Various Filters and Techniques for:

# Serial Data Analysis

# Part 2, TASS Protocol

**11 July 2008**

**Eric Hamilton**


## Contents



[1] `$Header: d:/Binder4/sda2/RCS/Sda2Tass.tex,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $`

This page intentionally left blank

# 1 Filters for TASS Protocol Processing

## 1.1 .BAT files for processing TASS data

### 1.1.1 KTass.bat

```
1  @echo off
2  rem $Header: d:/Binder4/sda2/RCS/KTass.bat,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
3  rem copy original\%1.txt %1.cap
4  makesecs %1.txt %1.out
5  tassfix  %1.out %1.fix
6  tassdecd %1.fix %1.dcd
7  tassintp %1.dcd %1.int
8  l %1.int
```

[2] $Header: d:/Binder4/sda2/RCS/Tass.inc,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $

### 1.1.2 tasssamp.1 Test file

```
  1  FTS capture buffer (6/6/2008 12:06:52 PM)
  2  Event 1 (6/6/2008 11:25:40.704785 AM) through
  3  Event 1,887 (6/6/2008 12:03:37.753288 PM)
  4
  5  Sd  Timestamp                  Hx
  6  DTE 6/6/2008 11:25:40.704785 AM f8
  7  DTE 6/6/2008 11:25:40.705817 AM 41
  8  DTE 6/6/2008 11:25:40.706854 AM 2a
  9  DTE 6/6/2008 11:25:40.707893 AM 01
 10  DTE 6/6/2008 11:25:40.708932 AM 1f
 11  DTE 6/6/2008 11:25:40.709969 AM 02
 12  DTE 6/6/2008 11:25:40.711011 AM 52
 13  DTE 6/6/2008 11:25:40.712046 AM 53
 14  DTE 6/6/2008 11:25:40.713087 AM 86
 15  DTE 6/6/2008 11:26:07.382281 AM f8
 16  DTE 6/6/2008 11:26:07.383312 AM 41
 17  DTE 6/6/2008 11:26:07.384351 AM 2a
 18  DTE 6/6/2008 11:26:07.385387 AM 01
 19  DTE 6/6/2008 11:26:07.386432 AM 1f
 20  DTE 6/6/2008 11:26:07.387463 AM 02
 21  DTE 6/6/2008 11:26:07.388506 AM 41
 22  DTE 6/6/2008 11:26:07.389558 AM 57
 23  DTE 6/6/2008 11:26:07.390419 AM 81
 24  DTE 6/6/2008 11:26:12.425191 AM f8
 25  DTE 6/6/2008 11:26:12.426227 AM 41
 26  DTE 6/6/2008 11:26:12.427267 AM 2a
 27  DTE 6/6/2008 11:26:12.428305 AM 01
 28  DTE 6/6/2008 11:26:12.429340 AM 1f
 29  DTE 6/6/2008 11:26:12.430383 AM 02
 30  DTE 6/6/2008 11:26:12.431417 AM 41
 31  DTE 6/6/2008 11:26:12.432458 AM 57
 32  DTE 6/6/2008 11:26:12.433493 AM 81
 33  DTE 6/6/2008 11:26:17.471860 AM f8
 34  DTE 6/6/2008 11:26:17.472888 AM 41
 35  DTE 6/6/2008 11:26:17.473931 AM 2a
 36  DTE 6/6/2008 11:26:17.474967 AM 01
 37  DTE 6/6/2008 11:26:17.476007 AM 1f
 38  DTE 6/6/2008 11:26:17.477043 AM 02
 39  DTE 6/6/2008 11:26:17.478085 AM 41
 40  DTE 6/6/2008 11:26:17.479120 AM 57
 41  DTE 6/6/2008 11:26:17.480160 AM 81
 42  DTE 6/6/2008 11:26:31.015916 AM f8
 43  DTE 6/6/2008 11:26:31.016947 AM 41
 44  DTE 6/6/2008 11:26:31.017988 AM 2a
 45  DTE 6/6/2008 11:26:31.019023 AM 01
 46  DTE 6/6/2008 11:26:31.020065 AM 1f
 47  DTE 6/6/2008 11:26:31.021100 AM 02
 48  DTE 6/6/2008 11:26:31.022142 AM 53
 49  DTE 6/6/2008 11:26:31.023176 AM 48
 50  DTE 6/6/2008 11:26:31.024219 AM 8c
 51  DTE 6/6/2008 11:26:36.065136 AM f8
 52  DTE 6/6/2008 11:26:36.066172 AM 41
 53  DTE 6/6/2008 11:26:36.067208 AM 2a
 54  DTE 6/6/2008 11:26:36.068250 AM 01
 55  DTE 6/6/2008 11:26:36.069285 AM 1f
 56  DTE 6/6/2008 11:26:36.070327 AM 02
 57  DTE 6/6/2008 11:26:36.071195 AM 53
 58  DTE 6/6/2008 11:26:36.072403 AM 48
 59  DTE 6/6/2008 11:26:36.073438 AM 8c
 60  DTE 6/6/2008 11:26:41.111908 AM f8
 61  DTE 6/6/2008 11:26:41.112938 AM 41
 62  DTE 6/6/2008 11:26:41.113980 AM 2a
 63  DTE 6/6/2008 11:26:41.115015 AM 01
 64  DTE 6/6/2008 11:26:41.116056 AM 1f
 65  DTE 6/6/2008 11:26:41.117093 AM 02
 66  DTE 6/6/2008 11:26:41.118142 AM 53
 67  DTE 6/6/2008 11:26:41.119169 AM 48
 68  DTE 6/6/2008 11:26:41.120211 AM 8c
 69  DTE 6/6/2008 11:26:51.434734 AM f8
 70  DTE 6/6/2008 11:26:51.435770 AM 41
 71  DTE 6/6/2008 11:26:51.436806 AM 2a
 72  DTE 6/6/2008 11:26:51.437849 AM 01
 73  DTE 6/6/2008 11:26:51.438884 AM 1f
 74  DTE 6/6/2008 11:26:51.439925 AM 02
 75  DTE 6/6/2008 11:26:51.440959 AM 49
 76  DTE 6/6/2008 11:26:51.442001 AM 3f
 77  DTE 6/6/2008 11:26:51.443026 AM 81
 78  DTE 6/6/2008 11:26:56.470845 AM f8
 79  DTE 6/6/2008 11:26:56.471875 AM 41
 80  DTE 6/6/2008 11:26:56.473085 AM 2a
 81  DTE 6/6/2008 11:26:56.474118 AM 01
 82  DTE 6/6/2008 11:26:56.475162 AM 1f
 83  DTE 6/6/2008 11:26:56.476194 AM 02
 84  DTE 6/6/2008 11:26:56.477238 AM 49
 85  DTE 6/6/2008 11:26:56.478271 AM 3f
 86  DTE 6/6/2008 11:26:56.479314 AM 81
 87  DTE 6/6/2008 11:27:01.518061 AM f8
 88  DTE 6/6/2008 11:27:01.519094 AM 41
 89  DTE 6/6/2008 11:27:01.520135 AM 2a
 90  DTE 6/6/2008 11:27:01.521171 AM 01
 91  DTE 6/6/2008 11:27:01.522212 AM 1f
 92  DTE 6/6/2008 11:27:01.523247 AM 02
 93  DTE 6/6/2008 11:27:01.524287 AM 49
 94  DTE 6/6/2008 11:27:01.525329 AM 3f
 95  DTE 6/6/2008 11:27:01.526364 AM 81
 96  DTE 6/6/2008 11:27:09.788213 AM f8
 97  DTE 6/6/2008 11:27:09.789262 AM 41
 98  DTE 6/6/2008 11:27:09.790287 AM 2a
 99  DTE 6/6/2008 11:27:09.791328 AM 01
100  DTE 6/6/2008 11:27:09.792363 AM 1f
101  DTE 6/6/2008 11:27:09.793405 AM 02
102  DTE 6/6/2008 11:27:09.794439 AM 47
103  DTE 6/6/2008 11:27:09.795481 AM 3f
104  DTE 6/6/2008 11:27:09.796517 AM 8f
105  DTE 6/6/2008 11:27:14.830058 AM f8
106  DTE 6/6/2008 11:27:14.831096 AM 41
107  DTE 6/6/2008 11:27:14.832130 AM 2a
108  DTE 6/6/2008 11:27:14.833170 AM 01
109  DTE 6/6/2008 11:27:14.834207 AM 1f
110  DTE 6/6/2008 11:27:14.835249 AM 02
111  DTE 6/6/2008 11:27:14.836283 AM 47
112  DTE 6/6/2008 11:27:14.837325 AM 3f
113  DTE 6/6/2008 11:27:14.838361 AM 8f
114  DTE 6/6/2008 11:27:19.877096 AM f8
115  DTE 6/6/2008 11:27:19.878133 AM 41
116  DTE 6/6/2008 11:27:19.879167 AM 2a
117  DTE 6/6/2008 11:27:19.880210 AM 01
118  DTE 6/6/2008 11:27:19.881245 AM 1f
119  DTE 6/6/2008 11:27:19.882286 AM 02
120  DTE 6/6/2008 11:27:19.883320 AM 47
121  DTE 6/6/2008 11:27:19.884363 AM 3f
122  DTE 6/6/2008 11:27:19.885398 AM 8f
123  DTE 6/6/2008 11:27:27.614383 AM f8
124  DTE 6/6/2008 11:27:27.615420 AM 41
125  DTE 6/6/2008 11:27:27.616455 AM 2a
126  DTE 6/6/2008 11:27:27.617497 AM 01
127  DTE 6/6/2008 11:27:27.618532 AM 1f
128  DTE 6/6/2008 11:27:27.619573 AM 02
129  DTE 6/6/2008 11:27:27.620608 AM 44
130  DTE 6/6/2008 11:27:27.621651 AM 3f
131  DTE 6/6/2008 11:27:27.622686 AM 8c
132  DTE 6/6/2008 11:27:32.658155 AM f8
133  DTE 6/6/2008 11:27:32.659184 AM 41
134  DTE 6/6/2008 11:27:32.660226 AM 2a
135  DTE 6/6/2008 11:27:32.661267 AM 01
136  DTE 6/6/2008 11:27:32.662302 AM 1f
137  DTE 6/6/2008 11:27:32.663344 AM 02
138  DTE 6/6/2008 11:27:32.664379 AM 44
```

## 1.2 Filters for processing TASS protocol data

### 1.2.1 TassDecd

TassDecd is used to decode TASS protocol messages that have been processed by TassFix.

1. This filter is written in Flex.
2. This filter is suitable for use with TASS protocol only.
3. This filter requires preprocessing by either MakeSecs.
4. A typical use of this filter is shown in: `KTass.bat` (Section 1.1.1, page 3)
5. **Listing of: `htass01.fix` a typical input file.**

```
 1  $ Id: MakeSecs.l,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
 2  $Header: d:/Binder4/sda2/RCS/Htass01.fix,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
 3  FTS capture buffer (6/6/2008 12:06:52 PM)
 4  Event 1 (6/6/2008 11:25:40.704785 AM) through
 5  Event 1,887 (6/6/2008 12:03:37.753288 PM)
 6
 7
 8       1,       1:  DTE     0.000000   0.000000 f8 41 2a 01 1f 02 52 53 86
 9       1,      10:  DTE    26.677496  26.669194 f8 41 2a 01 1f 02 41 57 81
10       1,      19:  DTE    31.720406   5.034772 f8 41 2a 01 1f 02 41 57 81
11       1,      28:  DTE    36.767075   5.038367 f8 41 2a 01 1f 02 41 57 81
12       1,      37:  DTE    50.311131  13.535756 f8 41 2a 01 1f 02 53 48 8c
13       1,      46:  DTE    55.360351   5.040917 f8 41 2a 01 1f 02 53 48 8c
14       1,      55:  DTE    60.407123   5.038470 f8 41 2a 01 1f 02 53 48 8c
15       1,      64:  DTE    70.729949  10.314523 f8 41 2a 01 1f 02 49 3f 81
16       1,      73:  DTE    75.766060   5.027819 f8 41 2a 01 1f 02 49 3f 81
17       1,      82:  DTE    80.813276   5.038747 f8 41 2a 01 1f 02 49 3f 81
18       1,      91:  DTE    89.083428   8.261849 f8 41 2a 01 1f 02 47 3f 8f
19       1,     100:  DTE    94.125273   5.033541 f8 41 2a 01 1f 02 47 3f 8f
20       1,     109:  DTE    99.172311   5.038735 f8 41 2a 01 1f 02 47 3f 8f
```

6. **Listing of: `htass01.dcd` a typical output file.**

```
 1  $ Id: MakeSecs.l,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
 2  $Header: d:/Binder4/sda2/RCS/Htass01.dcd,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
 3  FTS capture buffer (6/6/2008 12:06:52 PM)
 4  Event 1 (6/6/2008 11:25:40.704785 AM) through
 5  Event 1,887 (6/6/2008 12:03:37.753288 PM)
 6
 7
 8       1,       1:  DTE     0.000000   0.000000 f8 41 2a 01 1f 02 52 53 86 :  T=1.1 * G=1 F=31 L=2:  RS
 9       1,      10:  DTE    26.677496  26.669194 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW
10       1,      19:  DTE    31.720406   5.034772 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW
11       1,      28:  DTE    36.767075   5.038367 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW
12       1,      37:  DTE    50.311131  13.535756 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH
13       1,      46:  DTE    55.360351   5.040917 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH
14       1,      55:  DTE    60.407123   5.038470 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH
15       1,      64:  DTE    70.729949  10.314523 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I?
16       1,      73:  DTE    75.766060   5.027819 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I?
17       1,      82:  DTE    80.813276   5.038747 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I?
18       1,      91:  DTE    89.083428   8.261849 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G?
19       1,     100:  DTE    94.125273   5.033541 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G?
20       1,     109:  DTE    99.172311   5.038735 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G?
```

### 1.2.2 Listing of TassDecd.l

```
%{
//$ Id: MakeSecs.l,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
//$Header: d:/Binder4/sda2/RCS/TassDecd.l,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
//FTS capture file: E:\Capture\newtest2.cfa (12/15/2005 7:56:53 AM)
//Event 1 (12/15/2005 7:34:01.760340 AM) through
//Event 1,152 (12/15/2005 7:55:41.497004 AM)
//
//     1,       1: DCE      0.000000   0.000000 f8 21 2a 01 1f 02 52 53 86
//     1,      10: DCE     29.868193  29.859879 f8 21 2a 01 1f 02 41 57 81
//     1,      19: DCE     30.880875   1.004420 f8 21 2a 01 1f 02 41 57 81
//     1,      28: DCE     31.890736   1.001552 f8 21 2a 01 1f 02 41 57 81
//123456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
//
// $Header: d:/Binder4/sda2/RCS/TassDecd.l,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
// TassDecd is used to decode TASS protocol messages that have been
//          processed by TassFix.
//
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#undef yywrap

void GetDAddress(void);
void GetGcu(void);
void GetSAddress(void);
void GetLength(void);
void GetChecksum(void);
void GetMessage(void);

int DAddress;
int Gcu;
int SAddress;
int Length;
int Checksum;

char OutputMessage[10000];

%}
%x GETLENGTH MESSAGE
%x GETDADDRESS GETSTAR GETGCU GETSADDRESS
%%
^"$".*\n     |
^"E".*\n     |
^"F".*\n     |
^"T".*\n     |
^\n          ECHO;

^.*"f8"                     ECHO; OutputMessage[0] = '\0'; BEGIN GETDADDRESS;
<GETDADDRESS>" "[0-9a-f]{2} fprintf(yyout," %c%c",yytext[1],yytext[2]); GetDAddress(); BEGIN GETSTAR;
<GETSTAR>" "[0-9a-f]{2}     fprintf(yyout," %c%c",yytext[1],yytext[2]); GetMessage(); BEGIN GETGCU;
<GETGCU>" "[0-9a-f]{2}      fprintf(yyout," %c%c",yytext[1],yytext[2]); GetGcu(); BEGIN GETSADDRESS;
<GETSADDRESS>" "[0-9a-f]{2} fprintf(yyout," %c%c",yytext[1],yytext[2]); GetSAddress(); BEGIN GETLENGTH;
<GETLENGTH>" "[0-9a-f]{2}   fprintf(yyout," %c%c",yytext[1],yytext[2]); GetLength(); BEGIN MESSAGE;

<MESSAGE>" "[0-9a-f]{2}     fprintf(yyout," %c%c",yytext[1],yytext[2]); GetMessage();
<MESSAGE>" 8"[0-9a-f]$      fprintf(yyout," 8%c",yytext[2]); GetChecksum(); BEGIN 0;

.|\n     ;
%%


void GetDAddress(void)
{
char tempstring[50];
int temp;

```

```
69          DAddress = strtol(yytext,(char**)NULL,16);
70  //fprintf(yyout,".%d.",DAddress);
71
72          temp = 0x1F & DAddress;
73          sprintf(tempstring,":  T=%d.",temp);
74          strcat(OutputMessage,tempstring);
75
76          temp = (0xE0 & DAddress) > 5;
77          sprintf(tempstring,"%d ",temp);
78          strcat(OutputMessage,tempstring);
79  }
80
81
82  void GetGcu(void)
83  {
84  char tempstring[50];
85
86          Gcu = strtol(yytext,(char**)NULL,16);
87  //fprintf(yyout,".%d.",Gcu);
88
89          sprintf(tempstring," G=%d",Gcu);
90          strcat(OutputMessage,tempstring);
91  }
92
93
94  void GetSAddress(void)
95  {
96  char tempstring[50];
97
98          SAddress = strtol(yytext,(char**)NULL,16);
99  //fprintf(yyout,".%d.",SAddress);
100         sprintf(tempstring," F=%d ",SAddress);
101         strcat(OutputMessage,tempstring);
102 }
103
104
105 void GetLength(void)
106 {
107 char tempstring[50];
108
109         Length = strtol(yytext,(char**)NULL,10);
110 //fprintf(yyout,".%d.",Length);
111
112         Length = strtol(yytext,(char**)NULL,16);
113         sprintf(tempstring,"L=%d:  ",Length);
114         strcat(OutputMessage,tempstring);
115 }
116
117
118 void GetChecksum(void)
119 {
120         Checksum = strtol(yytext,(char**)NULL,16);
121 //fprintf(yyout,".%d.",Checksum);
122         fprintf(yyout," %s\n",OutputMessage);
123 }
124
125
126 void GetMessage(void)
127 {
128 char tempstring[5];
129 int  temp;
130
131         temp = strtol(yytext,(char**)NULL,16);
132 //fprintf(yyout,".%d ",temp);
133
134         if (isprint(temp)) sprintf(tempstring,"%c",temp);
135         else
136         sprintf(tempstring," 0x%02X ",temp);
137
138         strcat(OutputMessage,tempstring);
139 }
140
```

```
141
142
143 // yywrap is for End Of File processing, 1 = done
144 int yywrap(void)
145 {
146         return 1;
147 }
148
149 main(int argc, char *argv[])
150 {
151         fprintf(stderr,"$Id: TassDecd.l,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $\n");
152
153         // Do we have to deal with stdin/stdout?
154         // 3 = one for the program name and one each for input and output names
155         if (argc == 3) {
156             // Nope
157             // Get over the program, etc., name
158             argc--;
159             argv++;
160             yyin = fopen(argv[0],"r");
161             if (yyin == NULL) {
162                 fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
163                                 argv[0]);
164                 exit(EXIT_FAILURE);
165             }
166
167             // Now get over the input file name and get to the output file name
168             argc--;
169             argv++;
170             yyout = fopen(argv[0],"w");
171             if (yyout == NULL) {
172                 fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
173                                 argv[0]);
174                 exit(EXIT_FAILURE);
175             }
176         }
177         else {
178             // Woops we are using std... type io
179             yyin  = stdin;
180             yyout = stdout;
181         }
182
183         yylex();
184         exit(EXIT_SUCCESS);
185 }
186
187 // $Log: TassDecd.l,v $
188 // Revision 1.1  2008-06-10 09:25:20-07  Hamilton
189 // Initial revision
190 //
191 // Revision 1.2  2008-06-06 10:26:36-07  Hamilton
192 // Normal middle of the day data savings.
193 //
194 // Revision 1.1  2008-06-06 08:44:07-07  Hamilton
195 // Initial revision
196 //
197 // Revision 1.10  2007-02-08 06:45:52-08  Hamilton
198 // Normal end of day data saving
199 //
200
```

### 1.2.3 TassFix

TassFix is used to break down the output of Makesecs into TASS messages.

TassIntp is used to interpert the output of TassDecd to make the output codes somewhat more understandable.

1. This filter is written in Flex.
2. This filter is suitable for use with TASS protocol only.
3. This filter requires preprocessing by either MakeSecs.
4. A typical use of this filter is shown in: `KTass.bat` (Section 1.1.1, page 3)
5. **Listing of: `htass01.out` a typical input file.**

```
 1  $ Id: MakeSecs.l,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
 2  $Header: d:/Binder4/sda2/RCS/Htass01.out,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
 3  FTS capture buffer (6/6/2008 12:06:52 PM)
 4  Event 1 (6/6/2008 11:25:40.704785 AM) through
 5  Event 1,887 (6/6/2008 12:03:37.753288 PM)
 6
 7
 8       1,       1:  DTE     0.000000   0.000000 f8
 9       1,       2:  DTE     0.001032   0.001032 41
10       1,       3:  DTE     0.002069   0.001037 2a
11       1,       4:  DTE     0.003108   0.001039 01
12       1,       5:  DTE     0.004147   0.001039 1f
13       1,       6:  DTE     0.005184   0.001037 02
14       1,       7:  DTE     0.006226   0.001042 52
15       1,       8:  DTE     0.007261   0.001035 53
16       1,       9:  DTE     0.008302   0.001041 86
17       1,      10:  DTE    26.677496  26.669194 f8
18       1,      11:  DTE    26.678527   0.001031 41
19       1,      12:  DTE    26.679566   0.001039 2a
20       1,      13:  DTE    26.680602   0.001036 01
```

6. **Listing of: `htass01.fix` a typical output file.**

```
 1  $ Id: MakeSecs.l,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
 2  $Header: d:/Binder4/sda2/RCS/Htass01.fix,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
 3  FTS capture buffer (6/6/2008 12:06:52 PM)
 4  Event 1 (6/6/2008 11:25:40.704785 AM) through
 5  Event 1,887 (6/6/2008 12:03:37.753288 PM)
 6
 7
 8       1,       1:  DTE     0.000000   0.000000 f8 41 2a 01 1f 02 52 53 86
 9       1,      10:  DTE    26.677496  26.669194 f8 41 2a 01 1f 02 41 57 81
10       1,      19:  DTE    31.720406   5.034772 f8 41 2a 01 1f 02 41 57 81
11       1,      28:  DTE    36.767075   5.038367 f8 41 2a 01 1f 02 41 57 81
12       1,      37:  DTE    50.311131  13.535756 f8 41 2a 01 1f 02 53 48 8c
13       1,      46:  DTE    55.360351   5.040917 f8 41 2a 01 1f 02 53 48 8c
14       1,      55:  DTE    60.407123   5.038470 f8 41 2a 01 1f 02 53 48 8c
15       1,      64:  DTE    70.729949  10.314523 f8 41 2a 01 1f 02 49 3f 81
16       1,      73:  DTE    75.766060   5.027819 f8 41 2a 01 1f 02 49 3f 81
17       1,      82:  DTE    80.813276   5.038747 f8 41 2a 01 1f 02 49 3f 81
18       1,      91:  DTE    89.083428   8.261849 f8 41 2a 01 1f 02 47 3f 8f
19       1,     100:  DTE    94.125273   5.033541 f8 41 2a 01 1f 02 47 3f 8f
20       1,     109:  DTE    99.172311   5.038735 f8 41 2a 01 1f 02 47 3f 8f
```

### 1.2.4 Listing of TassFix.l

```
%{
//$ Id: MakeSecs.l,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
//$Header: d:/Binder4/sda2/RCS/TassFix.l,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
//FTS capture file: E:\Capture\newtest2.cfa (12/15/2005 7:56:53 AM)
//Event 1 (12/15/2005 7:34:01.760340 AM) through
//Event 1,152 (12/15/2005 7:55:41.497004 AM)
//
//     1,       1: DCE      0.000000   0.000000 f8
//     1,       2: DCE      0.001035   0.001035 21
//     1,       3: DCE      0.002078   0.001043 2a
//     1,       4: DCE      0.003120   0.001042 01
//     1,       5: DCE      0.004162   0.001042 1f
//     1,       6: DCE      0.005168   0.001006 02
//     1,       7: DCE      0.006171   0.001003 52
//     1,       8: DCE      0.007273   0.001102 53
//     1,       9: DCE      0.008314   0.001041 86
//
//123456789 123456789 123456789 123456789 123456789
//
//     1,    1152: DCE   1299.736664   0.001042 81
//
//There were a total of     1152 bytes transferred
//
//There were a total of     1152 DCE bytes transferred
//The first DCE byte came in at 0.000000 seconds from the start of data collection
//The last DCE byte was at 1299.736664 seconds from the start of data collection
//
//There were a total of        0 DTE bytes transferred
//The first DTE byte came in at 0.000000 seconds from the start of data collection
//The last DTE byte was at 0.000000 seconds from the start of data collection
//
// $Header: d:/Binder4/sda2/RCS/TassFix.l,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
// TassFix is used to break down the output of Makesecs into TASS messages.
#include <stdio.h>
#include <stdlib.h>
#undef yywrap
int i;
%}
%x MESSAGE
%%

^"$".*\n           |
^"E".*\n           |
^"F".*\n           |
^"T".*\n           |
^\n                ECHO;

^.*"f8"            fprintf(yyout,"%s",yytext); BEGIN MESSAGE;
<MESSAGE>^.*"f8"   fprintf(yyout,"\n%s",yytext);
<MESSAGE>.*       {i = strlen(yytext);
                   fprintf(yyout," %c%c",yytext[i-2],yytext[i-1]);
                  }

<MESSAGE>^"T".*\n  fprintf(yyout,"\n%s",yytext); BEGIN 0;

<MESSAGE>^\n       ECHO;
<MESSAGE>\n        ;

.|\n               ;
%%
// yywrap is for End Of File processing, 1 = done
int yywrap(void)
{
        return 1;
}

main(int argc, char *argv[])
{
```

```
69          fprintf(stderr,"$Id: TassFix.l,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $\n");
70
71          // Do we have to deal with stdin/stdout?
72          // 3 = one for the program name and one each for input and output names
73          if (argc == 3) {
74              // Nope
75              // Get over the program, etc., name
76              argc--;
77              argv++;
78              yyin = fopen(argv[0],"r");
79              if (yyin == NULL) {
80                  fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
81                                  argv[0]);
82                  exit(EXIT_FAILURE);
83              }
84
85              // Now get over the input file name and get to the output file name
86              argc--;
87              argv++;
88              yyout = fopen(argv[0],"w");
89              if (yyout == NULL) {
90                  fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
91                                  argv[0]);
92                  exit(EXIT_FAILURE);
93              }
94          }
95          else {
96              // Woops we are using std... type io
97              yyin  = stdin;
98              yyout = stdout;
99          }
100
101         yylex();
102         exit(EXIT_SUCCESS);
103 }
104
105 // $Log: TassFix.l,v $
106 // Revision 1.1  2008-06-10 09:25:20-07  Hamilton
107 // Initial revision
108 //
109 // Revision 1.1  2008-06-06 07:19:58-07  Hamilton
110 // Initial revision
111 //
112 // Revision 1.10  2007-02-08 06:45:52-08  Hamilton
113 // Normal end of day data saving
114 //
115
```

### 1.2.5 TassIntp

TassIntp is used to interpert the output of TassDecd to make the output codes somewhat more understandable.

1. This filter is written in Flex.
2. This filter is suitable for use with TASS protocol only.
3. This filter requires preprocessing by either MakeSecs.
4. A typical use of this filter is shown in: `KTass.bat` (Section 1.1.1, page 3)
5. **Listing of: `htass01.dcd` a typical input file.**

```
  1  $ Id: MakeSecs.l,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
  2  $Header: d:/Binder4/sda2/RCS/Htass01.dcd,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
  3  FTS capture buffer (6/6/2008 12:06:52 PM)
  4  Event 1 (6/6/2008 11:25:40.704785 AM) through
  5  Event 1,887 (6/6/2008 12:03:37.753288 PM)
  6
  7
  8       1,       1:  DTE     0.000000   0.000000 f8 41 2a 01 1f 02 52 53 86 :  T=1.1 * G=1 F=31 L=2:  RS
  9       1,      10:  DTE    26.677496  26.669194 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW
 10       1,      19:  DTE    31.720406   5.034772 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW
 11       1,      28:  DTE    36.767075   5.038367 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW
 12       1,      37:  DTE    50.311131  13.535756 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH
 13       1,      46:  DTE    55.360351   5.040917 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH
 14       1,      55:  DTE    60.407123   5.038470 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH
 15       1,      64:  DTE    70.729949  10.314523 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I?
 16       1,      73:  DTE    75.766060   5.027819 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I?
 17       1,      82:  DTE    80.813276   5.038747 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I?
 18       1,      91:  DTE    89.083428   8.261849 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G?
 19       1,     100:  DTE    94.125273   5.033541 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G?
 20       1,     109:  DTE    99.172311   5.038735 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G?
```

6. **Listing of: `htass01.int` a typical output file.**

```
  1  $ Id: MakeSecs.l,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
  2  $Header: d:/Binder4/sda2/RCS/Htass01.int,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
  3  FTS capture buffer (6/6/2008 12:06:52 PM)
  4  Event 1 (6/6/2008 11:25:40.704785 AM) through
  5  Event 1,887 (6/6/2008 12:03:37.753288 PM)
  6
  7
  8       1,       1:  DTE     0.000000   0.000000 f8 41 2a 01 1f 02 52 53 86 :  T=1.1 * G=1 F=31 L=2:  RS -- Reset
Receiver
  9       1,      10:  DTE    26.677496  26.669194 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW -- Request
awake status (ping)
 10       1,      19:  DTE    31.720406   5.034772 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW -- Request
awake status (ping)
 11       1,      28:  DTE    36.767075   5.038367 f8 41 2a 01 1f 02 41 57 81 :  T=1.1 * G=1 F=31 L=2:  AW -- Request
awake status (ping)
 12       1,      37:  DTE    50.311131  13.535756 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH -- Request
State of Health
 13       1,      46:  DTE    55.360351   5.040917 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH -- Request
State of Health
 14       1,      55:  DTE    60.407123   5.038470 f8 41 2a 01 1f 02 53 48 8c :  T=1.1 * G=1 F=31 L=2:  SH -- Request
State of Health
 15       1,      64:  DTE    70.729949  10.314523 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I? -- Request
ID Information
 16       1,      73:  DTE    75.766060   5.027819 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I? -- Request
ID Information
 17       1,      82:  DTE    80.813276   5.038747 f8 41 2a 01 1f 02 49 3f 81 :  T=1.1 * G=1 F=31 L=2:  I? -- Request
ID Information
```

```
  18       1,      91:  DTE    89.083428   8.261849 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G? -- Query
Groups
  19       1,     100:  DTE    94.125273   5.033541 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G? -- Query
Groups
  20       1,     109:  DTE    99.172311   5.038735 f8 41 2a 01 1f 02 47 3f 8f :  T=1.1 * G=1 F=31 L=2:  G? -- Query
Groups
```

### 1.2.6 Listing of TassIntp.l

```
 1  %{
 2  // $Header: d:/Binder4/sda2/RCS/TassIntp.l,v 1.1 2008-06-10 09:25:21-07 Hamilton Exp Hamilton $
 3  // TassIntp is used to interpert the output of TassDecd to make the
 4  //          output codes somewhat more understandable.
 5  //
 6  #include <stdio.h>
 7  #include <stdlib.h>
 8  #undef yywrap
 9  char string[1000];
10  %}
11  %x RESTOFLINE
12  %%
13  ^"$".*\n     |
14  ^"E".*\n     |
15  ^"F".*\n     |
16  ^"T".*\n     |
17  ^\n          ECHO;
18
19
20  ^.*"RC"          ECHO; strcat(string,"Recalibrate"); BEGIN RESTOFLINE;
21  ^.*"IC"          ECHO; strcat(string,"Iris/Aperture Close"); BEGIN RESTOFLINE;
22  ^.*"DC"          ECHO; strcat(string,"Set Compression Level"); BEGIN RESTOFLINE;
23
24  ^.*"#"         ECHO; strcat(string,"Set Device ID"); BEGIN RESTOFLINE;
25
26  ^.*"A"[0-9A-F] ECHO; strcat(string,"Set Auto-Move Speed"); BEGIN RESTOFLINE;
27  ^.*"AL"[0-9]   ECHO; strcat(string,"Select List"); BEGIN RESTOFLINE;
28  ^.*"AQ"        ECHO; strcat(string,"Protected Area Query"); BEGIN RESTOFLINE;
29  ^.*"AS"[0-9]   ECHO; strcat(string,"Begin Auto-Scan Mode"); BEGIN RESTOFLINE;
30  ^.*"AT"        ECHO; strcat(string,"Default Dwell Time"); BEGIN RESTOFLINE;
31  ^.*"AW"        ECHO; strcat(string,"Request awake status (ping)"); BEGIN RESTOFLINE;
32
33  ^.*"B"[0-2]    ECHO; strcat(string,"Color Balance Select auto/manual mode"); BEGIN RESTOFLINE;
34  ^.*"B?"        ECHO; strcat(string,"Request Maximum Data Rate"); BEGIN RESTOFLINE;
35  ^.*"B"[0-9A-F]{1,2}"P" ECHO; strcat(string,"Button, Push");    BEGIN RESTOFLINE;
36  ^.*"B"[0-9A-F]{1,2}"R" ECHO; strcat(string,"Button, Release"); BEGIN RESTOFLINE;
37  ^.*"b"         ECHO; strcat(string,"Set Brightness (Level) value"); BEGIN RESTOFLINE;
38
39  ^.*"C"         ECHO; strcat(string,"Set Communication Parameters"); BEGIN RESTOFLINE;
40
41  ^.*"D?"        ECHO; strcat(string,"Query Devices in a Group"); BEGIN RESTOFLINE;
42  ^.*"DG"        ECHO; strcat(string,"Grab Image"); BEGIN RESTOFLINE;
43  ^.*"DH"        ECHO; strcat(string,"Grab"); BEGIN RESTOFLINE;
44  ^.*"DI"        ECHO; strcat(string,"Detect change"); BEGIN RESTOFLINE;
45  ^.*"DP"        ECHO; strcat(string,"Compress Image Uses"); BEGIN RESTOFLINE;
46  ^.*"DQ"        ECHO; strcat(string,"Set Quality Factor"); BEGIN RESTOFLINE;
47  ^.*"DR"        ECHO; strcat(string,"Image Recognition"); BEGIN RESTOFLINE;
48  ^.*"DS"        ECHO; strcat(string,"Send Image"); BEGIN RESTOFLINE;
49
50  ^.*"E"[0-9A-F] ECHO; strcat(string,"Set Speed for Tilt Manual"); BEGIN RESTOFLINE;
51  ^.*"EM"        ECHO; strcat(string,"Extended Message to Device"); BEGIN RESTOFLINE;
52
53  ^.*"FF"        ECHO; strcat(string,"Focus Far"); BEGIN RESTOFLINE;
54  ^.*"FN"        ECHO; strcat(string,"Focus Near"); BEGIN RESTOFLINE;
55  ^.*"FS"        ECHO; strcat(string,"Focus Stop"); BEGIN RESTOFLINE;
56
57  ^.*"G?"        ECHO; strcat(string,"Query Groups"); BEGIN RESTOFLINE;
58  ^.*"G"         ECHO; strcat(string,"Set Group ID"); BEGIN RESTOFLINE;
59  ^.*"g"         ECHO; strcat(string,"Set Contrast (Gain) value"); BEGIN RESTOFLINE;
60
```

```
 61 ^.*"H"[0-9]    ECHO; strcat(string,"Go to a Preset"); BEGIN RESTOFLINE;
 62 ^.*"H?"        ECHO; strcat(string,"Preset Position Status"); BEGIN RESTOFLINE;
 63 ^.*"HB"        ECHO; strcat(string,"Set Polarity Black = Hot"); BEGIN RESTOFLINE;
 64 ^.*"HW"        ECHO; strcat(string,"Set Polarity White = Hot"); BEGIN RESTOFLINE;
 65
 66 ^.*"I?"        ECHO; strcat(string,"Request ID Information"); BEGIN RESTOFLINE;
 67 ^.*"IA"        ECHO; strcat(string,"Auto Contrast/Brightness Mode"); BEGIN RESTOFLINE;
 68 ^.*"IM"        ECHO; strcat(string,"Manual Contrast/Brightness"); BEGIN RESTOFLINE;
 69 ^.*"IO"        ECHO; strcat(string,"Iris/Aperture Open"); BEGIN RESTOFLINE;
 70 ^.*"IS"        ECHO; strcat(string,"Iris/Aperture Stop"); BEGIN RESTOFLINE;
 71
 72 ^.*"JL"[0-9A-F]{2}"D"[0-9A-F]{2} ECHO; strcat(string,"Joy Stick Left, Down"); BEGIN RESTOFLINE;
 73 ^.*"JL"[0-9A-F]{2}"U"[0-9A-F]{2} ECHO; strcat(string,"Joy Stick Left, Up"); BEGIN RESTOFLINE;
 74 ^.*"JR"[0-9A-F]{2}"D"[0-9A-F]{2} ECHO; strcat(string,"Joy Stick Right, Down"); BEGIN RESTOFLINE;
 75 ^.*"JR"[0-9A-F]{2}"U"[0-9A-F]{2} ECHO; strcat(string,"Joy Stick Right, Up"); BEGIN RESTOFLINE;
 76
 77 ^.*"K?"        ECHO; strcat(string,"24-bit Pan/Tilt Position Request"); BEGIN RESTOFLINE;
 78 ^.*"k"         ECHO; strcat(string,"24-bit Pan/Tilt Go-To Command"); BEGIN RESTOFLINE;
 79
 80 ^.*"L1"        ECHO; strcat(string,"Color Balance Select auto/manual mode"); BEGIN RESTOFLINE;
 81 ^.*"LN"[0-9]   ECHO; strcat(string,"Set Medium Field-of-View"); BEGIN RESTOFLINE;
 82 ^.*"L?"        ECHO; strcat(string,"Latch/Status Request"); BEGIN RESTOFLINE;
 83 ^.*"LA"        ECHO; strcat(string,"Set Auto Iris Mode"); BEGIN RESTOFLINE;
 84 ^.*"LB"        ECHO; strcat(string,"Set Manual Iris Mode"); BEGIN RESTOFLINE;
 85 ^.*"LL"        ECHO; strcat(string,"Toggle Lens Speed"); BEGIN RESTOFLINE;
 86 ^.*"LM"        ECHO; strcat(string,"Toggle Auto/Manual Iris"); BEGIN RESTOFLINE;
 87 ^.*"LN"        ECHO; strcat(string,"Set Narrow Field-of-View"); BEGIN RESTOFLINE;
 88 ^.*"LO"        ECHO; strcat(string,"Set Lens Speed Slow"); BEGIN RESTOFLINE;
 89 ^.*"LP"        ECHO; strcat(string,"Toggle Device Power On/Off"); BEGIN RESTOFLINE;
 90 ^.*"LS"        ECHO; strcat(string,"DSP Low Power Sleep"); BEGIN RESTOFLINE;
 91 ^.*"LT"        ECHO; strcat(string,"Set Lens Speed Fast"); BEGIN RESTOFLINE;
 92 ^.*"LW"        ECHO; strcat(string,"Set Wide Field-of-View"); BEGIN RESTOFLINE;
 93 ^.*"l"[1-3]    ECHO; strcat(string,"Set Latch 1"); BEGIN RESTOFLINE;
 94
 95 ^.*"P"[0-9]    ECHO; strcat(string,"Store Preset Position"); BEGIN RESTOFLINE;
 96 ^.*"P?"        ECHO; strcat(string,"Pan/Tilt Position Request"); BEGIN RESTOFLINE;
 97 ^.*"PA"        ECHO; strcat(string,"Store Auto-Scan Position A"); BEGIN RESTOFLINE;
 98 ^.*"PB"        ECHO; strcat(string,"Store Auto-Scan Position B"); BEGIN RESTOFLINE;
 99 ^.*"PF"        ECHO; strcat(string,"Turn Device Power Off"); BEGIN RESTOFLINE;
100 ^.*"PL"        ECHO; strcat(string,"Pan Left"); BEGIN RESTOFLINE;
101 ^.*"PN"        ECHO; strcat(string,"Turn Device Power On"); BEGIN RESTOFLINE;
102 ^.*"PR"        ECHO; strcat(string,"Pan Right"); BEGIN RESTOFLINE;
103 ^.*"PS"        ECHO; strcat(string,"Pan Stop"); BEGIN RESTOFLINE;
104 ^.*"p"         ECHO; strcat(string,"Pan/Tilt Go-To Command"); BEGIN RESTOFLINE;
105
106 ^.*"R"[1-9]    ECHO; strcat(string,"Reticule Select"); BEGIN RESTOFLINE;
107 ^.*"RO"        ECHO; strcat(string,"Reticule/Graticule Off"); BEGIN RESTOFLINE;
108 ^.*"RF"        ECHO; strcat(string,"Range Finder Command"); BEGIN RESTOFLINE;
109 ^.*"RS"        ECHO; strcat(string,"Reset Receiver"); BEGIN RESTOFLINE;
110 ^.*"r"[1-3]    ECHO; strcat(string,"Reset (clear) Latch 1"); BEGIN RESTOFLINE;
111
112 ^.*"S"[0-9A-F] ECHO; strcat(string,"Set Speed for Pan"); BEGIN RESTOFLINE;
113 ^.*"S?"        ECHO; strcat(string,"Request Imager Status"); BEGIN RESTOFLINE;
114 ^.*"SH"         ECHO; strcat(string,"Request State of Health"); BEGIN RESTOFLINE;
115 ^.*"SI"        ECHO; strcat(string,"Insert Shutter"); BEGIN RESTOFLINE;
116 ^.*"SR"        ECHO; strcat(string,"Remove Shutter"); BEGIN RESTOFLINE;
117
118 ^.*"TD"        ECHO; strcat(string,"Tilt Down"); BEGIN RESTOFLINE;
119 ^.*"TF"        ECHO; strcat(string,"Disable Test Mode"); BEGIN RESTOFLINE;
120 ^.*"TM"        ECHO; strcat(string,"Enable Test Mode"); BEGIN RESTOFLINE;
121 ^.*"TS"        ECHO; strcat(string,"Tilt Stop"); BEGIN RESTOFLINE;
122 ^.*"TU"        ECHO; strcat(string,"Tilt Up At constant speed"); BEGIN RESTOFLINE;
123
124 ^.*"V?"        ECHO; strcat(string,"Lens Position Request"); BEGIN RESTOFLINE;
125
126 ^.*"X"         ECHO; strcat(string,"Binary Message to Device"); BEGIN RESTOFLINE;
127
128 ^.*"v"         ECHO; strcat(string,"Lens Go-To Command"); BEGIN RESTOFLINE;
129
130 ^.*"YD"        ECHO; strcat(string,"24-bit Protected Area Definition"); BEGIN RESTOFLINE;
131
132 ^.*"Z?"        ECHO; strcat(string,"Lens Setting Request"); BEGIN RESTOFLINE;
```

```
133 ^.*"ZI"        ECHO; strcat(string,"Zoom In"); BEGIN RESTOFLINE;
134 ^.*"ZO"        ECHO; strcat(string,"Zoom Out"); BEGIN RESTOFLINE;
135 ^.*"ZS"        ECHO; strcat(string,"Zoom Stop"); BEGIN RESTOFLINE;
136 ^.*"z"         ECHO; strcat(string,"Lens Setting Go-To Command"); BEGIN RESTOFLINE;
137
138 <RESTOFLINE>.*$   {ECHO;fprintf(yyout," -- %s",string);
139                    string[0] = '\0';
140                    BEGIN 0;
141                   }
142 %%
143 // yywrap is for End Of File processing, 1 = done
144 int yywrap(void)
145 {
146         return 1;
147 }
148
149 main(int argc, char *argv[])
150 {
151         fprintf(stderr,"$Id: TassIntp.l,v 1.1 2008-06-10 09:25:21-07 Hamilton Exp Hamilton $\n");
152
153         // Do we have to deal with stdin/stdout?
154         // 3 = one for the program name and one each for input and output names
155         if (argc == 3) {
156             // Nope
157             // Get over the program, etc., name
158             argc--;
159             argv++;
160             yyin = fopen(argv[0],"r");
161             if (yyin == NULL) {
162                 fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
163                                 argv[0]);
164                 exit(EXIT_FAILURE);
165             }
166
167             // Now get over the input file name and get to the output file name
168             argc--;
169             argv++;
170             yyout = fopen(argv[0],"w");
171             if (yyout == NULL) {
172                 fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
173                                 argv[0]);
174                 exit(EXIT_FAILURE);
175             }
176         }
177         else {
178             // Woops we are using std... type io
179             yyin  = stdin;
180             yyout = stdout;
181         }
182
183         yylex();
184         exit(EXIT_SUCCESS);
185 }
186
187 // $Log: TassIntp.l,v $
188 // Revision 1.1  2008-06-10 09:25:21-07  Hamilton
189 // Initial revision
190 //
191 // Revision 1.2  2008-06-09 10:29:44-07  Hamilton
192 // Normal end of day data saving
193 //
194 // Revision 1.1  2008-06-09 07:41:10-07  Hamilton
195 // Initial revision
196 //
197 // Revision 1.10  2007-02-08 06:45:52-08  Hamilton
198 // Normal end of day data saving
199 //
200
```

# Index