AD F6 0A JNZ VETERI

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG100
=====
>>>>>> STORE() create array header <<<<<<<<
3490 71AF 7E 2C A1 LDM R76,R54 ROM 4 B
3491 71B2 CE FF FF JSB *INITIAL
3492 71B5 6E 91 VECTOR ISN R56 OVERFLOW?
3493 71B7 F7 03 JZR VCTER2 JIF NO
3494 71B9 6D AD FF VCTER1 LDB R55,*377 SET SIGN BIT
3495 71BE 6E 2C A1 VCTER2 LDM R56,R54
3496 71BF 7C 30 A1 LDM R74,R60 RESTORE ROW COL
3497 71C2 9C RTN

```

NOMAS

NOI Manufacturer Supported
recipient agrees NOI to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG102
=====
>>>>>> STREMW, env allocate string arrays <<<<<<<
3521 71F2
3522 71F2
3523 71F2
3524 71F2 STREMW: environmentally allocates string arrays
3525 71F2
3526 71F2
3527 71F2
3528 71F2 53 C8 CO STREMW CMB R23,*110000000b
3529 71F5 FA CD RMC If nonzero then remote access
3530 71F7
3531 71F7 CE 55 62 JSB *HANDLE SAVE FIRST NAME BYTE
3532 71FA 52 10 B5 0C LDM R22,R20,E.ROM get the ROM number
3533 71FE 00
3534 7201 7A 1E E1 IFZR
3535 7204 5E E1 PDM R72,*R34 get some information
3536 7206 FO OE PDM R36,*R34 get the relative pointer
3537 7208 7A A9 7A 1C ELSE
3538 720C E1 5E E1 9E LDM R72,*7AH,1CH,0E1H,5EH,0E1H,9EH
3539 7210 B3 E1 81 STND R72,*IMPHEH store code to ram
3540 7216 JSB *FINCOO the code
3541 7216 52 2E A1 EMOIF
3542 7219 LDM R22,R56 save max length
3543 7219
3544 7219
3545 7219 old code that didn't work out of ROM
3546 7219
3547 7219 pmd r72,*r34
3548 7219 pmd r36,*r34
3549 7219
3550 7219 SE 91 ISN R36 test for negloc in vpa
3551 721B F4 R5 RMC return if stalled prgm
3552 721D 10 C3 ROM R36,R20 pointer absolute
3553 721F 7E CF FF 7F RMI R76,*87FFFH and out the option base
3554 7223 76 93 CLH R66 clear for len field
3555 7225 8B DCM R66 decrement for snit
3556 7226 70 A8 20 LDB R60,*BLANK blanks for the string field
3557 7229
3558 7229 LOOP
3559 7229 76 1E E5 PDM R66,*R36 push in FF FF to len field
3560 722C 7E 8B DCM R76 decrem the total len
3561 722E 8B DCM R76 twice
3562 722F 70 1E E4 1H PDM R60,*R36 put in a blank
3563 7232 7E 8B DCM R76 dec the total len
3564 7234 52 8B DCM R22 dec the len of string
3565 7236 F6 F7 JNZ 1B until none left

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG101
=====
>>>>>> ENIs & ENIs <<<<<<<
3499 71C3 A entry points
3500 71C3
3501 71C3
3502 71C3
3503 71C3
3504 71C3
3505 71C3 The following bytes are here
3506 71C3 to allow external access to
3507 71C3 code that must be executed out
3508 71C3 of RAM
3509 71C3
3510 71C3
3511 71C3 99 50 10 16 RUNCOO byt 99h,50h,10h,0c6h,00h,00h,9eh
3512 71C7 00 00 9E
3513 71C8 CE 94 46 00 ROMADR BYT 0CEN,94H,46H,0,0,0,0,9EH
3514 71CE 00 00 00 9E
3515 71D2 CE 41 47 CE ROMADR BYT 0CEN,41H,47H,0CEN,5EH,47H,0CEN,0E1H
3516 71D6 5E 47 CE E1
3517 71DA B1 9E ROMADR2 BYT 81H,9EH
3518 71DC 50 08 E0 9E R10COO BYT 50H,08H,0E0H,9EH
3519 71E0 78 26 E1 5E SCRAT1 byt 78h,26h,0e1h,5eh,0e1h,9eh
3520 71E4 E1 9E
3521 71E6 52 26 E1 78 SCRAT2 byt 52h,26h,0e1h,78h,0e1h,5eh,0e1h,9eh
3522 71EA E1 5E E1 9E
3523 71EE 72 1C E1 9E SCRAT3 byt 72h,1ch,0e1h,9eh
3524 71F2

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG103
=====
>>>>>> STREMW, env allocate string arrays <<<<<<<
3566 723B 7E 91 ISN R76 are we done
3567 723A F7 05 JZR 2F yes then out
3568 723C 52 2E A1 LDM R22,R56 load up string len again
3569 723F FO E8 UNRP
3570 7241 9E 2H RTN
3571 7242

```



```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE SRC=RV8103 OBJ=RVX103 12/13/1983 11:38 AM PG104
*****
SUBENV, allocates the SUB statement
*****
3573 7242
3574 7242
3575 7242
3576 7242
3577 7242
3578 7242
3579 7242
3580 7242
3581 7242
3582 7242 CE 55 52 SUBENV JSD =HANDLE set handle flag
3583 7245 4E C8 01 LDB R16,=C85TS is the prgm halted
3584 7248 F6 49 JFEQ yes,
3585 7249 52 2C B1 LDB R22,R54 get the name
3586 7249 CF FF 3F RMB R22,=OFFH,3FH out the remote bits
3587 7250 A3 STR R22,R54 restore block
3588 7251 6C 1C E7 PUMB R54,=R34 put name back as non-remote
3589 7254 E1 POMB R54,=R34 restore r34
3590 7255 52 2C A0 LDB R22,R54 get name again
3591 7258 F5 00 IFNG string,
3592 725A B4 LDB R22 array bit
3593 725B 90 TSB R22 check str array
3594 725C F5 04 IFNG yes,
3595 725E 5C C8 06 00 RDB R34,=SIN r34 to ptr
3596 7262 ENDDIF
3597 7262 6E 1C E1 POMB R56,=R34 get the rel ptr
3598 7265 F0 09 ELSE numeric
3599 7267 B4 LDB R22 get array bit
3600 7268 90 TSB R22 numeric array
3601 7269 F5 07 IFNG yes,
3602 726B 5C C8 04 00 RDB R34,=FOUR r34 to ptr
3603 726F 6C 1C E1 POMB R56,=R34 get ptr
3604 7272 ENDDIF
3605 7272 ENDDIF
3606 7272 76 B1 51 82 LDB R66,=FIMVARS get fuvars
3607 7276 40 36 B5 02 LDB R0,X66,E.PREV get prev env len
3608 727B 76 2E C3 RDB R66,R56 r66 ptr to remote ptr
3609 727E 42 36 A5 LDB R2,R66 get remote ptr
3610 7281 00 C5 SBN R2,R0 add ptr-prev env len
3611 7283 2E A3 SIN R2,R56 this is the correct loc
3612 7285 6E 1C E7 PUMB R56,=R34 put into vpa
3613 7288 E1 POMB R56,=R34 restore r34
3614 7289 9E RIM all done
3615 728A * this horrible entry was so implemented
3616 728A * to get the JZR to be valid
3617 728A 9B ERRCHK BIN
3618 728B 4F 90 TSB R17 any errors set
3619 728D F5 55 JPS ACTION no okay, go on
3620 728F 44 A9 0B 75 CIO CLEANAL else clean up everything
3621 7293

```

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE SRC=RV8103 OBJ=RVX103 12/13/1983 11:38 AM PG105
*****
SUBENV, allocates the SUB statement
*****
3622 7293
3623 7293 ENDDIF
3624 7293 5A 06 E5 PUMB R32,=R6 save end of VPA
3625 7296 6E E5 PUMB R56,=R6 save rel ptr
3626 7298 50 E5 PUMB R20,=R6 save value ptr
3627 729A 5C E5 PUMB R34,=R6 save VPA ptr
3628 729C CE 4A 75 JSB =CHKRON RMflag is neg if ROM
3629 729F F8 10 JEN INARM in rom don't do much
3630 72A1 50 1C A1 LDB R20,R34 r20 now vpa ptr
3631 72A4 C0 04 00 SBN R20,=FOUR r20 ptr to name form
3632 72A7 B3 B8 83 SIN R20,=ENVFLG store vpa name loc
3633 72AA 6D 90 TSB R55 test name form
3634 72AC F5 10 IFNG access already set
3635 72AE 42 A8 01 LDB R2,=ONE flag already PRL
3636 72B1 40 B0 5E B2 LDB R0,=FLAG1
3637 72B5 02 94 ORB R0,R2
3638 72B7 B2 5E B2 STB R0,=FLAG1
3639 72BA 6D B8 83 SIN R55,=ENVFLG save VRL or REF
3640 72BE * 11 is val
3641 72BE * 10 is ref
3642 72BE ENDDIF
3643 72BE 50 B1 51 82 INARM LDB R20,=FIMVARS get fuvars
3644 72C1 B1 B8 82 SIN R20,=SUBFIV this is the subs fuvars
3645 72C5 52 10 B5 02 LDB R22,X20,E.PREV get len of previous env
3646 72C9 00
3647 72CA 50 12 C5 SBN R20,R22 r20 now prev FIMVARS
3648 72CB B1 51 82 SIN R20,=FIMVARS reset old fuvars
3649 72CD 48 10 B5 16 LDB R10,X20,E.RIM get r10 in call statement
3650 72D4 00
3651 72D5 D3 47 82 RDB R10,=RMFILE
3652 72D8 48 A8 4F LDB R2,=CALL1K load up
3653 72DB 0D 54 B4 CMB R2,=token compare to latest
3654 72DE F7 04 IFNE not doing call
3655 72E0 44 A9 0B 75 CIO CLEANAL then there is no reason to be here
3656 72E4 ENDDIF
3657 72E4 50 93 ACTION CMB R20 make absolute
3658 72E6 B0 5E B2 LDB R20,=FLAG1 caller in ROM
3659 72E9 B4 LDB R20,=SUBFIV make rel to sub fuvars
3660 72EA 90 TSB R20 put into environment
3661 72EB F5 05 IFNG yes,
3662 72ED CE B8 75 JSB =R10B get the token
3663 72EF F0 02 ELSE no,
3664 72F0 08 E0 POMB R20,=R10 get the token
3665 72F4 ENDDIF
3666 72F4 STB R20,=TOKEN store into token
3667 72F7 1F CMB R20,=ERRCHK is this external?
3668 72F8 9B IFNE no,
3669 72FA 9B BIN set binary
3670 72FB 78 B1 C8 71 LDB R70,=ROMADR load up r70 with
3671 72FD 50 B5 LDB R20 double the token
3672 72FE 10 B5 FF LDB R20,X20,BTAB.R load the run table
3673 7307 B8 A3 STR R20,R73 store in subroutine

```

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE SRC=RV8103 OBJ=RVX103 12/13/1983 11:38 AM PG106
*****
SUBENV, allocates the SUB statement
*****
3673 7309 B1 51 82 LDB R20,=FIMVARS get fuvars
3674 730C 10 B5 0C 00 LDB R20,X20,E.ROM get the basic in ROM
3675 7310 30 A3 STB R20,r75 put in for ROMJSB
3676 7312 A9 64 B4 LDB R20,=trpms3 r20 pts to code
3677 7315 78 B3 64 B4 SIN R20,=trpms3 store code in ram
3678 7319 F0 4D JMP JNPSUB go ex code
3679 731B ELSE yes,
3680 731B 80 5E B2 LDB R20,=FLAG1 caller in ROM
3681 731E B4 LDB R20,=SUBFIV
3682 731F 90 TSB R20
3683 7320 F5 05 IFNG yes,
3684 7322 CE B3 75 JSB =R10B get the token
3685 7325 F0 02 ELSE no,
3686 7327 08 E1 POMB R20,=R10 get the token
3687 7329 ENDDIF
3688 7329 50 C9 65 40 CMB R20,=MYROM is this our rom
3689 732D F7 04 IFNE ifne
3690 732F 08 E0 POMB R20,=R10 set r10
3691 7331 F0 13 JMP STUFF no then execute
3692 7333 ENDDIF
3693 7333 80 5E B2 LDB R20,=FLAG1 caller in ROM?
3694 7336 B4 LDB R20,=SUBFIV
3695 7337 90 TSB R20
3696 7338 F5 05 IFNG yes,
3697 733A CE B8 75 JSB =R10B get the token
3698 733D F0 02 ELSE no,
3699 733F 08 E0 POMB R20,=R10 get the token
3700 7341 ENDDIF
3701 7341 50 C8 2A CMB R20,=PARVAL are we env allocating
3702 7344 F8 2C IFNE no then
3703 7346 48 C0 03 00 00 STUFF shr r10,=three
3704 734A 50 B3 CMB R20 clear token
3705 734C A8 B4 LDB R20,=ERRCHK load errchk
3706 734E 98 BTM set binary for shift
3707 734F A5 LDB R20 double token
3708 7350 10 B5 FF FF LDB R20,X20,BTAB.R load from basic table
3709 7354 40 B0 5E B2 LDB R0,=flag1
3710 7358 B4 LDB R0,=flag1
3711 7359 90 TSB R0
3712 735A F5 0C IFNG
3713 735C 79 B1 C3 71 LDB R71,runcod
3714 7360 B3 E1 81 STND R71,=trpms3
3715 7363 CE B3 75 JSB =fincod
3716 7366 F0 06 ELSE
3717 7368 99 BCD bcd for math stuff
3718 7369 50 DRP R20
3719 736A 10 C6 00 00 JSB R20 go do the token
3720 736E ENDDIF
3721 736E 44 A9 89 72 GTO ERRCHK check for errors
3722 7372 ENDDIF
3723 7372 C8 B1 CMB R20,=BITK yes,
3724 7374 F6 04 IFEQ no more params
3725 7376 44 A9 0B 75 GTO CLEANAL error out

```

```

KARMA 08/31/82-----
ITEM  LOC  OBJECT CODE SRC=RV8103 OBJ=RVX103 12/13/1983 11:38 AM PG107
*****
SUBENV, allocates the SUB statement
*****
3726 737A ENDDIF
3727 737A CE 2A SBN R20,=PARVAL no,
3728 737C F0 52 JMP TABLE subtract offset
3729 737E ENDDIF go do env alloc
3730 737E
3731 737E
3732 737E 24 BYT 44
3733 737F 66 0A E3 SREF. POMB R46,=R12 get garbage off the stack
3734 7382 E3 POMB R46,=R12
3735 7383 2E C1 CMB R46,=R56
3736 7385 FA 39 JNC okaye
3737 7387 SE 0A E3 POMB R36,=R12
3738 738A CE FF FF JSB =GETADR go do and reference
3739 738D F0 08 JMP ENDDIF
3740 738F
3741 738F 24 BYT 44
3742 7390 SE 0A E3 MREF. POMB R36,=R12 get the name
3743 7393 CE FF FF JSB =GETADR get the address
3744 7396 5C C0 02 00 SBN R34,=FIMV get r34 back to ptr
3745 739A 40 B0 5E B2 ENDDIF LDB R0,=FLAG1 get the ROM flag
3746 739E F5 05 IFNG neg, then in ROM
3747 73A0 CE 39 75 JSB =ERRCHK get the rel ptr
3748 73A3 F0 20 ELSE pos, in ram
3749 73A5 90 TSB R0 get the recursive flag
3750 73A6 F2 10 IFEV 0 then no recursion
3751 73A8 54 B1 B8 83 LDB R24,=ENVFLG get the loc of name form
3752 73AC 6E 14 A5 LDB R56,R24 get name form
3753 73AF EB 00 B0 RDB R56,=B8000H put in remote bit
3754 73B2 CF FF BF RMB R56,=OFFH,OBFH and out val bit
3755 73B5 A7 STB R56,R24 store in vpa
3756 73B6 F0 0A ELSE NO then recursion
3757 73B8 43 B0 B8 83 LDB R3,=ENVFLG get the name form
3758 73BC CF C0 RMB R3,=OCOH isolate remote bits
3759 73BE C8 B0 CMB R3,=BOH compare to rel var
3760 73C0 F6 76 JNE RESTR report error
3761 73C2 ENDDIF
3762 73C2 6E 1C E1 POMB R56,=R34 pop the rel ptr
3763 73C5 ENDDIF
3764 73C5 6E 1C C3 RDB R56,R20 make absolute
3765 73C8 5E 05 B8 B2 SBN R36,=SUBFIV make rel to sub fuvars
3766 73CC 2E E5 PUMB R36,=R56 put into environment
3767 73CE F0 7A JMP RESET reset all pointers
3768 73D0
3769 73D0 50 12 A2 TABLE STB R20,R22 store byte
3770 73D3 5C E6 E3 POMB R34,=R6 get vpa ptr
3771 73D6 50 E3 POMB R20,=R6 get env ptr
3772 73D8 6E E3 POMB R56,=R6 get rel env ptr
3773 73DA 52 90 F7 B2 JB22=0 MREF. jump for num ref
3774 73DC E8 01 F7 90 JB22=1 SREF. jump for str ref
3775 73DE C8 02 F7 15 JB22=2 AREF. jump for array ref
3776 73E0 C8 03 F7 29 JB22=3 SAREF. jump for str array ref
3777 73E2 C8 04 F7 2C JB22=4 MVAL. jump for num val
3778 73E4 C8 05 F7 5B JB22=5 SVAL. jump for str val

```

```

KARMA 08/31/82:*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RY2103 12/13/1983 11:38 AM PG108
*****
SUBENV, allocates the SUB statement
3779 7362 C8 09 F7 24 JB22=9H MVAL. jump for (R)
3780 7366 C8 0A F7 53 JB22=9H MVAL. jump for (R8)
3781 73FA BYT 44
3782 73FB 24 MREF. cje
3783 73FD 9D 9E jsb =chdjm
3784 73FD CE RE 75 jne okaye
3785 7400 F6 0E arfin tab r23
3786 7402 53 90 ifng
3787 7404 F5 06 addd r36,+fuvars
3788 7406 5E D3 51 B2 LDND R36,R36 by load from env
3789 740A 1E AS ENDIF
3790 740C RDND R36,+FUWARS make absolute
3791 740E 5E D3 51 B2 jmp ENDREF do common end ref
3792 7410 F0 0E
3793 7412 BYT 44
3794 7412 SAMEF. cje
3795 7412 24 jsb =chdjm
3796 7413 9D jne okaye
3797 7414 CE RE 75 arfin jmp ENDREF make absolute
3798 7417 F0 E7
3799 7419 BYT 44
3800 7419 24 LDND R0,-FLAG1 get the ROM flag
3801 741A 40 B0 5E B2 MVAL. if in ram
3802 741E F4 1A IFPS tab r0
3803 7420 90 ifev 0 not recursion
3804 7421 F2 0D LDND R24,-ENVFLG get the name form loc
3805 7423 54 B1 8B B3 LDND R46,R24 get the name form
3806 7427 66 14 A5 RDND R46,-B8000H put in remote bit
3807 7428 C8 00 B0 STND R46,R24 store to vpa
3808 742D A7 ELSE NO then recursion
3809 742E F0 0A LDND R3,-ENVFLG get the name form
3810 7430 43 B0 B8 B3 RDND R3,-OEOH isolate the remote bits
3811 7434 CF 00 CHD R3,-OEOH compare to val var
3812 7436 C8 00 JNE RES1 no, then ref
3813 7438 16 A4 RESTR JNE RES1
3814 743A ENDIF
3815 743A ENDIF
3816 743A 6E 3E A3 SIM R56,R76 save for inc
3817 743D 10 C3 ADND R56,R20 make absolute
3818 743F 7E 1B 02 00 RDND R76,R2 values are remote by 2
3819 7441 2E E5 PUND R76,+R56 put into env
3820 7445 0A 0A E3 PUND R40,-R12 get the value
3821 7448 2E E5 PUND R40,-R56 put value into env
3822 744A F0 6D RESET+ JNP RESET common reset
3823 744C BYT 44
3824 744C 24 BYT 44
3825 744D 52 0A B5 FC MVAL. LDND r22,x12,(0-4) get the length
3826 7451 FF LDND r32,r56
3827 7452 5A 2E A1 LDM r32,r22
3828 7455 12 E1 LDC r32,r22
3829 7457 F8 06 ifnc
3830 7459 CE FF FF jsb =errr
3831 745C 2A byt 42d

```

```

KARMA 08/31/82:*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RY2103 12/13/1983 11:38 AM PG110
*****
SUBENV, allocates the SUB statement
3884 74D1 6F C8 40 CMB R57,+40H is this also remote
3885 74D4 F8 1E IFNC yes,
3886 74D6 40 B0 5E B2 DOME LDND R0,-FLAG1 call in ROM?
3887 74D8 84 lib r0
3888 74D8 90 tab r0
3889 74DC F5 0B ifng yes,
3890 74DE 7C A9 6C 08 LDND R74,-ECHO,OBH,ORSH,SEN
3891 74E2 A5 9E JSB =STF go get next token str
3892 74E7 F0 03 ELSE no,
3893 74E9 6C 08 A5 LDND R54,R10 get next token block
3894 74EE ENDMIF compare to end of param list
3895 74EE 6C C9 CMB R54,-
3896 74EE 04 val eronth
3897 74EF 65 40 def myromM
3898 74F1 31 byt 31H
3899 74F2 F6 1E JNE CLEAR- nope then error
3900 74F4 ENDMIF
3901 74F4 52 B1 51 B2 LDND R22,+FUWARS get fuvars
3902 74F8 48 D5 47 B2 SDND R10,-RNFLE make r10 rel
3903 74FC 12 B7 16 00 SIND R10,R22,E.RTN store in a.rtn
3904 7500 01 A8 B2 LDND R10,-SUBFLW get sub fuvars
3905 7503 03 51 B2 SIND R10,-FUWARS restore fuvars
3906 7506
3907 7506 * CHANGE R10 TO RN TO ENABLE BUG FIX
3908 7506 SEICIK
3909 7506 A8 4F LDND rM,-scalith get the call stuff
3910 7508 B2 54 B4 STND rM,-TOKEN store in token
3911 750B 9E RTN leave
3912 750C
3913 750C 7A 06 E3 CLEARND POND R72,-R6 clear r34,r20,r56 from stack
3914 750F 7E 06 E3 CLEARND POND R76,-R6 clear r32, this is for multi
3915 7512 58 B1 8A B2 CLEARND LDND R30,-SUBFLW r30 ptr to sub env
3916 7516 5A 18 A5 LDND R32,R30 r32 has sub env size
3917 7519 CE FF FF JSB =DELETE blast it away
3918 751C 5B B1 51 B2 LDND R30,-FUWARS get the calls fuvars
3919 7520 40 18 B5 18 LDND R0,R30,E.PCR get the pcr
3920 7524 00
3921 7525 03 47 B2 ADND R0,-RNFLE make it absolute
3922 7528 B3 40 B2 SIND R0,-PCR store in pcr
3923 752B C8 03 00 RDND R0,-R3 add three for r10
3924 752E 08 A3 SIND R0,R10 make r10 ptr to call
3925 7530 46 CD 15 00 SIND R6,-R21 take 21 bytes off r6
3926 7534 * stark, don't go back to
3927 7534 CE FF FF JSB =ERR1 HANDIO or ENVALD
3928 7537 7F BYT 127D report our own
3929 7538 9E RTN bad parameter value
3930 7539
3931 7539 7C B1 EE 71 EMERAND LDND R74,-scrut3 store the code
3932 753C 0C 04 wbb r74,-four get ROM to r22 for getrn2
3933 753F 7C B3 E1 81 title SIND R74,-IMPHEM
3934 7543 52 10 B5 0C LDND R22,R20,E.ROM

```

```

KARMA 08/31/82:*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RY2103 12/13/1983 11:38 AM PG109
*****
SUBENV, allocates the SUB statement
3831 745D F0 25 jmp rest
3832 745F ENDMIF
3833 745F 40 B0 5E B2 LDND R0,-FLAG1 get the ROM flag
3834 7463 F5 05 IFNG neg, in ROM
3835 7465 CE 39 75 JSB =EXERRM get the rel ptr
3836 7468 F0 21 ELSE pos, in RAM
3837 746A 90 tab r0
3838 746B F2 0D ifev 0 not recursion
3839 746D 54 B1 8B B3 LDND R24,-ENVFLG get name form loc
3840 7471 66 14 A5 LDND R46,R24 get name form
3841 7474 C8 00 B0 RDND R46,-B8000H put in remote bit
3842 7477 A7 STND R46,R24 store to vpa
3843 747B F0 0E ELSE NO then recursion
3844 747A 43 B0 B8 B3 LDND R3,-ENVFLG get the name form
3845 747E CF 00 RDND R3,-OEOH isolate remote bits
3846 7480 C8 00 CHD R3,-OEOH compare to val
3847 7482 F7 04 ifne
3848 7484 4A A9 0E 75 RES1 glo CLEAR no, then
3849 7488 ENDMIF
3850 7488 ENDMIF
3851 7488 6E 1C E1 POND R56,+R34 get the rel ptr
3852 748B ENDMIF
3853 748B 6E 1E A3 STND R56,R36 store for remote
3854 748E 10 C3 ADND R56,R20 make absolute
3855 7490 5E 1B 02 00 RDND R36,R2 values are remote by 2
3856 7494 2E E5 PUND R36,+R56 put into env
3857 7496 5A 18 LDM R24,R56 get the blank ptr
3858 7498 C8 02 00 RDND R24,-R2 point to value area
3859 749B 60 A8 20 LDND R40,-BLANK load up with blanks
3860 749E LOOP
3861 749F 60 14 E4 PUND R40,-R24 blank 8 bytes
3862 74A1 5A B8 DEN R32 four times
3863 74A3 F6 F9 MAND
3864 74A5 5E DRP R36 set the addr
3865 74A6 CE FF FF JSB =GETADND get the addr into r36
3866 74A9 52 0A E3 POND R22,-R12 get the str length
3867 74AC 2E E5 PUND R22,+R56 put into actual val field
3868 74AE F7 09 JZR RESET jump if null str
3869 74B0 LOOP no, put in env
3870 74B0 42 1E E0 POND R2,-R36 get a char
3871 74B3 2E E4 PUND R2,-R56 put in env
3872 74B5 52 B8 DEN R22 until done
3873 74B7 F6 F7 MAND
3874 74B9 5A 06 E3 RESET POND R32,-R6 retrieve vpa end
3875 74BE 1C C1 CMB R32,R34
3876 74BF F7 16 JEQ DOME
3877 74C0 40 B0 5E B2 LDND R0,-FLAG1 sub in ROM?
3878 74C4 F5 09 ifng yes,
3879 74C6 CE 39 75 JSB =EXERRM get next vpa name
3880 74C9 5C 1B 02 00 SDND R34,-R20 move r34 back
3881 74CD F0 02 ELSE no,
3882 74CF 6F A5 LDND R56,R34 get vpa name
3883 74D1 ENDMIF

```

```

KARMA 08/31/82:*****
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RY2103 12/13/1983 11:38 AM PG111
*****
SUBENV, allocates the SUB statement
3934 7547 00 jmp #INHI
3935 7548 F0 52 jmp #INHI
3936 754A LDND R40,-RNAME get the caller
3937 754A 60 B1 75 B2 CHNDND JSB =INRAM?
3938 754E CE 74 75 DRP I2
3939 7551 ann r2,-40h,00h
3940 7551 CF 40 00 LDND r0,-FLAG1
3941 7554 40 B0 5E B2 ann r0,-02h,00h
3942 7558 CF 02 00 orb r0,r2
3943 755B 02 94 stbd r0,-FLAG1
3944 755D B2 5E B2 LDND R40,-PRNAME get the file name
3945 7560 60 B1 63 B2 JSB =INRAM?
3946 7564 CE 74 75 DRP I2
3947 7567 ann r2,-80h,00h
3948 7567 CF 80 00 LDND r0,-FLAG1
3949 756A 40 B0 5E B2 orb r0,r2
3950 756E 02 94 stbd r0,-FLAG1
3951 7570 B2 5E B2 RTN return with e valid
3952 7573 9E
3953 7574
3954 7574 58 06 E5 INRAM? PUND R30,-R6 save R30
3955 7577 CE FF FF JSB =FOPEH in RAM?
3956 757A 58 06 E3 POND R30,-R6 restore R30
3957 757D 42 92 CLD R2 clear flag
3958 757F F9 F2 REZ yes,
3959 7581 A8 DCD R2 no, set flag
3960 7582 9E RTN
3961 7583
3962 7583
3963 7583 7C B1 DE 71 R10H LDND R74,-R10CDD get the POND R20,-R10
3964 7587 7E B8 ICB R76 make POND
3965 7589 F0 04 JMP STF go ahead
3966 758B 7C B1 DC 71 R10B LDND R74,-R10CDD get POND R20,-R10
3967 758F 7C B3 E1 81 SIF STND R74,-IMPHEM store code in ram
3968 7593 40 B1 51 B2 FIMCOO LDND R0,-FUWARS we need FUWARS
3969 7597 52 00 B5 0C LDND R22,R0,E.ROM to get the basic in ROM #
3970 759B 00
3971 759C 78 B1 D2 71 F1MHI LDND R70,-ROMAD1 get the SAVENE stuff
3972 75A0 B3 0D B2 STND R70,-IMPHEM store in impmem
3973 75A3 7E B1 DA 71 LDND R76,-ROMAD2 get the rest
3974 75A7 B3 15 B2 STND R76,-(IMPHEM2+B1) store appropriately
3975 75AB CE 0D B2 JSB =IMPHEM2 go do sub
3976 75AE RTN
3977 75AE 5C CD 02 00 CKDIN stn r34,-r20
3978 75B2 66 0A E3 POND R46,-r12
3979 75B5 D3 47 B2 addd r46,-rnfle
3980 75B8 40 B0 5E B2 LDND r0,-flag1
3981 75BC B4 lib r0
3982 75BD 90 tab r0
3983 75BE F5 14 ifng
3984 75C0 F9 06 ifen
3985 75C2 78 B1 E0 71 LDND r70,-scrut1

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG112
*****
>>>>>> SUBENV, allocates the SUB statement <<<<<<<
3986 75C6 F0 04 else
3987 75C8 78 01 E6 71 ldnd r70,=scrat2
3988 75CC endif
3989 75CC 03 E1 81 atnd r70,=trpnen
3990 75CC CE 93 75 jnb =fincod
3991 75D2 F0 0A else
3992 75D4 F8 03 ifez
3993 75D6 52 26 E1 pond r22,r46
3994 75D9 endif
3995 75D9 78 26 E1 pond r70,r46
3996 75DC 5E P* pond r36,r46
3997 75DE endif
3998 75DE 40 00 5E 82 ldnd r0,=flag1
3999 75E2 F5 17 ifnz
4000 75E4 78 06 E5 pond r70,r6
4001 75E7 52 E5 pond r22,r6
4002 75E9 7C 01 EE 71 ldnd r74,=scrat3
4003 75ED F8 02 ifaz
4004 75EF CC 02 stb r74,=two
4005 75F1 endif
4006 75F1 CE 3F 75 jnb =tattle
4007 75F4 52 06 E3 pond r22,r6
4008 75F7 78 E3 pond r70,r6
4009 75F9 F0 06 else
4010 75FB 70 drp 60
4011 75FC F9 01 ifen
4012 75FE 72 drp 62
4013 75FF endif
4014 75FF 1C E1 pond r8,r34
4015 7601 endif
4016 7601 78 drp 70
4017 7602 30 arp 60
4018 7603 F9 02 ifen
4019 7605 78 drp 72
4020 7606 32 arp 62
4021 7607 endif
4022 7607 C1 cmv r8,r8
4023 7608 F6 03 rne
4024 760A F9 03 ifen
4025 760C 52 38 A1 ldh r22,r70
4026 760F endif
4027 760F 42 92 clb r2
4028 7611 9E rtn
4029 7612

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG114
*****
>>>>>> VDLLOC deallocates a variable <<<<<<<
4036 7612
4037 7612 *****
4038 7612 A
4039 7612 A VDLLOC:
4040 7612 A THIS ROUTINE IS CALLED TO DEALLOCATE A
4041 7612 A VARIABLE. IF RUN, THE ADDRESS OF THE
4042 7612 A VARIABLE IS RELATIVE TO THE VARIABLE AREA
4043 7612 A OF THE CURRENT PROGRAM.
4044 7612 A
4045 7612 A INPUT:
4046 7612 A R23 = TOKEN
4047 7612 A R24 = PTR TO ADDR IN STACK
4048 7612 A R26 = FWA VARIABLES (IF RUN)
4049 7612 A R30 = LWA + 1 OF VARIABLES (IF RUN)
4050 7612 A R40-R44 = PCB 7-11 (IF RUN)
4051 7612 A R44-R47 = LINE # AND BYTES (IF RUN)
4052 7612 A
4053 7612 A INTERNAL:
4054 7612 A R34 = SCRATCH
4055 7612 A R36 = SCRATCH
4056 7612 *****
4057 7612 CE FF FF VDL01 JSB =ASCNAM
4058 7618 CE 55 62 PUND R36,#24
4059 761B 9E DALSUB JSB =HANDLE
4060 761C RTN

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG113
*****
>>>>>> ENLrns and EXLrns <<<<<<<
4031 7612
4032 7612
4033 7612
4034 7612

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG115
*****
>>>>>> ENI'S AND EXT'S <<<<<<<
4062 761C
4063 761C
4064 761C
4065 761C
4066 761C
4067 761C MAXLEN EQU 2
4068 761C MAXROW EQU 4
4069 761C MAXCOL EQU 6
4070 761C TOTLEN EQU 80
4071 761C STRLOC EQU 100
4072 761C
4073 761C 72 14 E1 18 C0DY1 by1 72H,14H,0E1H,18H,0E5H,14H,0E1H,18H
4074 7620 E5 14 E1 18
4075 7629 E5 9E C0DY2 BY1 0E5H,9EH

```

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG116
*****
>>>>>> DIN1R store 1-dim string array <<<<<<<
4077 7626
4078 7626 40 B1 51 82 TRACK ldd r0,=fuvars and fuvars
4079 7628 00 05 0C 00 ldd r0,x0,e,row the current ROW
4080 762E F7 EB rcr
4081 7630 58 A9 33 82 ldr r30,=PROTEM get destination
4082 7634 78 B1 1C 76 ldd r70,=cdy1 get first code
4083 7638 A3 E1 84 etnd r70,=INPTEM store for ex
4084 763B 01 24 76 ldd r70,=cdy2 get end of code
4085 763E 83 E9 81 etnd r70,=((INPTEM+8) store for ex
4086 7641 CE 93 75 jnb =FIMCDD go ex in RAM
4087 7644 5A A9 33 82 ldr r24,=protem get dest pointer
4088 7648 9E
4089 7649
4090 7649 1A BYT 32
4091 7649 CE FF FF DIN1R JSB =DMEB job to oneb
4092 764D CE E1 76 JSB =TEST1 test for neg subscr
4093 7650 F8 F6 REN return if neg
4094 7652 CE F1 76 JSB =MYSUB check max col
4095 7655 F6 0C JNE 9F if not then something wrong
4096 7657 76 14 05 04 LDM R66,X24,MAXROW get the max row
4097 765D 00
4098 765C 66 3E A3 stm r46,r76
4099 765F 76 C1 cmn R66,R76 compare to input
4100 7661 FB 04 snc
4101 7663 CE FF FF 9H jnb =err1+ input to large then
4102 7666 1B byt 27d
4103 7667
4104 7667 CE E7 76 JSB =CHECK check the option base
4105 766A 76 14 05 02 4H LDM R66,X24,MAXLEN get the len of str
4106 766E 00
4107 766F 89 ICR R66 incren twice for the
4108 7670 89 ICR R66 length field
4109 7671 CE FF FF JSB =INIMUL go multiply those
4110 7674 6E 2E A1 LDM R56,R54 answer to r56
4111 7677 5E 14 05 0A 7H LDM R34,X24,STRLOC get the string loc
4112 767B 00
4113 767C 42 A5 LDM R2,R24 get the name form
4114 767E 43 90 ISB R3 is this remote
4115 7680 F5 06 IFNG yes,
4116 7682 5C D3 51 82 ADD R34,=FUWARS get the ptr to env
4117 7686 1E A5 LDM R34,R34 load the remote addr
4118 7688
4119 7688 5C D3 51 82 ADD R34,=FUWARS make ptr absolute
4120 768C 2E C3 ADD R34,R56 add the offset to array
4121 768E A3 STM R34,R56 r56 has base loc
4122 7690 54 0A E3 pond r24,-r12
4123 7692 6E DRP 56
4124 7693 CE FF FF JSB =PUTADR put relativized
4125 7696 54 0A E5 PUND R24,=R12 put to stack
4126 7699 76 88 DEN R66 decrement the
4127 769B 8B DEN R66 string len by two
4128 769E 00 PUND R66,=R12 put string length
4129 76A0 00 ICR R34 r34 ptr to str base
4130 76A5
4131 76A5
4132 76A5

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG118
*****
>>>>>> DIN2R store 2-dim string array <<<<<<
4134 76A5
4135 76A5
4136 76A5
4137 76A5 1A BYT 32
4138 76A6 CE FF FF DIN2R JSB =IMOB get two values
4139 76A9 CE D0 76 JSB =TEST2 test for neg subscr
4140 76AC F8 F6 REN return if neg
4141 76AE CE F1 76 JSB =MYSUB check max col
4142 76B1 F7 80 JEQ 98 error if 1-dim
4143 76B3 DMP 166 set drp
4144 76B3 26 C1 CMN R66,R46 compare to input col value
4145 76B5 FA AC JMC 98 input to large jump back
4146 76B7 14 B5 04 00 LDM R66,X24,MAXROW get the maximum row val
4147 76BB 2E C1 CMN R66,R56 compare to input row value
4148 76BD FA A4 JMC 98 input to large jump back
4149 76BF CE FB 76 JSB =FIN11 get the max col back
4150 76C2 7E 2E A1 LDM R76,R56 load for INIMUL
4151 76C5 CE E7 76 JSB =CHECK check option base
4152 76C8 F5 02 IFNG if option base 0
4153 76CA 76 89 ICR R66 then increment the
4154 76CC EMDIF max col value
4155 76CC CE FF FF JSB =INIMUL multiply those bears
4156 76CF 7E 2E A1 LDM R76,R54 load answer to r76
4157 76D2 50 91 TSM R20 test for option base 1
4158 76D4 F4 02 IFPS
4159 76D6 66 8B DCN R46 if so dec the col
4160 76D8 EMDIF
4161 76D8 7E 26 C3 RDM R76,R46 add the offset into
4162 76DB F0 8D JNP 4B the row and finish up
4163 76DD
4164 76DD
4165 76DD 6E 91 TEST2 TSM R56 test the col number
4166 76DF F4 82 FOR JMC 9b jump if neg input
4167 76E1 90 TEST1 CLE signal all okay
4168 76E2 66 91 TSM R46 test the row value
4169 76E4 F4 F9 JMG FOR jump if neg input
4170 76E6 9E RTH return positive
4171 76E7
4172 76E7 50 14 05 08 CHECK LDM R20,X24,TOTLEN get the total length
4173 76EB 00
4174 76EC F4 F8 RMG return if option base 0
4175 76EE 7E 8B DCN R76 else decrement for 1
4176 76F0 9E RTH
4177 76F1
4178 76F1 54 0A E3 MYSUB POND R24,=R12 rel ptr to VPA
4179 76F4 E5 PUND R24,=R12 rel ptr to VPA
4180 76F5 D3 47 82 ADD R24,=RMFILE make it absolute
4181 76F8 CE 26 76 jnb =track
4182 76FB 76 14 05 06 FIXIT LDM R66,X24,MAXCOL get the MAX COL
4183 76FF 00
4184 7700 C9 FF FF CMN R66,=377,377 compare to non-init
4185 7703 9E RTH

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG117
*****
>>>>>> DIN1R store 1-dim string array <<<<<<
4127 769F 89 ICR R34 past the len field
4128 76A0 CE FF FF JSB =PUTREL relativise and put on the
4129 76A3 90 CLE all okay
4130 76A4 9E RTH r12 stack
4131 76A5
4132 76A5

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG119
*****
>>>>>> DIN2R store 2-dim string array <<<<<<
4185 7704

```

NOTAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG120
*****
>>>>>> DIMIV. string array value 1-dim <<<<<<<
4187 7704
4188 7704 1A BY1 J2
4189 7705 CE 4A 76 DIMIV. JSB =DIMIA. go do the messy stuff
4190 7708 F8 F9 5h REN return if errors
4191 770A CE FF FF JSB =GE100R get the addr that dimia
4192 770D 52 1C E3 POND R22,-R34 load the act length
4193 7710 E5 POND R22,+R34 get r34 back to the string
4194 7711 C9 FF FF CMI R22,-377,377 compare to non-init len
4195 7714 F6 06 sfaq
4196 7716 CE FE EF job error
4197 7719 07 byt 7d
4198 771A 52 C cfm r22
4199 771C endif
4200 771E 62 0A E3 BLACH1 POND R42,-R12 clean up stuff from DIMIA.
4201 771F 52 E5 POND R22,+R12 push r20
4202 7721 CE FF FF JSB =PUTREL relativise the pointer
4203 7724 9E RTN
4204 7725

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG122
*****
>>>>>> entries and externals <<<<<<<
4211 772B
4212 772B * entry points
4213 772B
4214 772B
4215 772B

```

NOMAS

Not Manufacturer Supported
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG121
*****
>>>>>> DIM2V. string array value 2-dim <<<<<<<
4206 7725
4207 7725 1A BY1 J2
4208 7726 CE A6 76 DIM2V. JSB =DIM2A. do the two dim stuff
4209 7729 F0 D0 JNP 5h jump back and out

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG123
*****
>>>>>> COMMON, the runtime routine for common <<<<<<<
4217 772B
4218 772B *****
4219 772B *
4220 772B * COMMON: creates the .common environment if not already
4221 772B * created, copies the necessary VPA entries and
4222 772B * and variable values into the .common env
4223 772B *
4224 772B * INPUT: R10 -- ptr to common list
4225 772B *
4226 772B * OUTPUT: .common file updated
4227 772B *
4228 772B * VOLATILES: R14/15, R20, R30/31, R34/37, R40/47
4229 772B *
4230 772B * EXT ROUTINES: FCREAT, CREATE, MYCALC, CHKENV,
4231 772B * MYCOPY
4232 772B *
4233 772B * ERRORS: none ; reported in chkenv
4234 772B *
4235 772B *****
4236 772B 63 6F 6D CHNAME ASC 'common '
4237 772F 6F 6E 20 20
4238 7733
4239 7733 A1 BY1 241
4240 7734 98 COM. BIN set bin mode
4241 7735 DRP 120 drp is 20 from interp
4242 7735 93 CLM R20 clear for file type
4243 7736 60 B1 20 77 LDMD R40,=CHNAME load the file name
4244 773A CE FF FF JSB =FCREAT create the file
4245 773D 53 18 A5 LDMD R30,R30 file pointer
4246 7740 F8 0E JFEZ created, so
4247 7742 CE 8B 77 JSB =CREATE build VPA and environ
4248 7745 F8 D0 REN not enough room
4249 7747 92 CLB RN clear flag
4250 7748 B2 5E 82 STBD RN,=FLAG1 flag copy to prevw
4251 774B 8A DCB RN flag is FF
4252 774C 1E E4 PUSD RN,+R36 end of VPA flag
4253 774E F0 0A ELSE is file there
4254 7750 9B DEC E=1, file already there
4255 7751 F8 D1 REN E=2, not enough room
4256 7753 42 92 CLB R2 clear flag
4257 7755 8B ICB R2 flag non-zero
4258 7756 88 lcb r2
4259 7757 B2 5E 82 STBD R2,=FLAG1 flag copy to env
4260 775A ENDF
4261 775A 58 89 COM- ICM R30 get the file ptr past
4262 775C 89 ICM R30 the two bytes for DALLDC
4263 775D 1E A3 STM R30,R36 file is guaranteed at R30
4264 775F DB RMD R30,R36 R36 pointer to environment
4265 7760 5E 89 ICM R36 increment twice

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG124
*****
>>>>>> COMMON, the runtime routine for common <<<<<<<<
4266 7762 B9 ICM R36 to get past env ptr
4267 7763 4C 08 R1 LDM R14,R10 save r10
4268 7766 0D 04 00 SBN R14,=FOUR point back to common tk
4269 7769 50 08 E0 NEXTK PODB R20,=R10 get the token
4270 7761 C8 0E CMB R20,=EOL10K end of line?
4271 7761 F7 13 JED SETR10 yes then reset r10
4272 7770 CE 58 78 JSB -CHKENV check the match
4273 7773 F8 12 JEN PURGE a mismatch set r10 to common
4274 7775 5C 12 R5 LDM R34,R22 r34 rel ptr
4275 7778 D3 51 R2 ARMD R34,=FIVARS r34 ptr to env loc
4276 7778 CE 91 78 JSB -NYCOPY copy from r34 to r30
4277 777E 58 1A E3 RDM R30,R32 update R30, copy restores
4278 7781 F0 E6 JMP NEXTK do next
4279 7783 50 08 E2 NEXTK PODB R20,=R10 reset r10 to coltok
4280 7786 9E RTN return
4281 7787 4C 08 R3 PURGE STN R14,R10 reset r10 to common
4282 778A 9E RTN return with error
4283 778B

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG126
*****
>>>>>> CREATE, builds the .common file <<<<<<<<
4335 77D4 00 C3 RDM R34,R0 env ptr to next env loc
4336 77D6 F0 E4 JMP NEXT do next token
4337 77D8 66 1E E5 UPPERS PUMB R46,=R36 put in the max len
4338 77D8 50 C8 C0 CMB R20,=EOCH str array?
4339 77DE FA 02 IFLY yes,
4340 77E0 6A E5 PUMB R52,=R36 put in row and col
4341 77E2 ENDTF no,
4342 77E2 F0 ED JMP PTR put in the env ptr etc.
4343 77E4

```

NOMAS

NOT Manufacturer Supported
Recipient agrees NOT to contact Manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG125
*****
>>>>>> CREATE, builds the .common file <<<<<<<<
4285 778D *****
4286 778D "
4287 778D "
4288 778D " CREATE: builds the .common file
4289 778D "
4290 778D " INPUT: R10 -- ptr to common list
4291 778D "
4292 778D " OUTPUT: creates the .common file with
4293 778D " the VPA validated, the env is not
4294 778D " touched.
4295 778D "
4296 778D " VOLATILES: R20, R24/25, R32/37
4297 778D "
4298 778D " EXT ROUTINES: MYCALL
4299 778D "
4300 778D " ERRORS: none
4301 778D "
4302 778D "
4303 778D 5A 93 CREATE CLM R32 clear alloc count
4304 778D 5C 93 CLM R34 clear VPA size
4305 778F 48 14 R3 STN R10,R24 R24 now the token ptr
4306 7792 LOOP get a token
4307 7792 50 14 E0 PODB R20,=R24 get the token
4308 7795 C8 0E CMB R20,=EOL10K end of line?
4309 7797 F7 0C JED CREAR1 build VPA
4310 7799 CE E4 77 JSB -MYCALL get the lengths
4311 779C 5A 00 C3 RDM R32,R0 add on the alloc size
4312 779F 02 C3 RDM R32,R2 add on the VPA size
4313 77A1 5C C3 RDM R34,R2 add on the VPA entry
4314 77A3 F0 ED JMP
4315 77A5 5A CB 05 00 CREAR1- RDM R32,=FIVE null, size and divider
4316 77A9 CE FF FF JSB -ALLDC alloc total size
4317 77AC F8 DC REM not enough room!
4318 77AE 58 1E R3 STN R30,R36 R36 will build VPA
4319 77B1 50 93 CLM R20 clear allocated flag
4320 77B3 E5 PUMB R20,=R36 put in for PIA10?
4321 77B4 5C CB 03 00 RDM R34,=VHREE R34 now has beginning of env
4322 77B8 E5 PUMB R34,=R36 put in the 1st byte of file
4323 77B9 48 14 R3 STN R10,R24 R24 now token pointer
4324 77BC 50 14 E0 NEXT PODB R20,=R24 get the token
4325 77BF C8 0E CMB R20,=EOL10K end of line?
4326 77C1 F7 E7 REQ all done
4327 77C3 CE E4 77 JSB -MYCALL get the information
4328 77C6 50 1E E4 PUMB R20,=R36 put in the var type
4329 77C9 FA 0D JNG UPPERS jump if a string
4330 77CD C8 02 CMB R20,=EOH as it a numeric array?
4331 77CD FA 02 IFLY yes,
4332 77CF 6A E5 PUMB R52,=R36 put in row and col
4333 77D1 ENDTF no,
4334 77D1 5C 1E E5 PTR PUMB R34,=R36 put in the env ptr

```

```

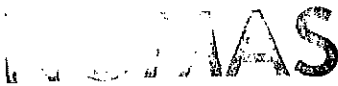
KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYZ103 12/13/1983 11:38 AM PG127
*****
>>>>>> MYCALL, gets necessary info on the var <<<<<<<<
4345 77E4 *****
4346 77E4 "
4347 77E4 "
4348 77E4 " MYCALL: examines the var token and returns the
4349 77E4 " necessary information to build the VPA
4350 77E4 "
4351 77E4 " INPUT: the arp must point at var pointer
4352 77E4 "
4353 77E4 " OUTPUT: R0/1 -- env length
4354 77E4 " R2/3 -- VPA length
4355 77E4 " R20 -- var type
4356 77E4 " R22/23 -- ptr to VPA entry in prgm
4357 77E4 " R46/47 -- max len if a string
4358 77E4 " R52/57 -- len, row, col if num array
4359 77E4 " R52/57 -- row, col, len if str array
4360 77E4 "
4361 77E4 " VOLATILES: R0/1, R2/3, R20, R22/23, R46/47,
4362 77E4 " R52/57
4363 77E4 "
4364 77E4 " EXT ROUTINES: none
4365 77E4 "
4366 77E4 " ERRORS: none
4367 77E4 "
4368 77E4 "
4369 77E4 52 E1 MYCALL POMB R22,=R0 get the rel VPA addr
4370 77E6 D3 47 R2 ARMD R22,=RMFILE make absolute
4371 77E9 50 C8 03 CMB R20,=THREE compare the token
4372 77EC FB 20 IFNC simple numeric,
4373 77EE 12 A4 LDM R20,R22 get the type byte
4374 77F0 CF F0 00 RMA R20,=BOOFON isolate the type
4375 77F3 F6 06 IFZR real,
4376 77F5 40 A9 08 00 LDM R0,=EIGHT env is eight
4377 77F9 F0 0E ELSE int or short,
4378 77FB C8 20 CMB R20,=20H com to short
4379 77FD FB 06 IFNC nope int,
4380 77FF 40 A9 03 00 LDM R0,=THREE env is three
4381 7803 F0 04 ELSE yes short,
4382 7805 40 A9 04 00 LDM R0,=FOUR env is four
4383 7809 ENDTF
4384 7809 ENDTF
4385 7809 42 A9 03 00 LDM R2,=THREE VPA size is three
4386 780D 9E RTN
4387 780E ENDTF
4388 780E F6 11 IFEQ a string,
4389 7810 A8 80 LDM R20,=BOH set the type
4390 7812 66 12 B5 02 LDM R46,R22,1MD get max len
4391 7816 00
4392 7817 40 26 A1 LDM R0,R46 ready for env len
4393 781A 89 ICM R0 increment twice
4394 781B 89 ICM R0 for act len field

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG128
*****
>>>>>> MYCALC, gets necessary info on the var *****
4394 781C 42 A9 05 00 LDR R2,-FIVE VPA size is five
4395 7820 9E RTN
4396 7821 ENDF
4397 7821 C8 85 CMB R20,-ORSH numeric array?
4398 7823 FB 1F IFNC no str array,
4399 7825 52 E0 POND R22,-RAN get past the token
4400 7827 E1 POND R22,-RAN get the rel addr
4401 7828 D3 47 82 ADDD R22,-RANLE make absolute
4402 7828 40 12 85 08 LDM R0,R22,EIGHT get the total size
4402 782F 00
4403 7830 EF FF 7F RAN R0,-87FFFH and out option bit
4404 7833 66 87 02 00 LDM R46,R22,TWO get the max len
4405 7837 6A 85 04 00 LDM R52,R22,FOUR get row col and len
4406 7838 42 A9 08 00 LDM R2,-ELEVEN VPA size is 11
4407 783F 50 A8 C0 LDR R20,-OLEOH set the type
4408 7842 F0 13 ELSE numeric array,
4409 7844 40 12 85 02 LDM R0,R22,TWO get the total len
4409 7848 00
4410 7849 CF FF 7F RAN R0,-87FFFH and out option bit
4411 784C 6A 85 02 00 LDM R52,R22,TWO get len row and col
4412 7850 42 A9 09 00 LDM R2,-NINE VPA size is 9
4413 7854 50 A8 40 LDR R20,-40H set the type
4414 7857 ENDF
4415 7857 9E RTN
4416 7858

```


 NOI MANUFACTURER SUPPORTED
 recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG130
*****
>>>>>> CHKEW, checks the var list *****
4468 7891 42 80 SE 82 MYCOPY LDD R2,-FLAG1 check source flag
4469 7895 86 lrb r2
4470 7896 90 lrb r2
4471 7897 F3 07 IFOD
4472 7899 58 10 A3 STR R30,R20 copy p-env to env
4473 789C 1C A1 LDM R30,R34 reverse the source
4474 789E 50 A3 STR R20,R34 and destination
4475 78A0 ENDF pointers
4476 78A0 CE FF FF JSB -COPY copy env to p-env
4477 78A3 42 80 SE 82 LDD R2,-FLAG1 go do the copy
4478 78A7 86 lrb r2 change source and dest
4479 78A8 90 lrb r2
4480 78A9 F3 E1 Rev
4481 78AB 58 10 A3 STR R30,R20 no
4482 78AC 1C A1 LDM R30,R34 yes swap the source
4483 78B0 50 A3 STR R20,R34 and destination
4484 78B2 9E RTN pointers
4485 78B3
4486 78B3 A1 BYT 241
4487 78B4 60 81 28 77 CMPURG LDM R40,-CNAME time to get
4488 78B8 CE FF FF JSB -FPLURG rld of the common
4489 78BB 9E RTN environment
4490 78BC

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG129
*****
>>>>>> CHKEW, checks the var list *****
4418 7858 *****
4419 7858 *
4419 7858 *
4420 7858 *
4421 7858 * CHKEW: makes sure the common list matches
4422 7858 * the VPA in .common
4423 7858 *
4424 7858 * INPUT: arp must be pointing to the var pointer
4425 7858 *
4426 7858 * OUTPUT: E0 then mismatch
4427 7858 * E0 then
4428 7858 * R22 -- ptr to var env ptr
4429 7858 * R36 -- ptr to next vpa entry in .common
4430 7858 * R32 -- size of environment to move
4431 7858 *
4432 7858 * VOLATILES: R2/3, R22/23, R36/37, R50/51
4433 7858 *
4434 7858 * EXT ROUTINES: ERR1+, MYCALC
4435 7858 *
4436 7858 * ERRORS: type mismatch
4437 7858 *
4437 7858 *****
4438 7858
4439 7858 90 CHKEW CLE
4440 7859 CE E4 77 JSB -MYCALC get the info
4441 785C 50 1E D8 CMBD R20,R36 get the info
4442 785F F6 2C JNE ER151 type match
4443 7861 5E 89 IEM R36 nope, error
4444 7863 42 C8 05 CMB R2,-FIVE point to var info
4445 7866 FA 1A JNC SETUP type str or num
4446 7868 F6 06 IFEQ num so done
4447 786A 66 D9 CMBD R46,R36 str,
4448 786C F6 1F JNE ER151 compare max lens
4449 786E F0 12 ELSE not equal, error
4450 7870 C8 0B CMB R2,-ELEVEN else,
4451 7872 FB 06 IFNC str array?,
4452 7874 6A D9 CMBD R52,R36 no, num array
4453 7876 F6 15 JNE ER151 compare info
4454 7878 F0 08 ELSE a mismatch
4455 787A 66 28 A3 STR R46,R50 str array,
4456 787D 68 1E D9 CMBD R50,R36 move tot len
4457 7880 F6 08 JNE ER151 compare info
4458 7882 ENDF nope, error
4459 7882 ENDF
4460 7882 42 88 SETUP DCM R2 dec the VPA size
4461 7884 5E 02 C3 RDM R36,R2 update r36 to next
4462 7887 52 C3 RDM R22,R2 R22 to rel ptr
4463 7889 5A 00 A1 LDM R32,R0 R32 has env size
4464 788C 9E RTN
4465 788D CE FF FF ER151 JSB -ERR1+
4466 7890 7E BYT 126D
4467 7891

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&I03 OBJ=RY&I03 12/13/1983 11:38 AM PG131
*****
>>>>>> entry points and externals *****
4492 78BC * entry points
4493 78BC
4494 78BC
4495 78BC
4496 78BC

```

[illegible]

```

KAMMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY&ID3 OBJ=RY&ID3 12/13/1983 11:38 AM PG135
-----
>>>>>> CHRM., runtime routine for chain
4652 79E8 CE FF FF job =NELJSB
4653 79E8 FF FF def VFHI establish conn
4654 79E8 FF 18 jen trymen no good out
4655 79E8 CE FF FF job =NELJSB
4656 79F2 FF FF def FUFIND find the file
4657 79F4 9A SRO
4658 79F5 CE FF FF JSB =NELJSB
4659 79F8 FF FF DEF VFBYE
4660 79FA 9F PAD
4661 79FB F6 0A ifzr
4662 79FD 42 06 E3 CLEARI pond r2,-r6
4663 7A00 64 E3 pond r44,-r6
4664 7A02 1A A3 STA R44,R32
4665 7A04 40 92 clb r0
4666 7A06 9E rtn
4667 7A07 ELSE
4668 7A07 CE FF FF TRYMEN JSB =CLRERR
4669 7A0A 42 06 E3 POND R2,-R6
4670 7A0B 64 E3 POND R44,-R6
4671 7A0F E5 POND R44,+R6
4672 7A10 42 E5 POND R2,+R6
4673 7A12 64 1A A3 STA R44,R32
4674 7A15 5A 0A E5 POND R32,+R12
4675 7A18 CE FF FF JSB =PUFREL
4676 7A1B CE FF FF JSB =HANDI
4677 7A1E 2F VAL V.EXST
4678 7A1F 3E BYT 62D
4679 7A20 4F 90 TSB R17
4680 7A22 F5 D9 JPS CLEARI
4681 7A24 endfi
4682 7A24 62 06 E3 clrmc pond r42,-r6
4683 7A27 9E rtn
4684 7A28 got it return
clear

```



```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG136
=====
>>>>>> CALL, entry points and externals <<<<<<<
4686 7A28
4687 7A28
4688 7A28
4689 7A28
4690 7A28
4691 7A28
4692 7A28 21 BYT 41
4693 7A29 ZUTS
4694 7A29 OUR(.
4695 7A29 9E OUR). RTN
4696 7A2A
4697 7A2A 10 20 BYT 20,55
4698 7A2C SSWAL.
4699 7A2C 9E SSWAL. RTN
4700 7A2D
4701 7A2D EF BYT 357
4702 7A2E 78 81 CA 71 MSUB. ldrd r70,=ROMADR get the job from job
4703 7A2E 50 A9 FF FF ldrd r20,=SKIPD to routine SKIPD
4704 7A2E 38 A3 stn r20,r73 load in skipd
4705 7A2E B1 51 82 ldrd r20,=FIVARMS get the env block
4706 7A2E 10 B5 0C 00 ldrd r20,x20.E.ROMM get the basic in ROM #
4707 7A2F 30 A3 stn r20,r75 store into code
4708 7A2F A9 64 84 ldr r20,=tmpw3 get the location
4709 7A2F 78 B3 64 84 stnd r70,=tmpw3 store code to ram
4710 7A2F CE 64 84 job =TMPW3 go do that stuff
4711 7A2F 9E rtn should be enough
4712 7A2C
4713 7A2C
4714 7A2C
4715 7A2C
4716 7A2C
4717 7A2C
4718 7A2C
4719 7A2C
4720 7A2C
4721 7A2C
4722 7A2C
4723 7A2C 40 B0 54 84 GETCAL ldrd r0,=TOKEN get the current token
4724 7A2C 18 4F cnd r0,=CALLTK is this a call
4725 7A2C F7 03 ifne no,
4726 7A2C A8 ldr r0,= then reset the
4727 7A2C 31 val V.FILE the HANDI event
4728 7A2C 9E rtn
4729 7A2C endif
4730 7A2C
4731 7A2C 6C 34 A3 SIN r54,r64
4732 7A2C
4733 7A2C CE FF FF job =RESEDM reserve memory
4734 7A2C 0D byt 13d 11 bytes for file spec
4735 7A2C F8 F6 ren no memory return

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG138
=====
>>>>>> ALLOCATION for SUB statement <<<<<<<
4774 7A22
4775 7A22
4776 7A22
4777 7A22
4778 7A22
4779 7A22
4780 7A22
4781 7A22
4782 7A22
4783 7A22 50 10 A5 D3 VPRCDD BYT 50H,10H,ORSH,003H,47H,82H
4784 7A22 47 82
4785 7A22 85 02 00 9E LODCDD BYT 0B5H,02H,00H,9EH
4786 7A22 50 10 A5 D3 TYPEDD BYT 50H,10H,ORSH,003H,47H,82H,ORSH,9EH
4787 7A22 47 82 A5 9E
4788 7A22 65 08 A5 9E HEPR BYT 65H,08H,ORSH,9EH
4789 7A22

```

NOMAS

NOT MANUFACTURER SUPPORTED
recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG137
=====
>>>>>> CALL, entry points and externals <<<<<<<
4736 7A60
4737 7A60 40 A8 20 ldr r0,=40 set up transfer reg
4738 7A63 56 14 A3 stn r26,r24 set r24 to dest
4739 7A66
4740 7A66
4741 7A66 42 01 A0 loop
4742 7A69 C8 20 ldr r2,r4 get a char
4743 7A69 F7 08 cnd r2,=BLANK if blank then
4744 7A6A 14 E4 jng gotnam we are at end of name
4745 7A6A 40 88 pubd r2,r24 else push char
4746 7A71 C8 28 scb r0 anc transfer reg
4747 7A73 F6 F1 cnd r0,=50 eight chars
4748 7A75
4749 7A75 42 A8 3A gotnam ldr r2,=: put a colon
4750 7A78 14 E4 pubd r2,r24 into the string
4751 7A7A 24 E5 PUND R64,=R24
4752 7A7C 54 16 C5 sbn r24,r26 get the length
4753 7A7F 0A E5 punb r24,r12 onto param stack
4754 7A81 56 E5 punb r26,r12 and the loc (source)
4755 7A83 54 CD 05 00 sbn r24,=FIVE sbn for dest length
4756 7A87 E5 punb r24,r12 onto the param stack
4757 7A88 56 E5 punb r26,r12 and the loc(dest)
4758 7A8A 40 A8 7C ldr r0,=COPYIK get the copy token
4759 7A8D B2 54 84 stbd r0,=TOKEN store into TOKEN for SEIH
4760 7A90 60 06 E5 punb r40,r6 go do the
4761 7A93 CE FF FF job =SYSJSB copy into RAM
4762 7A96 FF FF def COPY.
4763 7A98
4764 7A98 CE 55 62 JSB =HANDLE
4765 7A98
4766 7A98
4767 7A98 CE 06 75 JSB =SEICTX
4768 7A98
4769 7A98 60 06 E3 pond r40,=-r6
4770 7A91 9E rtn
4771 7A92
4772 7A92

```

```

KARMA 08/31/82=====
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RY&103 12/13/1983 11:38 AM PG139
=====
>>>>>> MPTRLO, ptr allocation routine <<<<<<<
4790 7A88
4791 7A88
4792 7A88
4793 7A88
4794 7A88
4795 7A88
4796 7A88
4797 7A88
4798 7A88
4799 7A88
4800 7A88
4801 7A88
4802 7A88
4803 7A88
4804 7A88
4805 7A88
4806 7A88 4C B0 5E 82 VORTIT LDBD R14,=FLAG1 caller in ROM?
4807 7A8C 84 11b r14
4808 7A8D 90 11b r14
4809 7A8E F5 0F ifng yes,
4810 7A8C 53 06 E4 pubd r23,r6
4811 7A8C 7C B1 B4 7A ldrd r74,=hepr
4812 7A8C CE 8F 75 job =STF
4813 7A8C 53 06 E2 pubd r23,-r6
4814 7A8C F0 03 ELSE
4815 7A8C 65 08 A5 ldrd r45,r10 get a token
4816 7A8C ENDIF
4817 7A8C 9E RTN
4818 7A8C
4819 7A8C 9D RORVAL CLE clear flag
4820 7A8C CE B8 7A JSB =VORTIT
4821 7A8C 65 DRP 45
4822 7A8C CE FF FF JSB =CREQL?
4823 7A8C F9 03 IFEM
4824 7A8C 9D CLE
4825 7A8C F0 03 ELSE
4826 7A8C 9D 9C STE
4827 7A8C 9E RTN
4828 7A8C ENDIF
4829 7A8C CE B8 7A VORT JSB =VORTIT
4830 7A8C 48 89 sch r10
4831 7A8C 65 C9 cnd r45,=
4832 7A8C B4 val aronk
4833 7A8C 65 40 def nyronk
4834 7A8C F6 F4 jne vort
4835 7A8C 4C 90 11b r14
4836 7A8C F5 12 ifng
4837 7A8C 53 06 E4 pubd r23,r6
4838 7A8C 7C B1 B4 7A ldrd r74,=hepr
4839 7A8C 7E A8 E1 ldr r76,=OEth

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG140
*****
***** MPTRLO, ptr allocation routine *****
4840 780D CE 8F 75 job =SIF
4841 780D 53 06 E2 pabd r23,-r6
4842 7803 FO 03 else
4843 7805 65 08 E1 pabd r45,-r10
4844 7808 endif
4845 7808 67 E8 2A cnb r47,-parval
4846 7808 FA 06 jnc vort
4847 780D C8 31 cnb r47,-01K
4848 780F F6 02 IFED yes,
4849 7811 9C IFED signal end of params
4850 7812 9C RIN
4851 7813 ENOIF
4852 7813 C8 2E CMB R47,-SEMPVAL compare to value
4853 7815 40 dmp 0
4854 7816 F8 03 IFNC no, reference
4855 7818 92 CLD R0 clear r0 for test
4856 7819 FO 02 ELSE yes, value
4857 7818 A8 01 LDB R0,-DNE set r0 for test
4858 781D ENOIF
4859 781D 90 TSB R0 test to set Z or NZ
4860 781E 9E RIN
4861 781F ENDOF
4862 781F 53 14 E2 ENDOF? PDBD R23,-R24 put back r24
4863 7822 B0 5E 82 LDBD R23,-FLAG1 get the ROM flag
4864 7825 F4 06 JNG SET3P rtn if basic in ROM
4865 7827 CE 03 7A JSB =RORVAL get the next token
4866 782A F9 0B JET zzzzzz e=0 if not eol
4867 782C 90 CIE clear e
4868 782D 42 93 SET3P CMB R2
4869 782F B3 91 83 SIMD R2,-ERRSTP
4870 7832 9E RIN
4871 7833 CE FF FF ER154 =ERR1
4872 7836 81 BVT 1290 return
4873 7837
4874 7837 44 A9 E0 7B zzzzzz gto err151
4875 7838
4876 7838 CE 4A 75 MPTRLO JSB =CHKRON load up for compare
4877 783E 42 A8 4F LDB R2,-callth load up for compare
4878 7841 D0 54 84 CMBD R2,-TOKEN make the compare
4879 7844 F6 03 JFCQ if not equal then
4880 7846 92 CLD R2 clear for flag
4881 7847 FO 02 else
4882 7849 A8 80 ldb r2,-80h
4883 784B ENOIF
4884 784B 40 B0 5E 82 LDBD R0,-FLAG1
4885 784F 02 94 orb r0,r2
4886 7851 B2 5E 82 stbD R0,-FLAG1
4887 7854 42 90 tdb r2
4888 7856 F6 0C stfr
4889 7858 40 B1 51 82 LDBD R0,-FUNARS jump if we are calling
4890 785C 48 00 85 16 LDBD R10,-D.E.RIN get funars
4891 7860 00 ADDD R10,-RNFIL make it absolute

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG142
*****
***** MPTRLO, ptr allocation routine *****
4945 780D 52 92 plus6 c1b r22
4946 780D CE FF FF job =SYSJSB
4947 780E FF FF def VSTOR
4948 780E 54 89 VNFIX ich r24
4949 7804 7E 01 8F 83 ldd r76,-varptr
4950 7808 2E C5 sbm r76,-r56
4951 780A C8 02 00 adn r76,-two
4952 780D B3 8F 83 stnd r76,-varptr
4953 780F 40 1E A1 arrend ldn r0,-r36
4954 780F C0 08 00 sbi r0,-height
4955 780F F0 B8 set- jmp setpgm
4956 780F
4957 780F F0 6A tok3+ jmp tok3
4958 780F
4959 780F F7 66 TOKB5 jng tokb4+ save token
4960 780F 42 B0 5E 82 ldd r2,-flag1
4961 780C F4 BF er151- jng er151
4962 7802 53 06 E4 pabd r23,-r6
4963 7805 A8 02 LDB R23,-SAVTON load array token
4964 7807 CE 74 7D JSB =nanfrm go form name
4965 780A 53 06 E2 PDBD R23,-R6 restore token
4966 780C 0D 03 7A JSB =RORVAL ref or val
4967 7810 F8 BD er151a JEN ER151 error is eol
4968 7812 67 C8 2C CMB R47,-AREFTK compare to aref th
4969 7815 F6 BD er151b JNE ER151 error if not
4970 7817 CE 15 7E JSB =CKREAL check for real type
4971 781A F6 BD jnz ER154b EMO then not real
4972 781C 50 08 A1 LDM R20,R10 get ptr to call prgm
4973 781F C0 06 00 SBH R20,-R6 r20 ptr to var ptr
4974 7822 40 B0 5E 82 LDBD R0,-FLAG1 caller in ROM?
4975 7826 B4 lib r0
4976 7827 90 tdb r0
4977 7828 F5 05 IFNG yes,
4978 782A CE D9 7D JSB =HUM2 do the following in ram
4979 782D F0 0A ELSE no,
4980 782F 50 10 A5 LDBD R20,R20 get the ptr to r20
4981 7832 D3 47 82 ADDD R20,-RNFIL make absolute
4982 7835 7A B5 02 00 LDBD R72,R20,TUD get the array info
4983 7839 ENOIF
4984 7839 5E 38 A3 BACKEN STM R36,R70 store the name
4985 783C 40 A8 38 LDB R0,-70 set up lfer reg
4986 783F 6E A9 02 00 LDM R56,-62 set env size to two
4987 7843 50 A9 08 00 LDM R20,-68 vpa size is eight
4988 7847 53 90 TSB R23 test the token
4989 7849 F3 0B IFOD 1-dim,
4990 784B 7E C9 FF FF CMB R76,-OFFH,OFFH check for bad dim
4991 784F F6 C4 er151c JNE ER151b error if not 1-dim
4992 7851 F0 0A ELSE 2-dim
4993 7853 7E C9 FF FF CMB R76,-OFFH,OFFH check for bad dim
4994 7857 F6 04 ifeq
4995 7859 90 9C er151v etc
4996 785B F0 B3 er151d jmp ER151a error if 1-dim
4997 785D endif

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG141
*****
***** MPTRLO, ptr allocation routine *****
4892 7864
4893 7864 CE 97 7D AFSE1 JSB =NAME? R10 now ptr to calling prgm
4894 7867 F8 C9 REN check name and loc
4895 7869 54 C8 04 00 ROM R24,-FOUR get past the zuts token
4896 786D 53 14 E0 CONH1? PDBD R23,-R24 get a token
4897 7870 54 B3 91 83 SIMD R24,-ERRSTP get a token
4898 7874 53 C8 0E CMB R23,-EDLTOK end of line?
4899 7877 F7 A6 JEO ENDOF? jump if no
4900 7879 C8 04 cnb r23,-zeronth
4901 787B F8 7D jcy TOKB5
4902 787D 53 C8 02 TOK1 cnb r23,-two
4903 7880 F8 43 jcy tok2
4904 7882 CE 74 7D JSB =nanfrm form internal name
4905 7885 F4 20 JNG MVAL if in ROM then value
4906 7887 CE 03 7A JSB =RORVAL in ram so check val vs ref
4907 788A F8 35 JEN ER151 error, hit eol
4908 788C F6 10 IFZR reference.
4909 788E 67 C8 2A CMB R47,-SREFTK compare to num ref token
4910 7891 F6 2E JNE ER151 nope then error
4911 7893 CE 15 7E JSB =CKREAL check for real type
4912 7896 F6 9B jnz ER154 EMO then not real
4913 7898 6E A9 02 00 LDM R56,-TUD else, env size is two
4914 789C F0 0D ELSE value.
4915 789E 67 C8 2E CMB R47,-MVALTK compare to val token
4916 78A1 F7 04 JED MVAL yes then process
4917 78A3 C8 33 CMB R47,-R1K no, compare to (R)
4918 78A5 F6 1A JNE ER151 no, then error
4919 78A7 6E A9 0A 00 MVAL LDM R56,-TEN yes, size is ten
4920 78AB ENOIF
4921 78AB CE FF FF JSB =R10SHV go to mainframe also routine
4922 78AE 40 1E A1 LDM R0,-R36 get the vpa pointer
4923 78B1 8B DCM R0 decrement twice to
4924 78B2 8B DCM R0 pt to vpa entry
4925 78B3 4F 90 SETPGH tab r17
4926 78B5 F5 01 ilng
4927 78B7 9E rtn
4928 78B8 ENOIF
4929 78B8 40 D5 43 82 SBMD R0,-PRFILE make rel to file
4930 78BC BB 91 83 strk r0,-errstp put into program
4931 78BF FO AC jmp cont?
4932 78C1
4933 78C1 CE FF FF ER151 jng =err1
4934 78C4 7E byt 126d
4935 78C5
4936 78C5 F6 31 TOK2 jng tok3+
4937 78C7 CE 74 7D jng =nanfrm
4938 78CA F4 0F jng plus6
4939 78CC CE D3 7A jng =rorval
4940 78CF FB FO er151e jng er151
4941 78D1 67 C8 2C cnb r47,-areftk
4942 78D4 F6 EB er151n jne er151
4943 78D6 CE 15 7E jsb =ckreal
4944 78D9 F6 BB er154b jnz er154a

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RY8103 OBJ=RYX103 12/13/1983 11:38 AM PG143
*****
***** MPTRLO, ptr allocation routine *****
4998 78C0
4999 78C0 CE FF FF JSB =RLVENT go to main frame routine
5000 78C0 F0 8E arr- jmp arrend get the vpa ptr
5001 78C2
5002 78C2 F0 66 tokb4+ jmp tokb4
5003 78C4
5004 78C4 CE 74 7D TOK3 jng =nanfrm
5005 78C7 F4 38 JNG STVAL in ram then value
5006 78C9 CE 03 7A JSB =RORVAL text for val or ref
5007 78CB F8 ED ER151f JEN ER151d error, hit eol
5008 78CE F6 2B IFZR reference
5009 78D0 67 C8 2B er151h JNE ER151c compare to ref tok
5010 78D3 F6 DA jsb =INIEG no, then error
5011 78D5 CE 82 7D jnz c1n56
5012 78D7 F7 10 LDM R20,R10 get the call ptr to r20
5013 78DA 50 08 A1 SBH R20,-R6 r20 ptr to var ptr
5014 78DD C0 06 00 LDBD R0,-FLAG1 caller in ROM?
5015 78E0 40 B0 5E 82 lib r0
5016 78E4 B4 tdb r0
5017 78E5 90 IFNG yes,
5018 78E6 F5 05 JSB =HUM1 do the following in ram
5019 78E8 CE D4 7D ELSE no,
5020 78EB F0 0A LDBD R20,R20 r20 has var ptr
5021 78ED 50 10 A5 ADDD R20,-RNFIL now absolute to VPA
5022 78F0 D3 47 82 LDBD R76,R20,TUD get the str len
5023 78F3 7E B5 02 00 ENOIF
5024 78F7 C1M R56 clear for remote size
5025 78F9 6E 93 ELSE value,
5026 7899 F0 19 CMB R47,-SVALTK str val token?
5027 789B 67 C8 2F JEQ STVAL yes go do it
5028 789E F7 04 CMB R47,-R1K (R6) token?
5029 78A0 C8 34 er151i JNE ER151h no then error
5030 78A2 F6 CF jng =INIEG default str len
5031 78A4 CE 82 7D strval ldn r76,-STRLEN
5032 78A7 F7 04 alnz
5033 78A9 7E A9 20 00 LDM R76,-STRLEN
5034 78AD ENOIF
5035 78AD 7E 2E A3 STM R76,R56 load to env size
5036 78B0 6E C8 02 00 ADDD R56,-62 add two for act len
5037 78B4 ENOIF
5038 78B4 C8 02 00 ROM R56,-62 add two for remote addr
5039 78B7 50 A9 04 00 LDM R20,-64 VPA size is four
5040 78BA 40 A8 3C LDB R0,-74 R74 is transfer reg
5041 78BC 5E 3C A3 STM R36,R74 store the name
5042 78C1 CE FF FF JSB =RYCALL go to mainframe routine
5043 78C4 5E C8 04 00 adn r36,-four get the vpa ptr
5044 78C8 F0 96 jmp arr- pt to vpa entry
5045 78CA
5046 78CA
5047 78CA
5048 78CA 42 14 E1 TOKB4 PDBD R2,-R24 trash rom W
5049 78CD 53 E0 PDBD R23,-R24 get str array token
5050 78CF 54 B3 91 83 SIMD R24,-ERRSTP

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RYX103 12/13/1983 11:38 AM PG144
*****
>>>>>>> MPTRLO, ptr allocation routine <<<<<<<<<
5051 7C03 53 C8 31 CMB R23,=0(1K
5052 7C06 F6 08 IF00
5053 7C08 54 C8 04 00 RDM R24,=R4
5054 7C0C 44 A9 6C 78 gto cont17
5055 7CE0 ENDF
5056 7CE0 06 E4 PDBD R23,=R6 save token
5057 7CE2 R8 03 LDB R23,=THREE fool namer load str
5058 7CE4 CE FF FF JSB =FRANM form str name
5059 7CE8 53 06 E2 PDBD R23,=R6 get back token
5060 7CEA 42 A3 40 40 LDM R2,=40H,40H get array and remote bits
5061 7CEE 1E 96 RDM R2,=R36 or into name
5062 7CF0 A3 STM R2,=R36
5063 7CF1 80 5E 82 LDBD R2,=FLAG1
5064 7CF4 F4 0A JNG FLAG1
5065 7CF6 CE 03 7A JSB =R0RVNL get the type ref or val
5066 7CF9 F8 57 JNE ER151J not so error
5067 7CFB 67 C8 20 CMB R47,=SARFTK str array ref token
5068 7CFE F6 A2 JNE ER151J no, error
5069 7D00 53 C8 21 ER151J
5070 7D03 F8 14 FLAG1
5071 7D05 52 32 ifnc
5072 7D07 CE E2 70 clb r22
5073 7D0A JAB =STRARR
5074 7D0A CE FF 07 DRP R56
5075 7D0D 42 14 A4 ANN R56,=807FFH
5076 7D10 C8 10 LDBD R2,=R24
5077 7D12 F7 01 CMB R2,=10H
5078 7D14 E2 ifnc
5079 7D15 PDBD R2,=R24
5080 7D15 44 A9 E1 78 ENDF
5081 7D19 gto varfix
5082 7D19 42 80 5E 82 ENDF
5083 7D1D F4 33 LDBD R2,=FLAG1 get the rom flag
5084 7D1F 50 08 A1 JNG ER151J jump if in rom
5085 7D22 C0 06 00 LDM R20,R10 get the call prgm ptr
5086 7D25 40 80 5E 82 SBM R20,=R6 r20 ptr to var ptr
5087 7D29 B4 LDBD R0,=FLAG1 caller in ROM?
5088 7D2A 90 JLB R0
5089 7D2B F5 05 TAB R0
5090 7D2C CE F1 70 IFNG JSB =MUM3 yes,
5091 7D30 F0 0E ELSE go do following in ram
5092 7D32 50 10 R5 LDM R20,R20 no,
5093 7D35 03 47 82 RDM R20,=RNFIL R20 <-- var ptr
5094 7D38 7A 05 02 00 LDM R72,R20,TWD now absolute
5095 7D3C 6E 05 08 00 LDM R56,R20,EIGHT get the str array info
5096 7D40 ENDF get the total length
5097 7D40 53 90 THRU LDB R23 test the dimension
5098 7D42 F3 08 RF0D 1-dim
5099 7D44 7E C9 FF FF CMB R76,=OFFH,OFFH col=ffff
5100 7D48 F6 04 JNE ER151J no, error
5101 7D4A F0 0A ELSE 2-dim
5102 7D4C 7E C9 FF FF CMB R76,=OFFH,OFFH col=ffff
5103 7D50 F6 04 IFNG

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RYX103 12/13/1983 11:38 AM PG146
*****
>>>>>>> MPTRLO, ptr allocation routine <<<<<<<<<
5157 7DC3 A8 20 LDB R0,=40 reg to r0
5158 7DC5 CE FF FF JSB =UPRC* upper case name
5159 7DC8 60 D1 63 82 CMB R40,=PRNAME compare to filename
5160 7DCE F6 02 JNE ER153 okay return
5161 7DCE 90 CLE success flag
5162 7DCF 9E RTN
5163 7D00 CE FF FF ER153 JSB =ERR1* report error
5164 7D03 80 BVI 128D invalid subname
5165 7D04
5166 7D04 72 A8 7E MUM1 LDB R62,=7EH drp 76
5167 7D07 F0 03 JMP DORF go do it
5168 7D09 72 A8 7A MUM2 LDB R62,=7AH drp 72
5169 7D0C 7A 81 A2 7A DORF LDM R72,=VPRC00 load up some code
5170 7D0D 83 E1 81 STMD R72,=TMPNEM store code to ram
5171 7D0E 74 81 A8 7A LDM R64,=LDC000 get more code
5172 7D0F 73 32 A0 LDB R63,R62 set the correct drp
5173 7D0A 83 E7 81 STMD R63,=(TMPNEM+06H) store to ram
5174 7D0D CE 93 75 FDOVF JSB =FINC00 sets up romjsb etc.
5175 7D0F 9E RTN
5176 7D0F
5177 7D0F 7A 81 A2 7A MUM3 LDM R72,=VPRC00 get code
5178 7D0F 83 E1 81 STMD R72,=TMPNEM store in ram
5179 7D0B 7A 89 7A B5 LDM R70,=7AH,0B5H,02H,00H,6EH,0B5H,08H,00H
5179 7D0E 02 00 6E B5
5179 7D0D 08 00
5180 7E02 83 E7 81 STMD R70,=(TMPNEM+06H) store above code into ram
5181 7E05 A8 9E LDB R70,=9EH get a rtn
5182 7E07 82 EF 81 STMD R70,=(TMPNEM+0EH) store to ram
5183 7E0A F0 E1 JMP FDOVF finish it
5184 7E0C
5185 7E0C 78 81 AC 7A RM1 LDM R70,=TYPC00 load the ram code
5186 7E10 83 E1 81 STMD R70,=TMPNEM store at TMPNEM
5187 7E13 F0 D8 JMP FDOVF execute the code
5188 7E15
5189 7E15 50 08 A1 ckreal LDM R20,R10 get the ptr to caller
5190 7E18 C0 06 00 SBM R20,=R6 r20 pts to var pointer
5191 7E1B 40 80 5E 82 LDBD R0,=FLAG1 is this in ROM?
5192 7E1F B4 JLB R0
5193 7E20 90 TAB R0
5194 7E21 F5 05 IFNG yes,
5195 7E23 CE 0C 7E JSB =RM1 go execute RAM code
5196 7E26 F0 07 ELSE no,
5197 7E28 50 10 A5 LDM R20,R20 get the var ptr
5198 7E2B 03 47 82 RDM R20,=RNFIL make it absolute
5199 7E2E A5 LDM R20,R20 get the name form
5200 7E2F ENDF
5201 7E2F 50 CF 30 00 RDM R20,=30H,00H isolate the type bits
5202 7E33 9E RTN
5203 7E34

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RYX103 12/13/1983 11:38 AM PG145
*****
>>>>>>> MPTRLO, ptr allocation routine <<<<<<<<<
5104 7D52 46 91 ER151J tab r6
5105 7D54 F0 F2 JMP ER151J yes, error
5106 7D56 ENDF
5107 7D56
5108 7D56 5E 38 A3 STM R36,R70 store name to tfer reg
5109 7D59 6E 06 E5 PDBD R56,=R6 save the total len
5110 7D5C A9 02 00 LDM R56,=R2 env size is ?
5111 7D5F 50 A9 0A 00 LDM R20,=R10 VPR size is 10*2
5112 7D63 CE 50 71 JSB =SUBSTR put into VPR
5113 7D66 6E 06 E3 PDBD R56,=R6 get back total len
5114 7D69 1E E5 PDBD R56,=R36 put into VPR entry
5115 7D6B 40 A1 LDM R0,R36
5116 7D6D C0 0A 00 SBM R0,=R10
5117 7D70 44 A9 B2 78 GTO SETPCM
5118 7D74
5119 7D74
5120 7D74 CE FF FF namfrm jsb =FRANM
5121 7D77 42 A9 00 40 LDM R2,=40,40h
5122 7D7B 1E 95 ORN R2,=R36
5123 7D7D A3 STM R2,=R36
5124 7D7E 80 5E 82 LDBD R2,=FLAG1
5125 7D81 9E RTN
5126 7D82
5127 7D82 42 14 A4 integ LDBD R2,=R24
5128 7D85 C8 1A CMB R2,=INT1H
5129 7D87 F6 F8 RNE
5130 7D89 5E 06 E5 PDBD R36,=R6
5131 7D8C CE FF FF JSB =GETINT
5132 7D8F 5E 06 E3 PDBD R36,=R6
5133 7D92 54 89 ICM R24
5134 7D94 42 93 CLN R2
5135 7D96 9E RTN
5136 7D97
5137 7D97
5138 7D97 42 B1 43 82 NAME* LDM R2,=PRFILE get file loc
5139 7D9B E8 0A 00 RDM R2,=PCBLEH get loc of first line
5140 7D9E D1 40 82 CMBD R2,=PCR compare to sub line
5141 7DA1 F6 20 JNE ER153 not equal then error
5142 7DA3 14 E0 PDBD R2,=R24 get the str token
5143 7DA5 E0 PDBD R2,=R24 get the str length
5144 7DA6 F7 28 JZR ER153 null str so invalid
5145 7DA8 03 A2 STB R2,R3
5146 7DA9 40 8B 20 LDB R0,=40 set tfer reg
5147 7DAB 60 8B 20 LDB R40,=BLANK blank r40
5148 7DAD 61 20 A1 LDM R41,R40 blank 41/47
5149 7DB1 LOOP
5150 7DB3 41 14 E0 PDBD R*,=R24 get name to r40
5151 7DB6 42 BA DCR R2 dec the count
5152 7DB8 F7 04 JZR ER0G go if done
5153 7DBA 40 88 ICB R0 incre tfer reg
5154 7DBE F0 F5 BBNP
5155 7DBE 43 8C FROG TCB R3 neg # of bytes for UPRC
5156 7D00 40 02 A1 LDM R0,R2 neg # to r1

```

```

KARMA 08/31/82*****
ITEM LOC OBJECT CODE SRC=RY&103 OBJ=RYX103 12/13/1983 11:38 AM PG147
*****
>>>>>>> entry points and externals <<<<<<<<<
5205 7E34
5206 7E34
5207 7E34
5208 7E34
5209 7E34
5210 7E34
5211 7E34
5212 7E34

```



 NOT MANUFACTURER SUPPORTED

 Recipient agrees NOT to contact manufacturer

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&103 OBJ=RVZ103 12/13/1983 11:38 AM PG148
*****
PRINT N USING: PRNUS.
*****
5214 7E34
5215 7E34
5216 7E34
5217 7E34
5218 7E34
5219 7E34
5220 7E34
5221 7E34
5222 7E34
5223 7E34
5224 7E34
5225 7E34
5226 7E34
5227 7E34
5228 7E34 CE FF FF ERR90 jnb =ERR1+ report invalid line #
5229 7E37 5A byt 90d
5230 7E38
5231 7E38 CE FF FF ERR68 jnb =ERR1+
5232 7E3D 44 byt 68d
5233 7E3C
5234 7E3E A1 byt 241
5235 7E3D 5A A9 0B 01 PRNUS. ldn r32,=8267 we need 12 bytes
5236 7E41 CE FF FF jnb =ROOM1 check, error if not
5237 7E44 F8 ED ren
5238 7E46 CE FF FF jnb =OMR12? what's on r12
5239 7E49 42 C8 10 cwb r2,=10h 16 bytes?
5240 7E4C FB 0F ifnc no, then just channel
5241 7E4E CE AE 68 jnb =CHAN open channel
5242 7E51 F8 E0 ren something's wrong
5243 7E53 F7 04 ifnz not in system
5244 7E55 44 A9 95 69 BACK GTO FILMAN eventually handi
5245 7E59
5246 7E59 78 92 cwb r70 set serial print flag
5247 7E5B F0 10 else yes, chan and line#
5248 7E5D CE E3 68 jnb =CHAN++ open channel save line#
5249 7E60 F8 D1 ren something went wrong
5250 7E62 F6 F1 jnz BACK not in system
5251 7E64 68 0A E5 pwnd r50,r12 put # on r12
5252 7E67 CE FF FF jnb =ONE1 get one integer
5253 7E6A 67 90 lcb r47 line # to large
5254 7E6C F6 16 jnz ERR90 yes, then error
5255 7E6E 6E 25 A1 ldn r56,=r5 get line # to r56
5256 7E71 6D 92 cwb r55 clear data pointer
5257 7E73 16 07 0B 00 stnd r55,x26,data# store in line #
5258 7E77 78 0B 01 lcb r70,rone set random write flag
5259 7E7A
5260 7E7A 40 18 B4 05 ldn r0,x30,dr.tnn get the file type
5261 7E7E 00
5262 7E7F C8 54 cwb r0,=tynlex not text then error
5263 7E81 F6 B5 jne ERR68 here it is

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&103 OBJ=RVZ103 12/13/1983 11:38 AM PG149
*****
PRINT N USING: PRNUS.
*****
5263 7E83 56 06 E5 pwnd r26,r6 save r26 from rsmen
5264 7E86 6E A9 FF 00 ldn r56,=8255 reserve 255 bytes
5265 7E88 CE FF FF jnb =RSMEN1 reserve then
5266 7E8D 56 2E A3 stn r26,r56 r56 has rsmen ptr
5267 7E90 06 E3 pwnd r26,-r6 restore sofile ptr
5268 7E92 F8 9F ren not enough memory
5269 7E94 6D 92 cwb r55 okay clear count
5270 7E96 CE FF FF jnb =10? open the sofile
5271 7E99 40 1E 04 02 ldn r0,x36,two get the line len
5272 7E9D 00
5273 7E9E C8 19 cwb r0,=25d have we been here
5274 7E9F 71 1C ifne no,
5275 7EA2 CE FF FF jnb =SKPLN r36 to next line
5276 7EA5 5A A9 0B 00 ldn r32,=813 12 bytes to open
5277 7EA9 5E 18 A3 stn r36,r30 at r30
5278 7EAC CE FF FF jnb =ALLDE open up
5279 7EAF F8 B2 ren not enough room
5280 7EB1 5A 08 19 ldb r32,=25d new size
5281 7EB4 18 B6 F3 FF stbd r32,x30,(0-13d) put in the line
5282 7EB8 56 C8 0B 00 adn r26,=813 move up r26
5283 7EBE F0 05 else yes,
5284 7ED0 C8 0F 00 ldn r30,r36 set r30
5285 7ED3 adn r30,=815 to our info
5286 7ED6 C8 16 05 03 ldn r40,x26,name# get the file name
5287 7ED7 00
5288 7ED8 18 A7 stnd r40,r30 store in line 0
5289 7ED9 6D 07 0B 00 stnd r55,x30,eight store count and rmen ptr
5290 7EE0 F6 39 lcb r70 serial,
5291 7EE2 40 16 04 0B ldn r0,x26,data# get the data#
5292 7EE6 00
5293 7EE9 7E B5 0C 00 ifnz already here,
5294 7EEA 7E B5 0C 00 ldn r76,x26,line# get the line#
5295 7EEB F0 2A jnz inc76 inc if 0
5296 7EEF jnz ahead not zero so go
5297 7EEF jnz ahead move up one
5298 7EEF jnz ahead get the line #
5299 7EEF jnz ahead save r30 from FOPEN
5300 7EEF jnz ahead open file r36
5301 7EEF jnz ahead restore r30
5302 7EEF jnz ahead seek to line #
5303 7EEF jnz ahead if there then
5304 7EEF jnz ahead ship to next
5305 7EEF jnz ahead
5306 7EEF jnz ahead get the line #
5307 7EEF jnz ahead andln ?
5308 7EEF jnz ahead yes,
5309 7EEF jnz ahead set bcd
5310 7EEF jnz ahead inc line #
5311 7EEF jnz ahead back to bin
5312 7EEF jnz ahead no,
5313 7EEF jnz ahead get the line #

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&103 OBJ=RVZ103 12/13/1983 11:38 AM PG150
*****
PRINT N USING: PRNUS.
*****
5313 7F06
5314 7F06 7E 16 07 0C SET stnd r76,x26,line# store in io line
5315 7F0A 00
5316 7F0B
5317 7F0B 7E 16 05 0C AHEAD ldn r76,x26,line# get the line #
5318 7F0F 00
5319 7F10 58 14 A3 stn r30,r24 r24 now pts to lin 0
5320 7F13 CE FF FF jnb =FLOPEN open file r36
5321 7F16 CE FF FF jnb =FSEEK seek to line #
5322 7F19 F9 1D ifen not there,
5323 7F1B 5A A9 03 00 ldn r32,=thre three bytes to open up
5324 7F1F 58 1E A1 ldn r30,r36 at r36
5325 7F22 CE FF FF jnb =ALLOC open up
5326 7F25 F9 08 ifen no mem,
5327 7F27 5E 14 A1 ldn r36,r24 restore sofile ptr
5328 7F2A CD 0F 00 adn r36,=815 r36 ptr to line 0
5329 7F2D F0 7D jnp delio delete the junk
5330 7F2F
5331 7F2F 7E 18 A7 stnd r76,r30 store in line #
5332 7F32 92 cwb r76 no bytes long
5333 7F33 86 02 00 stbd r76,x30,two store in stsize
5334 7F36 F0 15 else is there
5335 7F38 5A B3 cwb r32 clear for load
5336 7F3A 1E B4 02 00 ldn r32,x36,two load stsize
5337 7F3E 58 A1 ldn r30,r36 r30 for delete
5338 7F40 C8 03 00 adn r30,=three set to data
5339 7F43 CE FF FF jnb =DELETE delete this stuff
5340 7F46 drp r30 drp as 30
5341 7F46 ldn r30,r36 reset r30
5342 7F48 5B 18 B6 02 stbd r33,x30,two store 0 line len
5343 7F4C 00
5344 7F4D
5345 7F4D 8B adn r36,rone put in sofile line
5346 7F4E 16 B6 0B 00 stbi r36,=p_flag for null stuff
5347 7F52 0A 00 82 dcb r36 r36 = 0
5348 7F55 8A dcb r36 r36 = fff
5349 7F56 8A dcb r36 r36 = fff
5350 7F58 02 C8 82 stbd r36,=route put in route for handi call
5351 7F5A 5E 18 A1 ldn r36,r30 save r30 in r36
5352 7F5C CE FF FF jnb =FOPEN find beginning of file
5353 7F5E 5E 18 0D stbd r36,r30 make this line rel
5354 7F63 14 B7 0B 00 stnd r36,x24,eleven put loc into sofile
5355 7F67 9E rtn
5356 7F68
5357 7F68 24 byt 44
5358 7F69 CE FF FF pwnd. jnb =10? open sofile
5359 7F6C 5A B3 cwb r32 clear for load
5360 7F6E 1E B4 17 00 ldn r32,x36,$23 get the source count
5361 7F72 60 B5 0F 00 ldn r40,x36,$15 get the file name
5362 7F76 CE FF FF jnb =FOPEN open the file
5363 7F79 40 1E B5 1A ldn r0,x36,$26 get the rel offset of line
5364 7F7D 00

```

```

KARMA 08/31/82-----
ITEM LOC OBJECT CODE SRC=RV&103 OBJ=RVZ103 12/13/1983 11:38 AM PG151
*****
PRINT N USING: PRNUS.
*****
5362 7F7E 58 18 A5 ldn r30,r30 r30 --- beginning of file
5363 7F81 00 C3 adn r30,r0 r30 ptr to line
5364 7F83 C8 03 00 adn r30,=three ptr to text
5365 7F86 CE FF FF jnb =ALLOC alloc those bytes
5366 7F89 F9 0C ifen no memory
5367 7F8B drp r30
5368 7F8B C8 03 00 cwb r30,=three back to line #
5369 7F8E 5A A9 03 00 ldn r32,=three delete this null line
5370 7F92 CE FF FF jnb =DELETE
5371 7F95 F0 15 else
5372 7F97 56 1E B5 18 ldn r26,x36,$24 get rmen ptr
5373 7F9B 00
5374 7F9C 5A 18 B6 FF stbd r32,x30,(0-1) store the line size
5375 7FA0 FF
5376 7FA1 loop loop,
5377 7FA1 5A 8A dcb r32 dec the count
5378 7FA3 F4 07 jng DELIO all done
5379 7FA5 40 16 E0 pwnd r0,=r26 get a char
5380 7FA8 18 E4 pwnd r0,r30 put into file
5381 7FAB F0 F5 uhnp
5382 7FAC
5383 7FAC 5A 1E B4 02 DELIO ldn r32,x36,two get the line len
5384 7FBD 00
5385 7FBD C8 19 cwb r32,=25d if 22 then our stuff
5386 7FBD F6 B2 rne yes,
5387 7FBD 5A 0C 00 ldn r32,=812
5388 7FBD 86 02 00 stbd r32,x36,two
5389 7FBD 89 acn r32
5390 7FBD 5E C8 0F 00 adn r36,=815 r36 to our stuff
5391 7FBD 18 A3 stn r36,r30 r30 for delete
5392 7FBD CE FF FF jnb =DELETE
5393 7FBD drp r30
5394 7FBD cwb r30 clear flag
5395 7FBD stbd r30,=route store in route
5396 7FBD rtn

```

NOMAS

NOT MANUFACTURER SUPPORTED
recipient agrees NOT to contact manufacturer

KARMA 08/31/82-----
 ITEM LOC OBJECT CODE SRC-RY8103 OBJ-RYX103 12/13/1983 11:38 AM PG152

 >>>>>> entry points and externals <<<<<<<<
 5396 7FCR A entry points
 5397 7FCR

NOMAS

NO! Manufacturer Supported
 Customer agrees NO! to contact manufacturer

KARMA 08/31/82-----
 SYMBOL VALUE TYPE COUNT Symbol Table 12/13/1983 11:38 AM PG154

 8000H 000F EQU 1
 80010H 0010 EQU 0 <--- NOT REFERENCED??
 80018H 0018 EQU 0 <--- NOT REFERENCED??
 800F0H 00F0 EQU 1
 8077FH 07FF EQU 1
 810 000A EQU 4
 812 000C EQU 1
 813 000D EQU 2
 815 000F EQU 5
 82 0002 EQU 7
 821 0015 EQU 1
 823 0017 EQU 1
 824 0018 EQU 1
 8255 00FF EQU 1
 826 001A EQU 1
 8267 0108 EQU 1
 83 0003 EQU 1
 832 0020 EQU 1
 84 0004 EQU 3
 86 0006 EQU 4
 87E80H 7E80 EQU 1
 877FFH 7FFF EQU 3
 88 0008 EQU 1
 88000H 8000 EQU 4
 1H0001 62E6 LCL 1
 1H0002 6377 LCL 1
 1H0003 710F LCL 0 <--- NOT REFERENCED??
 1H0004 722F LCL 1
 2H0001 6322 LCL 1
 2H0002 7124 LCL 1
 2H0003 7241 LCL 1
 3H0001 62C4 LCL 1
 4H0001 766A LCL 1
 5H0001 7708 LCL 1
 7H0001 7677 LCL 0 <--- NOT REFERENCED??
 9H0001 7663 LCL 5
 A01JK 0034 EQU 2
 A11JK 0033 EQU 2
 AAAAAA 7837 LCL 1
 ACTION 72E4 LCL 1
 ADVAN 648D LCL 1
 ADVAN. 6FAB LCL 1
 ADVTK 0007 EQU 0 <--- NOT REFERENCED??
 AFSET 7864 LCL 0 <--- NOT REFERENCED??
 AGGIE 78F0 LCL 1
 AHERD 7F08 LCL 1
 ALCALL FFFF EXT 0 <--- NOT REFERENCED??
 ALINI FFFF EXT 0 <--- NOT REFERENCED??
 ALLDC FFFF EXT 10
 ALOSHV FFFF EXT 1
 ALVENT FFFF EXT 2
 AREF. 73FB LCL 2
 AREFJK 002C EQU 3
 ARFIN 7400 LCL 1

KARMA 08/31/82-----
 ITEM LOC OBJECT CODE SRC-RY8103 OBJ-RYX103 12/13/1983 11:38 AM PG153

 >>>>>> VCHAR., handle the V.CHAR. hand call <<<<<<<<
 5399 7FCR
 5400 7FCR
 5400 7FCR
 5401 7FCR
 5402 7FCR
 5403 7FCR
 5404 7FCR
 5405 7FCR
 5406 7FCR
 5407 7FCR
 5408 7FCR
 5409 7FCR
 5410 7FCR
 5411 7FCR
 5411 7FCR
 5412 7FCR
 5413 7FCR 60 06 E5 VCHAR. pund r40,r6 save the 40's
 5414 7FED 18 A1 ldn r40,r30 load the 30's
 5415 7FEF 06 E5 pund r40,r6 save the 30's
 5416 7FD1 CE FF FF jub =10? open logfile
 5417 7FD4 F8 1C jen DOWNER not there rtn
 5418 7FD6 5E CB 0F 00 adn r36,r815 r36 to our stuff
 5419 7FD8 58 1E B5 09 ldnd r30,x36,nine get the loc
 5419 7FDE 00
 5420 7FDF 5C 93 clr r34 clear for load
 5421 7FE1 B4 08 00 ldnd r34,x36,eight get the current len
 5422 7FE4 C8 FF cnb r34,r255D too big
 5423 7FE6 F7 0A jeq DOWNER yes thats all
 5424 7FE8 B8 scb r34 increment
 5425 7FE9 B6 08 00 stbd r34,x36,eight store new len
 5426 7FEC 8A dcb r34 dec for add
 5427 7FED 18 C3 adn r34,r30 r34 abs loc
 5428 7FEF A3 str r34,r30 r30 abs loc
 5429 7FF0 5A A6 stbd r32,r30 put away char
 5430 7FF2 60 06 E3 DOWNER pond r40,-r6 get the 30's
 5431 7FF5 18 A3 str r40,r30 into the 30's
 5432 7FF7 06 E3 pond r40,-r6 get the 40's
 5433 7FF9 9E rtn thats all
 5434 7FFA
 5435 7FFA
 5436 7FFA
 LEXEND
 fin

KARMA 08/31/82-----
 SYMBOL VALUE TYPE COUNT Symbol Table 12/13/1983 11:38 AM PG155

 ARNAME 6664 LCL 1
 ARR- 7E60 LCL 1
 ARRAY- 6675 LCL 2
 ARREF 65E0 LCL 2
 ARREND 78F0 LCL 1
 ASLEEN 60E4 LCL 1
 ASCHAM FFFF EXT 1
 AXCALL FFFF EXT 1
 BACH1 6DEF LCL 1
 BACH2 6DEA LCL 1
 BAKK 7E55 LCL 1
 BAKK10 7919 LCL 1
 BAKKER 7C39 LCL 0 <--- NOT REFERENCED??
 BACHM 68D4 LCL 2
 BODPAR 664C LCL 2
 BODPRM 6FEC LCL 1
 BODSTF 689A LCL 2
 BAR 7013 LCL 1
 BEGIN 6978 LCL 2
 BGIFIX 6211 LCL 1
 BJKINT FFFF EXT 0 <--- NOT REFERENCED??
 BLACH1 771C LCL 0 <--- NOT REFERENCED??
 BLAKK 0020 G EQU 4
 BLOF 6501 LCL 1
 BLORF 79A6 LCL 2
 BASTK 0006 EQU 1
 BRAHS1 6F15 LCL 1
 BRAHS2 6F5B LCL 1
 BRAHS3 6F5D LCL 1
 BTB.R FFFF EXT 2
 CALL 694E LCL 6
 CALCL 7194 LCL 1
 CALLTK 004F G EQU 5
 CALWMA FFFF EXT 1
 CALS22 71A1 LCL 0 <--- NOT REFERENCED??
 CHAIN 655C LCL 1
 CHAIN. 78D2 LCL 1
 CHAM 68AE LCL 10
 CHAM+ 68D8 LCL 2
 CHAM++ 68E3 LCL 3
 CHAMTK 0008 EQU 0 <--- NOT REFERENCED??
 CME 6637 LCL 1
 CHECK 76E7 LCL 2
 CMEWLM 67EE LCL 1
 CMEWHV 7858 LCL 1
 CHKROM 754A LCL 2
 CHKROM 683D LCL 2
 EKARR 65F7 LCL 1
 EKDIM 75AE LCL 2
 EKREHL 7E15 LCL 3
 EKSTAR 65B6 LCL 1
 EKSTR 659A LCL 1
 ELASH. 6A6B LCL 1
 ELATK 0001 EQU 0 <--- NOT REFERENCED??

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG156				
SYMBOL	VALUE	TYPE	COUNT						
CLEAR	000A	EQU	0		----	NOT REFERENCED??			
CLEAR	0001	G EQU	2						
CLEAR	750F	LCL	1						
CLEAR	7512	LCL	1						
CLEAR	750C	LCL	3						
CLEAR	640F	LCL	3						
CLEAR	6410	LCL	6						
CLEAR	79F0	LCL	1						
CLEAR	7C92	LCL	1						
CLEAR	6464	LCL	1						
CLEAR	FFFF	EXT	4						
CLEAR	7A24	LCL	1						
CLEAR	6844	LCL	3						
CLEAR	78D4	LCL	1						
CLEAR	7728	LCL	3						
CLEAR	761C	LCL	1						
CLEAR	7624	LCL	1						
CLEAR	775A	LCL	1						
CLEAR	7734	LCL	1						
CLEAR	002C	G EQU	5						
CLEAR	66F2	LCL	1						
CLEAR	6386	LCL	3						
CLEAR	64F9	LCL	2						
CLEAR	6409	LCL	1						
CLEAR	0009	EQU	0		----	NOT REFERENCED??			
CLEAR	FFFF	EXT	1						
CLEAR	786D	LCL	2						
CLEAR	68DA	LCL	1						
CLEAR	FFFF	EXT	1						
CLEAR	FFFF	EXT	2						
CLEAR	007C	G EQU	2						
CLEAR	691F	LCL	3						
CLEAR	6906	LCL	3						
CLEAR	0000	G EQU	2						
CLEAR	77A5	LCL	1						
CLEAR	7788	LCL	1						
CLEAR	FFFF	EXT	1						
CLEAR	68E6	LCL	1						
CLEAR	FFFF	EXT	1						
CLEAR	7618	LCL	1						
CLEAR	000B	G EQU	27						
CLEAR	679F	LCL	8						
CLEAR	0086	EQU	2						
CLEAR	622F	LCL	0		----	NOT REFERENCED??			
CLEAR	6A87	LCL	1						
CLEAR	63F8	LCL	1						
CLEAR	FFFF	EXT	10						
CLEAR	7FAC	LCL	2						
CLEAR	0002	EQU	0		----	NOT REFERENCED??			
CLEAR	FFFF	EXT	1						
CLEAR	764A	LCL	2						
CLEAR	7705	LCL	1						
CLEAR	76A6	LCL	2						
CLEAR	7726	LCL	1						

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG157				
SYMBOL	VALUE	TYPE	COUNT						
DIMEN	6341	LCL	1						
DIMIE	FFFF	EXT	1						
DIMCR	FFFF	EXT	1						
DIMEN	66E7	LCL	1						
DIMEN	74D6	LCL	1						
DIMEN	70DC	LCL	1						
DIMEN	7FF2	LCL	2						
DIMEN	0005	G DRD	2						
DIMEN	0004	G DRD	1						
DIMEN	0080	EQU	32						
DIMEN	000E	G DRD	1						
DIMEN	0013	G DRD	1						
DIMEN	0002	G DRD	2						
DIMEN	000C	G DRD	7						
DIMEN	0016	G DRD	3						
DIMEN	001A	G DRD	1						
DIMEN	0003	G EQU	7						
DIMEN	000B	EQU	3						
DIMEN	A999	G DRD	8						
DIMEN	7B1F	LCL	1						
DIMEN	739A	LCL	2						
DIMEN	FFFF	EXT	1						
DIMEN	838B	G DRD	8						
DIMEN	6650	LCL	1						
DIMEN	664E	LCL	2						
DIMEN	665C	LCL	5						
DIMEN	6787	LCL	2						
DIMEN	000E	G EQU	9						
DIMEN	788D	LCL	4						
DIMEN	7C00	LCL	0		----	NOT REFERENCED??			
DIMEN	7C10	LCL	1						
DIMEN	7C15	LCL	1						
DIMEN	7C4F	LCL	1						
DIMEN	7C5B	LCL	1						
DIMEN	78CF	LCL	1						
DIMEN	7C6E	LCL	0		----	NOT REFERENCED??			
DIMEN	7C73	LCL	1						
DIMEN	7C82	LCL	1						
DIMEN	7C9E	LCL	1						
DIMEN	7B4A	LCL	1						
DIMEN	7B04	LCL	1						
DIMEN	7D52	LCL	2						
DIMEN	7C59	LCL	0		----	NOT REFERENCED??			
DIMEN	7D00	LCL	3						
DIMEN	7B33	LCL	1						
DIMEN	7B96	LCL	1						
DIMEN	7B09	LCL	1						
DIMEN	70AF	LCL	1						
DIMEN	00B4	G EQU	12						
DIMEN	613D	LCL	3						
DIMEN	FFFF	EXT	3						
DIMEN	FFFF	EXT	13						
DIMEN	7B01	LCL	7						
DIMEN	633A	LCL	3						

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG158				
SYMBOL	VALUE	TYPE	COUNT						
ERR3	6337	LCL	2						
ERR68	7E38	LCL	1						
ERR78	6466	LCL	6						
ERR78	6469	LCL	6						
ERR78A	648B	LCL	1						
ERR89	64F5	LCL	3						
ERR89E	6606	LCL	1						
ERR90	7E34	LCL	1						
ERRCHK	728A	LCL	1						
ERRLEN	615F	LCL	1						
ERROR	FFFF	EXT	16						
ERROR	FFFF	EXT	14						
ERROUT	6F37	LCL	2						
ERRSTP	8391	G DRD	4						
ERR	6275	LCL	3						
ERR	FFFF	EXT	0		----	NOT REFERENCED??			
ERR	7539	LCL	3						
ERR	FFFF	EXT	0		----	NOT REFERENCED??			
ERR	FFFF	EXT	1						
ERR	7D6D	LCL	2						
ERR	00FF	G EQU	2						
ERR	0017	EQU	0		----	NOT REFERENCED??			
ERR	6996	LCL	12						
ERR	0018	EQU	0		----	NOT REFERENCED??			
ERR	7593	LCL	5						
ERR	673C	LCL	2						
ERR	759C	LCL	1						
ERR	0005	G EQU	5						
ERR	76F8	LCL	1						
ERR	825E	G DRD	34						
ERR	7D00	LCL	1						
ERR	FFFF	EXT	1						
ERR	FFFF	EXT	3						
ERR	70CE	LCL	2						
ERR	70C4	LCL	0		----	NOT REFERENCED??			
ERR	FFFF	EXT	0		----	NOT REFERENCED??			
ERR	FFFF	EXT	1						
ERR	FFFF	EXT	5						
ERR	FFFF	EXT	1						
ERR	76D9	LCL	1						
ERR	0027	EQU	2						
ERR	0028	EQU	2						
ERR	0027	EQU	2						
ERR	70D3	LCL	1						
ERR	79D0	LCL	1						
ERR	0004	G EQU	17						
ERR	FFFF	EXT	2						
ERR	FFFF	EXT	0		----	NOT REFERENCED??			
ERR	FFFF	EXT	3						
ERR	70BE	LCL	1						
ERR	65BD	LCL	5						
ERR	FFFF	EXT	8						
ERR	8251	G DRD	18						
ERR	FFFF	EXT	0		----	NOT REFERENCED??			

KARMA 08/31/82-----				Symbol Table 12/13/1983 11:38 AM PG159			
SYMBOL	VALUE	TYPE	COUNT				
G10R2M	FFFF	EXT	1				
GCHAR	FFFF	EXT	8				
GETHR	FFFF	EXT	5				
GETHR	FFFF	EXT	3				
GETHR	794C	LCL	1				
GETHR	FFFF	EXT	1				
GETHR	78BD	LCL	1				
GETHR	FFFF	EXT	4				
GETHR	FFFF	EXT	2				
GETHR	FFFF	EXT	1				
GETHR	FFFF	EXT	0		----	NOT REFERENCED??	
GETHR	7376	LCL	0		----	NOT REFERENCED??	
GETHR	FFFF	EXT	2				
GETHR	FFFF	EXT	0		----	NOT REFERENCED??	
GETHR	7A75	LCL	1				
GROU1Y	6782	LCL	1				
HT10	6477	LCL	1				
HT12	688E	LCL	1				
HAND1	FFFF	EXT	2				
HAND10	FFFF	EXT	0		----	NOT REFERENCED??	
HAND1D	8375	G DRD	1				
HAND1E	6255	LCL	10				
HAN11	6ECB	LCL	1				
HAN12	6ED0	LCL	2				
HAN14	6EEF	LCL	1				
HEL11	674E	LCL	1				
HEPR	7ABA	LCL	2				
HL10	681E	LCL	2				
HL10*	6847	LCL	1				
HL10*	68D3	LCL	1				
HL101*	6885	LCL	1				
HL102*	68E8	LCL	1				
HL11	6AF1	LCL	3				
HL110	6886	LCL	1				
HL111	6E30	LCL	4				
HL111*	6E90	LCL	2				
HL112	6EBE	LCL	1				
HL12	6B45	LCL	1				
HL12*	6B83	LCL	1				
HL14	6C0B	LCL	2				
HL15	6C4F	LCL	1				
HL16	6C6B	LCL	2				
HL17	6D16	LCL	3				
HL18	6CCA	LCL	1				
HL19	6D3F	LCL	2				
JCM36	693A	LCL	1				
INOPE	65AC	LCL	1				
ILLDIN	715F	LCL	2				
INC76	7EFE	LCL	1				
INCPTR	67C3	LCL	2				
INCTEX	67D5	LCL	3				
INDEX*	698F	LCL	1				
INDIK	0019	EQU	0		----	NOT REFERENCED??	
INILEX	61AA	LCL	1				

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG160				
SYMBOL	VALUE	TYPE	COUNT						
INHTK	0011	EDU	1						
IMMUN	68E8	LCL	1						
IMMUN?	7574	LCL	2						
IMROM	72BE	LCL	1						
INER	6417	LCL	2						
INER-	641B	LCL	1						
INSERT	FFFF	EXT	0	<--- NOT REFERENCED??					
INSTK	000E	EDU	1						
INSTOK	0003	EDU	0	<--- NOT REFERENCED??					
INSTA	6B27	LCL	1						
INTEG	7082	LCL	2						
INTMUL	FFFF	EXT	4						
INTTK	001A	G EDU	8						
INXADV	0016	EDU	1						
IO?	FFFF	EXT	5						
IEN.	6808	LCL	1						
ITERIK	001A	EDU	0	<--- NOT REFERENCED??					
JEZER	6365	LCL	1						
JMPSUB	7368	LCL	1						
JNEERR	637A	LCL	1						
KLUDGE	8249	G DAD	5						
LBACKE	00DE	G EDU	2						
LEMM	0002	G EDU	1						
LEXBGM	6016	LCL	2						
LEXEND	7FFA	LCL	1						
LIMEN	000C	G EDU	13						
LIMEN.	68F2	LCL	1						
LIMETK	001C	EDU	0	<--- NOT REFERENCED??					
LIMLM.	6818	LCL	1						
LIST.	FFFF	EXT	0	<--- NOT REFERENCED??					
LISZ2	69C4	LCL	1						
LISZ3	69F8	LCL	1						
LISZ4	69D8	LCL	2						
LLENIK	0010	EDU	0	<--- NOT REFERENCED??					
LODCOD	7AAB	LCL	1						
LPA	6804	LCL	2						
LPR.	67FF	LCL	2						
LSTLIK	001B	EDU	0	<--- NOT REFERENCED??					
MAXCOL	0006	EDU	1						
MAXLEN	0002	EDU	1						
MAXROW	0004	EDU	2						
MCALL	6566	LCL	1						
MCALIK	000B	EDU	0	<--- NOT REFERENCED??					
MELJSB	FFFF	EXT	3						
MORE	65C7	LCL	3						
MOREE	65F5	LCL	1						
MOZ1	6E6A	LCL	0	<--- NOT REFERENCED??					
MOZ2	6E4F	LCL	1						
MOZ3	6E72	LCL	1						
MPTALO	781B	LCL	1						
MSUB	66A2	LCL	1						
MSUB.	7A2E	LCL	1						
MSUBIK	000C	EDU	3						
MSURD	FFFF	EXT	1						

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG162				
SYMBOL	VALUE	TYPE	COUNT						
OUT	701F	LCL	4						
OUTM	6A58	LCL	2						
OUTR	701C	LCL	4						
PAD	65D3	LCL	1						
PARRAS	6580	LCL	1						
PAREAR	82B8	G DAD	4						
PARSE	6280	LCL	1						
PARSE-	6285	LCL	0	<--- NOT REFERENCED??					
PART2	7920	LCL	2						
PART3	7974	LCL	0	<--- NOT REFERENCED??					
PART4	7988	LCL	1						
PART5	79CC	LCL	0	<--- NOT REFERENCED??					
PARTAL	002A	EDU	3						
PCBLEM	000A	G DAD	2						
PCR	8240	G DAD	3						
PEND.	7F69	LCL	1						
PENDIK	0036	EDU	1						
PLUS6	780B	LCL	1						
POINIR	6968	LCL	2						
POPEMY	FFFF	EXT	1						
POSIT	6732	LCL	10						
POSIT.	6795	LCL	1						
POSITI	6790	LCL	5						
POSUM	FFFF	EXT	2						
PRITOK	000D	EDU	0	<--- NOT REFERENCED??					
PRMUS	66FF	LCL	1						
PRMUS.	7E3D	LCL	1						
PRFILE	8243	G DAD	3						
PRINT.	FFFF	EXT	1						
PRITOK	0096	EDU	2						
PRNAME	8263	G DAD	2						
PRNIM	FFFF	EXT	1						
PROTEN	8233	G DAD	7						
PSLEX	6090	LCL	1						
PTALO?	FFFF	EXT	0	<--- NOT REFERENCED??					
PTR	7701	LCL	1						
PU.	6462	LCL	1						
PU-	640D	LCL	1						
PU-R6	640A	LCL	1						
PUMING	FFFF	EXT	2						
PURAGE	6529	LCL	2						
PURGE	7787	LCL	1						
PUS.	64B3	LCL	0	<--- NOT REFERENCED??					
PUSHM	63E9	LCL	3						
PUSHAS	FFFF	EXT	1						
PUSHUS	63E8	LCL	6						
PUSIR	6538	LCL	2						
PUTADR	FFFF	EXT	1						
PUREL	FFFF	EXT	8						
PFLAG	82D0	G DAD	1						
QUOTK	0005	EDU	1						
RIOB	758B	LCL	2						
RIOCOD	710C	LCL	2						
RION	2583	LCL	2						

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG161				
SYMBOL	VALUE	TYPE	COUNT						
MYCALC	77E4	LCL	3						
MYCOPY	7891	LCL	1						
MYLEMM	000C	EDU	1						
MYMUN	63B3	LCL	6						
MYMUN.	63B8	LCL	6						
MYROW	4065	EDU	12						
MYROM	4063	EDU	2						
MYRIM	625C	LCL	26						
MYSTR	63E1	LCL	3						
MYSUB	78F1	LCL	2						
NAME	0003	G EDU	2						
NAME-	6E3A	LCL	2						
NAME?	7D97	LCL	1						
NAMER	6E37	LCL	0	<--- NOT REFERENCED??					
NAMEAD	7D74	LCL	1						
NAMEAN	6697	LCL	1						
NAMEIK	0029	EDU	0	<--- NOT REFERENCED??					
NAMEI	77BC	LCL	1						
NAMEIK	7769	LCL	1						
NAME	0009	G EDU	5						
NOCLC	FFFF	EXT	2						
NONAME	7939	LCL	1						
NOPE	70C6	LCL	1						
NORMAL	66F9	LCL	1						
REF.	7390	LCL	2						
REFIK	002A	EDU	2						
RHM1	7D04	LCL	1						
RHM2	7D09	LCL	1						
RHM3	7D11	LCL	1						
RHMIDF	000E	EDU	1						
RHMERT	653D	LCL	1						
RHMSCH	6821	LCL	1						
RHMV	6595	LCL	1						
RHMV.	6604	LCL	1						
RHMVA	FFFF	EXT	3						
RHMVAL	FFFF	EXT	2						
RVAL	78A7	LCL	2						
RVAL.	741A	LCL	3						
RVALIK	002E	EDU	2						
RXILIM	FFFF	EXT	0	<--- NOT REFERENCED??					
O(JK	0031	EDU	4						
O(JK	0032	EDU	1						
OKAYE	73C0	LCL	2						
OKSTF	68BF	LCL	1						
OLOSTF	6423	LCL	2						
ON/OFF	FFFF	EXT	1						
ONE	0001	G EDU	10						
ONEB	FFFF	EXT	5						
ONEI	FFFF	EXT	5						
ONR12?	FFFF	EXT	3						
OPTBAS	8282	G DAD	6						
ORIGIN	83B2	G DAD	3						
OUR.	7A29	LCL	1						
OUR.	7A29	LCL	1						

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG163				
SYMBOL	VALUE	TYPE	COUNT						
ABRAC1	00DF	G EDU	2						
REALIK	0004	EDU	4						
RECDH	78E2	LCL	0	<--- NOT REFERENCED??					
REFER	65C1	LCL	2						
RELMAR	FFFF	G DAD	1						
REPIK	0004	EDU	0	<--- NOT REFERENCED??					
RESCOM	FFFF	EXT	1						
RESET	74B9	LCL	2						
RESET.	74A4	LCL	1						
REST	74B4	LCL	2						
RESTEM	FFFF	EXT	1						
RESTA	7438	LCL	1						
RJCHS	6838	LCL	1						
RRI	7E0C	LCL	1						
RHMERD	1CE3	G DAD	1						
RNTLE	8247	G DAD	15						
RNLST?	FFFF	EXT	0	<--- NOT REFERENCED??					
RNNAME	8275	G DAD	2						
ROMAD1	71D2	LCL	1						
ROMAD2	71DA	LCL	1						
ROMADR	71CA	LCL	2						
ROMCLA	FFFF	EXT	0	<--- NOT REFERENCED??					
ROMCPY	FFFF	EXT	1						
ROMFIX	66CA	LCL	2						
ROMJSB	FFFF	EXT	2						
ROMVPA	FFFF	EXT	1						
ROM/OF	63C2	LCL	0	<--- NOT REFERENCED??					
ROOM?	FFFF	EXT	4						
ROMVAL	7AD3	LCL	6						
ROUTE	82CD	G DAD	3						
RPNTK	0012	EDU	0	<--- NOT REFERENCED??					
RPNUM	601F	LCL	1						
RPSTK	000F	EDU	0	<--- NOT REFERENCED??					
RPSTR	6C52	LCL	1						
RSNEN-	FFFF	EXT	2						
ASTRA.	625E	LCL	6						
RUM.	FFFF	EXT	1						
RUMC0D	71C3	LCL	1						
RUMLEK	6022	LCL	1						
RUMST\$	0002	G EDU	1						
SARFUM	FFFF	EXT	1						
SARSTR	FFFF	EXT	1						
SAREF.	7413	LCL	2						
SARFTK	0020	EDU	2						
SAYTOK	0002	G EDU	1						
SCRAT1	71E0	LCL	1						
SCRAT2	71E6	LCL	1						
SCRAT3	71EE	LCL	2						
SEARC	646D	LCL	1						
SEARTK	0005	EDU	0	<--- NOT REFERENCED??					
SEEK	64B5	LCL	1						
SEEKA.	6EF8	LCL	1						
SEEKR.	6F64	LCL	1						
SEETK	0006	EDU	0	<--- NOT REFERENCED??					

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG164				
SYMBOL	VALUE	TYPE	COUNT						
SEKATK	0014	EQU	1						
SEKATK	0015	EQU	1						
SEMI	0027	G EQU	3						
SEPVAL	002E	EQU	1						
SERW.	6E93	LCL	1						
SERW.	6DB7	LCL	1						
SERWTK	0013	EQU	0		----	NOT REFERENCED??			
SERWTK	0010	EQU	0		----	NOT REFERENCED??			
SET	7F06	LCL	0		----	NOT REFERENCED??			
SET-	7BF6	LCL	0		----	NOT REFERENCED??			
SETLW	FFFF	EXT	0		----	NOT REFERENCED??			
SETLW	6872	LCL	2						
SETCIX	7506	LCL	1						
SETED	FFFF	EXT	2						
SETPGM	78B3	LCL	2						
SETPR	FFFF	EXT	0		----	NOT REFERENCED??			
SETRIO	7783	LCL	1						
SETRI2	FFFF	EXT	3						
SETSTP	7829	LCL	1						
SETUP	7882	LCL	1						
SEVEN	0007	G EQU	1						
STPUSH	666F	LCL	4						
STSCW	FFFF	EXT	15						
SINT	63BC	LCL	3						
SIX	0006	G EQU	1						
SIZER	1F84	EQU	0		----	NOT REFERENCED??			
SKIPO	FFFF	EXT	1						
SKPLW	FFFF	EXT	0		----	NOT REFERENCED??			
SKPLW	FFFF	EXT	8						
SKPVR	FFFF	EXT	0		----	NOT REFERENCED??			
SMLINI	FFFF	EXT	1						
SMVAL	702C	LCL	1						
SREF	737F	LCL	2						
SREFTK	0028	EQU	2						
SSVAL	702C	LCL	1						
STAR1K	001F	EQU	0		----	NOT REFERENCED??			
STAR2K	0020	EQU	0		----	NOT REFERENCED??			
STARLOC	7047	LCL	1						
STARIR	693D	LCL	1						
STARZE	7176	LCL	2						
STCHM1	6860	LCL	2						
STF	758F	LCL	4						
STOR1	7163	LCL	2						
STOSV	FFFF	EXT	1						
STRARR	70E2	LCL	2						
STRCH	FFFF	EXT	1						
STRDIF	0008	EQU	1						
STREW	71F2	LCL	1						
STREVP	FFFF	EXT	0		----	NOT REFERENCED??			
STRLEN	0020	G EQU	1						
STRLOC	000A	EQU	1						
STRSCH	680F	LCL	1						
STRUP	65B1	LCL	1						
STSIZE	8255	G DAD	1						

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG166				
SYMBOL	VALUE	TYPE	COUNT						
TYPCD	79AC	LCL	1						
TYPEN	000E	G EQU	1						
TYPENI	6723	EXT	1						
TYPER	6787	LCL	12						
TYPIK	001E	EQU	0		----	NOT REFERENCED??			
UPPER	689F	LCL	1						
UPPER5	77D8	LCL	1						
UPRCA	FFFF	EXT	1						
USTOK	00CB	G EQU	1						
V.WEIN	004D	G EQU	1						
V.ALLO	0008	G EQU	1						
V.CHAR	0038	G EQU	1						
V.DALO	0009	G EQU	1						
V.DEC	000A	G EQU	1						
V.DIM	000B	G EQU	1						
V.EALO	0007	G EQU	1						
V.ENDL	0039	G EQU	1						
V.EXST	002F	G EQU	1						
V.FILE	0031	G EQU	2						
V.SIAR	0016	G EQU	1						
VAL	6608	LCL	2						
VALUE	6614	LCL	1						
VARFIX	78E2	LCL	1						
VARPTA	838F	G DAD	2						
VCHAR	7FCA	LCL	1						
VCIER1	71B9	LCL	1						
VCIER2	71BC	LCL	1						
VDALO1	7612	LCL	1						
VECTOR	71B5	LCL	1						
VEBYE	FFFF	EXT	1						
VFIH	FFFF	EXT	1						
VORI	7AE3	LCL	2						
VORTI1	7AB8	LCL	2						
VPRC00	7AB2	LCL	2						
VSREN	FFFF	EXT	1						
VS10R	FFFF	EXT	1						
WNRW	FFFF	EXT	0		----	NOT REFERENCED??			
ZUIS	7A29	LCL	1						
ZUISIK	0035	EQU	1						
if0001	61B3	LCL	1						
if0002	61C9	LCL	1						
if0004	61C5	LCL	1						
if0005	610E	LCL	1						
if0006	610A	LCL	1						
if0007	61EB	LCL	1						
if0008	6201	LCL	1						
if0009	61FD	LCL	1						
if0010	620D	LCL	1						
if0011	621A	LCL	1						
if0012	622F	LCL	1						
if0013	6253	LCL	1						
if0014	6254	LCL	1						
if0015	626E	LCL	1						
if0016	6271	LCL	1						

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG165				
SYMBOL	VALUE	TYPE	COUNT						
STUFF	7346	LCL	1						
STVAL	7E94	LCL	2						
SUBENW	7242	LCL	1						
SUBFW	82B8	G DAD	4						
SUBSTA	7150	LCL	1						
SVAL	744D	LCL	3						
SVALIK	002F	EQU	2						
SYSJSA	FFFF	EXT	20						
TABLE	7300	LCL	1						
TEMP2	8238	G DAD	5						
TEM	000A	G EQU	1						
TEST1	76E1	LCL	1						
TEST2	76DD	LCL	1						
TEXTAD	702C	LCL	1						
TEXTDE	6AF9	LCL	1						
TEXTJ1	69DC	LCL	1						
TEXTLP	6AFC	LCL	1						
TEXTRP	6CCE	LCL	1						
TEXTIS	6F39	LCL	1						
TEXTIS	6E3F	LCL	1						
TEXTIS	6887	LCL	1						
TEXTIS	6F39	LCL	1						
THREE	0003	G EQU	20						
THRU	7D40	LCL	9		----	NOT REFERENCED??			
TITLE	753F	LCL	1						
TRAPM1	81E1	G DAD	13						
TRAPM2	820D	G DAD	5						
TRAPM3	8464	G DAD	5						
TOK1	7B7D	LCL	0		----	NOT REFERENCED??			
TOK2	7BC5	LCL	1						
TOK3	7L64	LCL	1						
TOK3+	7BF8	LCL	1						
TOKB4	7LCA	LCL	1						
TOKB4+	7L62	LCL	1						
TOKB5	7BFA	LCL	1						
TOKEN	8454	G DAD	8						
TOS	8257	G DAD	1						
TOTLEN	0008	EQU	1						
TRACK	7626	LCL	1						
TRYMEX	7A07	LCL	1						
TRYPGM	7A81	LCL	2						
TSTUP	708C	LCL	0		----	NOT REFERENCED??			
TSTENO	FFFF	EXT	1						
TSTRN	FFFF	EXT	0		----	NOT REFERENCED??			
TUIVE	000C	EQU	0		----	NOT REFERENCED??			
TUO	0002	G EQU	43						
TUOB	FFFF	EXT	1						
TUULEN	6A5C	LCL	1						
TUASC	42FE	G DAD	1						
TYEDT	0020	G EQU	1						
TYLST	0010	G EQU	1						
TYNBS	0042	G EQU	1						
TYULEN	004C	G EQU	1						
TYMEX	0054	G EQU	1						

KARMA 08/31/82-----

Symbol Table 12/13/1983 11:38 AM PG167

=====

SYMBOL	VALUE	TYPE	COUNT	
if0018	6294	LCL	1	
if0019	6298	LCL	1	
if0020	62EF	LCL	1	
if0021	62E6	LCL	1	
if0022	62ED	LCL	1	
if0023	62FF	LCL	1	
if0024	62F8	LCL	1	
if0025	62FF	LCL	1	
if0027	632F	LCL	1	
if0028	6331	LCL	1	
if0029	6393	LCL	1	
if0030	6399	LCL	1	
if0031	6440	LCL	1	
if0032	6443	LCL	1	
if0033	644C	LCL	1	
if0034	6458	LCL	1	
if0035	6462	LCL	1	
if0036	64D7	LCL	1	
if0037	64E9	LCL	1	
if0039	651E	LCL	1	
if0040	654F	LCL	1	
if0041	657B	LCL	1	
if0042	65F5	LCL	1	
if0043	6612	LCL	1	
if0044	662F	LCL	1	
if0045	6632	LCL	1	
if0046	6641	LCL	1	
if0049	6689	LCL	1	
if0050	6694	LCL	1	
if0051	6697	LCL	1	
if0055	660F	LCL	1	
if0056	66F2	LCL	1	
if0057	66E7	LCL	1	
if0058	6714	LCL	1	
if0059	6731	LCL	1	
if0060	675A	LCL	1	
if0061	6745	LCL	1	
if0062	6753	LCL	1	
if0063	6771	LCL	1	
if0067	67E8	LCL	1	
if0071	6830	LCL	1	
if0073	685D	LCL	1	
if0075	685F	LCL	1	
if0076	6871	LCL	1	
if0077	688R	LCL	1	
if0078	689R	LCL	1	
if0079	688B	LCL	1	
if0081	68D1	LCL	1	
if0082	68FC	LCL	1	
if0083	6910	LCL	1	
if0087	6958	LCL	1	
if0088	6961	LCL	1	
if0089	6978	LCL	1	
if0090	697A	LCL	1	

Not Manufacturer Supported
recipient agrees NOT to claim manufacturer

NOMAS

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG168
SYMBOL	VALUE	TYPE	COUNT		
1F0091	698C	LCL	1		
1F0093	699C	LCL	1		
1F0094	69AD	LCL	1		
1F0097	69D4	LCL	1		
1F0099	6A04	LCL	1		
1F0101	6A25	LCL	1		
1F0102	6A3A	LCL	1		
1F0103	6A48	LCL	1		
1F0105	6A96	LCL	1		
1F0106	6A9A	LCL	1		
1F0108	6A9C	LCL	1		
1F0109	6AE8	LCL	1		
1F0110	6AEC	LCL	1		
1F0111	6B0E	LCL	1		
1F0112	6B1E	LCL	1		
1F0114	6B32	LCL	1		
1F0115	6B49	LCL	1		
1F0116	6B67	LCL	1		
1F0117	6B83	LCL	1		
1F0118	6B74	LCL	1		
1F0119	6B83	LCL	1		
1F0120	6B8F	LCL	1		
1F0121	6BA1	LCL	1		
1F0122	6BBD	LCL	1		
1F0123	6BE4	LCL	1		
1F0124	6BC9	LCL	1		
1F0125	6BE4	LCL	1		
1F0127	6BF6	LCL	1		
1F0128	6BFF	LCL	1		
1F0129	6C1B	LCL	1		
1F0130	6C22	LCL	1		
1F0131	6C34	LCL	1		
1F0132	6C4D	LCL	1		
1F0133	6C4B	LCL	1		
1F0134	6C4C	LCL	1		
1F0135	6C5D	LCL	1		
1F0136	6C89	LCL	1		
1F0137	6CC8	LCL	1		
1F0138	6C9E	LCL	1		
1F0139	6CD8	LCL	1		
1F0140	6CF7	LCL	1		
1F0141	6CF7	LCL	1		
1F0143	6D2A	LCL	1		
1F0144	6D33	LCL	1		
1F0145	6D58	LCL	1		
1F0146	6D5F	LCL	1		
1F0147	6D8A	LCL	1		
1F0148	6D88	LCL	1		
1F0149	6D8C	LCL	1		
1F0150	6D8D	LCL	1		
1F0151	6D81	LCL	1		
1F0153	6DCF	LCL	1		
1F0154	6DFF	LCL	1		
1F0155	6E21	LCL	1		

NOMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG170
SYMBOL	VALUE	TYPE	COUNT		
1F0217	73C2	LCL	1		
1F0218	740C	LCL	1		
1F0219	743A	LCL	1		
1F0220	743D	LCL	1		
1F0221	743A	LCL	1		
1F0224	745F	LCL	1		
1F0225	746A	LCL	1		
1F0226	748B	LCL	1		
1F0227	747A	LCL	1		
1F0228	748B	LCL	1		
1F0229	748B	LCL	1		
1F0232	74CF	LCL	1		
1F0233	74D1	LCL	1		
1F0234	74F4	LCL	1		
1F0235	74E9	LCL	1		
1F0236	74EC	LCL	1		
1F0239	7504	LCL	1		
1F0240	75C8	LCL	1		
1F0241	75CC	LCL	1		
1F0242	75DE	LCL	1		
1F0243	75D9	LCL	1		
1F0244	75F8	LCL	1		
1F0245	75F1	LCL	1		
1F0246	7601	LCL	1		
1F0247	75FF	LCL	1		
1F0248	7607	LCL	1		
1F0250	760F	LCL	1		
1F0254	7667	LCL	1		
1F0255	7688	LCL	1		
1F0257	76CC	LCL	1		
1F0251	76DB	LCL	1		
1F0261	771C	LCL	1		
1F0262	7750	LCL	1		
1F0264	775A	LCL	1		
1F0267	77D1	LCL	1		
1F0268	77E2	LCL	1		
1F0269	780E	LCL	1		
1F0270	77FB	LCL	1		
1F0271	7809	LCL	1		
1F0272	7805	LCL	1		
1F0273	7809	LCL	1		
1F0274	7821	LCL	1		
1F0275	7844	LCL	1		
1F0276	7857	LCL	1		
1F0277	7870	LCL	1		
1F0278	7882	LCL	1		
1F0279	787A	LCL	1		
1F0280	7882	LCL	1		
1F0281	788D	LCL	1		
1F0284	78C8	LCL	1		
1F0285	78CC	LCL	1		
1F0286	7901	LCL	1		
1F0287	7920	LCL	1		
1F0288	7953	LCL	1		

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG169
SYMBOL	VALUE	TYPE	COUNT		
1F0156	6E4A	LCL	1		
1F0157	6E81	LCL	1		
1F0159	6E94	LCL	1		
1F0160	6E9D	LCL	1		
1F0161	6EDC	LCL	1		
1F0162	6EEA	LCL	1		
1F0163	6EE8	LCL	1		
1F0165	6F0E	LCL	1		
1F0166	6F2D	LCL	1		
1F0168	6F5D	LCL	1		
1F0170	6F82	LCL	1		
1F0171	6F8B	LCL	1		
1F0172	6F93	LCL	1		
1F0173	6F9B	LCL	1		
1F0174	6FA0	LCL	1		
1F0175	6FCC	LCL	1		
1F0176	6FF4	LCL	1		
1F0178	7011	LCL	1		
1F0179	7004	LCL	1		
1F0180	704D	LCL	1		
1F0181	70B1	LCL	1		
1F0182	70B1	LCL	1		
1F0183	70E1	LCL	1		
1F0184	70ED	LCL	1		
1F0185	712B	LCL	1		
1F0186	713B	LCL	1		
1F0187	714D	LCL	1		
1F0188	716E	LCL	1		
1F0189	718B	LCL	1		
1F0190	7193	LCL	1		
1F0192	7208	LCL	1		
1F0193	7216	LCL	1		
1F0195	7293	LCL	1		
1F0196	7267	LCL	1		
1F0197	7262	LCL	1		
1F0198	7272	LCL	1		
1F0199	7272	LCL	1		
1F0200	728C	LCL	1		
1F0201	72E4	LCL	1		
1F0202	72F2	LCL	1		
1F0203	72F4	LCL	1		
1F0204	731B	LCL	1		
1F0205	7327	LCL	1		
1F0206	7329	LCL	1		
1F0207	7333	LCL	1		
1F0208	733F	LCL	1		
1F0209	7341	LCL	1		
1F0210	7372	LCL	1		
1F0211	7368	LCL	1		
1F0212	736E	LCL	1		
1F0213	737A	LCL	1		
1F0214	7385	LCL	1		
1F0215	73C5	LCL	1		
1F0216	73B8	LCL	1		

KARMA 08/31/82-----					Symbol Table 12/13/1983 11:38 AM PG171
SYMBOL	VALUE	TYPE	COUNT		
1F0290	7955	LCL	1		
1F0291	7974	LCL	1		
1F0292	799A	LCL	1		
1F0293	79B2	LCL	1		
1F0294	79E9	LCL	1		
1F0295	7A07	LCL	1		
1F0296	7A57	LCL	1		
1F0299	7AFC	LCL	1		
1F0300	7B02	LCL	1		
1F0301	7BEO	LCL	1		
1F0302	7BEC	LCL	1		
1F0303	7B05	LCL	1		
1F0304	7B08	LCL	1		
1F0305	7B13	LCL	1		
1F0306	7B1B	LCL	1		
1F0307	7B1D	LCL	1		
1F0308	7B49	LCL	1		
1F0309	7B4B	LCL	1		
1F0310	7B64	LCL	1		
1F0312	7B9E	LCL	1		
1F0313	7BAB	LCL	1		
1F0314	7BB8	LCL	1		
1F0315	7C2F	LCL	1		
1F0316	7C39	LCL	1		
1F0317	7C53	LCL	1		
1F0318	7C5D	LCL	1		
1F0319	7C5D	LCL	1		
1F0320	7C9B	LCL	1		
1F0321	7C8D	LCL	1		
1F0322	7C97	LCL	1		
1F0323	7CB4	LCL	1		
1F0324	7CD0	LCL	1		
1F0325	7CE0	LCL	1		
1F0326	7D19	LCL	1		
1F0327	7D15	LCL	1		
1F0328	7D32	LCL	1		
1F0329	7D40	LCL	1		
1F0330	7D4C	LCL	1		
1F0331	7D56	LCL	1		
1F0332	7D56	LCL	1		
1F0338	7E28	LCL	1		
1F0339	7E2F	LCL	1		
1F0341	7E5D	LCL	1		
1F0342	7E59	LCL	1		
1F0343	7E7A	LCL	1		
1F0344	7EBE	LCL	1		
1F0347	7EC3	LCL	1		
1F0348	7F0B	LCL	1		
1F0349	7EE1	LCL	1		
1F0350	7EF6	LCL	1		
1F0351	7F04	LCL	1		
1F0352	7F06	LCL	1		
1F0353	7F38	LCL	1		
1F0354	7F2F	LCL	1		

KARMA 08/31/82-----				Symbol Table 12/13/1983 11:38 AM PG172
SYMBOL	VALUE	TYPE	COUNT	
rfo355	7F4D	LCL	1	
rfo356	7F97	LCL	1	
rfo357	7FAC	LCL	1	
fo0070	6804	LCL	1	
fo0085	6928	LCL	1	
fo0095	6985	LCL	1	
fo0098	69C5	LCL	1	
fo0167	6F3D	LCL	1	
fo0177	6FF4	LCL	1	
fo0194	7229	LCL	1	
fo0230	749E	LCL	1	
fo0231	74B0	LCL	1	
fo0265	7792	LCL	1	
fo0283	78B0	LCL	1	
fo0298	7A66	LCL	1	
fo0334	7D83	LCL	1	
fo0360	7FA1	LCL	1	
rtn003	61B2	LCL	6	
rtn017	627F	LCL	1	
rtn026	6336	LCL	3	
rtn038	6565	LCL	3	
rtn047	65DF	LCL	1	
rtn048	6658	LCL	1	
rtn052	6674	LCL	1	
rtn053	66A1	LCL	1	
rtn054	66C9	LCL	2	
rtn064	6759	LCL	1	
rtn066	679E	LCL	3	
rtn069	67ED	LCL	5	
rtn072	6820	LCL	1	
rtn074	683C	LCL	1	
rtn080	68AD	LCL	1	
rtn084	690F	LCL	2	
rtn085	693C	LCL	2	
rtn092	698C	LCL	1	
rtn096	6998	LCL	1	
rtn100	69DB	LCL	2	
rtn104	6A58	LCL	1	
rtn107	6A85	LCL	1	
rtn113	6B25	LCL	2	
rtn126	6BF5	LCL	2	
rtn142	6D1D	LCL	1	
rtn152	6E2F	LCL	2	
rtn158	6E8F	LCL	2	
rtn164	6EF6	LCL	1	
rtn169	6F62	LCL	1	
rtn191	71C2	LCL	2	
rtn237	7573	LCL	1	
rtn249	75AD	LCL	1	
rtn251	7618	LCL	1	
rtn253	7648	LCL	1	
rtn256	76A4	LCL	1	
rtn259	76E6	LCL	1	
rtn260	7703	LCL	1	

HEADING	Table of Contents	PAGE	PG174
entries and externals	1		
parse time	15		
utilities	16		
PUSHUS, utility to push token and other stuff	17		
DELET, parse routine for DELETE	18		
CLEAR, parse routine for CLEAR ASSIGN W	19		
INSER, parse routine for INSERT, REPLACE and S	20		
SEEK, parse routine for the SEEK statement	22		
COMMON, parse routine for COMMON	23		
CHAIN, parse routine for chain	25		
PARSE CALL, entry points and externals	26		
MCALL, parse routine for CALL	27		
PRINT W parsing entry points and externals	32		
here is the Print W Using parsing code	33		
utility group 2 entries and externals	34		
POSIT, position the data pointer	35		
POSITL, position data pointer in text	37		
DATLIN, get the next data line	38		
TYPER, determine the file type	39		
INCPTR, pre-increment data pointer	40		
INCTX, inc data pointer in text file	41		
CHEMLN, checks text file ptr for 9999	42		
LPR, LPR+, loop move data routines	43		
STRSCH, str search routine	44		
MUMSCH, num search routine	45		
CHKIOK, get next valid data item	46		
SETCOM, sets the parameters for search	47		
UPPER, corrects the iofile data	48		
CHAR, CHAR+, and CHAR++	49		
LINE#	51		
COUNT#R, data item counter	52		
COUNT#, count data items in line	53		
STARTR, open the iofile	54		
CRCL, calculates the size of the data item	55		
POINTR, get data pointer for search	56		
BEGIN, beginning for SEARCH	57		
functions; entry points and externals	58		
FILES., return channel name	59		
INDEX., returns the absolute loc of data ptr	60		
ITEM., return the relative position of data pt	62		
IVP., the type function rtn data type	63		
LIMLEN, W of data items in line	65		
File Commands, entry points and externals	67		
ELASR., clear the iofile	67		
DELW., delete the next data item in W	68		
INSERT W runtimes, INSIR & INNUM	70		
INNUM, insert num data	73		
REPLACE W runtimes; RPSIR and RPNUM	75		
RPNUM, replace num data item	78		
SEARCH runtimes; SEAR#, and SEARW.	80		
SEARW., search for num item	83		
SEEK runtimes; SEEK#, and SEEKR.	85		
SEKR., seek to rel position	87		
ADVAN., advanced the pointer	89		

KARMA 08/31/82-----				Symbol Table 12/13/1983 11:38 AM PG173
SYMBOL	VALUE	TYPE	COUNT	
rtn263	7724	LCL	2	
rtn266	7789	LCL	2	
rtn282	788C	LCL	1	
rtn289	791F	LCL	1	
rtn297	7A56	LCL	1	
rtn311	7B32	LCL	1	
rtn333	7D81	LCL	1	
rtn340	7E33	LCL	5	
rtn361	7F67	LCL	1	

NUMAS

NOT Manufacturer Supported
recipient agrees NOT to contact manufacturer

HEADING	Table of Contents	PAGE	PG175
entries and externals	91		
STRING ARRAY STUFF	92		
STRLOC initialization	93		
look in the last program	94		
STRLOC in program	96		
allocate the variable	97		
STOR() create array header	99		
EXTs & EXTs	101		
STRENV, env allocate string arrays	102		
SUBENV, allocates the SUB statement	104		
ENIRYS and ENIRNLS	113		
VDALOC deallocates a variable	114		
EXT'S AND EXT'S	115		
DIMIR store 1-dim string array	116		
DIMIR, store 2-dim string array	118		
DIMIV, string array value 1-dim	120		
DIMIV, string array value 2-dim	121		
entries and externals	122		
COMMON, the runtime routine for common	123		
CREATE, builds the .common file	125		
HYCAL, gets necessary info on the var	127		
CHKENV, checks the var list	129		
entry points and externals	131		
CHAIN., runtime routine for chain	132		
CALL, entry points and externals	136		
ALLLOCATION for SUB statement	138		
MPARLO, ptr allocation routine	139		
entry points and externals	147		
PRINT W USING: PRNUS.	148		
entry points and externals	152		
VECHR., handle the V.CHAR handi call	153		

Scan Copyright ©
The Museum of HP Calculators
www.hpmuseum.org

Original content used with permission.

Thank you for supporting the Museum of HP
Calculators by purchasing this Scan!

Please to not make copies of this scan or
make it available on file sharing services.