

RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFF	FFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFF	FFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFF	FFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFF	FFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFF	FFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFF	FFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	

```
RRRRRRRR  NN  NN  000000  DDDDDDDD  AAAAAA  TTTTTTTTTT
RRRRRRRR  NN  NN  000000  DDDDDDDD  AAAAAA  TTTTTTTTTT
RR  RR  NN  NN  00  00  DD  DD  AA  AA  TT
RR  RR  NN  NN  00  00  DD  DD  AA  AA  TT
RR  RR  NNNN  NN  00  00  DD  DD  AA  AA  TT
RR  RR  NNNN  NN  00  00  DD  DD  AA  AA  TT
RRRRRRRR  NN  NN  NN  00  00  DD  DD  AA  AA  TT
RRRRRRRR  NN  NN  NN  00  00  DD  DD  AA  AA  TT
RR  RR  NN  NNNN  00  00  DD  DD  AAAAAAAAAA  TT
RR  RR  NN  NNNN  00  00  DD  DD  AAAAAAAAAA  TT
RR  RR  NN  NN  00  00  DD  DD  AA  AA  TT
RR  RR  NN  NN  00  00  DD  DD  AA  AA  TT
RR  RR  NN  NN  00  00  DD  DD  AA  AA  TT
RR  RR  NN  NN  000000  DDDDDDDD  AA  AA  TT
RR  RR  NN  NN  000000  DDDDDDDD  AA  AA  TT
```

```
....
....
....
....
```

```
LL  IIIIII  SSSSSSSS
LL  IIIIII  SSSSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SSSSSS
LL  II  SSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```



```
1 0001 0 MODULE RNODAT ( !
2 0002 0 IDENT = 'V04-000'
3 P 0003 0 %BLISS32[
4 P 0004 0 ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE, NONEXTERNAL=LONG_RELATIVE)
5 0005 0 ]
6 0006 0 ) =
7 0007 1 BEGIN
8 0008 1
9 0009 1 |
10 0010 1 *****
11 0011 1 *
12 0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
13 0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
14 0014 1 * ALL RIGHTS RESERVED.
15 0015 1 *
16 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21 0021 1 * TRANSFERRED.
22 0022 1 *
23 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25 0025 1 * CORPORATION.
26 0026 1 *
27 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29 0029 1 *
30 0030 1 *
31 0031 1 *****
32 0032 1
33 0033 1 ++
34 0034 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
35 0035 1
36 0036 1 ABSTRACT: Obtains the system date and time in a standard format.
37 0037 1
38 0038 1
39 0039 1 ENVIRONMENT: Specific code for each of: TOPS-10, VMS, TOPS-20, and RSX-11M
40 0040 1
41 0041 1 AUTHOR: R.W.Friday CREATION DATE: February, 1979
42 0042 1 |
```

RNODAT
V04-000

Revision History

N 2
16-Sep-1984 01:35:58
14-Sep-1984 13:07:52

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]RNODAT.BLI;1

Page 2
(2)

:	44	0043	1	%SBTTL 'Revision History'	
:	45	0044	1		
:	46	0045	1	MODIFIED BY:	
:	47	0046	1		
:	48	0047	1	008	KAD00008 Keith Dawson 07-Mar-1983
:	49	0048	1		Global edit of all modules. Updated module names, ids, copyright dates. Changed require files to BLISS library.
:	50	0049	1		
:	51	0050	1		
:	52	0051	1	--	


```

54      0052 1 %SBTTL 'Module Level Declarations'
55      0053 1
56      0054 1 TABLE OF CONTENTS:
57      0055 1
58      0056 1
59      0057 1
60      0058 1 INCLUDE FILES:
61      0059 1
62      U 0060 1 %IF %BLISS(BLISS36) %THEN
63      UU 0061 1 %IF %SWITCHES(TOPS20) %THEN
64      UU 0062 1 LIBRARY 'SYS:MONSYM'; ! TOPS-20 system symbol definitions
65      UU 0063 1 REQUIRE 'REQ:JSYSDEF';
66      UU 0064 1 %ELSE
67      UU 0065 1 LIBRARY 'BLI:UUOSYM';
68      U 0066 1 %FI
69      0067 1 %FI
70      0068 1
71      U 0069 1 %IF %BLISS(BLISS16) %THEN
72      U 0070 1 LIBRARY 'RSX:RSX11M';
73      0071 1 %FI
74      0072 1
75      0073 1 MACROS:
76      0074 1
77      0075 1
78      0076 1
79      0077 1 EQUATED SYMBOLS:
80      0078 1
81      0079 1
82      0080 1
83      0081 1 OWN STORAGE:
84      0082 1
85      0083 1
86      0084 1
87      0085 1 EXTERNAL REFERENCES:
88      0086 1
89      0087 1 %IF %BLISS(BLISS32) %THEN
90      0088 1 EXTERNAL ROUTINE
91      0089 1 SYSS$NUMTIM;
92      0090 1 %FI
93      0091 1
94      0092 1 GLOBAL ROUTINE rnodat (pieces) : NOVALUE =
95      0093 1
96      0094 1 ++
97      0095 1 FUNCTIONAL DESCRIPTION:
98      0096 1
99      0097 1 This routine obtains the system date and time, breaks it
100     0098 1 up into its constituents, and returns the results.
101     0099 1
102     0100 1 FORMAL PARAMETERS:
103     0101 1
104     0102 1 PIECES is a vector in which the results are returned.
105     0103 1
106     0104 1 IMPLICIT INPUTS: None
107     0105 1
108     0106 1 IMPLICIT OUTPUTS: None
109     0107 1
110     0108 1 ROUTINE VALUE:
```


Module Level Declarations

```
111 0109 1 | COMPLETION CODES:      None
112 0110 1 |
113 0111 1 | SIDE EFFECTS: None
114 0112 1 |
115 0113 1 | --
116 0114 1 |
117 0115 2 | BEGIN
118 0116 2 |
119 0117 2 | %IF %BLISS(BLISS32) %THEN
120 0118 2 | LOCAL
121 0119 2 |     time_32 : VECTOR [7,2];
122 0120 2 | %FI
123 0121 2 |
124 U 0122 2 | %IF %BLISS(BLISS16) %THEN
125 U 0123 2 | LOCAL
126 U 0124 2 |     time_16 : VECTOR [8];
127 U 0125 2 | %FI
128 0126 2 |
129 0127 2 | MAP
130 0128 2 |     pieces : REF VECTOR [6];
131 0129 2 |
132 0130 2 | BIND
133 0131 2 |     year      = pieces [0],
134 0132 2 |     month     = pieces [1],
135 0133 2 |     monthday  = pieces [2],
136 0134 2 |     hours     = pieces [3],
137 0135 2 |     minutes   = pieces [4],
138 0136 2 |     seconds   = pieces [5];
139 0137 2 |
140 U 0138 2 | %IF %BLISS(BLISS36) %THEN
141 U 0139 2 | %IF %SWITCHES(TOPS20)
142 U 0140 2 | %THEN
143 U 0141 2 |     MACRO
144 U 0142 2 |         lh = 18,18 %,
145 U 0143 2 |         rh = 0,18 %;
146 U 0144 2 |
147 U 0145 2 |     LOCAL
148 U 0146 2 |         year_month,      ! Year,,month # (month 0 = January)
149 U 0147 2 |         monthday_weekday, ! Day of month,,day of week (day 0 = Monday)
150 U 0148 2 |         status_time,     ! status,,time (time in seconds since 0 hrs.)
151 U 0149 2 |         temp;
152 U 0150 2 |
153 U 0151 2 |     jsys_odcnv(-1,0 : year_month,monthday_weekday,status_time );
154 U 0152 2 |     year      = .(year_month)<lh>;
155 U 0153 2 |     month     = .(year_month)<rh> + 1;
156 U 0154 2 |     monthday  = .(monthday_weekday)<lh> + 1;
157 U 0155 2 |     temp      = .(status_time)<rh>;
158 U 0156 2 |     hours     = .temp/3600;
159 U 0157 2 |     temp      = .temp - (.hours * 3600);
160 U 0158 2 |     minutes   = .temp/60;
161 U 0159 2 |     seconds   = .temp - (.minutes * 60);
162 U 0160 2 | %ELSE
163 U 0161 2 |     LOCAL
164 U 0162 2 |         system_date,
165 U 0163 2 |         system_mstime;
166 U 0164 2 |     MACRO
167 U 0165 2 |         t10_mstime =
```

!Map PIECES into its constituents.


```

168      U 0166      2      BEGIN
169      UU 0167      2      BUILTIN uuo;
170      UU 0168      2      REGISTER milliseconds;
171      UU 0169      2      uuo (0, MTIME (milliseconds));
172      UU 0170      2      .milliseconds
173      UU 0171      2      END %;
174      UU 0172      2      MACRO
175      UU 0173      2      t10_date =
176      UU 0174      2      BEGIN
177      UU 0175      2      BUILTIN uuo;
178      UU 0176      2      REGISTER current_date;
179      UU 0177      2      uuo (0, DATE (current_date));
180      UU 0178      2      .current_date
181      UU 0179      2      END %;
182      U 0180      2      %FI
183      0181      2      %FI
184      0182      2
185      U 0183      2      %IF %BLISS(BLISS36) %THEN
186      UU 0184      2      %IF %SWITCHES(TOPS20)
187      UU 0185      2      %THEN
188      UU 0186      2      %ELSE
189      UU 0187      2      system_date      = t10_date;      !Fetch system date.
190      UU 0188      2      system_mstime    = t10_mstime;    !Fetch system time.
191      UU 0189      2
192      UU 0190      2      seconds = .system_mstime/1000;    !Convert to seconds,
193      UU 0191      2      seconds = .seconds MOD 60;        !and compute seconds into current minute.
194      UU 0192      2
195      UU 0193      2      system_mstime = .system_mstime/(1000*60); !Toss away odd milliseconds and seconds.
196      UU 0194      2
197      UU 0195      2      MINUTES = .system_mstime MOD 60;    !Minutes into current hour.
198      UU 0196      2
199      UU 0197      2      hours = .system_mstime/60;        !Current hour.
200      UU 0198      2
201      UU 0199      2      monthday = 1 + (.system_date MOD 31); !Day of current month.
202      UU 0200      2
203      UU 0201      2      system_date = .system_date/31;    !Drop days.
204      UU 0202      2
205      UU 0203      2      month = 1 + (.system_date MOD 12); !Current month.
206      UU 0204      2
207      UU 0205      2      year = 1964 + .system_date/12;    !Current year.
208      U 0206      2      %FI
209      0207      2      %FI
210      0208      2
211      U 0209      2      %IF %BLISS(BLISS16) %THEN
212      UU 0210      2      GTIMSS (time_16);                !Get binary date as an 8-word vector.
213      UU 0211      2
214      UU 0212      2      INCR i FROM 0 TO 5 DO
215      UU 0213      2          pieces[i] = .time_16[i];        !Copy it to PIECES.
216      UU 0214      2      %FI
217      0215      2
218      UU 0216      2      %IF %BLISS(BLISS32) %THEN
219      UU 0217      2      SYSSNUMTIM (time_32, 0);          !Get binary date as a short-word vector.
220      UU 0218      2
221      UU 0219      2      INCR i FROM 0 TO 5 DO
222      UU 0220      2          pieces[i] = .time_32[i];        !Convert from short-word to long word.
223      UU 0221      2      %FI
224      0222      2

```

RNODAT
V04-000

Module Level Declarations

E 3
16-Sep-1984 01:35:58
14-Sep-1984 13:07:52

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]RNODAT.BLI;1

Page 6
(3)

```
: 225      0223 2   RETURN;
: 226      0224 2
: 227      0225 1   END;

                                !End of RNODAT
```

.TITLE RNODAT
.IDENT \V04-000\

.EXTRN SYSSNUMTIM

.PSECT \$CODE\$,NOWRT,2

.ENTRY RNODAT, Save R2

SUBL2 #16, SP

MOVL PIECES, R2

CLRL -(SP)

PUSHAB TIME_32

CALLS #2, SYSSNUMTIM

CLRL I

MOVZWL TIME_32[I], (R2)[I]

AOBLEQ #5, I, 1\$

RET

: 0092

: 0131

: 0217

: 0219

: 0220

: 0225

; Routine Size: 33 bytes, Routine Base: \$CODE\$ + 0000

```
: 228      0226 1
: 229      0227 1 END
: 230      0228 0 ELUDOM

                                !End of module
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	33	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:RNODAT/OBJ=OBJ\$:RNODAT MSRC\$:RNODAT/UPDATE=(ENH\$:RNODAT)

```
: Size:      33 code + 0 data bytes
: Run Time:   00:02.3
: Elapsed Time: 00:06.0
: Lines/CPU Min: 6026
: Lexemes/CPU-Min: 21806
: Memory Used: 23 pages
```


RNODAT
V04-000

Module Level Declarations

F 3
16-Sep-1984 01:35:58

VAX-11 Bliss-32 V4.0-742

Page 7

; Compilation Complete

RNI
VO

0348 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

0348	0349	0350	0351	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399																		
0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459										
0460	0461	0462	0463	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511	0512	0513	0514	0515	0516	0517	0518	0519										
0520	0521	0522	0523	0524	0525	0526	0527	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575	0576	0577	0578	0579										
0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639										
0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699										
0700	0701	0702	0703	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751	0752	0753	0754	0755	0756	0757	0758	0759										
0760	0761	0762	0763	0764	0765	0766	0767	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815	0816	0817	0818	0819										
0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879										
0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939										
0940	0941	0942	0943	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991	0992	0993	0994	0995	0996	0997	0998	0999										
1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069
1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139
1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199										
1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269
1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339
1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399										
1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469
1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535	1536	1537	1538	1539
1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599										
1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663	1664	1665	1666	1667	1668	1669
1670	1671	1672	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695	1696	1697	1698	1699																																								