

```

1  /*
2  *****
3  * COPYRIGHT AND WARRANTY INFORMATION
4  *
5  * Copyright 2001, International Telecommunications Union, Geneva
6  *
7  * DISCLAIMER OF WARRANTY
8  *
9  * These software programs are available to the user without any
10 * license fee or royalty on an "as is" basis. The ITU disclaims
11 * any and all warranties, whether express, implied, or
12 * statutory, including any implied warranties of merchantability
13 * or of fitness for a particular purpose. In no event shall the
14 * contributor or the ITU be liable for any incidental, punitive, or
15 * consequential damages of any kind whatsoever arising from the
16 * use of these programs.
17 *
18 * This disclaimer of warranty extends to the user of these programs
19 * and user's customers, employees, agents, transferees, successors,
20 * and assigns.
21 *
22 * The ITU does not represent or warrant that the programs furnished
23 * hereunder are free of infringement of any third-party patents.
24 * Commercial implementations of ITU-T Recommendations, including
25 * shareware, may be subject to royalty fees to patent holders.
26 * Information regarding the ITU-T patent policy is available from
27 * the ITU Web site at http://www.itu.int.
28 *
29 * THIS IS NOT A GRANT OF PATENT RIGHTS - SEE THE ITU-T PATENT POLICY.
30 *****
31 */
32
33 /*!
34 *****
35 * \file
36 *   parset.c
37 * \brief
38 *   Picture and Sequence Parameter set generation and handling
39 * \date 25 November 2002
40 * \author
41 *   Main contributors (see contributors.h for copyright, address and affiliation details)
42 *   - Stephan Wenger      <stewe@cs.tu-berlin.de>
43 *
44 *****
45 */
46 #include <assert.h>
47 #include <stdio.h>
48 #include <stdlib.h>
49 #include <memory.h>
50
51 #include "parsetcommon.h"

```

```

52  #include "memalloc.h"
53  /*!
54  *****
55  * \brief
56  *   Allocates memory for a pps
57  *
58  * \param none
59  *
60  * \return
61  *   pointer to a pps
62  *****
63  */
64
65  pic_parameter_set_rbsp_t *AllocPPS ()
66  {
67      pic_parameter_set_rbsp_t *p;
68
69      if ((p=calloc (sizeof (pic_parameter_set_rbsp_t), 1)) == NULL)
70          no_mem_exit ("AllocPPS: PPS");
71      if ((p->slice_group_id = calloc (SIZEslice_group_id, 1)) == NULL)
72          no_mem_exit ("AllocPPS: slice_group_id");
73      return p;
74  }
75
76
77  /*!
78  *****
79  * \brief
80  *   Allocates memory for am sps
81  *
82  * \param none
83  *
84  * \return
85  *   pointer to a sps
86  *****
87  */
88
89  seq_parameter_set_rbsp_t *AllocSPS ()
90  {
91      seq_parameter_set_rbsp_t *p;
92
93      if ((p=calloc (sizeof (seq_parameter_set_rbsp_t), 1)) == NULL)
94          no_mem_exit ("AllocSPS: SPS");
95      return p;
96  }
97
98
99  /*!
100 *****
101 * \brief
102 *   Frees a pps

```

```

103  *
104  * \param pps to be freed
105  *
106  * \return
107  *   none
108  *****
109  */
110
111  void FreePPS (pic_parameter_set_rbsp_t *pps)
112  {
113      assert (pps != NULL);
114      if (pps->slice_group_id != NULL) free (pps->slice_group_id);
115      free (pps);
116  }
117
118
119  /*!
120  *****
121  * \brief
122  *   Frees a sps
123  *
124  * \param sps to be freed
125  *
126  * \return
127  *   none
128  *****
129  */
130
131  void FreeSPS (seq_parameter_set_rbsp_t *sps)
132  {
133      assert (sps != NULL);
134      free (sps);
135  }

```