
QCAP SDK
1.1.0.140.4, 2014.12.05

1 SDK Introduction 26

1.1 Overview 26

1.2 Device Workflow in Software View 30

1.3 Programming Language Supported 31

1.4 Sample Code 31

1.5 Header Files and Link Library 31

1.6 SDK Package Organization 32

1.7 Platform and Thread Safety 32

2 System Function API 33

2.1 QCAP_GET_VERSION 33

2.2 QCAP_SET_SYSTEM_CONFIGURATION 35

2.3 QCAP_QUERY_ENCODER_TYPE_CAP 39

2.4 QCAP_QUERY_DECODER_TYPE_CAP 42

2.5 QCAP_QUERY_ENCODER_STATUS 45

3 Device Function API 48

3.1 QCAP_DEVICE_ENUMERATION 48
3.2 QCAP_GET_DEVICE_ENUMERATION_ITEM_INFO 54
3.3 QCAP_CREATE 58
3.4 QCAP_DESTROY 62
3.5 QCAP_RUN 64
3.6 QCAP_STOP 66
3.7 QCAP_GET_VIDEO_INPUT 68
3.8 QCAP_SET_VIDEO_INPUT 70
3.9 QCAP_GET_AUDIO_INPUT 73
3.10 QCAP_SET_AUDIO_INPUT 75
3.11 QCAP_GET_VIDEO_STANDARD 78
3.12 QCAP_SET_VIDEO_STANDARD 80
3.13 QCAP_GET_VIDEO_CURRENT_CONTENT_PROTECTION_STATUS 83
3.14 QCAP_GET_VIDEO_CURRENT_INPUT_FORMAT 85
3.15 QCAP_GET_AUDIO_CURRENT_INPUT_FORMAT 88
3.16 QCAP_GET_VIDEO_DEFAULT_OUTPUT_FORMAT 90
3.17 QCAP_SET_VIDEO_DEFAULT_OUTPUT_FORMAT 93

3.18 QCAP_GET_AUDIO_DEFAULT_OUTPUT_FORMAT 96
3.19 QCAP_SET_AUDIO_DEFAULT_OUTPUT_FORMAT 98
3.20 QCAP_GET_VIDEO_DEINTERLACE_TYPE 101
3.21 QCAP_SET_VIDEO_DEINTERLACE_TYPE 103
3.22 QCAP_GET_VIDEO_DEINTERLACE 105
3.23 QCAP_SET_VIDEO_DEINTERLACE 107
3.24 QCAP_GET_VIDEO_REGION_DISPLAY 110
3.25 QCAP_SET_VIDEO_REGION_DISPLAY 112
3.26 QCAP_GET_VIDEO_MIRROR 114
3.27 QCAP_SET_VIDEO_MIRROR 116
3.28 QCAP_GET_VIDEO_BRIGHTNESS 118
3.29 QCAP_SET_VIDEO_BRIGHTNESS 120
3.30 QCAP_GET_VIDEO_CONTRAST 122
3.31 QCAP_SET_VIDEO_CONTRAST 124
3.32 QCAP_GET_VIDEO_HUE 126
3.33 QCAP_SET_VIDEO_HUE 128
3.34 QCAP_GET_VIDEO_SATURATION 130

3.35 QCAP_SET_VIDEO_SATURATION 132
3.36 QCAP_GET_VIDEO_SHARPNESS 134
3.37 QCAP_SET_VIDEO_SHARPNESS 136
3.38 QCAP_GET_VIDEO_PREVIEW_PROPERTY_EX 138
3.39 QCAP_SET_VIDEO_PREVIEW_PROPERTY_EX 141
3.40 QCAP_GET_VIDEO_HARDWARE_ENCODER_PROPERTY 144
3.41 QCAP_SET_VIDEO_HARDWARE_ENCODER_PROPERTY 148
3.42 QCAP_GET_VIDEO_HARDWARE_ENCODER_PROPERTY_EX 153
3.43 QCAP_SET_VIDEO_HARDWARE_ENCODER_PROPERTY_EX 158
3.44 QCAP_GET_VIDEO_HARDWARE_ENCODER_VIDEOCOMPRESSION_PROPERTY 164
3.45 QCAP_SET_VIDEO_HARDWARE_ENCODER_VIDEOCOMPRESSION_PROPERTY 167
3.46 QCAP_GET_AUDIO_VOLUME 170
3.47 QCAP_SET_AUDIO_VOLUME 172
3.48 QCAP_GET_DEVICE_CUSTOM_PROPERTY 174
3.49 QCAP_SET_DEVICE_CUSTOM_PROPERTY 177
3.50 QCAP_GET_DEVICE_CUSTOM_PROPERTY_EX 180
3.51 QCAP_SET_DEVICE_CUSTOM_PROPERTY_EX 183

3.52 QCAP_COPY_VIDEO_PREVIEW_BUFFER 188
3.53 QCAP_COPY_AUDIO_PREVIEW_BUFFER 191
3.54 QCAP_LOCK_VIDEO_PREVIEW_BUFFER 194
3.55 QCAP_UNLOCK_VIDEO_PREVIEW_BUFFER 197
3.56 QCAP_LOCK_AUDIO_PREVIEW_BUFFER 199
3.57 QCAP_UNLOCK_VIDEO_PREVIEW_BUFFER 202

4 Snapshot Function API 204

4.1 QCAP_SNAPSHOT_BMP 204
4.2 QCAP_SNAPSHOT_JPG 207
4.3 QCAP_SNAPSHOT_BMP_EX 210
4.4 QCAP_SNAPSHOT_JPG_EX 214

5 Recording Function API 218

5.1 QCAP_GET_VIDEO_RECORD_PROPERTY 218
5.2 QCAP_SET_VIDEO_RECORD_PROPERTY 222
5.3 QCAP_GET_VIDEO_RECORD_PROPERTY_EX (CROPPING) 227
5.4 QCAP_SET_VIDEO_RECORD_PROPERTY_EX (CROPPING) 232

5.5 QCAP_GET_AUDIO_RECORD_PROPERTY 238
5.6 QCAP_SET_AUDIO_RECORD_PROPERTY 240
5.7 QCAP_GET_AUDIO_RECORD_PROPERTY_EX 243
5.8 QCAP_SET_AUDIO_RECORD_PROPERTY_EX 246
5.9 QCAP_GET_VIDEO_RECORD_SOFTWARE_ENCODER_PROPERTY 249
5.10 QCAP_SET_VIDEO_RECORD_SOFTWARE_ENCODER_PROPERTY 252
5.11 QCAP_START_CLONE_RECORD 255
5.12 QCAP_START_RECORD 260
5.13 QCAP_STOP_RECORD 265
5.14 QCAP_RESUME_RECORD 268
5.15 QCAP_PAUSE_RECORD 270

6 Synchronized Recording Function API 272

6.1 QCAP_START_SYNCHRONIZED_RECORD 272
6.2 QCAP_STOP_SYNCHRONIZED_RECORD 275
6.3 QCAP_RESUME_SYNCHRONIZED_RECORD 278
6.4 QCAP_PAUSE_SYNCHRONIZED_RECORD 281

7 OSD Function API 284

7.1 QCAP_GET_OSD_TEXT_BOUNDARY 284

7.2 QCAP_SET_OSD_TEXT 288

7.3 QCAP_SET_OSD_PICTURE 293

7.4 QCAP_SET_OSD_BUFFER 297

8 Clone Function API 303

8.1 QCAP_CREATE_CLONE 303

9 Share Recording Function API 308

9.1 Overview 308

9.2 QCAP_GET_VIDEO_SHARE_RECORD_PROPERTY 309

9.3 QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY 313

9.4 QCAP_GET_VIDEO_SHARE_RECORD_PROPERTY_EX 318

9.5 QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY_EX 323

9.6 QCAP_GET_AUDIO_SHARE_RECORD_PROPERTY 330

9.7 QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY 333

9.8 QCAP_GET_AUDIO_SHARE_RECORD_PROPERTY_EX 337
9.9 QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY_EX 340
9.10 QCAP_GET_VIDEO_SHARE_RECORD_SOFTWARE_ENCODER_PROPERTY 343
9.11 QCAP_SET_VIDEO_SHARE_RECORD_SOFTWARE_ENCODER_PROPERTY 346
9.12 QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER 349
9.13 QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER_EX 353
9.14 QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER 357
9.15 QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER_EX 360
9.16 QCAP_SET_VIDEO_SHARE_RECORD_COMPRESSION_BUFFER 363
9.17 QCAP_SET_AUDIO_SHARE_RECORD_COMPRESSION_BUFFER 366
9.18 QCAP_START_SHARE_RECORD 369
9.19 QCAP_STOP_SHARE_RECORD 375
9.20 QCAP_RESUME_SHARE_RECORD 378
9.21 QCAP_PAUSE_SHARE_RECORD 380
9.22 QCAP_SNAPSHOT_SHARE_RECORD_BMP 382
9.23 QCAP_SNAPSHOT_SHARE_RECORD_JPG 385
9.24 QCAP_SNAPSHOT_SHARE_RECORD_BMP_EX 388

9.25 QCAP_SNAPSHOT_SHARE_RECORD_JPG_EX 392
9.26 QCAP_GET_OSD_SHARE_RECORD_TEXT_BOUNDARY 396
9.27 QCAP_SET_OSD_SHARE_RECORD_TEXT 400
9.28 QCAP_SET_OSD_SHARE_RECORD_PICTURE 405
9.29 QCAP_SET_OSD_SHARE_RECORD_BUFFER 409
9.30 QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER 415
9.31 QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER_EX 419
9.32 QCAP_SET_VIDEO_3D_SHARE_RECORD_L_UNCOMPRESSION_BUFFER 423
9.33 QCAP_SET_VIDEO_3D_SHARE_RECORD_L_UNCOMPRESSION_BUFFER_EX 427
9.34 QCAP_SET_VIDEO_3D_SHARE_RECORD_R_UNCOMPRESSION_BUFFER 431
9.35 QCAP_SET_VIDEO_3D_SHARE_RECORD_R_UNCOMPRESSION_BUFFER_EX 435
9.36 QCAP_SET_VIDEO_3D_SHARE_RECORD_UNCOMPRESSION_BUFFER 439
9.37 QCAP_GET_AUDIO_MX_SHARE_RECORD_PROPERTY_EX 442
9.38 QCAP_SET_AUDIO_MX_SHARE_RECORD_PROPERTY_EX 445
9.39 QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXONG_UNCOMPRESSION_BUFFER 449
9.40 QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER 452
9.41 QCAP_SET_AUDIO_MX_SHARE_RECORD_COMPRESSION_BUFFER 455

10File Playback/Editing Function API 458

10.1 QCAP_OPEN_FILE 458

10.2 QCAP_DESTROY_FILE 463

10.3 QCAP_PLAY_FILE 465

10.4 QCAP_PAUSE_FILE 467

10.5 QCAP_STOP_FILE 469

10.6 QCAP_GET_FILE_POSITION 471

10.7 QCAP_SET_FILE_POSITION 473

10.8 QCAP_GET_FILE_PLAYBACK_SPEED 475

10.9 QCAP_SET_FILE_PLAYBACK_SPEED 477

10.10 QCAP_GET_VIDEO_FILE_REGION_DISPLAY 479

10.11 QCAP_SET_VIDEO_FILE_REGION_DISPLAY 482

10.12 QCAP_GET_VIDEO_FILE_MIRROR 484

10.13 QCAP_SET_VIDEO_FILE_MIRROR 486

10.14 QCAP_GET_AUDIO _FILE _VOLUME 488

10.15 QCAP_SET_AUDIO _FILE _VOLUME 490

10.16 QCAP_SNAPSHOT_FILE_BMP 492
10.17 QCAP_SNAPSHOT_FILE_JPG 495
10.18 QCAP_SNAPSHOT_FILE_BMP_EX 498
10.19 QCAP_SNAPSHOT_FILE_JPG_EX 502
10.20 QCAP_GET_OSD_FILE_TEXT_BOUNDARY 506
10.21 QCAP_SET_OSD_FILE_TEXT 510
10.22 QCAP_SET_OSD_FILE_PICTURE 514
10.23 QCAP_SET_OSD_FILE_BUFFER 518
10.24 QCAP_SCF_FILE_ENUMERATION 524
10.25 QCAP_OPEN_SCF_FILE 529
10.26 QCAP_SET_SCF_FILE_TIMER 532
10.27 QCAP_GET_GLOBAL_SCF_FILE_POSITION 534
10.28 QCAP_SET_GLOBAL_SCF_FILE_POSITION 536
10.29 QCAP_OPEN_3D_FILE 538
10.30 QCAP_GET_VIDEO_3D_FILE_DISPLAY_MODE 544
10.31 QCAP_SET_VIDEO_3D_FILE_DISPLAY_MODE 547
10.32 QCAP_GET_VIDEO _FILE _STREAM_BUFFER 549

10.33 QCAP_GET_ AUDIO_FILE_ STREAM_BUFFER 552
10.34 QCAP_EXPORT_FILE 554
10.35 QCAP_MERGE_FILE 557
10.36 QCAP_MERGE_FILES_EX 560
10.37 QCAP_BUILD_VIDEO_ON_DEMAND_FILE 563
10.38 QCAP_DIAGNOSE_FILE 566
10.39 QCAP_REPAIR_FILE 568

11 Broadcasting Server& Client Function API 571

11.1 Overview 571
11.2 QCAP_CREATE_BROADCAST_RTSP_SERVER 574
11.3 QCAP_CREATE_BROADCAST_RTMP_SERVER 578
11.4 QCAP_CREATE_BROADCAST_RTMP_SERVER_EX 582
11.5 QCAP_CREATE_BROADCAST_RTMP_WEB_PORTAL_SERVER 586
11.6 QCAP_CREATE_BROADCAST_RTMP_WEB_PORTAL_SERVER_EX 589
11.7 QCAP_CREATE_BROADCAST_MMS_WEB_PORTAL_SERVER 592
11.8 QCAP_CREATE_BROADCAST_HLS_SERVER 595

11.9 QCAP_DESTROY_BROADCAST_SERVER 599
11.10 QCAP_START_BROADCAST_SERVER 601
11.11 QCAP_STOP_BROADCAST_SERVER 604
11.12 QCAP_GET_VIDEO_BROADCAST_SERVER_PROPERTY 607
11.13 QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY 611
11.14 QCAP_GET_VIDEO_BROADCAST_SERVER_PROPERTY_EX 616
11.15 QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY_EX 621
11.16 QCAP_GET_AUDIO_BROADCAST_SERVER_PROPERTY 628
11.17 QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY 632
11.18 QCAP_GET_AUDIO_BROADCAST_SERVER_PROPERTY_EX 636
11.19 QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY_EX 640
11.20 QCAP_GET_VIDEO_BROADCAST_SERVER_SOFTWARE_ENCODER_PROPERTY 644
11.21 QCAP_SET_VIDEO_BROADCAST_SERVER_SOFTWARE_ENCODER_PROPERTY 647
11.22 QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER 650
11.23 QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER_EX 655
11.24 QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER 660
11.25 QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER_EX 663

11.26 QCAP_SET_VIDEO_BROADCAST_SERVER_COMPRESSION_BUFFER 667
11.27 QCAP_SET_AUDIO_BROADCAST_SERVER_COMPRESSION_BUFFER 670
11.28 QCAP_GET_SESSION_BROADCAST_SERVER_PROPERTY 673
11.29 QCAP_SET_SESSION_BROADCAST_SERVER_PROPERTY 676
11.30 QCAP_SNAPSHOT_BROADCAST_SERVER_BMP 679
11.31 QCAP_SNAPSHOT_BROADCAST_SERVER_JPG 682
11.32 QCAP_SNAPSHOT_BROADCAST_SERVER_BMP_EX 685
11.33 QCAP_SNAPSHOT_BROADCAST_SERVER_JPG_EX 689
11.34 QCAP_GET_OSD_BROADCAST_SERVER_TEXT_BOUNDARY 693
11.35 QCAP_SET_OSD_BROADCAST_SERVER_TEXT 697
11.36 QCAP_SET_OSD_BROADCAST_SERVER_PICTURE 702
11.37 QCAP_SET_OSD_BROADCAST_SERVER_BUFFER 706
11.38 QCAP_SET_VIDEO_3D_BRODCAST_SERVER_STEREO_UNCOMPRESSION_BUFFER 712
11.39QCAP_SET_VIDEO_3D_BROADCAST_SERVER_STEREO_UNCOMPRESSION_BUFFER_EX 716
11.40 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_L_UNCOMPRESSION_BUFFER 720
11.41 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_L_UNCOMPRESSION_BUFFER_EX 724
11.42 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_R_UNCOMPRESSION_BUFFER 728

11.43 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_R_UNCOMPRESSION_BUFFER_EX 732
11.44 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_UNCOMPRESSION_BUFFER 736
11.45 QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER 739
11.46 QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER 743
11.47 QCAP_CREATE_BROADCAST_CLIENT 746
11.48 QCAP_DESTROY_BROADCAST_CLIENT 749
11.49 QCAP_START_BROADCAST_CLIENT 751
11.50 QCAP_STOP_BROADCAST_CLIENT 754
11.51 QCAP_GET_VIDEO_BROADCAST_CLIENT_REGION_DISPLAY 756
11.52 QCAP_SET_VIDEO_BROADCAST_CLIENT_REGION_DISPLAY 758
11.53 QCAP_GET_VIDEO_BROADCAST_CLIENT_MIRROR 760
11.54 QCAP_SET_VIDEO_BROADCAST_CLIENT_MIRROR 763
11.55 QCAP_GET_ AUDIO_BROADCAST_CLIENT_ VOLUME 765
11.56 QCAP_SET_ AUDIO_BROADCAST_CLIENT_ VOLUME 767
11.57 QCAP_START_BROADCAST_CLIENT_RECORD 769
11.58 QCAP_STOP_BROADCAST_CLIENT_RECORD 773
11.59 QCAP_RESUME_BROADCAST_CLIENT_RECORD 775

11.60 QCAP_PAUSE_BROADCAST_CLIENT_RECORD 776
11.61 QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP 778
11.62 QCAP_SNAPSHOT_BROADCAST_CLIENT_JPG 781
11.63 QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP_EX 784
11.64 QCAP_SNAPSHOT_BROADCAST_CLIENT_JPG_EX 788
11.65 QCAP_GET_OSD_BROADCAST_CLIENT_TEXT_BOUNDARY 792
11.66 QCAP_SET_OSD_BROADCAST_CLIENT_TEXT 796
11.67 QCAP_SET_OSD_BROADCAST_CLIENT_PICTURE 801
11.68 QCAP_SET_OSD_BROADCAST_CLIENT_BUFFER 805
11.69 QCAP_GET_BROADCAST_CLIENT_CUSTOM_PROPERTY 811
11.70 QCAP_SET_BROADCAST_CLIENT_CUSTOM_PROPERTY 813
11.71 QCAP_CREATE_3D_BROADCAST_CLIENT 815
11.72 QCAP_GET_VIDEO_3D_BROADCAST_CLIENT_DISPLAY_MODE 819
11.73 QCAP_SET_VIDEO_3D_BROADCAST_CLIENT_DISPLAY_MODE 822
11.74 QCAP_CREATE_COMMUNICATION_ONVIF_SERVER 824
11.75 QCAP_DESTROY_COMMUNICATION_SERVER 827
11.76 QCAP_START_COMMUNICATION_SERVER 829

11.77 QCAP_STOP_COMMUNICATION_SERVER 831
11.78 QCAP_GET_COMMUNICATION_SERVER_VIDEO_SOURCE 833
11.79 QCAP_SET_COMMUNICATION_SERVER_VIDEO_SOURCE 836
11.80 QCAP_GET_COMMUNICATION_SERVER_AUDIO_SOURCE 839
11.81 QCAP_SET_COMMUNICATION_SERVER_AUDIO_SOURCE 842
11.82 QCAP_GET_COMMUNICATION_SERVER_VIDEO_ENCODER 845
11.83 QCAP_SET_COMMUNICATION_SERVER_VIDEO_ENCODER 849
11.84 QCAP_GET_COMMUNICATION_SERVER_AUDIO_ENCODER 854
11.85 QCAP_SET_COMMUNICATION_SERVER_AUDIO_ENCODER 857
11.86 QCAP_GET_COMMUNICATION_SERVER_PROFILE 860
11.87 QCAP_SET_COMMUNICATION_SERVER_PROFILE 863
11.88 An Broadcasting Software Programming Guides 866

12 Animation Function API 872

12.1 QCAP_CREATE_ANIMATION_CLIP 872
12.2 QCAP_DESTROY_ANIMATION_CLIP 875
12.3 QCAP_START_ANIMATION_CLIP 878

12.4 QCAP_STOP_ANIMATION_CLIP 880
12.5 QCAP_STEP_ANIMATION_CLIP 882
12.6 QCAP_SEEK_ANIMATION_CLIP 885
12.7 QCAP_GET_ANIMATION_CLIP_SPRITE_TRANSFORM_PROPERTY 888
12.8 QCAP_SET_ANIMATION_CLIP_SPRITE_TRANSFORM_PROPERTY 891
12.9 QCAP_GET_ANIMATION_CLIP_SPRITE_ALPHA_PROPERTY 893
12.10 QCAP_SET_ANIMATION_CLIP_SPRITE_ALPHA_PROPERTY 895
12.11 QCAP_SET_ANIMATION_CLIP_SPRITE_PICTURE 897
12.12 QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER 899
12.13 QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER_EX 903
12.14 QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE 907
12.15 QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE_EX 910
12.16 An Animation Clip Software Programming Guides 913

13 Virtual Camera Function API 915

13.1 QCAP_CREATE_VIRTUAL_CAMERA 915
13.2 QCAP_DESTROY_VIRTUAL_CAMERA 918

13.3 QCAP_GET_VIDEO_VIRTUAL_CAMERA_PROPERTY 921
13.4 QCAP_SET_VIDEO_VIRTUAL_CAMERA_PROPERTY 924
13.5 QCAP_GET_AUDIO_VIRTUAL_CAMERA_PROPERTY 928
13.6 QCAP_SET_AUDIO_VIRTUAL_CAMERA_PROPERTY 931
13.7 QCAP_SET_VIDEO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER 934
13.8 QCAP_SET_VIDEO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER_EX 938
13.9 QCAP_SET_AUDIO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER 942
13.10 QCAP_SET_AUDIO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER_EX 945
13.11 QCAP_START_VIRTUAL_CAMERA 948
13.12 QCAP_STOP_VIRTUAL_CAMERA 951
13.13 QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP 953
13.14 QCAP_SNAPSHOT_VIRTUAL_CAMERA_JPG 956
13.15 QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP_EX 959
13.16 QCAP_SNAPSHOT_VIRTUAL_CAMERA_JPG_EX 963
13.17 QCAP_GET_OSD_VIRTUAL_CAMERA_TEXT_BOUNDARY 967
13.18 QCAP_SET_OSD_VIRTUAL_CAMERA_TEXT 971
13.19 QCAP_SET_OSD_VIRTUAL_CAMERA_PICTURE 975

13.20 QCAP_SET_OSD_VIRTUAL_CAMERA_BUFFER 978

14 Callback Function API 989

14.1 QCAP_REGISTER_NO_SIGNAL_DETECTED_CALLBACK 989

14.2 QCAP_REGISTER_SIGNAL_REMOVED_CALLBACK 991

14.3 QCAP_REGISTER_FORMAT_CHANGED_CALLBACK 993

14.4 QCAP_REGISTER_SNAPSHOT_DONE_CALLBACK 995

14.5 QCAP_REGISTER_RECORD_DONE_CALLBACK 997

14.6 QCAP_REGISTER_SHARE_RECORD_DONE_CALLBACK 999

14.7 QCAP_REGISTER_SHARE_RECORD_SNAPSHOT_DONE_CALLBACK 1001

14.8 QCAP_REGISTER_BROADCAST_SERVER_GET_CUSTOM_PROPERTY_CALLBACK 1003

14.9 QCAP_REGISTER_BROADCAST_SERVER_SET_CUSTOM_PROPERTY_CALLBACK 1005

14.10 QCAP_REGISTER_BROADCAST_SERVER_SNAPSHOT_DONE_CALLBACK 1007

14.11 QCAP_REGISTER_BROADCAST_CLIENT_RECORD_DONE_CALLBACK 1009

14.12 QCAP_REGISTER_BROADCAST_CLIENT_CONNECTED_CALLBACK 1011

14.13 QCAP_REGISTER_BROADCAST_CLIENT_SNAPSHOT_DONE_CALLBACK 1013

14.14 QCAP_REGISTER_VIRTUAL_CAMERA_SNAPSHOT_DONE_CALLBACK 1015

14.15 QCAP_REGISTER_FILE_SNAPSHOT_DONE_CALLBACK 1017
14.16 QCAP_REGISTER_VIDEO_PREVIEW_CALLBACK 1019
14.17 QCAP_REGISTER_AUDIO_PREVIEW_CALLBACK 1024
14.18 QCAP_REGISTER_VIDEO_HARDWARE_ENCODER_CALLBACK 1028
14.19 QCAP_REGISTER_SNAPSHOT_STREAM_CALLBACK 1030
14.20 QCAP_REGISTER_SHARE_RECORD_SNAPSHOT_STREAM_CALLBACK 1032
14.21 QCAP_REGISTER_BROADCAST_SERVER_SNAPSHOT_STREAM_CALLBACK 1034
14.22 QCAP_REGISTER_BROADCAST_CLIENT_SNAPSHOT_STREAM_CALLBACK 1036
14.23 QCAP_REGISTER_VIRTUAL_CAMERA_SNAPSHOT_STREAM_CALLBACK 1038
14.24 QCAP_REGISTER_FILE_SNAPSHOT_STREAM_CALLBACK 1040
14.25 QCAP_REGISTER_VIDEO_RECORD_CALLBACK 1042
14.26 QCAP_REGISTER_AUDIO_RECORD_CALLBACK 1044
14.27 QCAP_REGISTER_MEDIA_RECORD_CALLBACK 1046
14.28 QCAP_REGISTER_VIDEO_SHARE_RECORD_CALLBACK 1048
14.29 QCAP_REGISTER_AUDIO_SHARE_RECORD_CALLBACK 1050
14.30 QCAP_REGISTER_AUDIO_MX_SHARE_RECORD_CALLBACK 1052
14.31 QCAP_REGISTER_MEDIA_SHARE_RECORD_CALLBACK 1054

14.32 QCAP_REGISTER_VIDEO_DECODER_SHARE_RECORD_CALLBACK....1056
14.33 QCAP_REGISTER_AUDIO_DECODER_SHARE_RECORD_CALLBACK....1058
14.34 QCAP_REGISTER_AUDIO_DECODER_MX_SHARE_RECORD_CALLBACK....1060
14.35 QCAP_REGISTER_VIDEO_BROADCAST_SERVER_CALLBACK....1062
14.36 QCAP_REGISTER_AUDIO_BROADCAST_SERVER_CALLBACK....1064
14.37 QCAP_REGISTER_VIDEO_DECODER_BROADCAST_SERVER_CALLBACK....1066
14.38 QCAP_REGISTER_AUDIO_DECODER_BROADCAST_SERVER_CALLBACK....1068
14.39 QCAP_REGISTER_VIDEO_BROADCAST_CLIENT_CALLBACK....1070
14.40 QCAP_REGISTER_AUDIO_BROADCAST_CLIENT_CALLBACK....1073
14.41 QCAP_REGISTER_VIDEO_DECODER_BROADCAST_CLIENT_CALLBACK....1076
14.42 QCAP_REGISTER_VIDEO_DECODER_3D_BROADCAST_CLIENT_CALLBACK....1078
14.43 QCAP_REGISTER_AUDIO_DECODER_BROADCAST_CLIENT_CALLBACK....1080
14.44 QCAP_REGISTER_VIDEO_DECODER_FILE_CALLBACK....1082
14.45 QCAP_REGISTER_VIDEO_DECODER_3D_FILE_CALLBACK....1084
14.46 QCAP_REGISTER_AUDIO_DECODER_FILE_CALLBACK....1086

15CD&DVD Burning Function API 1088

15.1 QCAP_CREATE_BURNING_DRIVE 1088
15.2 QCAP_DESTROY_BURNING_DRIVE 1091
15.3 QCAP_START_BURNING_DRIVE 1093
15.4 QCAP_STOP_BURNING_DRIVE 1095
15.5 QCAP_UNLOAD_DISC 1097
15.6 QCAP_LOAD_DISC 1099
15.7 QCAP_GET_DISC_INFO 1101
15.8 QCAP_ERASE_DISC 1105
15.9 CD&DVD Burning Software Programming Guides 1107

16 CameraFunction API 1108

16.1 QCAP_CAMERA_ENUMERATION 1108
16.2 QCAP_SET_DEFAULT_CAMERA 1111

17 Sound Card Function API 1114

17.1 QCAP_SOUNDCARD_ENUMERATION 1114
17.2 QCAP_SET_DEFAULT_SOUNDCARD 1117

18 Helper Function API 1120

18.1 QCAP_GET_H264_BUFFER_LAYER_ID 1120

18.2 QCAP_COLORSPACE_YUY2_TO_ABGR32 1122

18.3 QCAP_COLORSPACE_YV12_TO_ABGR32 1126

18.4 QCAP_COLORSPACE_YUY2_TO_BGR24 1130

18.5 QCAP_COLORSPACE_YV12_TO_BGR24 1134

18.6 QCAP_COLORSPACE_YUY2_TO_YV12 1138

18.7 QCAP_CONVERT_3D_STEREO_BUFFER 1142

18.8 QCAP_RESIZE_VIDEO_BUFFER 1146

18.9 QCAP_GET_ROTATE_VIDEO_BUFFER_BOUNDARY 1150

18.10 QCAP_ROTATE_VIDEO_BUFFER 1153

18.11 QCAP_LOAD_PICTURE_BUFFER 1157

18.12 QCAP_GET_AUDIO_BUFFER_VOLUME_DB 1160

18.13 QCAP_RESAMPLE_AUDIO_BUFFER 1163

18.14 QCAP_HELPER_OBJPTR 1167

1 SDK Introduction

1.1 Overview

The purpose of QCAP is different from original AMESDK library. It offers more high-level APIs for developer to easy access our capture card and to develop rich application.It is developed based on AMESDK's kerneland supports following features:

- Video and Audio Capture and Preview
- Video and Audio Clone Preview on Several Attached Windows
- Various Deinterlace Methods
- Alpha Blending OSD function for Text, Picture, and Buffer Objects
- Snapshot to BMP or JPG Format(Continuous, Scaling, and Cropping)
- Multiple Streams Channel Recording to AVI, MP4, ASF, FLV, TS, WMV, WAV and SCF (Scaling & Cropping)
- Multiple Streams Share Recording to AVI, MP4, ASF, FLV, TS, WMV, WAV and SCF (Scaling & Cropping)
- Multiple Channels Synchronized Recording
- File Playback andEditing Library (Export, Merge, and Insert)

About AAC License

This software development kit contains software programming code and libraries related to AAC function that is subject to your development use only.All the intellectual properties are held by individual license groups and/or the property holders.License and royalty fee should be charged by the license groups and/or property holders and paid by you if any of them is implemented into your commercial product.We will not be responsible for any illegal usage and/or any infringement of the rights of the individual owners of these intellectual properties.

Details of AAC could be found in the website:

http://www.vialicensing.com/licensing/aac-overview.aspx

- Multiple Streams Broadcasting RTSP Server for UDP/TCP/HTTP& RTSP Client
- Multiple Streams Broadcasting RTMP Server and RTMP Web Portal
- Multiple Streams Broadcasting HLS Server
- Callback Functions Registration
- Transition EffectAnimationEngine (Action Script File, Spline Transform, Alpha Blending, and Up/Down Scaling, Blinds)
- CD&DVD Burning
- Virtual Camera
- Add-on Sound Card and USB Camera Capture Supported
- Multiple Cards Supported
- 3D Function Support
- Video Mirror Support
- Audio Mixer Support

- MP4 Repair Support
- Color Key Support
- Audio resample Support
- RTSP Over Multicast Support
- Recording mp4 to One Vod (video on demand) File Support
- Region Display Function With All Devices Support
- Auto Network Client Reconnection Support
- Dynamic Setting Encoder Property Support
- Text Scrolling Support
- RTMP Account & Password Support *New*
- Change Default Output Colorspace to YV12 Support *New*
- Broadcast RTSP Client and File Playback with GPU decoder Support *New*

- Multi-3D Display Modes Support *New*
- Delay Live Streaming Support *New*

1.2 Device Workflow in Software View

Programming Step	Related API Functions
CoInitialize	
Create Device	QCAP_CREATE
Set Device Parameters	QCAP_SET_VIDEO_INPUT QCAP_SET_VIDEO_DEINTERLACE QCAP_SET_VIDEO_BRIGHTNESS, … QCAP_SET_AUDIO_VOLUME
Start Capturing	QCAP_RUN
Snapshot	QCAP_SNAPSHOT_BMP QCAP_SNAPSHOT_JPG
Start Recording	QCAP_SET_VIDEO_RECORD_PROPERTY QCAP_SET_AUDIO_RECORD_PROPERTY QCAP_START_RECORD
Stop Recording	QCAP_STOP_RECORD
Stop Capturing	QCAP_STOP
Delete Device	QCAP_DESTROY
CouninitializeCoUninitialize	

1.3 Programming Language Supported

The API supports VC, VB, C#, QT, Delphi, Powerbuilder and LabView development environments.

In this document, only how to use APIs in VC environment is illustrated.

Please refer to the sample codes to get examples of other languages.

For LabView developer, please add _QCAP_LABVIEW_IMPORT macro into the predefinition page during the library importing.

1.4 Sample Code

VC, VB, C#, QT, Powerbuilder, OPENCV and LabView sample code and sampleprogram are available in SDK package.

1.5 Header Files and Link Library

VC, LabView	**X86**	QCAP.H, QCAP.LIB, QCAP.DLL, AMESDK.DLL
	X64	QCAP.H, QCAP.X64.LIB, QCAP.X64.DLL, AMESDK.X64.DLL
C#, VB, Delphi	**X86**	QCAP.H, QCAP.LIB, QCAP.DLL, QCAP.NET.DLL, AMESDK.DLL
	X64	QCAP.H, QCAP.X64.LIB, QCAP.X64.DLL, QCAP.NET.X64.DLL, AMESDK.X64.DLL

Developer should find out 4 lib folders from SDK packet. If developer wants to access Intel hardware H.264 encoder (Intel Media SDK), you need use VC.GPU or NET.GPU library. Note!! Intel Media SDK is only support on Windows 7 system.

1.6 SDK Package Organization

DOC	SDK documents for capture devices
INC	Included files
LIB	Libraries
SAMPLES	Sample code and sample program

1.7 Platform and Thread Safety

X86 and X64 of Windows XP, Windows Vista, Windows 7 and Windows 8 are all supported by this SDK.

API calling can be used in the multiple process and the multiple threads.

2 System Function API

2.1 QCAP_GET_VERSION

```
QRESULT QCAP_GET_VERSION(  ULONG * pMajorVersion,
                           ULONG * pMinorVersion )
```

Parameter

Name	Description
pMajorVersion	**OUT**, Pointer of storage of major version
pMinorVersion	**OUT**, Pointer of storage of minor version

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function togetmajor and minor versions.

Example

```
QCAP_GET_VERSION( &nMajorVersion, &nMinorVersion );
```

2.2 QCAP_SET_SYSTEM_CONFIGURATION

```
QRESULT QCAP_SET_SYSTEM_CONFIGURATION(  BOOL      bEnableMultipleUsersAccess = TRUE,
                                        BOOL      bEnableVideoPreviewDevice= TRUE,
                                        BOOL      bEnableAudioPreviewDevice= TRUE,
                                        BOOL      bEnableVideoHardwareMainEncoderDevice= TRUE,
                                        BOOL      bEnableVideoHardwareSubEncoderDevice = TRUE,
                                        ULONG     nAutoInputDetectionTimeout = 3000,
                                        BOOL      bEnableSCF = FALSE,
                                        CHAR *    pszDB3 = NULL,
                                        BOOL      bEnableAsyncBackgroundSnapshot = FALSE,
                                        BOOL      bEnableEnhancedVideoRenderer = TRUE,
                                        BOOL      bEnableSystemTimeCallback = FALSE,
                                        BOOL      bEnableFileRepairFunction = TRUE,
                                        ULONG     bEnableNewRTSPLibrary = TRUE,
                                        CHAR *    pszWebServerRootFolderPath = NULL )
```

Parameter

Name	Description
bEnableMultipleUsersAccess	**IN**, If true, qcap will alllow multiple users to access the device
bEnableVideoPreviewDevice	**IN**, If true, the video capture device will be enabled
bEnableAudioPreviewDevice	**IN**, If true, the audio capture device will be enabled
bEnableHardwareMainEncoderDevice	**IN**, If true, the main hardware encoder will be enabled
bEnableHardwareSubEncoderDevice	**IN**, If true, the sub hardware encoder will be enabled
nAutoInputDetectionTimeout	**IN**, The parameter allows you to specify the timeout duration of the auto input detection, default is 3sec.
bEnableSCF	**IN**, If true, the SCF file recording function will be turned on.
pszDB3	**IN**, Specify the SQLITE database name (path + filename), the default path is local folder
bEnableAsyncBackgroundSnapshot	**IN**, If true, snapshot function will be moved into another thread
bEnableEnhancedVideoRenderer	**IN**, If true, QCAP will enable the DirectShow's Enhanced Video Renderer Filter
bEnableSystemTimeCallback	**IN**, If true, the preview's and encoder's callbacks will return one local system time at dSampleTime
bEnableFileRepairFunction	**IN**, If true, the recording file will own self-repaired ability
bEnableNewRTSPLibrary	**IN**, If true, the new rtsp library to replace old one default is True
pszWebServerRootFolderPath	**IN**, Specify the parameter allows you to specify the web server's root folder path

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function toset some global parameters of the SDK, the function could be modified in future.

Note

If bEnableSCF is enstable, we will recording SCF format for user, this format is for secure user.

Example

```
QCAP_SET_SYSTEM_CONFIGURATION( TRUE, TRUE, TRUE, TRUE, TRUE, 3000, FALSE, NULL, FALSE,

                               TRUE, FLASE, TRUE, TRUE, “C:\AppServ\www”  );
```

2.3 QCAP_QUERY_ENCODER_TYPE_CAP

```
QRESULT QCAP_QUERY_ENCODER_TYPE_CAP( ULONG  nEncoderType )
```

Parameter

Name	Description
nEncoderType	**IN**, Specify query encodertype. Available values are QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK QCAP_ENCODER_TYPE_NVIDIA_CUDA

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to query query encoder type, if exist in the system before you use GPU Encoder, we suggest you to use the function to check if GPU Encoder is enabled on the system.

Example

```
QCAP_QUERY_ENCODER_TYPE_CAP( nEncoderType );
```

2.4 QCAP_QUERY_DECODER_TYPE_CAP

QRESULT QCAP_QUERY_DECODER_TYPE_CAP(ULONG nDecoderType)

Parameter

Name	Description
nDecoderType	**IN**, Specify query decodertype. Available values are QCAP_DECODER_TYPE_SOFTWARE QCAP_DECODER_TYPE_INTEL_MEDIA_SDK

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to query query decoder type, if exist in the system before you use GPU Encoder, we suggest you to use the function to check if GPU Encoder is enabled on the system.

Example

```
QCAP_QUERY_DECODER_TYPE_CAP( nDecoderType );
```

2.5 QCAP_QUERY_ENCODER_STATUS

```
QRESULT QCAP_QUERY_ENCODER_STATUS( ULONG nEncoderType, ULONG * pExistInstances )
```

Parameter

Name	Description
nDecoderType	**IN**, Specify query decodertype. Available values are QCAP_DECODER_TYPE_SOFTWARE QCAP_DECODER_TYPE_INTEL_MEDIA_SDK
pExistInstances	**OUT**, Pointer to the address of exist instances

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Query encoder status helper API. User can use this function to check query decoder status.

Example

```
QCAP_QUERY_ENCODER_STATUS( nEncoderType, &p_ExistInstances );
```

3 Device Function API

3.1 QCAP_DEVICE_ENUMERATION

```
QRESULT QCAP_DEVICE_ENUMERATION( ULONGLONG * *          ppVideoDevicelist,
                                 ULONG *                pVideoDeviceSize,
                                 ULONGLONG * *          ppVideoEncoderDevicelist,
                                 ULONG *                pVideoEncoderDeviceSize,
                                 ULONGLONG * *          ppAudioDevicelist,
                                 ULONG *                pAudioDeviceSize,
                                 ULONGLONG * *          ppAudioEncoderDeviceList,
                                 ULONG *                pAudioEncoderDeviceSize,
                                 ULONG                  nDeviceEnumType = QCAP_ENUM_TYPE_SERIAL_NUMBER )
```

Parameter

Name	Description
ppVideoDevicelist	**OUT**, Pointer to the address of all video capture devices
pVideoDeviceSize	**OUT**, Pointer to a variable that stores the quantity of video capture device
ppVideoEncoderDevicelist	**OUT**, Pointer to the address of all video encoder device
pVideoEncoderDeviceSize	**OUT**, Pointer to a variable that stores the quantity of video encoder device
ppAudioDeviceList	**OUT**, Pointer to the address of all audio capture devices
pAudioDeviceSize	**OUT**, Pointer to a variable that stores the quantity of audio capture device
ppAudioEncoderDeviceList	**OUT**, Pointer to the address of all audio encoder devices
pAudioEncoderDeviceSize	**OUT**, Pointer to a variable that stores the quantity of audio encoder device
nDeviceEnumType	**IN**, Specify input device enumerate type. Available values are QCAP_ENUM_TYPE_DEVICE_NAME QCAP_ENUM_TYPE_SERIAL_NUMBER

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user select QCAP_ENUM_TYPE_SERIAL_NUMBER, this function is used to enumerate all capture device on the platform.

Here, every item of the returned list uses 8 bytes to describe the devices information.

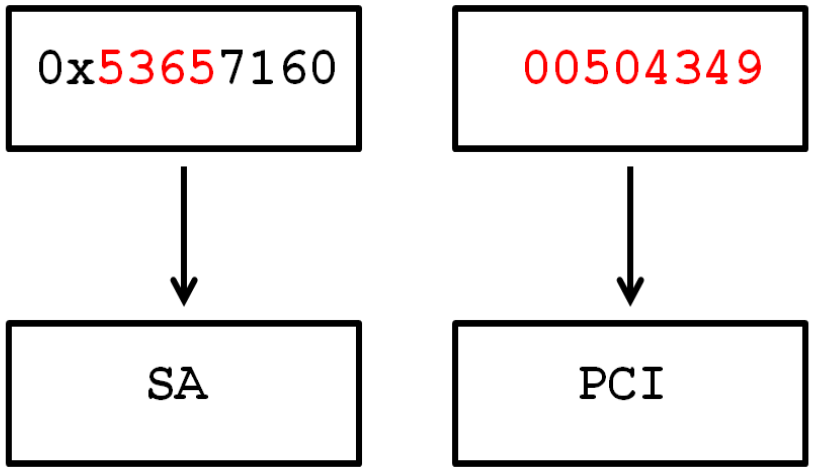
The lower 4 bytes are the devices serial number. The upper 4 bytes are device's number.

The upper 4 bytes are drivers product id. For instance, SC300N8 could return the 0x0000680212ABF308, from this function.

If user select QCAP_ENUM_TYPE_DEVICE_NAME, this function is used to list the names of all capture devices currently known to the user on the platform.

Although every item ofthe returned list of the latter is like the above mentioned brief explanation, nevertheless, the returned list it returns is to uses 8 bytes to describethe name of capture device currently known on the platform to the user.

For example, SC510N1 will return the value, 0x5365716000504349, from this function. And thenwe have to uses the ASCII character that corresponds to that ASCII value, so that the return value actually stands for or mean the device's name "SA7160 PCI".



Example

```
ULONGLONG * p_video_device_list = NULL;

ULONGLONG * p_video_encoder_device_list = NULL;

ULONGLONG * p_audio_device_list = NULL;

ULONGLONG * p_audio_encoder_device_list = NULL;

ULONG       n_video_device_list_size = 0;

ULONG       n_video_encoder_device_list_size = 0;

ULONG       n_audio_device_list_size = 0;

ULONG       n_audio_encoder_device_list_size = 0;
```

```
QCAP_DEVICE_ENUMERATION( &p_video_device_list, &n_video_device_list_size,
                         &p_video_encoder_device_list, &n_video_encoder_device_list_size,
                         &p_audio_device_list, &n_audio_device_list_size,
                         &p_audio_encoder_device_list, &n_audio_encoder_device_list_size,
                         QCAP_ENUM_TYPE_SERIAL_NUMBER );

for( ULONG i = 0 ; i<n_video_device_list_size ; i++ ) {

sprintf( "Live#%02d = 0x%llX", i, p_video_device_list[ i ] );
}
for( ULONG i = 0 ; i<n_video_encoder_device_list_size ; i++ ) {

sprintf( "Encoder#%02d = 0x%llX", i, p_video_encoder_device_list[ i ] );
}
```

3.2 QCAP_GET_DEVICE_ENUMERATION_ITEM_INFO

```
QRESULT QCAP_GET_DEVICE_ENUMERATION_ITEM_INFO( UINT          iDevNum,
                                               ULONGLONG *   pDeviceList,
                                               ULONG     *   pDeviceInfoH,
                                               ULONG     *   pDeviceInfoL )
```

Parameter

Name	Description
iDevNum	**IN**, Device index specified according to pszDevName, start from 0
pDeviceList	**IN**, Specify a device list
pDeviceInfoH	**OUT**, Pointer to a variable that stores the detailed information of device
pDeviceInfoL	**OUT**, Pointer to a variable that stores the simple information of device

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This is a help function for C# develop. User can use this function to get device enumeration item information.

Example

```
uint m_pVideoDeviceList = 0;

uint m_pVideoEncoderDeviceList = 0;

uint m_pAudioDeviceList = 0;

uint m_pAudioEncoderDeviceList = 0;

uint m_nVideoDeviceSize = 0;

uint m_nVideoEncoderDeviceSize = 0;

uint m_nAudioDeviceSize = 0;

uint m_nAudioEncoderDeviceSize = 0;

// ENUMERATE ALL CAPTURE DEVICES ON THE PLATFORM
EXPORTS.QCAP_DEVICE_ENUMERATION( ref m_pVideoDeviceList, ref m_nVideoDeviceSize,

                                 ref m_pVideoEncoderDeviceList, ref m_nVideoEncoderDeviceSize,

                                 ref m_pAudioDeviceList, ref m_nAudioDeviceSize,

                                 ref m_pAudioEncoderDeviceList, ref m_nAudioEncoderDeviceSize );
```



```
uint pDeviceInfoH = 0;

uint pDeviceInfoL = 0;

// GET DEVICE ITEM INFO
EXPORTS.QCAP_GET_DEVICE_ENUMERATION_ITEM_INFO( 0,  m_pVideoDeviceList, ref pDeviceInfoH,

                                                ref pDeviceInfoL );
```

3.3 QCAP_CREATE

```
QRESULT QCAP_CREATE( CHAR *    pszDevName,
                     UINT      iDevNum,
                     HWND      hAttachedWindow,
                     PVOID *   ppDevice,
                     BOOL      bThumbDraw = FALSE
                     BOOL      bMaintainAspectRatio = FALSE )
```

Parameter

Name	Description
pszDevName	**IN**, Device Name, Available values are "DC1150 USB"is for PD652 and PD652.3D "QP0204 USB"is for PD5A0*New* "UB658G USB"is for UB658G "CY3014 USB"is for UB530 and UB530G "TW6802 PCI"is for SC200, SC300, SC230 and SC330 "CX2581 PCI"is for SC310, SC340 "CX2385 PCI"is for SC350 "AH8400 PCI"is for SC290 and SC390 "FH8735 PCI"is for SC2A0 and SC3A0 "TW5864 PCI"is for SC2B0 and SC3B0 "SA7160 PCI"is for SC500, SC510, and TB510 "FH8735 PCI"is for SC580 "TW2809 PCI"is for SC590 "QP0203 PCI"is for SC5A0*New* "MZ0380 PCI"is for SC3C0 and SC5C0*New* "CAMERA"is for USB CAMERA "SOUNDCARD"is for SOUND CARD
iDevNum	**IN**, Device index specified according to pszDevName, start from 0
hAttachedWindow	**IN**, Handle of the window to show the preview video
ppDevice	**OUT**, Handle of the capture card object
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer
bMaintainAspectRatio	**IN**, Enable/Disable the maintain aspect ratio

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Running multiple devices simultaneously is supported. Use this function to get a handle which represents each card.

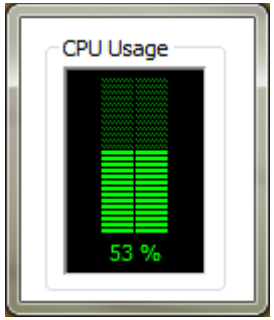
Controldifferent devices in the same process or in different process are all supported.

If attached window is NULL, we will enable device under background mode.

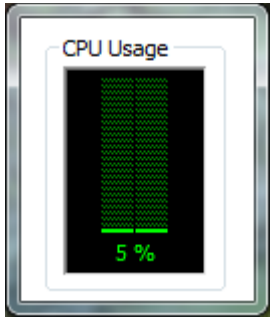
Developer can register both video and audio preview callbacks to obtain the raw frame buffers.

The thumb draw function can reduce the loading of video display, when display window size is small than the input video size.

For multiple channels developer, we suggest to turn on it in order to obtain the best rendering performance.



Normal Display



Thumb Draw Mode

Example

```
QCAP_CREATE( “SA7160 PCI”, 0, m_hWnd, &pDevice, FALSE, FALSE );
```

3.4 QCAP_DESTROY

```
QRESULT QCAP_DESTROY( PVOID pDevice )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Destroy capture card object to release its resource.

Example

```
QCAP_DESTROY( pDevice );
```

3.5 QCAP_RUN

```
QRESULT QCAP_RUN( PVOID pDevice )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Start video and audio capturing.

Example

```
QCAP_RUN( pDevice );
```

3.6 QCAP_STOP

```
QRESULT QCAP_STOP( PVOID pDevice )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Stop video and audio capturing.

Example

```
QCAP_STOP( pDevice );
```

3.7 QCAP_GET_VIDEO_INPUT

```
QRESULT QCAP_GET_VIDEO_INPUT( PVOID pDevice, ULONG * pInput )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pInput	**OUT**, Pointer of storage of input video source

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current video input source.

Example

```
QCAP_GET_VIDEO_INPUT( pDevice, &nInput );
```

3.8 QCAP_SET_VIDEO_INPUT

```
QRESULT QCAP_SET_VIDEO_INPUT( PVOID pDevice, ULONG nInput )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nInput	**IN**, Specify input video source. Available values are QCAP_INPUT_TYPE_COMPOSITE, QCAP_INPUT_TYPE_SVIDEO, QCAP_INPUT_TYPE_HDMI, QCAP_INPUT_TYPE_DVI_D, QCAP_INPUT_TYPE_COMPONENTS, QCAP_INPUT_TYPE_YCBCR, QCAP_INPUT_TYPE_DVI_A, QCAP_INPUT_TYPE_RGB, QCAP_INPUT_TYPE_VGA, QCAP_INPUT_TYPE_SDI, QCAP_INPUT_TYPE_DISPLAY_PORT, QCAP_INPUT_TYPE_AUTO

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set video source.

If nInput set to `QCAP_INPUT_TYPE_AUTO`, video source will be switched automatically if signal missed.

The duration of auto detection function between 2 inputs is 3 sec as default.

Developer can use `QCAP_SET_SYSTEM_CONFIGURATION` to modify it.

Note

If signal is unstable, we suggest custmer to set input directly.

Example

```
QCAP_SET_VIDEO_INPUT(  pDevice, QCAP_INPUT_TYPE_HDMI );
```

3.9 QCAP_GET_AUDIO_INPUT

```
QRESULT QCAP_GET_AUDIO_INPUT( PVOID pDevice, ULONG * pInput )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pInput	**OUT**, Pointer of storage of input audio source

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current audio input source.

Example

```
QCAP_GET_AUDIO_INPUT( pDevice, &nInput );
```

3.10 QCAP_SET_AUDIO_INPUT

```
QRESULT QCAP_SET_AUDIO_INPUT( PVOID pDevice, ULONG nInput )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nInput	**IN**, Specify input audio source. Available values are QCAP_INPUT_TYPE_EMBEDDED_AUDIO QCAP_INPUT_TYPE_LINE_IN QCAP_INPUT_TYPE_SOUNDCARD_MICROPHONE *New* QCAP_INPUT_TYPE_SOUNDCARD_LINE_IN *New*

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio input source.

Here, the embedded audio input means HDMI or SDI audio.

Note

Custome can use this function to change to soundcard input;

Moreover, developer alse can use `QCAP_CREATE` function

to create independent input sound card device by `QCAP_CREATE( “SOUNDCARD”, … )`.

Example

```
QCAP_SET_AUDIO_INPUT( pDevice, QCAP_INPUT_TYPE_EMBEDDED_AUDIO );
```

3.11 QCAP_GET_VIDEO_STANDARD

```
QRESULT QCAP_GET_VIDEO_STANDARD( PVOID     pDevice,
                                 ULONG  *  pStandard )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pStandard	**OUT**, Pointer to a variable that stores the standard. It cannot be NULL

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current video standard. This is function only support for SD capture card.

Example

```
QCAP_GET_VIDEO_STANDARD( pDevice, &nStandard );
```

3.12 QCAP_SET_VIDEO_STANDARD

```
QRESULT QCAP_SET_VIDEO_STANDARD( PVOID  pDevice,
                                 ULONG  nStandard )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nStandard	**IN**, Specify input video standard. Available values are QCAP_STANDARD_TYPE_NTSC_M QCAP_STANDARD_TYPE_NTSC_M_J QCAP_STANDARD_TYPE_NTSC_433 QCAP_STANDARD_TYPE_PAL_M QCAP_STANDARD_TYPE_PAL_60 QCAP_STANDARD_TYPE_PAL_B QCAP_STANDARD_TYPE_PAL_D QCAP_STANDARD_TYPE_PAL_G QCAP_STANDARD_TYPE_PAL_H QCAP_STANDARD_TYPE_PAL_I QCAP_STANDARD_TYPE_PAL_N QCAP_STANDARD_TYPE_PAL_N_COMBO QCAP_STANDARD_TYPE_AUTO (DEFAULT)

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set current video standard.

If nStandard set to `QCAP_STANDARD_TYPE_AUTO`, video standard will be switched automatically.

Note

It is only for PAL and NTSC format.

If your input signal is unstable, please disable atuo signal detection and set the standard directly.

This is function only support for SD capture card.

Example

```
QCAP_SET_VIDEO_STANDARD( pDevice, QCAP_STANDARD_TYPE_AUTO );
```

3.13 QCAP_GET_VIDEO_CURRENT_CONTENT_PROTECTION_STATUS

```
QRESULT QCAP_GET_VIDEO_CURRENT_CONTENT_PROTECTION_STATUS(   PVOID     pDevice,
                                                            ULONG  *  pStatus )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pWidth	**OUT**, Pointer to the storage of current content protection status

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current content input source protection status (hdcp & marcovision).

Example

```
QCAP_GET_VIDEO_CURRENT_CONTENT_PROTECTION_STATUS( pDevice, &nStatus );
```

3.14 QCAP_GET_VIDEO_CURRENT_INPUT_FORMAT

```
QRESULT QCAP_GET_VIDEO_CURRENT_INPUT_FORMAT(   PVOID     pDevice,
                                               ULONG  *   pColorSpaceType,
                                               ULONG  *   pWidth,
                                               ULONG  *   pHeight,
                                               BOOL   *   pIsInterleaved,
                                               double *   pFrameRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pColorSpaceType	**OUT**, Pointer to the storage of current color space type
pWidth	**OUT**, Pointer to the storage of current video width
pHeight	**OUT**, Pointer to the storage of current video height
pIsInterleaved	**OUT**, Pointer to the storage of current Is Interleaved
pFrameRate	**OUT**, Pointer to the storage of current frame rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current input video format information.

Example

```
QCAP_GET_VIDEO_CURRENT_INPUT_FORMAT( pDevice, &nColorSpaceType, &nWidth, &nHeight,

                                     &nIsInterleaved, &nFrameRate );
```

3.15 QCAP_GET_AUDIO_CURRENT_INPUT_FORMAT

```
QRESULT QCAP_GET_AUDIO_CURRENT_INPUT_FORMAT(   PVOID     pDevice,
                                               ULONG  *   pChannels,
                                               ULONG  *   pBitsPerSample,
                                               BOOL   *   pSampleFrequency )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pChannels	**OUT**, Pointer to the storage of current channels
pBitsPerSample	**OUT**, Pointer to the storage of current bit per sample
pSampleFrequency	**OUT**, Pointer to the storage of current sample frequency

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current default audio format parameters.

Example

```
QCAP_GET_AUDIO_CURRENT_INPUT_FORMAT( pDevice, &nChannels, &nBitsPerSample, &nSampleFrequency );
```

3.16 QCAP_GET_VIDEO_DEFAULT_OUTPUT_FORMAT

```
QRESULT QCAP_GET_VIDEO_DEFAULT_OUTPUT_FORMAT(  PVOID     pDevice,
                                               ULONG  *   pColorSpaceType,
                                               ULONG  *   pWidth,
                                               ULONG  *   pHeight,
                                               BOOL   *   pIsInterleaved,
                                               double *   pFrameRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pColorSpaceType	**OUT**, Pointer to the storage of current color space type
pWidth	**OUT**, Pointer to the storage of current video width
pHeight	**OUT**, Pointer to the storage of current video height
pIsInterleaved	**OUT**, Pointer to the storage of current Is Interleaved
pFrameRate	**OUT**, Pointer to the storage of current frame rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current default video format parameters.

Example

```
QCAP_GET_VIDEO_DEFAULT_OUTPUT_FORMAT( pDevice, &nWidth, &nHeight, &nColorSpaceType,

                                      &nIsInterleaved, &nFrameRate );
```

3.17 QCAP_SET_VIDEO_DEFAULT_OUTPUT_FORMAT

```
QRESULT QCAP_SET_VIDEO_DEFAULT_OUTPUT_FORMAT(  PVOID     pDevice,
                                               ULONG     nColorSpaceType,
                                               ULONG     nWidth,
                                               ULONG     nHeight,
                                               BOOL      bIsInterleaved,
                                               double    dFrameRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify video width
nHeight	**IN**, Specify video height
bIsInterleaved	**IN**, Specify video Is Interleaved

dFrameRate	**IN**, Specify video frame rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to change default video format output.

In original QCAP concept, the default format output is same as input signal format,

so customer doesn’t need setup output format from any function.

The function is used be advanced user who want to fixed special format output from capture card.

For example, the SC300 outputs 720x480i@60fps at NTSC as default.

You can use the function to change it to 704x480i@60fps.

The function is only opened by all SD capture cards currentlly.

Note

For example, LabView developer can use this to set video output format.

Example

```
QCAP_SET_VIDEO_DEFAULT_OUTPUT_FORMAT( pDevice, QCAP_COLORSPACE_TYEP_YUY2, 720, 480, 1, 60 );
```

3.18 QCAP_GET_AUDIO_DEFAULT_OUTPUT_FORMAT

```
QRESULT QCAP_GET_AUDIO_DEFAULT_OUTPUT_FORMAT(  PVOID     pDevice,
                                               ULONG  *   pChannels,
                                               ULONG  *   pBitsPerSample,
                                               BOOL   *   pSampleFrequency )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pChannels	**OUT**, Pointer to the storage of current channels
pBitsPerSample	**OUT**, Pointer to the storage of current bit per sample
pSampleFrequency	**OUT**, Pointer to the storage of current sample frequency

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current default output audio format parameters.

Example

```
QCAP_GET_AUDIO_DEFAULT_OUTPUT_FORMAT( pDevice, &nChannel, &nBitsPerSample, &nSampleFrequencyd );
```

3.19 QCAP_SET_AUDIO_DEFAULT_OUTPUT_FORMAT

```
QRESULT QCAP_SET_AUDIO_DEFAULT_OUTPUT_FORMAT(  PVOID  pDevice,
                                               ULONG  nChannels,
                                               ULONG  nBitsPerSample,
                                               BOOL   nSampleFrequency )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nChannels	**IN**, Specify current channels
nBitsPerSample	**IN**, Specify current bit per sample
nSampleFrequency	**IN**, Specify current sample frequency

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set current default output audio format parameters.

Note

For example, LabView developer can use this to set video output format.

Example

```
QCAP_SET_AUDIO_DEFAULT_OUTPUT_FORMAT( pDevice, 2, 16, 48000 );
```

3.20 QCAP_GET_VIDEO_DEINTERLACE_TYPE

```
QRESULT QCAP_GET_VIDEO_DEINTERLACE_TYPE( PVOID pDevice, ULONG * pType )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pType	**OUT**, Pointer to a variable that stores the de-interlace

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current video de-interlace method.

Example

```
QCAP_GET_VIDEO_DEINTERLACE_TYPE( pDevice, &Type );
```

3.21 QCAP_SET_VIDEO_DEINTERLACE_TYPE

```
QRESULT QCAP_SET_VIDEO_DEINTERLACE_TYPE( PVOID pDevice, ULONG nType )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nType	**IN**, Specify input video de-interlace. Available values are QCAP_SOFTWARE_DEINTERLACE_TYPE_BLENDING QCAP_SOFTWARE_DEINTERLACE_TYPE_MOTIONAD_APTIVE QCAP_SOFTWARE_DEINTERLACE_TYPE_FILTER_TRIANGLE QCAP_SOFTWARE_DEINTERLACE_TYPE_MEDICAL_RGB_REPACK

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set current video de-interlace method.

Example

```
QCAP_SET_VIDEO_DEINTERLACE_TYPE( pDevice, QCAP_SOFTWARE_DEINTERLACE_TYPE_BLENDING );
```

3.22 QCAP_GET_VIDEO_DEINTERLACE

```
QRESULT QCAP_GET_VIDEO_DEINTERLACE( PVOID pDevice, BOOL * pEnable )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pEnable	**OUT**, Returned the current status of this function

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get the current status if de-interlace function is enabled.

Example

```
QCAP_GET_VIDEO_DEINTERLACE( pDevice, &bEnable );
```

3.23 QCAP_SET_VIDEO_DEINTERLACE

```
QRESULT QCAP_SET_VIDEO_DEINTERLACE( PVOID pDevice, BOOL bEnable )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
bEnable	**IN**, Enable/Disable video deinterlace

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to enable/disable de-interlace function.

The function is only enabled for interleaved video format, such as 1920x1080i@60fps or 720x480i@60fps.

The default algorithm of de-interlace function is the belnding method.

User must use QCAP_SET_VIDEO_DEINTERLACE_TYPE to set deinterlace type first and then

user can use QCAP_SET_VIDEO_DEINTERACE to start deinterlace.

Example

```
QCAP_SET_VIDEO_DEINTERLACE( pDevice, TRUE );
```

3.24 QCAP_GET_VIDEO_REGION_DISPLAY

```
QRESULT QCAP_GET_VIDEO_REGION_DISPLAY(  PVOID     pDevice,
                                        ULONG *   pCropX,
                                        ULONG *   pCropY,
                                        ULONG *   pCropW,
                                        ULONG *   pCropH )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pCropX	**OUT**, Pointer to the storage of x-coordinate of the crop region display
pCropY	**OUT**, Pointer to the storage of y-coordinate of the crop region display
pCropW	**OUT**, Pointer to the storage of horizontal width of crop region display
pCropH	**OUT**, Pointer to the storage of vertical height of crop region display

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get the region display information with all devices.

Example

```
QCAP_GET_VIDEO_REGION_DISPLAY( pDevice, &bCropX, &bCropY, &bCropW, &bCropH );
```

3.25 QCAP_SET_VIDEO_REGION_DISPLAY

```
QRESULT QCAP_SET_VIDEO_REGION_DISPLAY(  PVOID  pDevice,
                                        ULONG  nCropX,
                                        ULONG  nCropY,
                                        ULONG  nCropW,
                                        ULONG  nCropH )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nCropX	**IN**, Specify the x-coordinate of the crop region display
nCropY	**IN**, Specify the y-coordinate of the crop region display
nCropW	**IN**, Specify the horizontal width of crop region display
nCropH	**IN**, Specify the vertical height of crop region display

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set the any region display with all devices.

Example

```
QCAP_SET_VIDEO_REGION_DISPLAY( pDevice, TRUE, 0, 0, 1280, 720 );
```

3.26 QCAP_GET_VIDEO_MIRROR

```
QRESULT QCAP_GET_VIDEO_MIRROR( PVOID pDevice, BOOL * pHorizontalMirror, BOOL * pVerticalMirror )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pHorizontalMirror	**OUT**, Returned the current horizontal of mirror
pVerticalMirror	**OUT**, Returned the current vertical of mirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out

QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get the current status of mirror of video.

Example

```
QCAP_GET_VIDEO_MIRROR( pDevice, &bHorizontalMirror, &bVerticalMirror );
```

3.27 QCAP_SET_VIDEO_MIRROR

```
QRESULT QCAP_SET_VIDEO_MIRROR( PVOID pDevice, BOOL  bHorizontalMirror, BOOL  bVerticalMirror )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
bHorizontalMirror	**IN**, Enable/Disable horizontal of mirror
bVerticalMirror	**IN**, Enable/Disable vertical of mirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set the status of mirror of video.

Example

```
QCAP_SET_VIDEO_MIRROR( pDevice, TRUE, TRUE );
```

3.28 QCAP_GET_VIDEO_BRIGHTNESS

```
QRESULT QCAP_GET_VIDEO_BRIGHTNESS( PVOID pDevice, ULONG * pValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pValue	**OUT**, Return the current brightness

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current brightness.

Example

```
QCAP_GET_VIDEO_BRIGHTNESS( pDevice, &nValue );
```

3.29 QCAP_SET_VIDEO_BRIGHTNESS

```
QRESULT QCAP_SET_VIDEO_BRIGHTNESS( PVOID pDevice, ULONG nValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nValue	**IN**, Specify the value for brightness, from 0~255

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set brightness.

Example

```
QCAP_SET_VIDEO_BRIGHTNESS( pDevice, 128 );
```

3.30 QCAP_GET_VIDEO_CONTRAST

```
QRESULT QCAP_GET_VIDEO_CONTRAST( PVOID pDevice, ULONG * pValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pValue	**OUT**, Return the current contrast

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current contrast.

Example

```
QCAP_GET_VIDEO_CONTRAST( pDevice, &nValue );
```

3.31 QCAP_SET_VIDEO_CONTRAST

```
QRESULT QCAP_SET_VIDEO_CONTRAST( PVOID pDevice, ULONG nValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nValue	**IN**, Specify the value for contrast, from 0~255

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set contrast.

Example

```
QCAP_SET_VIDEO_CONTRAST( pDevice, 128 );
```

3.32 QCAP_GET_VIDEO_HUE

```
QRESULT QCAP_GET_VIDEO_HUE( PVOID pDevice, ULONG * pValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pValue	**OUT**, Return the current hue

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get hue.

Example

```
QCAP_GET_VIDEO_HUE( pDevice, &nValue );
```

3.33 QCAP_SET_VIDEO_HUE

```
QRESULT QCAP_SET_VIDEO_HUE( PVOID pDevice, ULONG nValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nValue	**IN**, Specify the value for hue, from 0~255

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set hue.

Example

```
QCAP_SET_VIDEO_HUE( pDevice, 128 );
```

3.34 QCAP_GET_VIDEO_SATURATION

```
QRESULT QCAP_GET_VIDEO_SATURATION( PVOID pDevice, ULONG * pValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pValue	**OUT**, Return the current saturation

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get saturation.

Example

```
QCAP_GET_VIDEO_SATURATION( pDevice, &nValue );
```

3.35 QCAP_SET_VIDEO_SATURATION

```
QRESULT QCAP_SET_VIDEO_SATURATION( PVOID pDevice, ULONG nValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nValue	**IN**, Specify the value for saturation, from 0~255

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set saturation.

Example

```
QCAP_SET_VIDEO_SATURATION( pDevice, 128 );
```

3.36 QCAP_GET_VIDEO_SHARPNESS

```
QRESULT QCAP_GET_VIDEO_SHARPNESS( PVOID pDevice, ULONG * pValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pValue	**OUT**, Return the current sharpness

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get sharpness.

Example

```
QCAP_GET_VIDEO_SHARPNESS( pDevice, &nValue );
```

3.37 QCAP_SET_VIDEO_SHARPNESS

```
QRESULT QCAP_SET_VIDEO_SHARPNESS( PVOID pDevice, ULONG nValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nValue	**IN**, Specify the value for sharpness, from 0~255

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set sharpness.

Example

```
QCAP_SET_VIDEO_SHARPNESS( pDevice, 128 );
```

3.38 QCAP_GET_VIDEO_PREVIEW_PROPERTY_EX

```
QRESULT QCAP_GET_VIDEO_PREVIEW_PROPERTY_EX( PVOID pDevice, ULONG * pDownscaleMode )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pDownscaleMode	**OUT**, Pointer to the storage of current recording down scale mode

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is only support on SC590 capture card.

User can use this function to get video preview format size of SC590 capture card.

Example

```
QCAP_GET_VIDEO_PREVIEW_PROPERTY_EX( pDevice, DownscaleMode );
```

3.39 QCAP_SET_VIDEO_PREVIEW_PROPERTY_EX

QRESULT QCAP_SET_VIDEO_PREVIEW_PROPERTY_EX(PVOID pDevice, ULONG nDownscaleMode)

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object

nDownscaleMode	**IN**, Specify video preview down scale mode. Available values are
	QCAP_DOWNSCALE_MODE_OFF = 0
	QCAP_DOWNSCALE_MODE_2_3 = 1 // EX,1920X1080 -> 1280X720
	QCAP_DOWNSCALE_MODE_1_2 = 2 // EX,1920X1080 -> 960X540
	QCAP_DOWNSCALE_MODE_1_4 = 3 // EX,1920X1080 -> 480X270

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is only support on SC590 capture card.

User can use this function to set video preview format size of SC590 capture card.

Example

```
QCAP_SET_VIDEO_PREVIEW_PROPERTY_EX( pDevice, QCAP_DOWNSCALE_MODE_OFF );
```

3.40 QCAP_GET_VIDEO_HARDWARE_ENCODER_PROPERTY

```
QRESULT QCAP_GET_VIDEO_HARDWARE_ENCODER_PROPERTY( PVOID     pDevice,
                                                  UINT      iRecNum,
                                                  ULONG *   pEncoderFormat,
                                                  ULONG *   pRecordMode,
                                                  ULONG *   pQuality,
                                                  ULONG *   pBitRate,
                                                  ULONG *   pGOP,
                                                  ULONG *   pAspectRatioX,
                                                  ULONG *   pAspectRatioY,
                                                  ULONG *   pDownscaleMode,
                                                  ULONG *   pPostSkipFrameRate,
                                                  ULONG *   pPostAvgFrameRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pRecordProfile	**OUT**, Pointer to the storage of current recording profile
pRecordLevel	**OUT**, Pointer to the storage of current recording level
pRecordEntropy	**OUT**, Pointer to the storage of current recording entropy
pRecordMode	**OUT**, Pointer to the storage of current recording mode
pQuality	**OUT**, Pointer to the storage of current recording quality
pBitRate	**OUT**, Pointer to the storage of current recording bit rate
pGOP	**OUT**, Pointer to the storage of current recording GOP size
pAspectRatioX	**OUT**, Pointer to the storage of aspect ratio X axle
pAspectRatioY	**OUT**, Pointer to the storage of aspect ratio Y axle
pDownscaleMode	**OUT**, Pointer to the storage of current recording down scale mode
pPostSkipFrameRate	**OUT**, Pointer to the storage of post skip frame rate
pPostAvgFrameRate	**OUT**, Pointer to the storage of post average frame rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use hardware encode capture card, user can use this function to get video hardware recording parameters.

Example

```
QCAP_GET_VIDEO_RECORD_HARDWARE_ENCODER_PROPERTY(
                                    pDevice, 0, &nEncoderFormat,&nRecordProfile, &nRecordLevel,
                                    &nRecordEntropy, &nRecordMode, &nQuality, &nBitRate, &nGOP,
                                    &nAspectRatioX, &nAspectRatioY, &nDownscaleMode, &nPostSkipFrameRate,
                                    &nPostAvgFrameRate );
```

3.41 QCAP_SET_VIDEO_HARDWARE_ENCODER_PROPERTY

```
QRESULT QCAP_SET_VIDEO_RECORD_ENCODER_PROPERTY(    PVOID   pDevice,
                                                   UINT    iRecNum,
                                                   ULONG   nEncoderFormat,
                                                   ULONG   nRecordProfile,
                                                   ULONG   nRecordLevel,
                                                   ULONG   nRecordEntropy,
                                                   ULONG   nRecordMode,
                                                   ULONG   nQuality,
                                                   ULONG   nBitRate,
                                                   ULONG   nGOP,
                                                   ULONG   nAspectRatioX,
                                                   ULONG   nAspectRatioY,
                                                   ULONG   nDownscaleMode,
                                                   ULONG   nPostSkipFrameRate,
                                                   ULONG   nPostAvgFrameRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to set recording parameters

nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,

nRecordProfile	**IN**, Specify recording profile
	QCAP_RECORD_PROFILE_BASELINE = 0, // DEFAULT
	QCAP_RECORD_PROFILE_MAIN = 1,
	QCAP_RECORD_PROFILE_HIGH = 2,

nRecordLevel	**IN**, Specify recording level
	QCAP_RECORD_LEVEL_1 = 10,
	QCAP_RECORD_LEVEL_1B = 9,
	QCAP_RECORD_LEVEL_11 = 11,
	QCAP_RECORD_LEVEL_12 = 12,
	QCAP_RECORD_LEVEL_13 = 13,
	QCAP_RECORD_LEVEL_2 = 20,
	QCAP_RECORD_LEVEL_21 = 21,
	QCAP_RECORD_LEVEL_22 = 22,
	QCAP_RECORD_LEVEL_3 = 30,
	QCAP_RECORD_LEVEL_31 = 31,
	QCAP_RECORD_LEVEL_32 = 32,
	QCAP_RECORD_LEVEL_4 = 40,
	QCAP_RECORD_LEVEL_41 = 41, // DEFAULT
	QCAP_RECORD_LEVEL_42 = 42,
	QCAP_RECORD_LEVEL_50 = 50,

	QCAP_RECORD_LEVEL_51 = 51, QCAP_RECORD_LEVEL_52 = 52,

nRecordEntropy	**IN**, Specify recording entropy
	QCAP_RECORD_ENTROPY_CAVLC = 0, // DEFAULT
	QCAP_RECORD_ENTROPY_CABAC = 1,

nRecordMode	**IN**, Specify recording mode. Available values are QCAP_RECORD_MODE_VBR = 0, QCAP_RECORD_MODE_CBR = 1, QCAP_RECORD_MODE_ABR= 2,
nQuality	**IN**, Specify recording quality, from 0~10000. It is used for VBR and ABR.
nBitRate	**IN**, Specify recording bit rate. It is used for CBR and ABR. (Ex. 12Mbps is 12 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0~255
nAspectRatioX	**IN**, Specify the aspect ratio X axle, 0 is turned off
nAspectRatioY	**IN**, Specify the aspect ratio Y axle, 0 is turned off

nDownscaleMode	**IN**, Specify recording down scale mode. Available values are
	QCAP_DOWNSCALE_MODE_OFF = 0
	QCAP_DOWNSCALE_MODE_1_2 = 2 // EX,1920X1080 -> 960X540
	QCAP_DOWNSCALE_MODE_1_4 = 3 // EX,1920X1080 -> 480X270

nPostSkipFrameRate	**IN**, Specify the post skip frame rate
nPostAvgFrameRate	**IN**, Specify the post average frame rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use hardware encode capture card, user can use this function to set video hardware recording parameters.

Example

```
QCAP_SET_VIDEO_RECORD_HARDWARE_ENCODER_PROPERTY(
                                    pDevice, 0, &nEncoderFormat, &nRecordProfile, &nRecordLevel,
                                    &nRecordEntropy, &nRecordMode, &nQuality, &nBitRate,&nGOP,
                                    &nAspectRatioX, &nAspectRatioY, &nDownscaleMode, &nPostSkipFrameRate,
                                    &nPostAvgFrameRate );
```

3.42 QCAP_GET_VIDEO_HARDWARE_ENCODER_PROPERTY_EX

```
QRESULT QCAP_GET_VIDEO_HARDWARE_ENCODER_PROPERTY_EX( PVOID     pDevice,
                                                     UINT      iRecNum,
                                                     ULONG *   pEncoderFormat,
                                                     ULONG *   pRecordProfile,
                                                     ULONG *   pRecordLevel,
                                                     ULONG *   pRecordEntropy,
                                                     ULONG *   pRecordMode,
                                                     ULONG *   pQuality,
                                                     ULONG *   pBitRate,
                                                     ULONG *   Pgop,
                                                     ULONG *   pBFrames,
                                                     BOOL  *   pIsInterleaved,
                                                     ULONG *   pSlices,
                                                     ULONG *   pLayers,
                                                     ULONG *   pSceneCut,
                                                     ULONG *   pAspectRatioX,
                                                     ULONG *   pAspectRatioY,
                                                     ULONG *   pCropX,
                                                     ULONG *   pCropY,
                                                     ULONG *   pCropW,
                                                     ULONG *   pCropH,
                                                     ULONG *   pDownscaleMode,
```

```
ULONG *    pPostSkipFrameRate,
ULONG *    pPostAvgFrameRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pRecordProfile	**OUT**, Pointer to the storage of current recording profile
pRecordLevel	**OUT**, Pointer to the storage of current recording level
pRecordEntropy	**OUT**, Pointer to the storage of current recording entropy
pRecordMode	**OUT**, Pointer to the storage of current recording mode
pQuality	**OUT**, Pointer to the storage of current recording quality
pBitRate	**OUT**, Pointer to the storage of current recording bit rate
pGOP	**OUT**, Pointer to the storage of current recording GOP size
pBFrames	**OUT**, Pointer to the storage of current recording B frames
pIsInterleaved	**OUT**, Pointer to the storage of current recording Interleaved
pSlices	**OUT**, Pointer to the storage of current recording slices
pLayers	**OUT**, Pointer to the storage of current recording layers
pSceneCut	**OUT**, Pointer to the storage of current recording screen cut

pAspectRatioX	**OUT**, Pointer to the storage of aspect ratio X axle
pAspectRatioY	**OUT**, Pointer to the storage of aspect ratio Y axle
pCropX	**OUT**, Pointer to the storage of x-coordinate of the crop
pCropY	**OUT**, Pointer to the storage of y-coordinate of the crop
pCropW	**OUT**, Pointer to the storage of horizontal width of crop
pCropH	**OUT**, Pointer to the storage of vertical height of crop
pDownscaleMode	**OUT**, Pointer to the storage of current recording down scale mode
pPostSkipFrameRate	**OUT**, Pointer to the storage of post skip frame rate
pPostAvgFrameRate	**OUT**, Pointer to the storage of post average frame rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use hardware encode capture card, user can use this function to get video hardware recording parameters and crop parameters or Advance parameters from H.264 Encoder, such as Profile, Level, Entropy, B Frames and Scene cut.

The extern function get supports the cropping function and the various up/down scaling function.

Example

```
QCAP_GET_VIDEO_HARDWARE_ENCODER_PROPERTY_EX(
                                    pDevice, 0, &nEncoderFormat, &nRecordProfile, &nRecordLevel,
                                    &nRecordEntropy, &nRecordMode, &nQuality, &nBitRate, &nGOP,
                                    &nBFrames, &nIsInterleaved, &nSlices, &nSceneCut, &nLayers
                                    &nAspectRatioX, &nAspectRatioY, &nCropX, &nCropY, &nCropW, &nCropH,
                                    &nDownscaleMode &nPostSkipFrameRate, &nPostAvgFrameRate );
```

3.43 QCAP_SET_VIDEO_HARDWARE_ENCODER_PROPERTY_EX

```
QRESULT QCAP_SET_VIDEO_HARDWARE_ENCODER_PROPERTY_EX( PVOID   pDevice,
                                                     UINT    iRecNum,
                                                     ULONG   nEncoderFormat,
                                                     ULONG   nRecordProfile,
                                                     ULONG   nRecordLevel,
                                                     ULONG   nRecordEntropy,
                                                     ULONG   nRecordMode,
                                                     ULONG   nQuality,
                                                     ULONG   nBitRate,
                                                     ULONG   nGOP,
                                                     ULONG   nBFrames,
                                                     BOOL    bIsInterleaved,
                                                     ULONG   nSlices,
                                                     ULONG   nLayers,
                                                     ULONG   nSceneCut,
                                                     ULONG   nAspectRatioX,
                                                     ULONG   nAspectRatioY,
                                                     ULONG   nCropX,
                                                     ULONG   nCropY,
                                                     ULONG   nCropW,
                                                     ULONG   nCropH,
                                                     ULONG   nDownscaleMode,
```

```
ULONG   nPostSkipFrameRate,
ULONG   nPostAvgFrameRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to set recording parameters
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,

nRecordProfile	**IN**, Specify recording profile
	QCAP_RECORD_PROFILE_BASELINE = 0, // DEFAULT
	QCAP_RECORD_PROFILE_MAIN = 1,
	QCAP_RECORD_PROFILE_HIGH = 2,

nRecordLevel	**IN**, Specify recording level QCAP_RECORD_LEVEL_1 = 10, QCAP_RECORD_LEVEL_1B = 9, QCAP_RECORD_LEVEL_11 = 11, QCAP_RECORD_LEVEL_12 = 12, QCAP_RECORD_LEVEL_13 = 13, QCAP_RECORD_LEVEL_2 = 20, QCAP_RECORD_LEVEL_21 = 21,

	QCAP_RECORD_LEVEL_22 = 22,
	QCAP_RECORD_LEVEL_3 = 30,
	QCAP_RECORD_LEVEL_31 = 31,
	QCAP_RECORD_LEVEL_32 = 32,
	QCAP_RECORD_LEVEL_4 = 40,
	QCAP_RECORD_LEVEL_41 = 41, // DEFAULT
	QCAP_RECORD_LEVEL_42 = 42,
	QCAP_RECORD_LEVEL_50 = 50,
	QCAP_RECORD_LEVEL_51 = 51,
	QCAP_RECORD_LEVEL_52 = 52,

nRecordEntropy	**IN**, Specify recording entropy
	QCAP_RECORD_ENTROPY_CAVLC = 0, // DEFAULT
	QCAP_RECORD_ENTROPY_CABAC = 1,

nRecordMode	**IN**, Specify recording mode. Available values are QCAP_RECORD_MODE_VBR = 0 , QCAP_RECORD_MODE_CBR = 1, QCAP_RECORD_MODE_ABR= 2,
nQuality	**IN**, Specify recording quality, from 0~10000. It is used for VBR and ABR.
nBitRate	**IN**, Specify recording bit rate. It is used for CBR and ABR. (Ex. 12Mbps is 12 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0~255
nBFrames	**IN**, Specify recording B frames
bIsInterleaved	**IN**, Specify recording interleaved
nSlices	**IN**, Specify recording slices

nLayers	**IN**, Specify recording layers
nSceneCut	**IN**, Specify recording Scene Cut, recommedned value is 40， 0 is function turned off
nAspectRatioX	**IN**, Specify the aspect ratio X axle, 0 is turned off
nAspectRatioY	**IN**, Specify the aspect ratio Y axle, 0 is turned off
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop

nDownscaleMode	**IN**, Specify recording down scale mode. Available values are
	QCAP_DOWNSCALE_MODE_OFF = 0
	QCAP_DOWNSCALE_MODE_1_2 = 2 // EX,1920X1080 -> 960X540
	QCAP_DOWNSCALE_MODE_1_4 = 3 // EX,1920X1080 -> 480X270

nPostSkipFrameRate	**IN**, Specify the post skip frame rate
nPostAvgFrameRate	**IN**, Specify the post average frame rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use hardware encode capture card, user can use this function to set video hardware recording parameters and crop parameters or Advance parameters from H.264 Encoder, such as Profile, Level, Entropy, B Frames and SceneCut.

The extern function get supports the cropping function and the various up/down scaling function.

Example

```
QCAP_SET_VIDEO_HARDWARE_ENCODER_PROPERTY_EX(
                                    pDevice, 0, nEncoderFormat, nRecordProfile, nRecordLevel,
                                    nRecordEntropy, nRecordMode, nQuality, nBitRate, nGOP, nBFrames,
                                    nIsInterleaved, nSlices, nLayers, nSceneCut,
                                    nAspectRatioX, nAspectRatioY, nCropX, nCropY, nCropW, nCropH,
                                    0, nPostSkipFrameRate, nPostAvgFrameRate );
```

3.44 QCAP_GET_VIDEO_HARDWARE_ENCODER_VIDEOCOMPRESSION_PROPERTY

```
QRESULT QCAP_GET_VIDEO_HARDWARE_ENCODER_VIDEOCOMPRESSION_PROPERTY( PVOID     pDevice,
                                                                   UINT      iRecNum,
                                                                   ULONG     nProperty,
                                                                   ULONG *   pValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to set recording parameters
nProperty	**IN**, Specifies the property that will be gotten from the device.
pValue	**OUT**, Pointer to a variable that stores the specify property value. It cannot be NULL. The range of value is dependent on its property

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use hardware encode capture card, this function is used to get some custom device properties.

Example

For more custom property programming, please reference Yuan's Windows Media SDK and

product’s Extra Prgoramming Guide in SDK packet.

For example, we can obtain device serial number from it.

```
QCAP_GET_VIDEO_HARDWARE_ENCODER_VIDEOCOMPRESSION_PROPERTY(  pDevice, 0,
                                                            KSPROPERTY_CUSTOM_XET_GPIO_DATA,
                                                            &nDeviceSerialNumber );
```

3.45 QCAP_SET_VIDEO_HARDWARE_ENCODER_VIDEOCOMPRESSION_PROPERTY

```
QRESULT QCAP_SET_VIDEO_HARDWARE_ENCODER_VIDEOCOMPRESSION_PROPERTY( PVOID     pDevice,
                                                                   UINT      iRecNum,
                                                                   ULONG     nProperty,
                                                                   ULONG     nValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to set recording parameters
nProperty	**IN**, Specifies the property that will be gotten from the device.
nValue	**IN**, Specifies the property value. The range of value is dependent on its property.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use haredware encode capture card, this function is used to get some custom device properties.

Example

For more custom property programming, please reference Yuan's Windows Media SDK and

product’s Extra Prgoramming Guide in SDK packet.

For example, we can obtain device serial number from it.

```
QCAP_SET_VIDEO_HARDWARE_ENCODER_VIDEOCOMPRESSION_PROPERTY(  pDevice, 0,
                                                            KSPROPERTY_CUSTOM_SET_OSD_COLOR,
                                                            nOsdColor );
```

3.46 QCAP_GET_AUDIO_VOLUME

```
QRESULT QCAP_GET_AUDIO_VOLUME( PVOID pDevice, ULONG * pVolume )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pVolume	**OUT**, Pointer to the storage of audio volume

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get audio volume.

Example

```
QCAP_GET_AUDIO_VOLUME( pDevice, &nVolume );
```

3.47 QCAP_SET_AUDIO_VOLUME

```
QRESULT QCAP_SET_AUDIO_VOLUME( PVOID pDevice, ULONG nVolume )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nVolume	**IN**, Audio volume, from 0~100, 0 is mute

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio volume.

Example

```
QCAP_SET_AUDIO_VOLUME( pDevice, 50 );
```

3.48 QCAP_GET_DEVICE_CUSTOM_PROPERTY

```
QRESULT QCAP_GET_DEVICE_CUSTOM_PROPERTY(    PVOID     pDevice,
                                            ULONG     nProperty,
                                            ULONG *   pValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nProperty	**IN**, Specifies the property that will be gotten from the device.
pValue	**OUT**, Pointer to a variable that stores the specify property value. It cannot be NULL. The range of value is dependent on its property

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is used to get some custom device properties.

Example

For more custom property programming,please reference Yuan's Windows Media SDK and product’s Extra Prgoramming Guide in SDK packet.

For example, we can obtain device serial number from it.

```
QCAP_GET_DEVICE_CUSTOM_PROPERTY(  pDevice,
                                  KSPROPERTY_CUSTOM_XET_GPIO_DATA,
                                  &nDeviceSerialNumber );
```

3.49 QCAP_SET_DEVICE_CUSTOM_PROPERTY

```
QRESULT QCAP_GET_DEVICE_CUSTOM_PROPERTY(    PVOID  pDevice,
                                            ULONG  nProperty,
                                            ULONG  nValue )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nProperty	**IN**, Specifies the property that will be set to the device.
nValue	**IN**, Specifies the property value. The range of value is dependent on its property.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is used to set/change some custom device properties.

Example

For more custom property programming, please reference Yuan's Windows Media SDK and product’s Extra Prgoramming Guide in SDK packet.

```
QCAP_SET_DEVICE_CUSTOM_PROPERTY(  pDevice,
                                  KSPROPERTY_CUSTOM_SET_OSD_COLOR,
                                  nOsdColor );
```

3.50 QCAP_GET_DEVICE_CUSTOM_PROPERTY_EX

```
QRESULT QCAP_GET_DEVICE_CUSTOM_PROPERTY_EX(PVOID     pDevice,
                                           ULONG     nProperty,
                                           BYTE   *  pValue,
                                           ULONG     nBytes )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nProperty	**IN**, Specifies the property that will be gotten from the device.
pValue	**OUT**, Pointer to a variable that stores the specify property value. It cannot be NULL. The range of value is dependent on its property.
nBytes	**IN**, Specifies the size of property value.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is used to get some custom device properties.

Example

For more custom property programming, please reference Yuan's Windows Media SDK and

product’s Extra Prgoramming Guide in SDK packet.

For example, we can obtain analog video switch table from it.

```
QCAP_GET_DEVICE_CUSTOM_PROPERTY_EX( pDevice,
                                    KSPROPERTY_CUSTOM_XET_ANALOG_VIDEO_SWITCH_TABLE,
                                    &nDeviceVideoSwitchTable
                                    12 );
```

3.51 QCAP_SET_DEVICE_CUSTOM_PROPERTY_EX

```
QRESULT QCAP_SET_DEVICE_CUSTOM_PROPERTY_EX(PVOID     pDevice,
                                           ULONG     nProperty,
                                           BYTE *    pValue,
                                           ULONG     nBytes )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
nProperty	**IN**, Specifies the property that will be gotten from the device.
pValue	**IN**, Pointer to a variable that stores the specify property value. It cannot be NULL. The range of value is dependent on its property.
nBytes	**IN**, Specifies the size of property value buffer.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

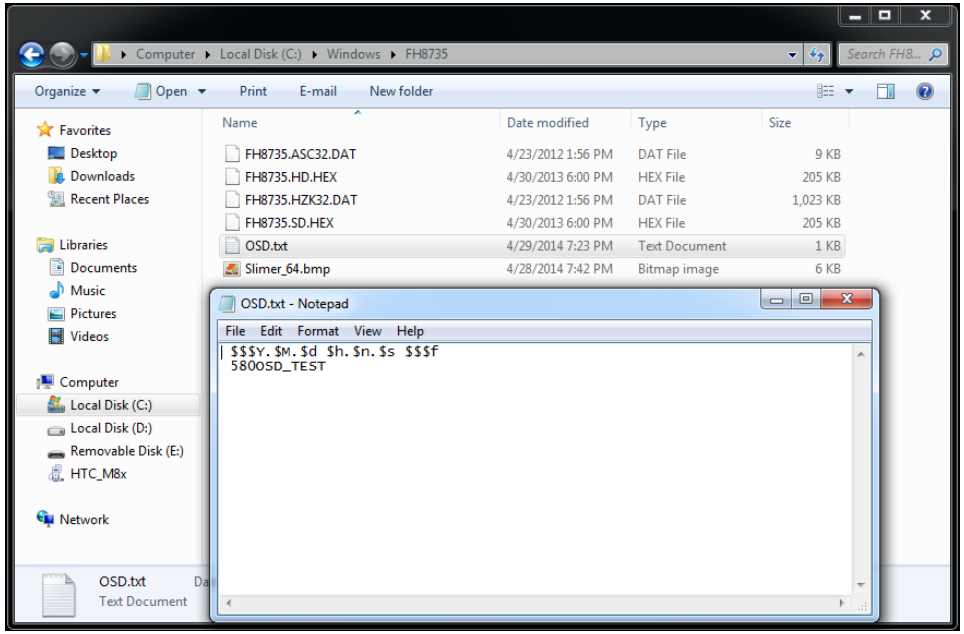
This function is used to set/change some custom device properties.

Example

For more custom property programming, please reference Yuan's Windows Media SDK and product’s Extra Prgoramming Guide in SDK packet.

There we give example for 580 OSD text and picture.

Example for 580 OSD text in hardware encoder.




```
QCAP_SET_DEVICE_CUSTOM_PROPERTY( pDevice, 929, 0x00000001 );

QCAP_SET_DEVICE_CUSTOM_PROPERTY( pDevice, 920, 0x00000000 );

CHAR path_text[] = "C:\\WINDOWS\\FH8735\\OSD.TXT";

QCAP_SET_DEVICE_CUSTOM_PROPERTY_EX( pDevice, 921, (BYTE *)(path_text), strlen(path_text) );
```

Example for 580 OSD picture in hardware encoder.

Note

580 allows software to load one 8 bits (256 colors) BMP file into its board memory.

```
ULONG params[ 4 ] = {   0,   /*Picture index is 0 or 1 */
                        1,   /*Picture start left position*/
                        1,   /*Picture start top position*/
                        255  /*Picture transparent is from 0~255*/
                    };

CHAR path_picture[] = "C:\\WINDOWS\\FH8735\\Slimer_64.bmp";

QCAP_SET_DEVICE_CUSTOM_PROPERTY_EX( pDevice, 970, (BYTE *)params, sizeof(params) );

QCAP_SET_DEVICE_CUSTOM_PROPERTY_EX( pDevice, 971, (BYTE *)(path_picture), strlen(path_picture) );
```

3.52 QCAP_COPY_VIDEO_PREVIEW_BUFFER

```
QRETURN QCAP_COPY_VIDEO_PREVIEW_BUFFER( PVOID         pDevice,
                                        DOUBLE *      pSampleTime,
                                        BYTE   *      pFrameBuffer,
                                        ULONG  *      pFrameBufferLen )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pSampleTime	**OUT**, Pointer to the storage of sample time
pFrameBuffer	**OUT**, Pointer to the storage of the frame buffer
pFrameBufferLen	**OUT**, Pointer to the input source buffer's size **IN**, Specify the input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is support for LABVIEW user, user can use this function to copy preview video data, and use SampleTime to control the Timestamp of video to get copy video data.

Note

This function is started after `QCAP_RUN`.

Example

```
QCAP_COPY_VIDEO_PREVIEW_BUFFER( pDevice, &Sampletime, &FrameBuffer, &FrameBufferlen );
```

3.53 QCAP_COPY_AUDIO_PREVIEW_BUFFER

```
QRETURN QCAP_COPY_AUDIO_PREVIEW_BUFFER( PVOID         pDevice,
                                        DOUBLE *      pSampleTime,
                                        BYTE   *      pFrameBuffer,
                                        ULONG  *      pFrameBufferLen )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pSampleTime	**OUT**, Pointer to the storage of sample time
pFrameBuffer	**OUT**, Pointer to the storage of the frame buffer
pFrameBufferLen	**OUT**, Pointer to the input source buffer's size **IN**, Specify the input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is support for LABVIEW user, user can use this function to copy preview audio data, and use SampleTime to control the timestamp of audio to copy audio data.

Note

This function is started after `QCAP_RUN`.

Example

```
QCAP_COPY_AUDIO_PREVIEW_BUFFER( pDevice, &Sampletime, &FrameBuffer, &FrameBufferlen );
```

3.54 QCAP_LOCK_VIDEO_PREVIEW_BUFFER

```
QRETURN QCAP_LOCK_VIDEO_PREVIEW_BUFFER( PVOID         pDevice,
                                        DOUBLE *      pSampleTime,
                                        BYTE   * *    ppFrameBuffer,
                                        ULONG  *      pFrameBufferLen )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pSampleTime	**OUT**, Pointer to the storage of sample time
ppFrameBuffer	**OUT**, Pointer to the storage of the frame buffer
pFrameBufferLen	**OUT**, Pointer to the input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is support for LABVIEW user, this function is same as video_preview_callback function

User can use this function to lock video data, and useSampleTime to control the timestamp of video to get lock video data.

Note

This function is started after `QCAP_RUN`.

Example

```
QCAP_LOCK_VIDEO_PREVIEW_BUFFER( pDevice, &Sampletime, &FrameBuffer, &FrameBufferlen );
```

3.55 QCAP_UNLOCK_VIDEO_PREVIEW_BUFFER

```
QRETURN QCAP_UNLOCK_VIDEO_PREVIEW_BUFFER( PVOID   pDevice )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is support for LABVIEW user, user can use this function to unlock video date,

Example

```
QCAP_UNLOCK_VIDEO_PREVIEW_BUFFER( pDevice );
```

3.56 QCAP_LOCK_AUDIO_PREVIEW_BUFFER

```
QRETURN QCAP_LOCK_AUDIO_PREVIEW_BUFFER( PVOID         pDevice,
                                        DOUBLE *      pSampleTime,
                                        BYTE   * *    ppFrameBuffer,
                                        ULONG  *      pFrameBufferLen )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pSampleTime	**OUT**, Pointer to the storage of sample time
ppFrameBuffer	**OUT**, Pointer to the storage of the frame buffer
pFrameBufferLen	**OUT**, Pointer to the input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is support for LABVIEW user, this function is same as audio_preview_callback function.

User can use this function to lock audio data, and use SampleTime to control the timestamp of audio to get audio data.

Note

This function is started after `QCAP_RUN`.

Example

```
QCAP_LOCK_AUDIO_PREVIEW_BUFFER( pDevice, &Sampletime, &FrameBuffer, &FrameBufferlen );
```

3.57 QCAP_UNLOCK_VIDEO_PREVIEW_BUFFER

```
QRETURN QCAP_UNLOCK_VIDEO_PREVIEW_BUFFER( PVOID   pDevice )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is support for LABVIEW user, user can use this function to unlock audio date,

Example

```
QCAP_UNLOCK_AUDIO_PREVIEW_BUFFER( pDevice );
```

4 Snapshot Function API

4.1 QCAP_SNAPSHOT_BMP

```
QRESULT QCAP_SNAPSHOT_BMP( PVOID      pDevice,
                           CHAR *     pszFilePathName,
                           BOOL       bIsAsync = TRUE,
                           ULONG      nMilliseconds = 0 )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with BMP format

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BMP( pDevice, "C:\\PICTURE1.BMP24" ); // PICTURE1.BMP24 ->PICTURE1.BMP
```

4.2 QCAP_SNAPSHOT_JPG

```
QRESULT QCAP_SNAPSHOT_JPG( PVOID     pDevice,
                           CHAR *    pszFilePathName,
                           ULONG     nQuality,
                           BOOL      bIsAsync = TRUE,
                           ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pszFilePathName	**IN**, Specify the file name to store JPG data
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with JPG format.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_JPG( pDevice, “C:\\PICTURE1.JPG”, 100 );
```

4.3 QCAP_SNAPSHOT_BMP_EX

```
QRESULT QCAP_SNAPSHOT_BMP_EX( PVOID     pDevice,
                              CHAR *    pszFilePathName,
                              ULONG     nCropX,
                              ULONG     nCropY,
                              ULONG     nCropW,
                              ULONG     nCropH,
                              ULONG     nDstW,
                              ULONG     nDstH,
                              BOOL      bIsAsync = TRUE,
                              ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with BMP format.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

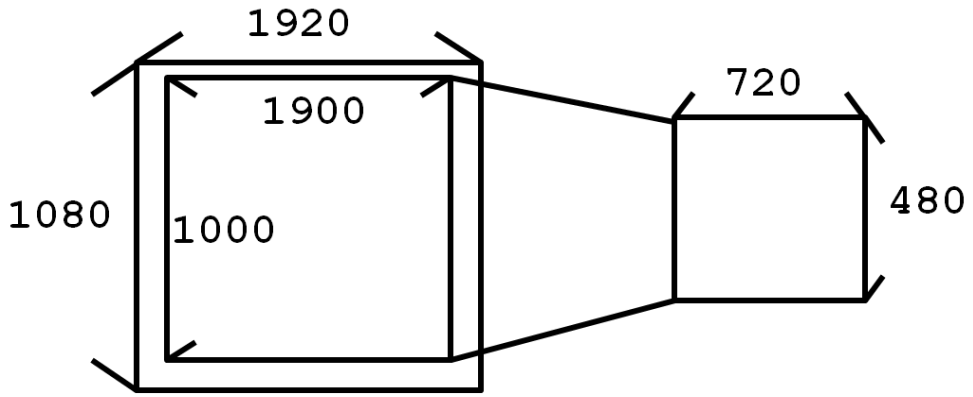
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BMP_EX( pDevice, "C:\\PICTURE1.BMP", 10, 40, 1900, 1000, 720, 480 );
```



4.4 QCAP_SNAPSHOT_JPG_EX

```
QRESULT QCAP_SNAPSHOT_JPG_EX(  PVOID     pDevice,
                               CHAR *    pszFilePathName,
                               ULONG     nCropX,
                               ULONG     nCropY,
                               ULONG     nCropW,
                               ULONG     nCropH,
                               ULONG     nDstW,
                               ULONG     nDstH,
                               ULONG     nQuality,
                               BOOL      bIsAsync = TRUE,
                               ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pszFilePathName	**IN**, Specify the file name to store JPG data
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with JPG format.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

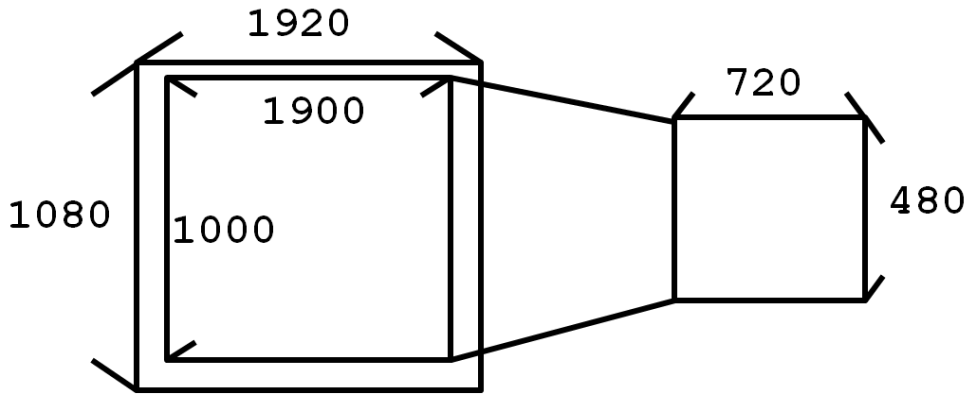
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_JPG_EX( pDevice, "C:\\PICTURE1.JPG", 10, 40, 1900, 1000, 720, 480, 100 );
```



5 Recording Function API

5.1 QCAP_GET_VIDEO_RECORD_PROPERTY

```
QRESULT QCAP_GET_VIDEO_RECORD_PROPERTY( PVOID     pDevice,
                                        UINT      iRecNum,
                                        ULONG *   pEncoderType,
                                        ULONG *   pEncoderFormat,
                                        ULONG *   pRecordMode,
                                        ULONG *   pQuality,
                                        ULONG *   pBitRate,
                                        ULONG *   pGOP,
                                        ULONG *   pAspectRatioX,
                                        ULONG *   pAspectRatioY,
                                        ULONG *   pDownscaleMode )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0. Max iRecNum is 0~3
pEncoderType	**OUT**, Pointer to the storage of current recording encoder type
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pRecordMode	**OUT**, Pointer to the storage of current recording mode
pQuality	**OUT**, Pointer to the storage of current recording quality
pBitRate	**OUT**, Pointer to the storage of current recording bit rate
pGOP	**OUT**, Pointer to the storage of current recording GOP size
pAspectRatioX	**OUT**, Pointer to the storage of aspect ratio X axle
pAspectRatioY	**OUT**, Pointer to the storage of aspect ratio Y axle
pDownscaleMode	**OUT**, Pointer to the storage of current recording down scale mode

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get video recording parameters.

Example

```
QCAP_GET_VIDEO_RECORD_PROPERTY( pDevice, 0,
                                &nEncoderType, &nEncoderFormat, &nRecordMode, &nQuality,
                                &nBitRate, &nGOP, &nAspectRatioX, &nAspectRatioY, &nDownscaleMode );
```

5.2 QCAP_SET_VIDEO_RECORD_PROPERTY

```
QRESULT QCAP_SET_VIDEO_RECORD_PROPERTY( PVOID   pDevice,
                                        UINT    iRecNum,
                                        ULONG   nEncoderType,
                                        ULONG   nEncoderFormat,
                                        ULONG   nRecordMode,
                                        ULONG   nQuality,
                                        ULONG   nBitRate,
                                        ULONG   nGOP,
                                        ULONG   nAspectRatioX,
                                        ULONG   nAspectRatioY,
                                        ULONG   nDownscaleMode )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0. Max iRecNum is 0~3
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,
nRecordMode	**IN**, Specify recording mode. Available values are QCAP_RECORD_MODE_VBR = 0, QCAP_RECORD_MODE_CBR = 1, QCAP_RECORD_MODE_ABR = 2,
nQuality	**IN**, Specify recording quality, from 0~10000. It is used for VBR and ABR.
nBitRate	**IN**, Specify recording bit rate. It is used for CBR and ABR. (Ex. 12Mbps is 12 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0~255
nAspectRatioX	**IN**, Specify the aspect ratio X axle, 0 is turned off
nAspectRatioY	**IN**, Specify the aspect ratio Y axle, 0 is turned off

nDownscaleMode	**IN**, Specify recording down scale mode. Available values are QCAP_DOWNSCALE_MODE_OFF = 0 QCAP_DOWNSCALE_MODE_2_3 = 1//EX,1920X1080 -> 1280X720 QCAP_DOWNSCALE_MODE_1_2 = 2// EX,1920X1080 -> 960X540 QCAP_DOWNSCALE_MODE_1_4 = 3// EX,1920X1080 -> 480X270

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set video recording parameters.

For the hardware compression card, SC290, SC390 and UB658G,

Then EncoderType is always `QCAP_ENCODER_TYPE_HARDWARE` at iRecNum 0 recording.

The parameter, nDownscaleMode, is `QCAP_DOWNSCALE_MODE_OFF`, too.

For the dual-stream hardware compression card, SC2B0, SC3B0, SC2A0, SC3A0, SC580 and SC590,

Then EncoderType is always `QCAP_ENCODER_TYPE_HARDWARE` at iRecNum 0 and iRecNum 1 recording.

Here, iRecNum 0 is used by main H.264 hardware encoder and iRecNum 1 is used by sub H.264 hardware encoder.
The parameter, nDownscaleMode, is `QCAP_DOWNSCALE_MODE_OFF` at iRecNum 0 and `QCAP_DOWNSCALE_MODE_1_2` at iRecNum 1.

Products	iRecNum 0	iRecNum 1
UB658G	QCAP_ENCODER_TYPE_HARDWARE + QCAP_DOWNSCALE_MODE_OFF	Software Video Encoder
SC290 / SC390		
SC2B0 / SC3B0		QCAP_ENCODER_TYPE_HARDWARE + QCAP_DOWNSCALE_MODE_1_2
SC2A0 / SC3A0		
SC580		
SC590		

Example

```
QCAP_SET_VIDEO_RECORD_PROPERTY(pDevice, 0, QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK, QCAP_ENCODER_FORMAT_H264,
                               QCAP_RECORD_MODE_CBR, 8000, 12*1024* 1024,
                               30, 4, 3, QCAP_DOWNSCALE_MODE_OFF );
```

5.3 QCAP_GET_VIDEO_RECORD_PROPERTY_EX (CROPPING)

```
QRESULT QCAP_GET_VIDEO_RECORD_PROPERTY_EX( PVOID      pDevice,
                                           UINT       iRecNum,
                                           ULONG  *   pEncoderType,
                                           ULONG  *   pEncoderFormat,
                                           ULONG  *   pRecordProfile,
                                           ULONG  *   pRecordLevel,
                                           ULONG  *   pRecordEntropy,
                                           ULONG  *   pRecordComplexity,
                                           ULONG  *   pRecordMode,
                                           ULONG  *   pQuality,
                                           ULONG  *   pBitRate,
                                           ULONG  *   pGOP,
                                           ULONG  *   pBFrames,
                                           BOOL   *   pIsInterleaved,
                                           ULONG  *   pSlices,
                                           ULONG  *   pLayers,
                                           ULONG  *   pSceneCut,
                                           ULONG  *   pAspectRatioX,
                                           ULONG  *   pAspectRatioY,
                                           ULONG  *   pCropX
                                           ULONG  *   pCropY
                                           ULONG  *   pCropW
```

```
ULONG  *  pCropH
ULONG  *  pDstW
ULONG  *  pDstH )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0. Max iRecNum is 0~3
pEncoderType	**OUT**, Pointer to the storage of current recording encoder type
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pRecordProfile	**OUT**, Pointer to the storage of current recording profile
pRecordLevel	**OUT**, Pointer to the storage of current recording level
pRecordEntropy	**OUT**, Pointer to the storage of current recording entropy
pRecordComplexity	**OUT**, Pointer to the storage of current recording complexity
pRecordMode	**OUT**, Pointer to the storage of current recording mode
pQuality	**OUT**, Pointer to the storage of current recording quality
pBitRate	**OUT**, Pointer to the storage of current recording bit rate
pGOP	**OUT**, Pointer to the storage of current recording GOP size
pBFrames	**OUT**, Pointer to the storage of current B frames
pIsInterleaved	**OUT**, Pointer to the storage of current interleave

pSlices	**OUT**, Pointer to the storage of current slices
pLayers	**OUT**, Pointer to the storage of current layers
pSceneCut	**OUT**, Pointer to the storage of current scene cut
pAspectRatioX	**OUT**, Pointer to the storage of aspect ratio X axle
pAspectRatioY	**OUT**, Pointer to the storage of aspect ratio Y axle
pCropX	**OUT**, Pointer to the storage of x-coordinate of the crop
pCropY	**OUT**, Pointer to the storage of y-coordinate of the crop
pCropW	**OUT**, Pointer to the storage of horizontal width of crop
pCropH	**OUT**, Pointer to the storage of vertical height of crop
pDstW	**OUT**, Pointer to the storage of horizontal width of current recording down scale
pDstH	**OUT**, Pointer to the storage of vertical height of current recording down scale

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get video recording parameters and crop parameters or advance parameters from H.264 Encoder, such as Profile, Level, Entropy, Complexity, B Frames and SceneCut.

The extern function get supports the cropping function and the various up/down scaling function.

Example

```
QCAP_GET_VIDEO_RECORD_PROPERTY_EX( pDevice, 0, &nEncoderType, &nEncoderFormat, &nRecordProfile,
                                   &nRecordLevel, &nRecordEntropy, &nRecordComplexity, &nRecordMode,
                                   &nQuality, &nBitRate, &nGOP, &nBFrames, &nIsInterleaved, &nSlices,
                                   &nLayers, &nSceneCut, &nAspectRatioX, &nAspectRatioY, &nCropX,
                                   &nCropY,&nCropW, &nCropH, &nDstW, &nDstH );
```

5.4 QCAP_SET_VIDEO_RECORD_PROPERTY_EX (CROPPING)

```
QRESULT QCAP_SET_VIDEO_RECORD_PROPERTY_EX( PVOID   pDevice,
                                           UINT    iRecNum,
                                           ULONG   nEncoderType,
                                           ULONG   nEncoderFormat,
                                           ULONG   nRecordProfile,
                                           ULONG   nRecordLevel,
                                           ULONG   nRecordEntropy,
                                           ULONG   nRecordComplexity,
                                           ULONG   nRecordMode,
                                           ULONG   nQuality,
                                           ULONG   nBitRate,
                                           ULONG   nGOP,
                                           ULONG   nBFrames,
                                           BOOL    bIsInterleaved,
                                           ULONG   nSlices,
                                           ULONG   nLayers,
                                           ULONG   nSceneCut,
                                           ULONG   nAspectRatioX,
                                           ULONG   nAspectRatioY,
                                           ULONG   nCropX,
                                           ULONG   nCropY,
                                           ULONG   nCropW,
```

```
ULONG   nCropH,
ULONG   nDstW,
ULONG   nDstH )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0. Max iRecNum is 0~3
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,

nRecordProfile	**IN**, Specify recording profile
	QCAP_RECORD_PROFILE_BASELINE = 0, // DEFAULT
	QCAP_RECORD_PROFILE_MAIN = 1,
	QCAP_RECORD_PROFILE_HIGH = 2,

nRecordLevel	**IN**, Specify recording level QCAP_RECORD_LEVEL_1 = 10,

	QCAP_RECORD_LEVEL_1B = 9,
	QCAP_RECORD_LEVEL_11 = 11,
	QCAP_RECORD_LEVEL_12 = 12,
	QCAP_RECORD_LEVEL_13 = 13,
	QCAP_RECORD_LEVEL_2 = 20,
	QCAP_RECORD_LEVEL_21 = 21,
	QCAP_RECORD_LEVEL_22 = 22,
	QCAP_RECORD_LEVEL_3 = 30,
	QCAP_RECORD_LEVEL_31 = 31,
	QCAP_RECORD_LEVEL_32 = 32,
	QCAP_RECORD_LEVEL_4 = 40,
	QCAP_RECORD_LEVEL_41 = 41, // DEFAULT
	QCAP_RECORD_LEVEL_42 = 42,
	QCAP_RECORD_LEVEL_50 = 50,
	QCAP_RECORD_LEVEL_51 = 51,
	QCAP_RECORD_LEVEL_52 = 52,

nRecordEntropy	**IN**, Specify recording entropy
	QCAP_RECORD_ENTROPY_CAVLC = 0, // DEFAULT
	QCAP_RECORD_ENTROPY_CABAC = 1,

nRecordComplexity	**IN**, Specify recording complexity
	QCAP_RECORD_COMPLEXITY_0 = 0, // BEST SPEED, DEFAULT
	QCAP_RECORD_COMPLEXITY_1 = 1,
	QCAP_RECORD_COMPLEXITY_2 = 2,
	QCAP_RECORD_COMPLEXITY_3 = 3,
	QCAP_RECORD_COMPLEXITY_4 = 4,

	QCAP_RECORD_COMPLEXITY_5 = 5, QCAP_RECORD_COMPLEXITY_6 = 6, // BEST QUALITY
nRecordMode	**IN**, Specify recording mode. Available values are QCAP_RECORD_MODE_VBR = 0 QCAP_RECORD_MODE_CBR = 1 QCAP_RECORD_MODE_ABR = 2
nQuality	**IN**, Specify recording quality, from 0~10000. It is used for VBR and ABR.
nBitRate	**IN**, Specify recording bit rate. It is used for CBR and ABR. (Ex. 12Mbps is 12 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0~255
nBFrames	**IN**, Specify recording B frames
bIsInterleaved	**IN**, Specify recording interleaved
nSlices	**IN**, Specify recording slices
nLayers	**IN**, Specify recording layers
nSceneCut	**IN**, Specify recording Scene Cut, recommedned value is 40， 0 is function turned off
nAspectRatioX	**IN**, Specify the aspect ratio X axle, 0 is turned off
nAspectRatioY	**IN**, Specify the aspect ratio Y axle, 0 is turned off
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop

nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set video recording parameters and crop parameters or advance parameters from H.264 encoder such as Profile, Level, Entropy, Complexity and B Frames.

The extern function set supports the cropping function and the various up/down scaling function.

Example

```
QCAP_SET_VIDEO_RECORD_PROPERTY_EX(  pDevice, 0, QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK,
                                    QCAP_ENCODER_FORMAT_H264, nRecordProfile, nRecordLevel,
                                    nRecordEntropy, 1, QCAP_RECORD_MODE_CBR, 8000,
                                    12*1024* 1024, 30, 0, FALUSE, 0, 0, 40,
                                    4, 3, 10, 40, 1900, 1000, 720, 480 );
```

5.5 QCAP_GET_AUDIO_RECORD_PROPERTY

```
QRESULT QCAP_GET_AUDIO_RECORD_PROPERTY( PVOID     pDevice,
                                        UINT      iRecNum,
                                        ULONG *   pEncoderType,
                                        ULONG *   pEncoderFormat )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0. Max iRecNum is 0~3
pEncoderType	**OUT**, Pointer to the storage of current recording encoder type
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get audio recording parameters.

Example

```
QCAP_GET_AUDIO_RECORD_PROPERTY( pDevice, 0, &nEncoderType, &nEncoderFormat );
```

5.6 QCAP_SET_AUDIO_RECORD_PROPERTY

```
QRESULT QCAP_SET_AUDIO_RECORD_PROPERTY( PVOID   pDevice,
                                        UINT    iRecNum,
                                        ULONG   nEncoderType,
                                        ULONG   nEncoderFormat )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0. Max iRecNum is 0~3
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_PCM = 0 QCAP_ENCODER_FORMAT_AAC = 1 QCAP_ENCODER_FORMAT_AAC_RAW = 1 QCAP_ENCODER_FORMAT_AAC_ADTS = 2

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio recording parameters.

For MP4 file recording, we suggest to use H.264 + AAC encoder. It is the most popular file format for Microsoft Windows 7 as default.

If customer doesn’t want to license AAC, the QCAP SDK also support H.264 + PCM file format for your software.

Note

Developer uses TS, FLV, RTMP and HLS must to set audio format as AAC_ADTS.

Example

```
QCAP_SET_AUDIO_RECORD_PROPERTY( pDevice, 0, QCAP_ENCODER_TYPE_SOFTWARE, QCAP_ENCODER_FORMAT_AAC);
```

About AAC License

This software development kit contains software programming code and libraries related to AAC function that is subject to your development use only. All the intellectual properties are held by individual license groups and/or the property holders. License and royalty fee should be charged by the license groups and/or property holders and paid by you if any of them is implemented into your commercial product. We will not be responsible for any illegal usage and/or any infringement of the rights of the individual owners of these intellectual properties.

Details of AAC could be found in the website:http://www.vialicensing.com/licensing/aac-overview.aspx

5.7 QCAP_GET_AUDIO_RECORD_PROPERTY_EX

```
QRESULT QCAP_GET_AUDIO_RECORD_ENCODER_EX(   PVOID      pDevice,
                                            UINT       iRecNum,
                                            ULONG *    pEncoderType,
                                            ULONG *    pEncoderFormat,
                                            ULONG *    pBitRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0. Max iRecNum is 0~3
pEncoderType	**OUT**, Pointer to the storage of current recording encoder type
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pBitRate	**OUT**, Pointer to the storage of current recording bit rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get extra current channel audio record property parameters.

Example

```
QCAP_GET_AUDIO_RECORD_PROPERTY_EX( pDevice, 0, &nEncoderType, &nEncoderFormat, &nBitRate );
```

5.8 QCAP_SET_AUDIO_RECORD_PROPERTY_EX

```
QRESULT QCAP_SET_AUDIO_RECORD_PROPERTY_EX( PVOID     pDevice,
                                           UINT      iRecNum,
                                           ULONG     nEncoderType,
                                           ULONG     nEncoderFormat,
                                           ULONG     nBitRate )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0. Max iRecNum is 0~3
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,
nBitRate	**IN**, Specify audio bit rate. Default is 128k and max is 496k

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set extra current channel audio record property parameters.

Example

```
QCAP_SET_AUDIO_RECORD_PROPERTY_EX( pDevice, 0, 0, 0, 12*1024*1024 );
```

5.9 QCAP_GET_VIDEO_RECORD_SOFTWARE_ENCODER_PROPERTY

```
QRESULT QCAP_GET_VIDEO_RECORD_SOFTWARE_ENCODER_PROPERTY(    PVOID     pDevice,
                                                            UINT      iRecNum,
                                                            ULONG *   pRecordMode,
                                                            ULONG *   pQuality,
                                                            ULONG *   pBitRate
                                                            ULONG *   pGOP )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0. Max iRecNum is 0~3
pRecordMode	**OUT**, Pointer to the storage of current recording mode
pQuality	**OUT**, Pointer to the storage of current recording quality of software encoder
pBitRate	**OUT**, Pointer to the storage of current recording bitrate of software encoder
pGOP	**OUT**, Pointer to the storage of current recording GOP of software encoder

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get auto current channel video record software encoder property parameters.

User can use this function to get dynamic bitrate & gop reconfiguration for all software encoders.

Example

```
QCAP_GET_VIDEO_RECORD_SOFTWARE_ENCODER_PROPERTY( pDevice, 0, &nRecordMode,
                                                 &nQuality, &nBitRate, &nGOP  );
```

5.10 QCAP_SET_VIDEO_RECORD_SOFTWARE_ENCODER_PROPERTY

```
QRESULT QCAP_SET_VIDEO_RECORD_SOFTWARE_ENCODER_PROPERTY(    PVOID     pDevice,
                                                            UINT      iRecNum,
                                                            ULONG     nRecordMode,
                                                            ULONG     nQuality,
                                                            ULONG     nBitRate,
                                                            ULONG     nGOP )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0. Max iRecNum is 0~3
nRecordMode	**IN**, Specify recording mode. Available values are QCAP_RECORD_MODE_VBR = 0, QCAP_RECORD_MODE_CBR = 1, QCAP_RECORD_MODE_ABR = 2,
nQuality	**IN**, Specify recording software encoder quality, from 0~10000. It is used for VBR and ABR.
nBitRate	**IN**, Specify recording software encoder bit rate. It is used for CBR and ABR. (Ex. 12Mbps is 12 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording software encoder GOP size, from 0~255

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set current channel video record software encoder property parameters.

User can use this function to set dynamic bitrate & gop reconfiguration for all software encoders.

Example

```
QCAP_SET_VIDEO_RECORD_SOFTWARE_ENCODER_PROPERTY( pDevice, 0, 1, 8000, 12*1024*1024, 30  );
```

5.11 QCAP_START_CLONE_RECORD

```
QRESULT QCAP_START_CLONE_RECORD( PVOID     pDevice,
                                 UINT      iRecNum,
                                 CHAR   *  pszFilePathNameA,
                                 CHAR   *  pszFilePathNameB,
                                 DWORD     dwFlags = QCAP_RECORD_FLAG_FULL,
                                 double    dVideoDelayTime = 0.0,
                                 double    dAudioDelayTime = 0.0,
                                 double    dSegmentDurationTime = 0.0,
                                 ULONG     nSegmentDurationSizeKB = 0 )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to start recording, start from 0. Max iRecNum is 0~3
pszFilePathNameA	**IN**, Specify the file name of storage for recording. Recording file format is AVI, if extension name is AVI. Recording file format is MP4, if extension name is MP4. Recording file format is ASF, if extension name is ASF. Recording file format is FLV, if extension name is FLV. Recording file format is TS, if extension name is TS. Recording file format is SCF, if extension name is SCF. Recording file format is WMV, if extension name is WMV.

	Recording file format is WAV, if extension name is WAV.
pszFilePathNameB	**IN**, Specify the file name of storage for recording. Recording file format is AVI, if extension name is AVI. Recording file format is MP4, if extension name is MP4. Recording file format is ASF, if extension name is ASF. Recording file format is FLV, if extension name is FLV. Recording file format is TS, if extension name is TS. Recording file format is SCF, if extension name is SCF. Recording file format is WMV, if extension name is WMV. Recording file format is WAV, if extension name is WAV.
dwFlags	**IN**, Specify the action flag of share recording. This parameter can also include any combination of the following flags. Available value are QCAP_RECORD_FLAG_FULL = 0x00000007 QCAP_RECORD_FLAG_FILE = 0x00000001 QCAP_RECORD_FLAG_ENCODE = 0x00000002 QCAP_RECORD_FLAG_DISPLAY = 0x00000004 QCAP_RECORD_FLAG_DECODE = 0x00000008 QCAP_RECORD_FLAG_VIDEO_ONLY = 0x00000010 QCAP_RECORD_FLAG_AUDIO_ONLY = 0x00000020 QCAP_RECORD_FLAG_VIDEO_USE_IDEAL_TIMESTAMP= 0x00000040 QCAP_RECORD_FLAG_AUDIO_USE_IDEAL_TIMESTAMP= 0x00000080 QCAP_RECORD_FLAG_INGNORE_FORMAT_CHANGED = 0x00000100
dVideoDelayTime	**IN**, Specify video delay time, default is 0.0
dAudioDelayTime	**IN**, Specify audio delay time, default is 0.0
dSegmentDurationTime	**IN**, Specify segment duration time

nSegmentDurationSizeKB	**IN**, Specify segment duration size, the uint is K Byte

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start clone recording and also can use this function to record two same files at same time.

Users can use dVideoDelayTime & dAudioDelayTime parameters to set video or audio delay time for control play speed.

QCAP support the multiple-streams recording at the same time. It means you can generate many recording files with different resolution or bit rate for one capture device. The dwFlags parameter allows you to turn on/off internal components of recording.

User can use dSegmentDurationTime parameter to set segment duration time of auto recording.

Note

#1 User must set QCAP_SET_VIDEO_RECORD_PROPERTY and QCAP_SET_AUDIO_RECORD_PROPERTY first, when user set QCAP_START_CLONE_RECORD.

#2 dSegmentDurationTime parameter offer more variables for recording file path name as show in below:

$Y, $M, $D, $h, $m, $h, $i for start system time

%Y, %M, %D, %h, %m, %h, %i for stop system time

#3 the dSegmentDurationTime and nSegmentDurationSizeKB can’t be set at same time.

Example

```
QCAP_START_CLONE_RECORD( pDevice, 0, "C:\\CH01.AVI", "D:\\CH01.AVI"  );

str1.Format( "%sMAIN_CH%02d.$Y.$M.$D.$h.$m.$ss.$i.mp4", g_scmProtypeEx.pszRecordFilePathName, i );

str2.Format( "%sMAIN_CH%02d.$Y.$M.$D.$h.$m.$ss.$i.mp4", g_scmProtypeEx.pszRecordFilePathName, i );

    QCAP_START_CLONE_RECORD ( pDevice,
                              0,
                             (CHAR*)(LPCSTR)str1 ,
                             (CHAR*)(LPCSTR)str2 ,
                             QCAP_RECORD_FLAG_FULL,
                             0.0 /*dVideoDelayTime = 0.0*/,
                             0.0/*dAudioDelayTime = 0.0*/,
                             10 );
```

5.12 QCAP_START_RECORD

```
QRESULT QCAP_START_RECORD( PVOID     pDevice,
                           UINT      iRecNum,
                           CHAR *    pszFilePathName,
                           DWORD     dwFlags = QCAP_RECORD_FLAG_FULL,
                           double    dVideoDelayTime = 0.0,
                           double    dAudioDelayTime = 0.0,
                           double    dSegmentDurationTime = 0.0,
                           ULONG     nSegmentDurationSizeKB = 0 )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to start recording, start from 0. Max iRecNum is 0~3
pszFilePathName	**IN**, Specify the file name of storage for recording. Recording file format is AVI, if extension name is AVI. Recording file format is MP4, if extension name is MP4. Recording file format is ASF, if extension name is ASF. Recording file format is FLV, if extension name is FLV. Recording file format is TS, if extension name is TS. Recording file format is SCF, if extension name is SCF. Recording file format is WMV, if extension name is WMV. Recording file format is WAV, if extension name is WAV.
dwFlags	**IN**, Specify the action flag of share recording. This parameter can also include any combination of the following flags. Available value are QCAP_RECORD_FLAG_FULL = 0x00000007 QCAP_RECORD_FLAG_FILE = 0x00000001 QCAP_RECORD_FLAG_ENCODE = 0x00000002 QCAP_RECORD_FLAG_DISPLAY = 0x00000004 QCAP_RECORD_FLAG_DECODE = 0x00000008 QCAP_RECORD_FLAG_VIDEO_ONLY = 0x00000010 QCAP_RECORD_FLAG_AUDIO_ONLY = 0x00000020 QCAP_RECORD_FLAG_VIDEO_USE_IDEAL_TIMESTAMP= 0x00000040 QCAP_RECORD_FLAG_AUDIO_USE_IDEAL_TIMESTAMP= 0x00000080 QCAP_RECORD_FLAG_INGNORE_FORMAT_CHANGED = 0x00000100

dVideoDelayTime	**IN**, Specify video delay time, default is 0.0
dAudioDelayTime	**IN**, Specify audio delay time, default is 0.0
dSegmentDurationTime	**IN**, Specify segment duration time
nSegmentDurationSizeKB	**IN**, Specify segment duration size, the uint is K Byte

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start recording.

Users can use dVideoDelayTime& dAudioDelayTime parameters to set video or audio delay time for control play speed.

QCAP support the multiple-streams recording at the same time. It means you can generate many recording files with different resolution or bit rate for one capture device. The dwFlags parameter allows you to turn on/off internal components of recording.

User can use dSegmentDurationTime parameter to set segment duration time of auto recording.

Note

#1 User must set QCAP_SET_VIDEO_RECORD_PROPERTY and QCAP_SET_AUDIO_RECORD_PROPERTY first, when user set QCAP_START_RECORD .

#2 dSegmentDurationTime parameter offer more variables for recording file path name as show in below:

$Y, $M, $D, $h, $m, $h, $i for start system time

%Y, %M, %D, %h, %m, %h, %i for stop system time

#3 the dSegmentDurationTime and nSegmentDurationSizeKB can't be set at same time.

Example

```
QCAP_START_RECORD( pDevice, 0, "D:\\CH01.AVI" ); // @ 1920X1080 ( QCAP_DOWNSCALE_MODE_OFF )


str1.Format( "%sMAIN_CH%02d.$Y.$M.$D.$h.$m.$ss.$i.mp4", g_scmProtypeEx.pszRecordFilePathName, i );

              QCAP_START_RECORD( pDevice,
              0,
              (CHAR*)(LPCSTR)str1,
              QCAP_RECORD_FLAG_FULL,
              0.0 /*dVideoDelayTime = 0.0*/,
              0.0/*dAudioDelayTime = 0.0*/,
              10 );
```

5.13 QCAP_STOP_RECORD

```
QRESULT QCAP_STOP_RECORD(  PVOID  pDevice,
                           UINT   iRecNum,
                           BOOL   bIsAsync = TRUE,
                           ULONG  nMilliseconds = 0 )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to stop recording, start from 0. Max iRecNum is 0~3
bIsAsync	**IN**, Synchronized stop recording flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop recording.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the recording file is closed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate recording file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_STOP_RECORD( pDevice, 0 );
```

5.14 QCAP_RESUME_RECORD

```
QRESULT QCAP_RESUME_RECORD(   PVOID  pDevice,
                              UINT   iRecNum )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to resume recording, start from 0. Max iRecNum is 0~3

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function toresume recording.

Example

```
QCAP_RESUME_RECORD( pDevice, 0 );
```

5.15 QCAP_PAUSE_RECORD

```
QRESULT QCAP_PAUSE_RECORD( PVOID  pDevice,
                           UINT   iRecNum )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index to pause recording, start from 0. Max iRecNum is 0~3

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to pause recording.

Example

```
QCAP_PAUSE_RECORD( pDevice, 0 );
```

6 Synchronized Recording Function API

6.1 QCAP_START_SYNCHRONIZED_RECORD

```
QRESULT QCAP_START_SYNCHRONIZED_RECORD( ULONG     nFileArgs,
                                        CHAR   *  pszFilePathName1,
                                        PVOID     pDevice1,
                                        UINT      iRecNum1,
                                        CHAR   *  pszFilePathName2,
                                        PVOID     pDevice2,
                                        UINT      iRecNum2, … )
```

Parameter

Name	Description
nFileArgs	**IN**, Set the devices number
pszFilePathName1	**IN**, Specify the file name of storage for recording. Recording file format is AVI, if extension name is AVI. Recording file format is MP4, if extension name is MP4. Recording file format is ASF, if extension name is ASF. Recording file format is FLV, if extension name is FLV. Recording file format is TS, if extension name is TS. Recording file format is SCF, if extension name is SCF. Recording file format is WMV, if extension name is WMV. Recording file format is WAV, if extension name is WAV.

pDevice1, pDevice2, …	**IN**, Handle of the capture card object
iRecNum1, iRecNum2, …	**IN**, Specify recorder index to start synchronized recording

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start synchronized recording. At this function, QCAP will give unified sample time to start multiple recording.

In other words, QCAP will provide same start sample time for many recording. User can use nFileArgs parameter to set number of devices.

Note

AVI format cannot support synchronized recording.

Example

```
QCAP_START_SYNCHRONIZED_RECORD( 4,
                                "SR_CH01.MP4",pDevices[ 0 ],
                                "SR_CH02.MP4", pDevices[ 1 ], 0,
                                "SR_CH03.MP4", pDevices[ 2 ], 0,
                                "SR_CH04.MP4", pDevices[ 3 ], 0, NULL, NULL, 0, NULL, NULL, 0 );
```

6.2 QCAP_STOP_SYNCHRONIZED_RECORD

```
QRESULT QCAP_STOP_SYNCHRONIZED_RECORD(  BOOL   bIsAsync,
                                        ULONG  nMilliseconds,
                                        ULONG  nFileArgs,
                                        PVOID  pDevice1,
                                        UINT   iRecNum1,
                                        PVOID  pDevice2,
                                        UINT   iRecNum2, …)
```

Parameter

Name	Description
bIsAsync	**IN**, Synchronized stop recording flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.
nFileArgs	**IN**, Set the devices number
pDevice1, pDevice2, …	**IN**, Handle of the capture card object
iRecNum1, iRecNum2, …	**IN**, Specify recorder index to stop synchronized recording

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop synchronized recording.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the recording file is closed.

User can use nFileArgs parameter to set number of devices.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_STOP_SYNCHRONIZED_RECORD( TRUE, 0, 4 ,pDevices[ 0 ], 0,
                                           pDevices[ 1 ], 0,
                                           pDevices[ 2 ], 0,
                                           pDevices[ 3 ], 0, NULL, 0 );
```

6.3 QCAP_RESUME_SYNCHRONIZED_RECORD

```
QRESULT QCAP_RESUME_SYNCHRONIZED_RECORD(    ULONG  nFileArgs,
                                            PVOID  pDevice1,
                                            UINT   iRecNum1,
                                            PVOID  pDevice2,
                                            UINT   iRecNum2, … )
```

Parameter

Name	Description
nFileArgs	**IN**, Set the devices number
pDevice1, pDevice2, …	**IN**, Handle of the capture card object
iRecNum1, iRecNum2, …	**IN**, Specify recorder index to resume synchronized recording

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to resume synchronized recording.

User can use nFileArgs parameter to set number of devices.

Example

```
QCAP_RESUME_SYNCHRONIZED_RECORD( 4, pDevice, 0 );
```

6.4 QCAP_PAUSE_SYNCHRONIZED_RECORD

```
QRESULT QCAP_PAUSE_SYNCHRONIZED_RECORD( ULONG  nFileArgs,
                                        PVOID  pDevice1,
                                        UINT   iRecNum1,
                                        PVOID  pDevice2,
                                        UINT   iRecNum2, … )
```

Parameter

Name	Description
nFileArgs	**IN**, Set the devices number
pDevice1, pDevice2, …	**IN**, Handle of the capture card object
iRecNum1, iRecNum2, …	**IN**, Specify recorder index to pause synchronized recording

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to pause synchronized recording.

User can use nFileArgs parameter to set number of devices.

Example

```
QCAP_PAUSE_SYNCHRONIZED_RECORD( 4, pDevice, 0 );
```

7 OSD Function API

7.1 QCAP_GET_OSD_TEXT_BOUNDARY

```
QRESULT QCAP_GET_OSD_TEXT_BOUNDARY(  PVOID      pDevice,
                                     UINT       iOsdNum,
                                     CHAR   *   pszString,
                                     CHAR   *   pszFontFamilyName,
                                     ULONG      nFontStyle,
                                     ULONG      nFontSize,
                                     ULONG  *   pBoundaryWidth,
                                     ULONG  *   pBoundaryHeight )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 127
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output
nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are QCAP_FONT_STYLE_REGULAR = 0x00000000, QCAP_FONT_STYLE_BOLD = 0x00000001, QCAP_FONT_STYLE_ITALIC = 0x00000002, QCAP_FONT_STYLE_BOLDITALIC = 0x00000003, QCAP_FONT_STYLE_UNDERLINE = 0x00000004, QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,
nFontSize	**IN**, Specify the font size used to display the text of OSD output
pBoundaryWidth	**OUT**, Pointer to the storage of boundary width
pBoundaryHeight	**OUT**, Pointer to the storage of boundary height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get size of OSD string.

For example, user can use boundary information to set OSD string location on display window.

Example

```
QCAP_GET_OSD_TEXT_BOUNDARY( hHandle, 0, “CH01”, “Arial”,
                            QCAP_FONT_STYLE_BOLD, 12, pBoundaryWidth, pBoundaryHeight );
```

7.2 QCAP_SET_OSD_TEXT

```
QRESULT QCAP_SET_OSD_TEXT(    PVOID      pDevice,
                              UINT       iOsdNum,
                              INT        x,
                              INT        y,
                              INT        width,
                              INT        height,
                              CHAR   *   pszString,
                              CHAR   *   pszFontFamilyName,
                              ULONG      nFontStyle,
                              ULONG      nFontSize,
                              DWORD      dwFontColor,
                              DWORD      dwBackgroundColor,
                              ULONG      nTransparent,
                              INT        nTextStartPosX = 0,
                              INT        nTextStartPosY = 0,
                              ULONG      nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOS )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 127
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output, if width is -1, QCAP will auto calculate width.
height	**IN**, Specify the vertical height of OSD output, if height is -1, QCAP will auto calculate height.
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output

nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are
	`QCAP_FONT_STYLE_REGULAR = 0x00000000,`
	`QCAP_FONT_STYLE_BOLD = 0x00000001,`
	`QCAP_FONT_STYLE_ITALIC = 0x00000002,`
	`QCAP_FONT_STYLE_BOLDITALIC = 0x00000003,`
	`QCAP_FONT_STYLE_UNDERLINE = 0x00000004,`
	`QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,`

nFontSize	**IN**, Specify the font size used to display the text of OSD output
dwFontColor	**IN**, Specify the font color used to display the text of OSD output
dwBackgroundColor	**IN**, Specify the background color used to display the text of OSD output

nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nTextStartPosX	**IN**, Specify the text scrolling start position X of the upper-left corner of OSD text
nTextStartPosY	**IN**, Specify the text scrolling start position Y of the upper-left corner of OSD text
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a text field objects used for onscreen display of information.

If you want to get a text field object auto calculated actual width and height value, you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Note

Color space of FontColoris ABGR, don’t forgot to set its alpha value.

Color space of BackgroundColor is ABGR, don’t forgot to set its alpha value.

Transparent is one global parameter for all colors.

Example

```
QCAP_SET_OSD_TEXT( hHandle, 0, 0, 0, -1, -1, “CH01”, “Arial”,
                   QCAP_FONT_STYLE_BOLD, 12, 0xFFFF0000,
                   0xFFFFFFFF, 128, 0, 0, QCAP_SEQUENCE_STYLE_FOREMOST );
```

7.3 QCAP_SET_OSD_PICTURE

```
QRESULT QCAP_SET_OSD_PICTURE( PVOID     pDevice,
                              UINT      iOsdNum,
                              INT       x,
                              INT       y,
                              INT       width,
                              INT       height,
                              CHAR *    pszFilePathName,
                              ULONG     nTransparent,
                              ULONG     nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 127
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify they-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output If width is -1, QCAP will use original picture width as default.
height	**IN**, Specify the vertical height of OSD output. If height is -1, QCAP will use original picture height as default.
pszFilePathName	**IN**, Specify the file name to display image OSD output, support a file extension of "BMP" as 24 or 32-bit,“JPG” and “PNG”
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create static image objects used for onscreen display of information.

If you want to get a static image object auto calculated actual width and height value,

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_PICTURE( hHandle, 0, 0, 0, -1, -1, "C:\\SAMPLE.PNG", 128, QCAP_SEQUENCE_STYLE_FOREMOST );
```

7.4 QCAP_SET_OSD_BUFFER

```
QRESULT QCAP_SET_OSD_BUFFER(      PVOID     pDevice,
                                  UINT      iOsdNum,
                                  INT       x,
                                  INT       y,
                                  INT       width,
                                  INT       height,
                                  ULONG     nColorSpaceType,
                                  BYTE *    pFrameBuffer,
                                  ULONG     nFrameWidth,
                                  ULONG     nFrameHeight,
                                  ULONG     nFramePitch,
                                  ULONG     nTransparent,
                                  DWORD     dwKeyColor = 0xFFFFFFFF,
                                  ULONG     nKeyColorThreshold = 25,
                                  ULONG     nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 127
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify they-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output
height	**IN**, Specify the vertical height of OSD output
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameWidth	**IN**, Specify the horizontal width of frame contained in a buffer
nFrameHeight	**IN**, Specify the vertical height of frame contained in a buffer
nFramePitch	**IN**, Specify the distance in bytes between the start of one scan line to the next. If it is 0, we will auto calculate it by width and color space format.

nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
dwKeyColor	**IN**, Specify the key color of OSD, default is0xFFFFFFFF
nKeyColorThreshold	**IN**, Specify the threshold of key color, the range of 0 to 128
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

* The performance of BGR type is slower than RGB in our alpha blending algorithm.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a frame buffer objects used for onscreen display of information.

The function’s purpose is used to point directly to a valid frame buffer whose entire pixel data, which is stored in available color space, and display it.

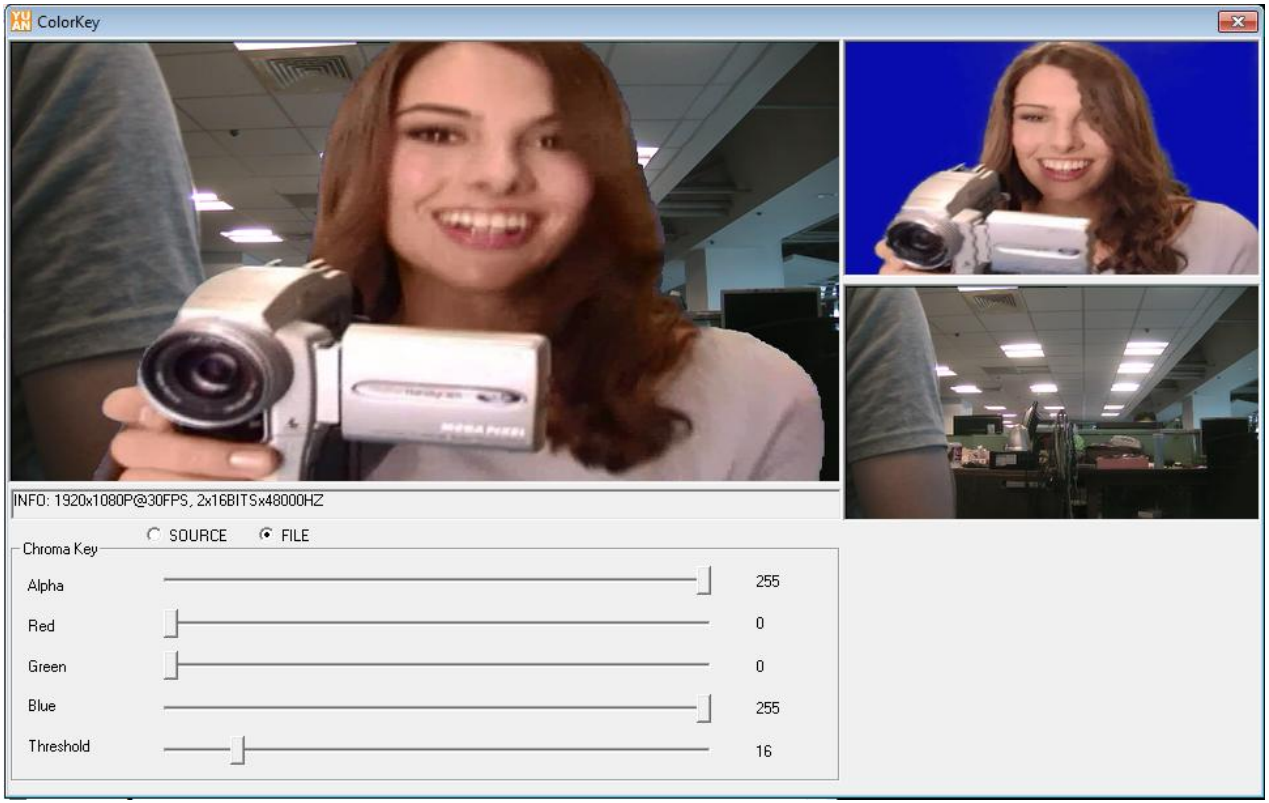
Beside, you also can adjust the width and height parameters to scale the OSD output

And OSD sequence style to video-streams video-recording the OSD output.

Note

User can use this dwKeyColor parament to select colorkey mode Green&Blue mode and RGB mode.

For instance the Blue mode way as show in below.



Example

```
QCAP_SET_OSD_BUFFER( hHandle, 0, 0, 0, 1920, 1080,
                     QCAP_COLORSPACE_TYEP_YUY2, pFrameBuffer, 800, 600, 0, 128,
                     0xFFFFFFFF, 25, QCAP_SEQUENCE_STYLE_FOREMOST );
```

8 Clone Function API

8.1 QCAP_CREATE_CLONE

```
QRESULT QCAP_CREATE_CLONE( PVOID     pDevice,
                           HWND      hAttachedWindow,
                           PVOID *   ppCloneDevice,
                           BOOL      bThumbDraw = FALSE,
                           BOOL      bMaintainAspectRatio = FALSE )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
hAttachedWindow	**IN**, Handle of the window to show the preview clone video
ppCloneDevice	**OUT**, Handle of the clone capture device
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer
bMaintainAspectRatio	**IN**, Enable/Disable the Maintain aspect ratio

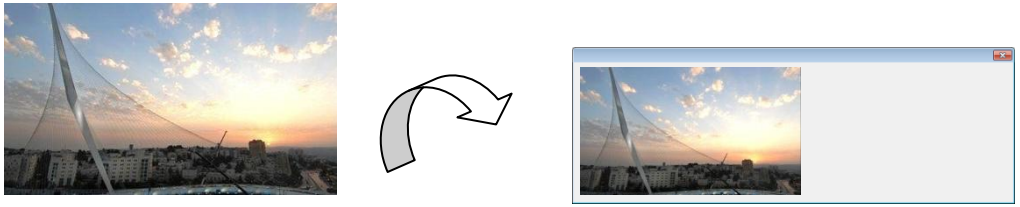
Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a clone device from current hardware device object.

The purpose of clone function is to help user to create multiple preview windows from one hardware capture device.



Physical Device Display Clone Display on Other Window

Developer also can use the clone device to access device's property function, such as QCAP_SET_VIDEO_INPUT.

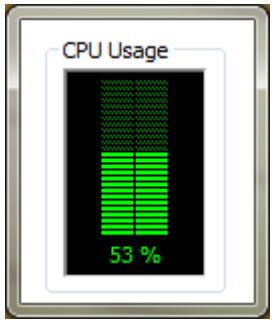
QCAP will be auto mapping to its physical device object. However, every clone device also can access own independently functions.

These independent functions are listed as below:

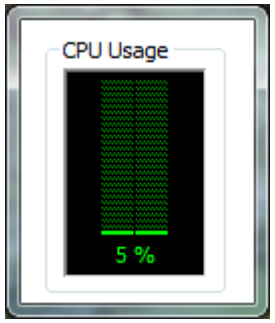
Related Clone API Functions
QCAP_CREATE_CLONE
QCAP_RUN
QCAP_STOP
QCAP_DESTROY
QCAP_GET_VOLUME / QCAP_SET_VOLUME

The thumb draw function can reduce the loading of video display, when display window size is small than the input video size.

For multiple channels developer, we suggest to turn on it in order to obtain the best rendering performance.



Normal Display



Thumb Draw Mode

Programming Workflow of Clone Device

Programming Step	Related API Functions
Create Device	`QCAP_CREATE( …, &pDevice )`
Start Capturing	`QCAP_RUN( pDevice )`
Create Clone Device	`QCAP_CREATE_CLONE(pDevice, …, &pClone, … )`
Start Clone Preview	`QCAP_RUN( pClone )`
Stop Clone Preview	`QCAP_STOP( pClone )`
Delete Clone Device	`QCAP_DESTROY( pClone )`
Stop Capturing	`QCAP_STOP( pDevice )`
Delete Device	`QCAP_DESTROY( pDevice )`

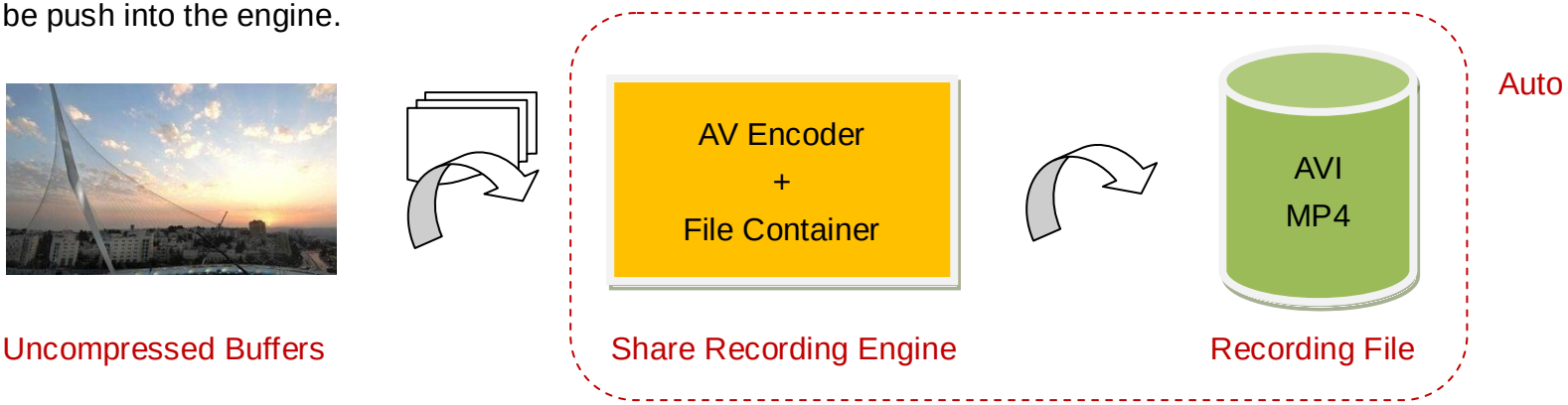
Example

```
QCAP_CREATE_CLONE( pDevice, m_hWnd, &pClone, FALSE, FALSE );
```

9 Share Recording Function API

9.1 Overview

The function of share recording can help you to create one customized recording fileeasily. It is different from channel recording at Chapter 5, the software developer can freedom combine any audio and video frame buffers into one recording engine. The engine will hide all complex actions for stream encoding and file saving. For one software developer, you just need to decide which frame buffer will be push into the engine.



For instance, we can combine CH1 and CH2 video into one H.264's MP4 file. The first 5min video is come from CH1 and remains are CH2.



9.2 QCAP_GET_VIDEO_SHARE_RECORD_PROPERTY

```
QRESULT QCAP_GET_VIDEO_SHARE_RECORD_PROPERTY(  UINT       iRecNum,
                                               ULONG  *   pEncoderType,
                                               ULONG  *   pEncoderFormat,
                                               ULONG  *   pColorSpaceType,
                                               ULONG  *   pWidth,
                                               ULONG  *   pHeight,
                                               double *   pFramate,
                                               ULONG  *   pRecordMode,
                                               ULONG  *   pQuality,
                                               ULONG  *   pBitRate,
                                               ULONG  *   pGOP,
                                               ULONG  *   pAspectRatioX,
                                               ULONG  *   pAspectRatioY,
                                               HWND *     pAttachedWindow = NULL,
                                               BOOL *     pThumbDraw = NULL,
                                               BOOL *     pMaintainAspectRatio = NULL )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0 Max Recnum is 0~31
pEncoderType	**OUT**, Pointer to the storage of current recording encoder type
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pColorSpaceType	**OUT**, Pointer to the storage of current recording encoder color space type
pWidth	**OUT**, Pointer to the storage of current recording encoder width
pHeight	**OUT**, Pointer to the storage of current recording encoder height
pFrameRate	**OUT**, Pointer to the storage of current recording encoder frame rate
pRecordMode	**OUT**, Pointer to the storage of current recording mode
pQuality	**OUT**, Pointer to the storage of current recording quality
pBitRate	**OUT**, Pointer to the storage of current recording bit rate
pGOP	**OUT**, Pointer to the storage of current recording GOP size
pAspectRatioX	**OUT**, Pointer to the storage of aspect ratio X axle
pAspectRatioY	**OUT**, Pointer to the storage of aspect ratio Y axle
pAttachedWindow	**OUT**, Pointer to the storage of window to show the recording video
pThumbDraw	**OUT**, Pointer to the enable flag of thumb draw renderer
pMaintainAspectRatio	**OUT**, Pointer to the enable flag of maintain aspect ratio

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get video share recording parameters.

Example

```
QCAP_GET_VIDEO_SHARE_RECORD_PROPERTY( 0, &nEncoderType, &nEncoderFormat,
                                      &nColorSpaceType, &nWidth, &nHeight, &nFrameRate,
                                      &nRecordMode, &nQuality, &nBitRate, &nGOP,
                                      &pAspectRatioX, &npAspectRatioY, &hAttachedWindow,
                                      &bThumbDraw, &MaintainAspectRatio );
```

9.3 QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY

```
QRESULT QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY(  UINT      iRecNum,
                                               ULONG     nEncoderType,
                                               ULONG     nEncoderFormat,
                                               ULONG     nColorSpaceType,
                                               ULONG     nWidth,
                                               ULONG     nHeight,
                                               double    dFrameRate,
                                               ULONG     nRecordMode,
                                               ULONG     nQuality,
                                               ULONG     nBitRate,
                                               ULONG     nGOP,
                                               ULONG     nAspectRatioX,
                                               ULONG     nAspectRatioY,
                                               HWND      hAttchedWindow= NULL,
                                               BOOL      bThumbDraw= FALSE,
                                               BOOL      bMaintainAspectRatio = FALSE )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0 Max Recnum is 0~31
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_I420
nWidth	**IN**, Specify encoder width.
nHeight	**IN**, Specify encoder height.

dFrameRate	**IN**, Specify encoder frame rate.

nRecordMode	**IN**, Specify recording mode. Available values are
	`QCAP_RECORD_MODE_VBR = 0,`
	`QCAP_RECORD_MODE_CBR = 1,`
	`QCAP_RECORD_MODE_ABR = 2,`

nQuality	**IN**, Specify recording quality, from 0~10000. It is used for VBR and ABR.
nBitRate	**IN**, Specify recording bit rate. It is used for CBR and ABR.(Ex. 12Mbps is 12 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0~255
nAspectRatioX	**IN**, Specify the aspect ratio X axle, 0 is turned off
nAspectRatioY	**IN**, Specify the aspect ratio Y axle, 0 is turned off
hAttachedWindow	**IN**, Handle of the window to show the preview video
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer
bMaintainAspectRatio	**IN**, Enable/Disable the maintain aspect ratio

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource

QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

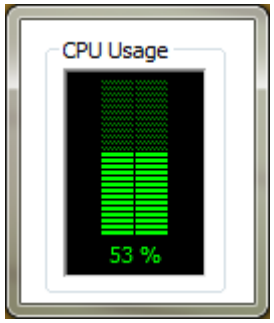
Introduction

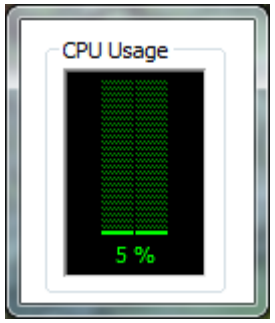
User can use this function to set video recording parameters.

By using the attached window, QCAP can help you to real-time display the share recording’s result.

The thumb draw function can reduce the loading of video display, when display window size is small than the input video size.

For multiple channels developer, we suggest to turn on it in order to obtain the best rendering performance.





Normal Display Thumb Draw Mode

Note

The encode type of this function is only support on SOFTWARE, INTEL and NVIDIA, so nEncoderType can not be QCAP_ENCODER_TYPE_HARDWARE, if user wants to obtained the bitstream from hardware compression card, should get it in QCAP_REGISTER_VIDEO_HARDWARE_ENCODER_CALLBACK.

Example

```
QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY(0, QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK, QCAP_ENCODER_FORMAT_H264,
                                     QCAP_COLORSPACE_TYEP_YV12, 1280, 720, 60, QCAP_RECORD_MODE_CBR, 8000,
                                     12*1024* 1024, 30, 4, 3, hAttachedWindow, FALSE, FALSE );
```

9.4 QCAP_GET_VIDEO_SHARE_RECORD_PROPERTY_EX

```
QRESULT QCAP_GET_VIDEO_SHARE_RECORD_PROPERTY_EX(  UINT       iRecNum,
                                                  ULONG  *   pEncoderType,
                                                  ULONG  *   pEncoderFormat,
                                                  ULONG  *   pColorSpaceType,
                                                  ULONG  *   pWidth,
                                                  ULONG  *   pHeight,
                                                  double *   pFrameRate,
                                                  ULONG  *   pRecordProfile,
                                                  ULONG  *   pRecordLevel,
                                                  ULONG  *   pRecordEntropy,
                                                  ULONG  *   pRecordComplexity,
                                                  ULONG  *   pRecordMode,
                                                  ULONG  *   pQuality,
                                                  ULONG  *   pBitRate,
                                                  ULONG  *   pGOP,
                                                  ULONG  *   pBFrames,
                                                  BOOL   *   pIsInterleaved,
                                                  ULONG  *   pSlices,
                                                  ULONG  *   pLayers,
                                                  ULONG  *   pSceneCut,
                                                  ULONG  *   pAspectRatioX,
                                                  ULONG  *   pAspectRatioY,
```

```
HWND    *   pAttachedWindow = NULL,
BOOL    *   pThumbDraw = NULL,
BOOL    *   pMaintainAspectRatio = NULL )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0 Max Recnum is 0~31
pEncoderType	**OUT**, Pointer to the storage of current recording encoder type
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pColorSpaceType	**OUT**, Pointer to the storage of current recording encoder color space type
pWidth	**OUT**, Pointer to the storage of current recording encoder width
pHeight	**OUT**, Pointer to the storage of current recording encoder height
dFrameRate	**OUT**, Pointer to the storage of current recording encoder frame rate
pRecordProfile	**OUT**, Pointer to the storage of current recording profile
pRecordLevel	**OUT**, Pointer to the storage of current recording level
pRecordEntropy	**OUT**, Pointer to the storage of current recording entropy
pRecordComplexity	**OUT**, Pointer to the storage of current recording complexity
pRecordMode	**OUT**, Pointer to the storage of current recording mode
pQuality	**OUT**, Pointer to the storage of current recording quality
pBitRate	**OUT**, Pointer to the storage of current recording bit rate

pGOP	**OUT**, Pointer to the storage of current recording GOP size
pBFrames	**OUT**, Pointer to the storage of current recording B frames
pIsInterleaved	**OUT**, Pointer to the storage of current recording Interleaved
pSlices	**OUT**, Pointer to the storage of current recording slices
pLayers	**OUT**, Pointer to the storage of current recording layers
pSceneCut	**OUT**, Pointer to the storage of current recording screen cut
pAspectRatioX	**OUT**, Pointer to the storage of aspect ratio X axle
pAspectRatioY	**OUT**, Pointer to the storage of aspect ratio Y axle
pAttachedWindow	**OUT**, Pointer to the storage of window to show the recording video
pThumbDraw	**OUT**, Pointer to the enable flag of thumb draw renderer
pMaintainAspectRatio	**OUT**, Pointer to the enable flag of maintain aspect ratio

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device

QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get video share recording parameters or even parameters from H.264 system such as Profile, Level, Entropy Complexity, B Frames and SceneCut.

Example

```
QCAP_GET_VIDEO_SHARE_RECORD_PROPERTY_EX( 0, &nEncoderType, &nEncoderFormat,
                                         &nColorSpaceType, &nWidth, &nHeight, &nFrameRate,
                                         &nRecordProfile,  &nRecordLevel, &nRecordEntropy,
                                         &nRecordComplexity, &nRecordMode, &nQuality, &nBitRate,
                                         &nGOP, &nBframe, &nIsInterleaved, &nSlices, &nLayers
                                         &nSceneCut, &nAspectRatioX, &npAspectRatioY, &hAttachedWindow,
                                         &nbThumbDraw, &nMaintainAspectRatio );
```

9.5 QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY_EX

```
QRESULT QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY_EX(  UINT    iRecNum,
                                                  ULONG   nEncoderType,
                                                  ULONG   nEncoderFormat,
                                                  ULONG   nColorSpaceType,
                                                  ULONG   nWidth,
                                                  ULONG   nHeight,
                                                  double  dFrameRate,
                                                  ULONG   nRecordProfile,
                                                  ULONG   nRecordLevel,
                                                  ULONG   nRecordEntropy,
                                                  ULONG   nRecordComplexity,
                                                  ULONG   nRecordMode,
                                                  ULONG   nQuality,
                                                  ULONG   nBitRate,
                                                  ULONG   nGOP,
                                                  ULONG   nBFrame,
                                                  BOOL    bIsInterleaved,
                                                  ULONG   nSlices,
                                                  ULONG   nLayers,
                                                  ULONG   nSceneCut,
                                                  ULONG   nAspectRatioX,
                                                  ULONG   nAspectRatioY,
```

```
HWND    hAttchedWindow = NULL,
BOOL    bThumbDraw = FALSE,
BOOL    bMaintainAspectRatio = FALSE )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0 Max Recnum is 0~31
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_YV1

	QCAP_COLORSPACE_TYEP_I420
nWidth	**IN**, Specify encoder width.
nHeight	**IN**, Specify encoder height.
dFrameRate	**IN**, Specify encoder frame rate.

nRecordProfile	**IN**, Specify recording profile
	QCAP_RECORD_PROFILE_BASELINE = 0, // DEFAULT
	QCAP_RECORD_PROFILE_MAIN = 1,
	QCAP_RECORD_PROFILE_HIGH = 2,

nRecordLevel	**IN**, Specify recording level QCAP_RECORD_LEVEL_1 = 10, QCAP_RECORD_LEVEL_1B = 9, QCAP_RECORD_LEVEL_11 = 11, QCAP_RECORD_LEVEL_12 = 12, QCAP_RECORD_LEVEL_13 = 13, QCAP_RECORD_LEVEL_2 = 20, QCAP_RECORD_LEVEL_21 = 21, QCAP_RECORD_LEVEL_22 = 22, QCAP_RECORD_LEVEL_3 = 30, QCAP_RECORD_LEVEL_31 = 31, QCAP_RECORD_LEVEL_32 = 32, QCAP_RECORD_LEVEL_4 = 40, QCAP_RECORD_LEVEL_41 = 41, // DEFAULT

	QCAP_RECORD_LEVEL_42 = 42, QCAP_RECORD_LEVEL_50 = 50, QCAP_RECORD_LEVEL_51 = 51, QCAP_RECORD_LEVEL_52 = 52,

nRecordEntropy	**IN**, Specify recording entropy
	QCAP_RECORD_ENTROPY_CAVLC = 0, // DEFAULT
	QCAP_RECORD_ENTROPY_CABAC = 1,

nRecordComplexity	**IN**, Specify recording complexity
	QCAP_RECORD_COMPLEXITY_0 = 0, // BEST SPEED, DEFAULT
	QCAP_RECORD_COMPLEXITY_1 = 1,
	QCAP_RECORD_COMPLEXITY_2 = 2,
	QCAP_RECORD_COMPLEXITY_3 = 3,
	QCAP_RECORD_COMPLEXITY_4 = 4,
	QCAP_RECORD_COMPLEXITY_5 = 5,
	QCAP_RECORD_COMPLEXITY_6 = 6, // BEST QUALITY

nRecordMode	**IN**, Specify recording mode. Available values are QCAP_RECORD_MODE_VBR = 0 QCAP_RECORD_MODE_CBR = 1 QCAP_RECORD_MODE_ABR = 2
nQuality	**IN**, Specify recording quality, from 0~10000. It is used for VBR and ABR.
nBitRate	**IN**, Specify recording bit rate. It is used for CBR and ABR. (Ex. 12Mbps is 12 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0~255
nBFrames	**IN**, Specify recording B frame

bIsInterleaved	**IN**, Enable/Disable the Interleaved
nSlices	**IN**, Specify recording slices, default is 0
nLayers	**IN**, Specify recording layers, default is 0
nSceneCut	**IN**, Specify recording Scene Cut, recommedned value is 40， 0 is function turned off
nAspectRatioX	**IN**, Specify the aspect ratio X axle, 0 is turned off
nAspectRatioY	**IN**, Specify the aspect ratio Y axle, 0 is turned off
hAttachedWindow	**IN**, Handle of the window to show the preview video
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer
bMaintainAspectRatio	**IN**, Enable/Disable the maintain aspect ratio

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support

QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

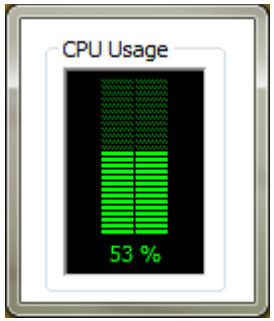
Introduction

User can use this function to set video share recording parameters or even parameters from H.264 system such as Profile, Level, Entropy Complexity, B Frames and SceneCut.

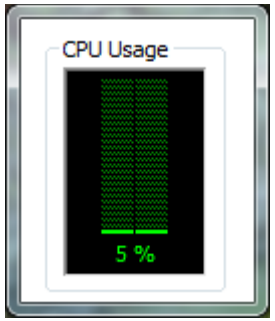
By using the attached window, QCAP can help you to real-time display the share recording's result.

The thumb draw function can reduce the loading of video display, when display window size is small than the input video size.

For multiple channels developer, we suggest to turn on it in order to obtain the best rendering performance.



Normal Display



Thumb Draw Mode



Example

```
QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY_EX( 0, QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK, QCAP_ENCODER_FORMAT_H264,
                                         QCAP_COLORSPACE_TYEP_YUY2, 1280, 720, 60, nRecordProfile,
                                         nRecordLevel, nRecordEntropy, QCAP_RECORD_COMPLEXITY_1,
                                         QCAP_RECORD_MODE_CBR, 8000, 12*1024*1024, 30, 0, FALUSE, 0, 0, 40,
                                         4, 3, hAttachedWindow, FALSE, FALSE);
```

9.6 QCAP_GET_AUDIO_SHARE_RECORD_PROPERTY

```
QRESULT QCAP_GET_AUDIO_SHARE_RECORD_PROPERTY(  UINT      iRecNum,
                                               ULONG *   pEncoderType,
                                               ULONG *   pEncoderFormat,
                                               ULONG *   pChannels,
                                               ULONG *   pBitsPerSample,
                                               ULONG *   pSampleFrequency,
                                               ULONG *   pVolume )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0 Max Recnum is 0~31
pEncoderType	**OUT**, Pointer to the storage of current recording encoder type
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pChannels	**OUT**, Pointer to the storage of current recording encoder channels
pBitsPerSample	**OUT**, Pointer to the storage of current recording encoder bits per sample
pSampleFrequency	**OUT**, Pointer to the storage of current recording encoder sample frequency
pVolume	**OUT**, Pointer to the storage of current audio renderer's volume

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get audio share recording parameters.

Example

```
QCAP_GET_AUDIO_SHARE_RECORD_PROPERTY( 0, &nEncoderType, &nEncoderFormat,
                                      &nChannels, &nBitsPerSample, &nSampleFrequency, &nVolume );
```

9.7 QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY

```
QRESULT QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY(  UINT    iRecNum,
                                               ULONG   nEncoderType,
                                               ULONG   nEncoderFormat,
                                               ULONG   nChannels,
                                               ULONG   nBitsPerSample,
                                               ULONG   nSampleFrequency,
                                               ULONG   nVolume )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0 Max Recnum is 0~31

nEncoderType	**IN**, Specify encoder type. Available value are
	`QCAP_ENCODER_TYPE_SOFTWARE = 0`
	`QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590`
	`QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2`
	`QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4`

nEncoderFormat	**IN**, Specify encoder format. Available value are
	`QCAP_ENCODER_FORMAT_PCM = 0`
	`QCAP_ENCODER_FORMAT_AAC = 1`
	`QCAP_ENCODER_FORMAT_AAC_RAW = 1`
	`QCAP_ENCODER_FORMAT_AAC_ADTS = 2`

nChannels	**IN**, Specify encoder channels.
nBitsPerSample	**IN**, Specify encoder bits per sample.
nSampleFrequency	**IN**, Specify encoder sample frequency.
nVolume	**IN**, Specify the volume of audio renderer, from 0~ 100

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio share recording parameters.

For MP4 file recording, we suggest to use H.264 + AAC encoder. It is the most popular file format for Microsoft Windows 7 as default.

If customer doesn’t want to license AAC, the QCAP SDK also support H.264 + PCM file format for your software.

Example

```
QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY( 0, QCAP_ENCODER_TYPE_SOFTWARE, QCAP_ENCODER_FORMAT_AAC,
                                      2, 16, 48000, 50 );
```

About AAC License

This software development kit contains software programming code and libraries related to AAC function that is subject to your development use only. All the intellectual properties are held by individual license groups and/or the property holders. License and royalty fee should be charged by the license groups and/or property holders and paid by you if any of them is implemented into your commercial product. We will not be responsible for any illegal usage and/or any infringement of the rights of the individual owners of these intellectual properties.

Details of AAC could be found in the website:

http://www.vialicensing.com/licensing/aac-overview.aspx

9.8 QCAP_GET_AUDIO_SHARE_RECORD_PROPERTY_EX

```
QRESULT QCAP_GET_AUDIO_SHARE_RECORD_PROPERTY_EX(  UINT      iRecNum,
                                                  ULONG  *  pEncoderType,
                                                  ULONG  *  pEncoderFormat,
                                                  ULONG  *  pChannels,
                                                  ULONG  *  pBitsPerSample,
                                                  ULONG  *  pSampleFrequency,
                                                  ULONG  *  pBitRate,
                                                  ULONG  *  pVolume )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0 Max Recnum is 0~31
pEncoderType	**OUT**, Pointer to the storage of current recording encoder type
pEncoderFormat	**OUT**, Pointer to the storage of current recording encoder format
pChannels	**OUT**, Pointer to the storage of current recording encoder channels
pBitsPerSample	**OUT**, Pointer to the storage of current recording encoder bits per sample
pSampleFrequency	**OUT**, Pointer to the storage of current recording encoder sample frequency
pBitRate	**OUT**, Pointer to the storage of current recording encoder bit rate
pVolume	**OUT**, Pointer to the storage of current audio renderer's volume

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get extra audio share recording parameters or even parameters such as bitrate.

Example

```
QCAP_GET_AUDIO_SHARE_RECORD_PROPERTY_EX( 0, &nEncoderType, &nEncoderFormat,
                                         &nChannels, &nBitsPerSample, &nSampleFrequency,
                                         &nBitRate, &nVolume );
```

9.9 QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY_EX

```
QRESULT QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY_EX(  UINT    iRecNum,
                                                  ULONG   nEncoderType,
                                                  ULONG   nEncoderFormat,
                                                  ULONG   nChannels,
                                                  ULONG   nBitsPerSample,
                                                  ULONG   nSampleFrequency,
                                                  ULONG   nBitRate,
                                                  ULONG   nVolume )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0 Max Recnum is 0~31
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_PCM = 0 QCAP_ENCODER_FORMAT_AAC = 1 QCAP_ENCODER_FORMAT_AAC_RAW = 1 QCAP_ENCODER_FORMAT_AAC_ADTS = 2
nChannels	**IN**, Specify encoder channels.

nBitsPerSample	**IN**, Specify encoder bits per sample.
nSampleFrequency	**IN**, Specify encoder sample frequency.
nBitRate	**IN**, Specify audio bit rate. Default is 128k and max is 496k
nVolume	**IN**, Specify the volume of audio renderer, from 0~ 100

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set extra audio share recording parameters.

For MP4 file recording, we suggest to use H.264 + AAC encoder. It is the most popular file format for Microsoft Windows 7 as default.

If customer doesn't want to license AAC, the QCAP SDK also support H.264 + PCM file format for your software.

Example

```
QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY_EX( 0, QCAP_ENCODER_TYPE_SOFTWARE, QCAP_ENCODER_FORMAT_AAC,
                                         2, 16, 48000, 128*512, 50 );
```

About AAC License

This software development kit contains software programming code and libraries related to AAC function that is subject to your development use only. All the intellectual properties are held by individual license groups and/or the property holders. License and royalty fee should be charged by the license groups and/or property holders and paid by you if any of them is implemented into your commercial product. We will not be responsible for any illegal usage and/or any infringement of the rights of the individual owners of these intellectual properties.

Details of AAC could be found in the website:

http://www.vialicensing.com/licensing/aac-overview.aspx

9.10 QCAP_GET_VIDEO_SHARE_RECORD_SOFTWARE_ENCODER_PROPERTY

```
QRESULT QCAP_GET_VIDEO_SHARE_RECORD_SOFTWARE_ENCODER_PROPERTY (UINT      iRecNum,
                                                               ULONG  *  pRecordMode,
                                                               ULONG  *  pQuality,
                                                               ULONG  *  pBitRate,
                                                               ULONG  *  pGOP )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0 Max Recnum is 0~31
pRecordMode	**OUT**, Pointer to the storage of the record mode for the i-th video streamsession
pQuality	**OUT**, Pointer to the storage of the quality for the i-th video streamsession
pBitRate	**OUT**, Pointer to the storage of the bit rate for the i-th video streamsession
pGOP	**OUT**, Pointer to the storage of the GOP size for the i-th video streamsession

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get video share recording software encoder parameters.

User can use this function to get dynamic bitrate & gop reconfiguration for all software encoders.

Example

```
QCAP_GET_VIDEO_SHARE_RECORD_SOFTWARE_ENCODER_PROPERTY(   0, &nRecordMode,
                                                         &nQuality, &nBitRate, &nGOP );
```

9.11 QCAP_SET_VIDEO_SHARE_RECORD_SOFTWARE_ENCODER_PROPERTY

```
QRESULT QCAP_SET_VIDEO_SHARE_RECORD_SOFTWARE_ENCODER_PROPERTY( UINT      iRecNum,
                                                               ULONG     nRecordMode,
                                                               ULONG     nQuality,
                                                               ULONG     nBitRate,
                                                               ULONG     nGOP )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set recording parameters, start from 0 Max Recnum is 0~31

nRecordMode	**IN**, Specify recording mode. Available values are
	`QCAP_RECORD_MODE_VBR = 0`
	`QCAP_RECORD_MODE_CBR = 1`
	`QCAP_RECORD_MODE_ABR = 2`

nQuality	**IN**, Specify video quality. It is used by VBR and ABR.
nBitRate	**IN**, Specify video bit rate. It is used by CBR and ABR. (Ex. 6Mbps is 6 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0 ~ 255

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set video share recording software encoder parameters.

User can use this function to set dynamic bitrate & gop reconfiguration for all software encoders.

Example

```
QCAP_SET_VIDEO_SHARE_RECORD_SOFTWARE_ENCODER_PROPERTY(   0, QCAP_RECORD_MODE_CBR,
                                                         8000, 12 * 1024 * 1024, 30 );
```

9.12 QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER( UINT     iRecNum ,
                                                          ULONG    nColorSpaceType,
                                                          ULONG    nWidth,
                                                          ULONG    nHeight,
                                                          BYTE *   pFrameBuffer,
                                                          ULONG    nFrameBufferLen,
                                                          double   dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nColorSpaceType	**IN**, Specify the input source buffer's color space type, Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify the input source buffer's width
nHeight	**IN**, Specify the input source buffer's height
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

* The performance of RGB type is slower than BGR in our color converter algorithm.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression video buffer into share recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

If the width or the height of source frame are different from the target frame size.

QCAP will enable one auto video scaler to resize it before encoding.

Example

```
QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                  pFrameBuffer, 1280 * 720 * 2 );
```

9.13 QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER_EX(    UINT      iRecNum ,
                                                                ULONG     nColorSpaceType,
                                                                ULONG     nWidth,
                                                                ULONG     nHeight,
                                                                BYTE *    pFrameBuffer,
                                                                ULONG     nFrameBufferLen,
                                                                ULONG     nCropX,
                                                                ULONG     nCropY,
                                                                ULONG     nCropW,
                                                                ULONG     nCropH,
                                                                BOOL      bForceKeyFrame = FALSE,
                                                                double    dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nColorSpaceType	**IN**, Specify the input source buffer's color space type, Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify the input source buffer's width
nHeight	**IN**, Specify the input source buffer's height
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
nCropX	**IN**, Specify the input source buffer's width
nCropY	**IN**, Specify the input source buffer's height
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
bForceKeyFrame	**IN**, Enforce the input source's frame is key-frame, the default is FALSE

dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression video buffer and set crop parameters into share recorder.

User can use bForceKeyFrame parameter to push key-frame immediately.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

If the width or the height of source frame are different from the target frame size.

QCAP will enable one auto video scaler to resize it before encoding.

Example

```
QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER_EX( 0, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                     pFrameBuffer, 1280 * 720 * 2, 0, 0, 720, 480, FALSE );
```

9.14 QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER( UINT     iRecNum,
                                                          BYTE *   pFrameBuffer,
                                                          ULONG    nFrameBufferLen,
                                                          double   dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression audio buffer into share recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

Example

```
QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, pFrameBuffer, 4096 );
```

9.15 QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER_EX(    UINT      iRecNum,
                                                                ULONG     nChannels,
                                                                ULONG     nBitsPerSample,
                                                                ULONG     nSampleFrequency,
                                                                BYTE *    pFrameBuffer,
                                                                ULONG     nFrameBufferLen,
                                                                double    dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nChannels	**IN**, Specify encoder channels for the i-th audio stream session
nBitsPerSample	**IN**, Specify encoder bits per sample for the i-th audio stream session
nSampleFrequency	**IN**, Specify encoder sample frequency for the i-th audio stream session
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This is audio help function user can use this to resamle audio format.

For example, 2Ch 16Bit 48000HZ transform to 1Ch 8Bit 48000HZ.

This function is setting catcher input audio sample format, not resample audio format.

Example

```
QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER_EX( 0, 2, 16, 48000, pFrameBuffer, nFrameBufferLen );
```

9.16 QCAP_SET_VIDEO_SHARE_RECORD_COMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_SHARE_RECORD_COMPRESSION_BUFFER(  UINT      iRecNum ,
                                                         BYTE *    pStreamBuffer,
                                                         ULONG     nStreamBufferLen,
                                                         ULONG     bIsKeyFrame,
                                                         double    dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
pStreamBuffer	**IN**, Specify the input source buffer
nStreamBufferLen	**IN**, Specify the input source buffer's size
bIsKeyFrame	**IN**, Specify the input source's frame whether it is key-frame or not
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push compression stream buffer directly into share recorder.

The function will not enable encoder.

Example

```
QCAP_SET_VIDEO_SHARE_RECORD_COMPRESSION_BUFFER( 0, pStreamBuffer, 4096, TRUE );
```

9.17 QCAP_SET_AUDIO_SHARE_RECORD_COMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_SHARE_RECORD_COMPRESSION_BUFFER(  UINT      iRecNum ,
                                                         BYTE *    pStreamBuffer,
                                                         ULONG     nStreamBufferLen,
                                                         double    dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
pStreamBuffer	**IN**, Specify the input source buffer
nStreamBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push compression stream buffer directly into share recorder.

The function will not enable encoder.

Example

```
QCAP_SET_AUDIO_SHARE_RECORD_COMPRESSION_BUFFER( 0, pStreamBuffer, 4096 );
```

9.18 QCAP_START_SHARE_RECORD

```
QRESULT QCAP_START_SHARE_RECORD( UINT      iRecNum,
                                 CHAR *    pszFilePathName,
                                 DWORD     dwFlags = QCAP_RECORD_FLAG_FULL,
                                 double    dVideoDelayTime = 0.0,
                                 double    dAudioDelayTime = 0.0,
                                 double    dSegmentDurationTime = 0.0,
                                 ULONG     nSegmentDurationSizeKB = 0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
pszFilePathName	**IN**, Specify the file name of storage for recording. Recording file format is AVI, if extension name is AVI. Recording file format is MP4, if extension name is MP4. Recording file format is ASF, if extension name is ASF. Recording file format is FLV, if extension name is FLV. Recording file format is TS, if extension name is TS. Recording file format is SCF, if extension name is SCF. Recording file format is WMV, if extension name is WMV. Recording file format is WAV, if extension name is WAV.
dwFlags	**IN**, Specify the action flag of share recording. This parameter can also include any combination of the following flags. Available value are QCAP_RECORD_FLAG_FULL = 0x00000007

	QCAP_RECORD_FLAG_FILE = 0x00000001 QCAP_RECORD_FLAG_ENCODE = 0x00000002 QCAP_RECORD_FLAG_DISPLAY = 0x00000004 QCAP_RECORD_FLAG_DECODE = 0x00000008 QCAP_RECORD_FLAG_VIDEO_ONLY = 0x00000010 QCAP_RECORD_FLAG_AUDIO_ONLY = 0x00000020 QCAP_RECORD_FLAG_VIDEO_USE_IDEAL_TIMESTAMP= 0x00000040 QCAP_RECORD_FLAG_AUDIO_USE_IDEAL_TIMESTAMP= 0x00000080 QCAP_RECORD_FLAG_INGNORE_FORMAT_CHANGED = 0x00000100
dVideoDelayTime	**IN**, Specify video delay time, default is 0.0
dAudioDelayTime	**IN**, Specify audio delay time, default is 0.0
dSegmentDurationTime	**IN**, Specify segment duration time
nSegmentDurationSizeKB	**IN**, Specify segment duration size, the uint is K Byte

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start share recording.

The dwFlags parameter allows you to turn on/off internal components of recording.

Users can use dVideoDelayTime& dAudioDelayTime parameters to set video or audio delay time for control play speed.

For example, if you don’t want to display video/audio stream, the dwFlags is `QCAP_RECORD_FLAG_FILE | QCAP_RECORD_FLAG_ENCODE`.

User can use dSegmentDurationTime parameter to set segment duration time of record file.

Note

`QCAP_RECORD_FLAG_DECODE` is new parameter. If user wants to show video date from compression buffer, the new parameter of dwFlags can be used to preview video date from compression buffer on the window.

For example, if you want to display video stream from compression buffer,

the dwFlags is `QCAP_RECORD_FLAG_FILE | QCAP_RECORD_FLAG_DECODE`.

User must set QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY and QCAP_SET_AUDIO_RECORD_PROPERTY first, when user set QCAP_START_SHARE_REOCRD.

The dSegmentDurationTime parameter offer more variables for recording file path name as show in below:

$Y, $M, $D, $h, $m, $h, $i for start system time

%Y, %M, %D, %h, %m, %h, %i for stop system time

The dSegmentDurationTime and nSegmentDurationSizeKB can’t be set at same time.

Programming Workflow of Share Recording

Programming Step	Related API Functions
Set Video Record Property	QCAP_SET_VIDEO_SHARE_RECORD_PROPERTY(idx, …)
Set Audio Record Property	QCAP_SET_AUDIO_SHARE_RECORD_PROPERTY(idx, …)
Start Share Recording	QCAP_START_SHARE_RECORD(idx, …)
Push Video Frame Buffer into Recorder*	QCAP_SET_VIDEO_SHARE_RECORD_UNCOMPRESSION_BUFFER(idx, …)
Push Audio Frame Buffer into Recorder*	QCAP_SET_AUDIO_SHARE_RECORD_UNCOMPRESSION_BUFFER(idx, …)
Stop Share Recording	QCAP_STOP_SHARE_RECORD(idx, …)

* The function is called for several times.

Example

```
QCAP_START_SHARE_RECORD(0, "D:\\CH01.AVI" );

str1.Format( "%sMAIN_CH%02d.$Y.$M.$D.$h.$m.$ss.$i.mp4", g_scmProtypeEx.pszRecordFilePathName, i );

    QCAP_START_SHARE_RECORD ( 0,
                              (CHAR*)(LPCSTR)str1 ,
                              QCAP_RECORD_FLAG_FULL,
                              0.0 /*dVideoDelayTime = 0.0*/,
                              0.0/*dAudioDelayTime = 0.0*/,
                              10 );
```

9.19 QCAP_STOP_SHARE_RECORD

```
QRESULT QCAP_STOP_RECORD(  UINT   iRecNum,
                           BOOL   bIsAsync = TRUE,
                           ULONG  nMilliseconds = 0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to stop recording, start from 0 Max Recnum is 0~31
bIsAsync	**IN**, Synchronized stop recording flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop share recording.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the recording file is closed.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_STOP_SHARE_RECORD( 0 );
```

9.20 QCAP_RESUME_SHARE_RECORD

```
QRESULT QCAP_RESUME_SHARE_RECORD(UINT iRecNum )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to resume recording, start from 0 Max Recnum is 0~31

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to resume share recording.

Example

```
QCAP_RESUME_SHARE_RECORD( 0 );
```

9.21 QCAP_PAUSE_SHARE_RECORD

```
QRESULT QCAP_PAUSE_SHARE_RECORD( UINT iRecNum )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to pause recording, start from 0 Max Recnum is 0~31

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to pause share recording.

Example

```
QCAP_PAUSE_SHARE_RECORD( 0 );
```

9.22 QCAP_SNAPSHOT_SHARE_RECORD_BMP

```
QRESULT QCAP_SNAPSHOT_SHARE_RECORD_BMP( UINT      iRecNum,
                                        CHAR *    pszFilePathName,
                                        BOOL      bIsAsync = TRUE,
                                        ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0 Max Recnum is 0~31
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with BMP format, when share record processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_SHARE_RECORD_BMP( 0, "C:\\PICTURE1.BMP24" ); // PICTURE1.BMP24 ->PICTURE1.BMP
```

9.23 QCAP_SNAPSHOT_SHARE_RECORD_JPG

```
QRESULT QCAP_SNAPSHOT_SHARE_RECORD_JPG( UINT      iRecNum,
                                        CHAR *    pszFilePathName,
                                        ULONG     nQuality,
                                        BOOL      bIsAsync = TRUE,
                                        ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0 Max Recnum is 0~31
pszFilePathName	**IN**, Specify the file name to store JPG data
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with JPG format, when share record processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_SHARE_RECORD_JPG( 0, “C:\\PICTURE1.JPG”, 100 );
```

9.24 QCAP_SNAPSHOT_SHARE_RECORD_BMP_EX

```
QRESULT QCAP_SNAPSHOT_SHARE_RECORD_BMP_EX(  PVOID     iRecNum,
                                            CHAR *    pszFilePathName,
                                            ULONG     nCropX,
                                            ULONG     nCropY,
                                            ULONG     nCropW,
                                            ULONG     nCropH,
                                            ULONG     nDstW,
                                            ULONG     nDstH,
                                            BOOL      bIsAsync = TRUE,
                                            ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0 Max Recnum is 0~31
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with BMP format, when share record processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

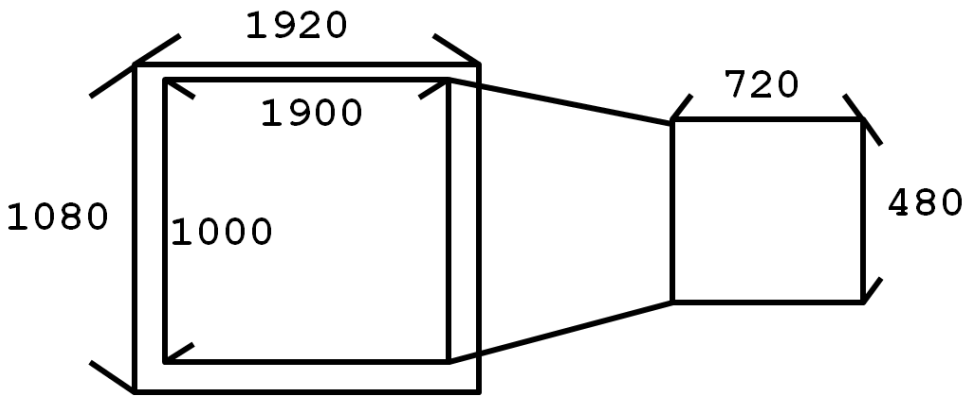
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_SHARE_RECORD_BMP_EX( 0, "C:\\PICTURE1.BMP", 10, 40, 1900, 1000, 720, 480 );
```



9.25 QCAP_SNAPSHOT_SHARE_RECORD_JPG_EX

```
QRESULT QCAP_SNAPSHOT_SHARE_RECORD_JPG_EX(  PVOID     iRecNum,
                                            CHAR *    pszFilePathName,
                                            ULONG     nCropX,
                                            ULONG     nCropY,
                                            ULONG     nCropW,
                                            ULONG     nCropH,
                                            ULONG     nDstW,
                                            ULONG     nDstH,
                                            ULONG     nQuality,
                                            BOOL      bIsAsync = TRUE,
                                            ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to get recording parameters, start from 0 Max Recnum is 0~31
pszFilePathName	**IN**, Specify the file name to store JPG data
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with JPG format, when share record processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

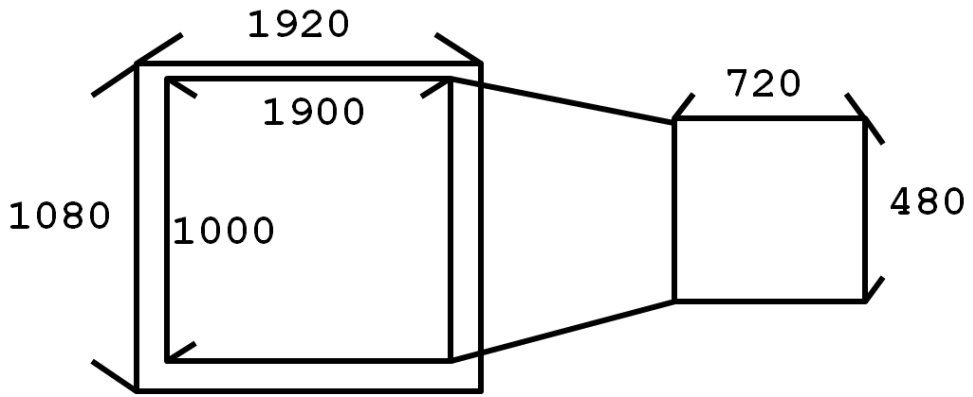
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_JPG_EX( 0, "C:\\PICTURE1.JPG", 10, 40, 1900, 1000, 720, 480, 100 );
```



9.26 QCAP_GET_OSD_SHARE_RECORD_TEXT_BOUNDARY

```
QRESULT QCAP_GET_OSD_SHARE_RECORD_TEXT_BOUNDARY(  UINT       iRecNum,
                                                  UINT       iOsdNum,
                                                  CHAR   *   pszString,
                                                  CHAR   *   pszFontFamilyName,
                                                  ULONG      nFontStyle,
                                                  ULONG      nFontSize,
                                                  ULONG  *   pBoundaryWidth,
                                                  ULONG  *   pBoundaryHeight )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set OSD output, start from 0 Max Recnum is 0~31
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 127
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output

nFontStyle	**IN**, Specify the font style used to display the text of OSD output
	Available values are
	`QCAP_FONT_STYLE_REGULAR = 0x00000000,`
	`QCAP_FONT_STYLE_BOLD = 0x00000001,`
	`QCAP_FONT_STYLE_ITALIC = 0x00000002,`
	`QCAP_FONT_STYLE_BOLDITALIC = 0x00000003,`
	`QCAP_FONT_STYLE_UNDERLINE = 0x00000004,`
	`QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,`

nFontSize	**IN**, Specify the font size used to display the text of OSD output
pBoundaryWidth	**OUT**, Pointer to the storage of boundary width
pBoundaryHeight	**OUT**, Pointer to the storage of boundary height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get size of OSD string, when the users use the share record function.

For example, user can use boundary information to set OSD string location on share record display window.

Example

```
QCAP_GET_OSD_SHARE_RECORD_TEXT_BOUNDARY( 0, 0, “CH01”, “Arial”,
                                         QCAP_FONT_STYLE_BOLD, 12,
                                         pBoundaryWidth, pBoundaryHeight );
```

9.27 QCAP_SET_OSD_SHARE_RECORD_TEXT

```
QRESULT QCAP_SET_OSD_SHARE_RECORD_TEXT( UINT      iRecNum,
                                        UINT      iOsdNum,
                                        INT       x,
                                        INT       y,
                                        INT       width,
                                        INT       height,
                                        CHAR *    pszString,
                                        CHAR *    pszFontFamilyName,
                                        ULONG     nFontStyle,
                                        ULONG     nFontSize,
                                        DWORD     dwFontColor,
                                        DWORD     dwBackgroundColor,
                                        ULONG     nTransparent,
                                        INT       nTextStartPosX = 0,
                                        INT       nTextStartPosY = 0,
                                        ULONG     nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set OSD output, start from 0 Max Recnum is 0~31
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 127
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify they-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output, if width is -1, QCAP will auto calculate width.
height	**IN**, Specify the vertical height of OSD output, if height is -1, QCAP will auto calculate height.
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output

nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are
	`QCAP_FONT_STYLE_REGULAR = 0x00000000,`
	`QCAP_FONT_STYLE_BOLD = 0x00000001,`
	`QCAP_FONT_STYLE_ITALIC = 0x00000002,`
	`QCAP_FONT_STYLE_BOLDITALIC = 0x00000003,`
	`QCAP_FONT_STYLE_UNDERLINE = 0x00000004,`
	`QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,`

nFontSize	**IN**, Specify the font size used to display the text of OSD output
dwFontColor	**IN**, Specify the font color used to display the text of OSD output
dwBackgroundColor	**IN**, Specify the background color used to display the text of OSD output

nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nTextStartPosX	**IN**, Specify the text scrolling start position X of the upper-left corner of OSD text
nTextStartPosY	**IN**, Specify the text scrolling start position Y of the upper-left corner of OSD text
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

We also support to create a text field objects used for onscreen display of information

when the users use the share record function to save record to a file.

If you want to get a text field object auto calculated actual width and height value;

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_SHARE_RECORD_TEXT(  0, 0, 0, 0, -1, -1, “CH01”, “Arial”,
                                 QCAP_FONT_STYLE_BOLD, 12, 0xFF000000,
                                 0xFFFFFFFF, 128, 0, 0, QCAP_SEQUENCE_STYLE_FOREMOST );
```

9.28 QCAP_SET_OSD_SHARE_RECORD_PICTURE

```
QRESULT QCAP_SET_OSD_SHARE_RECORD_PICTURE( UINT      iRecNum,
                                           UINT      iOsdNum,
                                           INT       x,
                                           INT       y,
                                           INT       width,
                                           INT       height,
                                           CHAR *    pszFilePathName,
                                           ULONG     nTransparent,
                                           ULONG     nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set OSD output, start from 0 Max Recnum is 0~31
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 127
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify they-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output If width is -1, QCAP will use original picture width as default.
height	**IN**, Specify the vertical height of OSD output. If height is -1, QCAP will use original picture height as default.
pszFilePathName	**IN**, Specify the file name to display image OSD output, support a file extension of "BMP" as 24 or 32-bit,“JPG” and “PNG”
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

We also support to create a static image objects used for onscreen display of information

when the users use the share record function to save record to a file.

If you want to get a text field object auto calculated actual width and height value;

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_SHARE_RECORD_PICTURE( 0, 0, 0, 0, -1, -1,
                                   "C:\\SAMPLE.PNG", 128, QCAP_SEQUENCE_STYLE_FOREMOST );
```

9.29 QCAP_SET_OSD_SHARE_RECORD_BUFFER

```
QRESULT QCAP_SET_OSD_SHARE_RECORD_BUFFER ( UINT      iRecNum,
                                           UINT      iOsdNum,
                                           INT       x,
                                           INT       y,
                                           INT       width,
                                           INT       height,
                                           ULONG     nColorSpaceType,
                                           BYTE *    pFrameBuffer,
                                           ULONG     nFrameWidth,
                                           ULONG     nFrameHeight,
                                           ULONG     nFramePitch,
                                           ULONG     nTransparent,
                                           DWORD     dwKeyColor = 0xFFFFFFFF,
                                           ULONG     nKeyColorThreshold = 25,
                                           ULONG     nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to set OSD output, start from 0 Max Recnum is 0~31
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 127
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify they-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output
height	**IN**, Specify the vertical height of OSD output
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameWidth	**IN**, Specify the horizontal width of frame contained in a buffer
nFrameHeight	**IN**, Specify the vertical height of frame contained in a buffer
nFramePitch	**IN**, Specify the distance in bytes between the start of one scan line to the next. If it is 0, we will auto calculate it by width and color space format.

nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
dwKeyColor	**IN**, Specify the key color of share record OSD, default is 0xFFFFFFFF
nKeyColorThreshold	**IN**, Specify the threshold of key color, the range of 0 to 128
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

We also support to create a frame buffer objects used for onscreen display of information

when the users use the share record function to save record to a file.

The function’s purpose is used to point directly to a valid frame buffer whose entire pixel data,

which is stored in available color space, and display it.

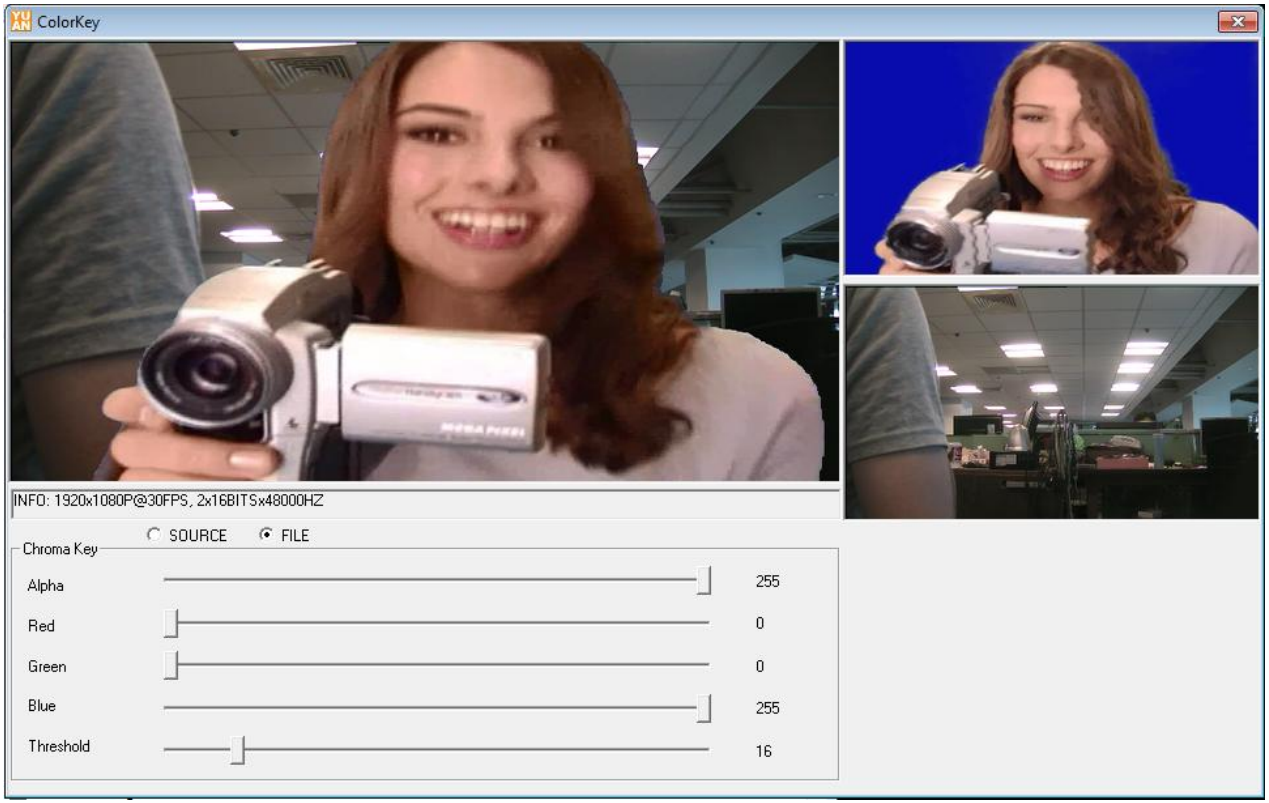
Beside, you also can adjust the width and height parameters to scale the OSD output

And OSD sequence style to video-streams recording the OSD output.

Note

User can use this dwKeyColor parament to select colorkey mode Green&Blue mode and RGB mode.

For instance the Blue mode way as show in below



Example

```
QCAP_SET_OSD_SHARE_RECORD_BUFFER( 0, 0, 0, 0, 1920, 1080,

                                  QCAP_COLORSPACE_TYEP_YUY2, pFrameBuffer, 800, 600, 0, 128,

                                  0xFFFFFFFF, 25, QCAP_SEQUENCE_STYLE_FOREMOST );
```

9.30 QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER(  UINT    iRecNum,
                                                                     ULONG    nColorSpaceType,
                                                                     ULONG    nWidth,
                                                                     ULONG    nHeight,
                                                                     BYTE *   pFrameBuffer,
                                                                     ULONG    nFrameBufferLen,
                                                                     ULONG    nStereoBufferType )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to 3D start recording, start from 0 Max Recnum is 0~31
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D input source buffer's width
height	**IN**, Specify the 3D input source buffer's height
pFrameBuffer	**IN**, Specify the 3D input source buffer
nFrameBufferLen	**IN**, Specify the 3D input source buffer's size
nStereoBufferType	**IN**, Specify 3D stereo mode. Available values are QCAP_3D_STEREO_BUFFER_SIDE_BY_SIDE = 0 QCAP_3D_STEREO_BUFFER_TOP_BOTTOM = 1 QCAP_3D_STEREO_BUFFER_LINE_BY_LINE = 2

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use one camera to recording 3D video stream.

User can use this function to push 3D un-compression video buffer into share recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

And dwFlags to video-streams recording the 3D stereo output.

Example

```
QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER( 0, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                            pFrameBuffer, nFrameBufferLen,
                                                            QCAP_3D_STEREO_BUFFER_SIDE_BY_SIDE );
```

9.31 QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER_EX(   UINT   iRecNum,
                                                                         ULONG  nColorSpaceType,
                                                                         ULONG  nWidth,
                                                                         ULONG  nHeight,
                                                                         BYTE * pFrameBuffer,
                                                                         ULONG  nFrameBufferLen,
                                                                         ULONG  nCropX,
                                                                         ULONG  nCropY,
                                                                         ULONG  nCropW,
                                                                         ULONG  nCropH,
                                                                         ULONG  nStereoBufferType )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D input source buffer's width
height	**IN**, Specify the 3D input source buffer's height
pFrameBuffer	**IN**, Specify the 3D input source buffer
nFrameBufferLen	**IN**, Specify the 3D input source buffer's size
nCropX	**IN**, Specify the x-coordinate of the crop of 3D video
nCropY	**IN**, Specify the y-coordinate of the crop of 3D video
nCropW	**IN**, Specify the horizontal width of crop of 3D video
nCropH	**IN**, Specify the vertical height of crop of 3D video
nStereoBufferType	**IN**, Specify 3D stereo mode. Available values are QCAP_3D_STEREO_BUFFER_SIDE_BY_SIDE = 0 QCAP_3D_STEREO_BUFFER_TOP_BOTTOM = 1

	QCAP_3D_STEREO_BUFFER_LINE_BY_LINE = 2

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use one camera to recording 3D video stream.

User can use this function to push 3D un-compression video buffer into share and set crop parameters into share recorder.

And User can use crop parameters to set any part of a video screen at will.

Example

```
QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER_EX( 0, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                               pFrameBuffer, nFrameBufferLen,
                                                               100, 150, 720, 480
                                                               QCAP_3D_STEREO_BUFFER_SIDE_BY_SIDE );
```

9.32 QCAP_SET_VIDEO_3D_SHARE_RECORD_L_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_3D_SHARE_RECORD_L_UNCOMPRESSION_BUFFER (UINT      iRecNum,
                                                               ULONG     nColorSpaceType,
                                                               ULONG     nWidth,
                                                               ULONG     nHeight,
                                                               BYTE *    pFrameBuffer,
                                                               ULONG     nFrameBufferLen )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D of left input source buffer's width
height	**IN**, Specify the 3D of left input source buffer's height
pFrameBuffer	**IN**, Specify the 3D of left input source buffer
nFrameBufferLen	**IN**, Specify the 3D of left input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D of left un-compression video buffer into share recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

Example

```
QCAP_SET_VIDEO_3D_SHARE_RECORD_L_UNCOMPRESSION_BUFFER( 0, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                       pFrameBuffer, nFrameBufferLen );
```

9.33 QCAP_SET_VIDEO_3D_SHARE_RECORD_L_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_3D_SHARE_RECORD_L_UNCOMPRESSION_BUFFER_EX(  UINT      iRecNum,
                                                                   ULONG     nColorSpaceType,
                                                                   ULONG     nWidth,
                                                                   ULONG     nHeight,
                                                                   BYTE *    pFrameBuffer,
                                                                   ULONG     nFrameBufferLen,
                                                                   ULONG     nCropX,
                                                                   ULONG     nCropY,
                                                                   ULONG     nCropW,
                                                                   ULONG     nCropH )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D of left input source buffer's width
height	**IN**, Specify the 3D of left input source buffer's height
pFrameBuffer	**IN**, Specify the 3D of left input source buffer
nFrameBufferLen	**IN**, Specify the 3D of left input source buffer's size
nCropX	**IN**, Specify the x-coordinate of the crop of 3D video
nCropY	**IN**, Specify the y-coordinate of the crop of 3D video
nCropW	**IN**, Specify the horizontal width of crop of 3D video
nCropH	**IN**, Specify the vertical height of crop of 3D video

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D of left un-compression video buffer into share recorder and set crop parameters into share recorder.

And User can use crop parameters to set any part of a video screen at will.

Example

```
QCAP_SET_VIDEO_3D_SHARE_RECORD_L_UNCOMPRESSION_BUFFER_EX( 0, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                          pFrameBuffer, nFrameBufferLen,
                                                          100, 150, 720, 480 );
```

9.34 QCAP_SET_VIDEO_3D_SHARE_RECORD_R_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_3D_SHARE_RECORD_R_UNCOMPRESSION_BUFFER (UINT      iRecNum,
                                                               ULONG     nColorSpaceType,
                                                               ULONG     nWidth,
                                                               ULONG     nHeight,
                                                               BYTE *    pFrameBuffer,
                                                               ULONG     nFrameBufferLen )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D of right input source buffer's width
height	**IN**, Specify the 3D of right input source buffer's height
pFrameBuffer	**IN**, Specify the 3D of right input source buffer
nFrameBufferLen	**IN**, Specify the 3D of right input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D of right un-compression video buffer into share recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

Example

```
QCAP_SET_VIDEO_3D_SHARE_RECORD_R_UNCOMPRESSION_BUFFER( 0, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                       pFrameBuffer, nFrameBufferLen );
```

9.35 QCAP_SET_VIDEO_3D_SHARE_RECORD_R_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_3D_SHARE_RECORD_R_UNCOMPRESSION_BUFFER_EX(  UINT      iRecNum,
                                                                   ULONG     nColorSpaceType,
                                                                   ULONG     nWidth,
                                                                   ULONG     nHeight,
                                                                   BYTE *    pFrameBuffer,
                                                                   ULONG     nFrameBufferLen,
                                                                   ULONG     nCropX,
                                                                   ULONG     nCropY,
                                                                   ULONG     nCropW,
                                                                   ULONG     nCropH )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D of right input source buffer's width
height	**IN**, Specify the 3D of right input source buffer's height
pFrameBuffer	**IN**, Specify the 3D of right input source buffer
nFrameBufferLen	**IN**, Specify the 3D of right input source buffer's size
nCropX	**IN**, Specify the x-coordinate of the crop of 3D video
nCropY	**IN**, Specify the y-coordinate of the crop of 3D video
nCropW	**IN**, Specify the horizontal width of crop of 3D video
nCropH	**IN**, Specify the vertical height of crop of 3D video

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D of right un-compression video buffer into share recorder and set crop parameters into share recorder.

And User can use crop parameters to set any part of a video screen at will.

Example

```
QCAP_SET_VIDEO_3D_SHARE_RECORD_R_UNCOMPRESSION_BUFFER_EX( 0, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                          pFrameBuffer, nFrameBufferLen,
                                                          100, 150, 720, 480 );
```

9.36 QCAP_SET_VIDEO_3D_SHARE_RECORD_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_3D_SHARE_RECORD_UNCOMPRESSION_BUFFER(    UINT      iRecNum,
                                                                ULONG     nStereoDisplayMode,
                                                                BOOL      bLeftRightSwap = FALSE,
                                                                BOOL      bForceKeyFrame = FALSE,
                                                                double    dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
nStereoDisplayMode	**IN**, Specify 3D display mode. Available values are QCAP_3D_STEREO_DISPLAY_MODE_SIDE_BY_SIDE = 0 QCAP_3D_STEREO_DISPLAY_MODE_TOP_BOTTOM = 1 QCAP_3D_STEREO_DISPLAY_MODE_LINE_BY_LINE = 2 QCAP_3D_STEREO_DISPLAY_MODE_LEFT_ONLY = 3 QCAP_3D_STEREO_DISPLAY_MODE_RIGHT_ONLY = 4
bLeftRightSwap	**IN**, Swap the input camera source's (right to left, left to right), the default is FALSE
bForceKeyFrame	**IN**, Enforce the input source's frame is key-frame, the default is FALSE
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D un-compression video buffer into share recorder.

User can use bForceKeyFrame parameter to push key-frame immediately.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

If the width or the height of source frame are different from the target frame size.

QCAP will enable one auto video scaler to resize it before encoding.

Example

```
QCAP_SET_VIDEO_3D_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, 0, FALSE, FALSE, 0 );
```

9.37 QCAP_GET_AUDIO_MX_SHARE_RECORD_PROPERTY_EX

```
QRESULT QCAP_GET_AUDIO_MX_SHARE_RECORD_PROPERTY_EX(  UINT      iRecNum,
                                                     ULONG *   pTracks,
                                                     ULONG *   pEncoderType,
                                                     ULONG *   pEncoderFormat,
                                                     ULONG *   pChannels,
                                                     ULONG *   pBitsPerSample,
                                                     ULONG *   pSampleFrequency,
                                                     ULONG *   pBitRate,
                                                     ULONG *   pVolume )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, start from 0 Max Recnum is 0~31
pTracks	**OUT**, Pointer to the storage of audio tracks
pEncoderType	**OUT**, Pointer to the storage of audio encoder type
pEncoderFormat	**OUT**, Pointer to the storage of audio encoder format
pChannels	**OUT**, Pointer to the storage of audio channels
pBitsPerSample	**OUT**, Pointer to the storage of audio bits per sample
pSampleFrequency	**OUT**, Pointer to the storage of audio sample frequency

pBitRate	**OUT**, Pointer to the storage of audio bit rate
pVolume	**OUT**, Pointer to the storage of audio volume

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get audio mixer information, when the users use the audio mixer function.

Example

```
QCAP_GET_AUDIO_MX_SHARE_RECORD_PROPERTY_EX( 0, &nTracks, &nBoundaryHeight, &nEncoderFormat, &nChannels,

                                            &nBitsPerSample, &nSampleFrequency, &nBitRate, &nVolume );
```

9.38 QCAP_SET_AUDIO_MX_SHARE_RECORD_PROPERTY_EX

```
QRESULT QCAP_SET_AUDIO_MX_SHARE_RECORD_PROPERTY_EX(  UINT      iRecNum,
                                                     ULONG     nTracks,
                                                     ULONG     nEncoderType,
                                                     ULONG     nEncoderFormat,
                                                     ULONG     nChannels,
                                                     ULONG     nBitsPerSample,
                                                     ULONG     nSampleFrequency,
                                                     ULONG     nBitRate,
                                                     ULONG     nVolume )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, rec num: 0 ~ 31
nTracks	**IN**, Specify audio track num, start from 0, track num: 0 ~ 3
nEncoderType	**IN**, Specify encoder type. Available value are `QCAP_ENCODER_TYPE_SOFTWARE = 0`

nEncoderFormat	**IN**, Specify encoder format. Available value are
	`QCAP_ENCODER_FORMAT_PCM = 0`
	`QCAP_ENCODER_FORMAT_AAC = 1`
	`QCAP_ENCODER_FORMAT_AAC_ADTS = 2`

nChannels	**IN**, Specify encoder channels for the i-th audio stream session
nBitsPerSample	**IN**, Specify encoder bits per sample for the i-th audio stream session
nSampleFrequency	**IN**, Specify encoder sample frequency for the i-th audio stream session
nBitRate	**IN**, Specify audio bit rate. Default is 128k and max is 496k
nVolume	**IN**, Specify the volume of audio renderer for the i-th audio stream session, from 0 ~ 100

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio mixer format property.

The nTracks is decide how many audio tracks to create, use can create three audio track at same time.

Example

```
QCAP_SET_AUDIO_MX_SHARE_RECORD_PROPERTY_EX( 0, 3, QCAP_ENCODER_TYPE_SOFTWARE, QCAP_ENCODER_FORMAT_AAC,
                                            2, 16, 48000, 100 );
```

9.39 QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXONG_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER(   UINT      iRecNum,
                                                                      UINT      iTrackNum,
                                                                      BYTE *    pFrameBuffer,
                                                                      ULONG     nFrameBufferLen )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, rec num: 0 ~ 31
iTrackNum	**IN**, Specify audio track num
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to mixing audio share record uncompression buffer on audio preview callback.

User must set `QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER` to mixer audio record first then call

`QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER.`

Example

```
if( nChannelNum == 0 ){
    QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER( 0, 0, pFrameBuffer, nFrameBufferLen );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER( 0, 1, pFrameBuffer, nFrameBufferLen );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, 0 );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, 1 );
}
if( nChannelNum == 1 ){
    QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER( 0, 0, pFrameBuffer, nFrameBufferLen );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER( 0, 2, pFrameBuffer, nFrameBufferLen );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, 2 );
}
```

9.40 QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER(   UINT      iRecNum,
                                                                      UINT      iTrackNum,
                                                                      double    dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, rec num: 0 ~ 31
iTrackNum	**IN**, Specify audio track num
dSampleTime	**IN**, Specify the timestamp of this frame.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio share record uncompression buffer on audio preview callback.

Example

```
if( nChannelNum == 0 ){
    QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER( 0, 0, pFrameBuffer, nFrameBufferLen );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER( 0, 1, pFrameBuffer, nFrameBufferLen );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, 0 );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, 1 );
}
if( nChannelNum == 1 ){
    QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER( 0, 0, pFrameBuffer, nFrameBufferLen );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_UNCOMPRESSION_BUFFER( 0, 2, pFrameBuffer, nFrameBufferLen );
    QCAP_SET_AUDIO_MX_SHARE_RECORD_UNCOMPRESSION_BUFFER( 0, 2 );
}
```

9.41 QCAP_SET_AUDIO_MX_SHARE_RECORD_COMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_MX_SHARE_RECORD_MIXING_COMPRESSION_BUFFER(  UINT      iRecNum,
                                                                   UINT      iTrackNum,
                                                                   BYTE *    pStreamBuffer,
                                                                   ULONG     nStreamBufferLen,
                                                                   double    dSampleTime = 0.0 )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index to start recording, rec num: 0 ~ 31
iTrackNum	**IN**, Specify audio track num
pStreamBuffer	**OUT**, Pointer to the storage of the input source buffer
pStreamBufferLen	**IN**, Specify the max input source buffer's size
pSampleTime	**OUT**, Pointer to the storage of sample time

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio share record compression buffer on audio share record callback.

Example

```
if( nChannelNum == 0 ){
    QCAP_SET_AUDIO_MX_SHARE_RECORD_COMPRESSION_BUFFER( 0, 0, pStreamBuffer, nStreamBufferLen );
}
if( nChannelNum == 1 ){
    QCAP_SET_AUDIO_MX_SHARE_RECORD_COMPRESSION_BUFFER( 0, 1, pStreamBuffer, nStreamBufferLen );
}
```

10 File Playback/Editing Function API

10.1 QCAP_OPEN_FILE

```
QRESULT QCAP_OPEN_FILE(    CHAR   *   pszFileName,
                           PVOID  *   ppFile,
                           ULONG      nDecoderType,
                           ULONG  *   pVideoFormat,
                           ULONG  *   pVideoWidth,
                           ULONG  *   pVideoHeight,
                           double *   pVideoFrameRate,
                           ULONG  *   pAudioFormat,
                           ULONG  *   pAudioChannels,
                           ULONG  *   pAudioBitsPerSample,
                           ULONG  *   pAudioSampleFrequency,
                           double *   pTotalDurationTimes,
                           ULONG  *   pTotalVideoFrames,
                           ULONG  *   pTotalAudioFrames,
                           HWND       hAttachedWindow,
                           BOOL       bThumbDraw = FALSE,
                           BOOL       bMaintainAspectRatio = FALSE )
```

Parameter

Name	Description
pszFileName	**IN**, Specify the video file name to display video output
ppFile	**OUT**, Handle of the file object

nDecoderType	**IN**, Specify decoder type. Available value are
	`QCAP_DECODER_TYPE_SOFTWARE = 0`
	`QCAP_DECODER_TYPE_INTEL_MEDIA_SDK = 2`

pVideoFormat	**OUT**, Pointer to the storage of the video format
pVideoWidth	**OUT**, Pointer to the storage of the video width
pVideoHeight	**OUT**, Pointer to the storage of the video height
pVideoFrameRate	**OUT**, Pointer to the storage of the video frame rate
pAudioFormat	**OUT**, Pointer to the storage of the audio format
pAudioChannels	**OUT**, Pointer to the storage of the audio channel
pAudioBitsPerSample	**OUT**, Pointer to the storage of the audio bit per sample
pAudioSampleFrequency	**OUT**, Pointer to the storage of the audio sample frequency
pTotalDurationTimes	**OUT**, Pointer to the storage of the total duration times
pTotalVideoFrames	**OUT**, Pointer to the storage of the total video frames
pTotalAudioFrames	**OUT**, Pointer to the storage of the audio frames
hAttachedWindow	**IN**, Handle of the window to show the preview video. If it is NULL, the file will be used at Editing mode only.
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer

bMaintainAspectRatio	**IN**, Enable/Disable the maintain aspect ratio

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

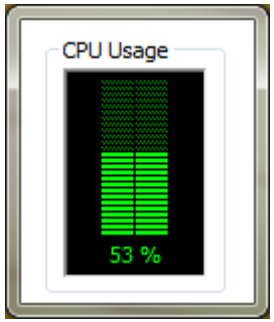
Introduction

User can use this function to create one video file used for on screen playback.

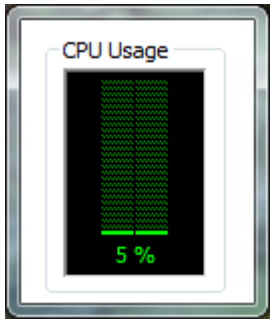
If you want to use editing mode to access this file, you can set the hAttachedWindowas NULL.

The thumb draw function can reduce the loading of video display, when display window size is small than the input video size.

For multiple channels developer, we suggest to turn on it in order to obtain the best rendering performance.



Normal Display



Thumb Draw Mode

Example

```
QCAP_OPEN_FILE( "D:\\CH01.AVI", &pFile, QCAP_DECODER_TYPE_SOFTWARE, &nVideoFormat, &nVideoWidth,
                &nVideoHeight, &nVideoFrameRate, &nAudioFormat, &nAudioChannels, &nAudioBitsPerSample,
                &nAudioSampleFrequency, &nTotalDurationTimes, &nTotalVideoFrames, &nTotalAudioFrames,
                hAttachedWindow );
```

10.2 QCAP_DESTROY_FILE

```
QRESULT QCAP_DESTROY_FILE( PVOID pFile )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Destroy file object to release its resource.

Example

```
QCAP_DESTROY_FILE( pFile );
```

10.3 QCAP_PLAY_FILE

```
QRESULT QCAP_PLAY_FILE( PVOID pFile )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to play file.

Example

```
QCAP_PLAY_FILE( pFile );
```

10.4 QCAP_PAUSE_FILE

```
QRESULT QCAP_PAUSE_FILE( PVOID pFile )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to pause file.

Example

```
QCAP_PAUSE_FILE( pFile )
```

10.5 QCAP_STOP_FILE

```
QRESULT QCAP_STOP_FILE( PVOID pFile )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop file.

Example

```
QCAP_STOP_FILE( pFile );
```

10.6 QCAP_GET_FILE_POSITION

```
QRESULT QCAP_GET_FILE_POSITION( PVOID pFile, double * pSampleTime )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pSampleTime	**OUT**, Pointer to the storage of sample time

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get video frame position such as video seek function.

Example

```
QCAP_GET_FILE_POSITION( pFile, &nSampleTime );
```

10.7 QCAP_SET_FILE_POSITION

```
QRESULT QCAP_SET_FILE_POSITION( PVOID pFile, double dSampleTime )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
dSampleTime	**IN**, Specify the timestamp of this frame.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set current file position such as video seek function.

Example

```
QCAP_SET_FILE_POSITION( pFile, SetSampleTime );
```

10.8 QCAP_GET_FILE_PLAYBACK_SPEED

```
QRESULT QCAP_GET_FILE_PLAYBACK_SPEED( PVOID pFile, double * pSpeed )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pSpeed	**OUT**, Pointer to the storage of speed

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current file playback speed.

Example

```
QCAP_GET_FILE_PLAYBACK_SPEED( pFile, &nSpeed );
```

10.9 QCAP_SET_FILE_PLAYBACK_SPEED

```
QRESULT QCAP_SET_FILE_PLAYBACK_SPEED( PVOID pFile, double dSpeed )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
dSpeed	**IN**, Specify the speed

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set and control file playback speed.

Example

```
QCAP_SET_FILE_PLAYBACK_SPEED( pFile, nSpeed );
```

10.10 QCAP_GET_VIDEO_FILE_REGION_DISPLAY

```
QRESULT QCAP_GET_VIDEO_FILE_REGION_DISPLAY(    PVOID     pFile,
                                               ULONG  *  pCropX,
                                               ULONG  *  pCropY,
                                               ULONG  *  pCropW,
                                               ULONG  *  pCropH )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pCropX	**OUT**, Pointer to the storage of x-coordinate of the crop region display
pCropY	**OUT**, Pointer to the storage of y-coordinate of the crop region display
pCropW	**OUT**, Pointer to the storage of horizontal width of crop region display
pCropH	**OUT**, Pointer to the storage of vertical height of crop region display

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get the current region display information of video file.

Example

```
QCAP_GET_VIDEO_FILE_REGION_DISPLAY( pFile, &bCropX, &bCropY, &bCropW, &bCropH );
```

10.11 QCAP_SET_VIDEO_FILE_REGION_DISPLAY

```
QRESULT QCAP_SET_VIDEO_FILE_REGION_DISPLAY(    PVOID  pFile,
                                               ULONG  nCropX,
                                               ULONG  nCropY,
                                               ULONG  nCropW,
                                               ULONG  nCropH  )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
nCropX	**IN**, Specify the x-coordinate of the crop region display
nCropY	**IN**, Specify the y-coordinate of the crop region display
nCropW	**IN**, Specify the horizontal width of crop region display
nCropH	**IN**, Specify the vertical height of crop region display

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set the current region display of video file.

Example

```
QCAP_SET_VIDEO_FILE_REGION_DISPLAY( pFile, 0, 0, 1280, 720 );
```

10.12 QCAP_GET_VIDEO_FILE_MIRROR

```
QRESULT QCAP_GET_VIDEO_FILE_MIRROR(  PVOID     pFile,
                                     BOOL *    pHorizontalMirror,
                                     BOOL *    pVerticalMirror )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
bHorizontalMirror	**OUT**, Pointer to the storage of current horizontal mirror
bVerticalMirror	**OUT**, Pointer to the storage of current vertical mirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get the status of mirror of file video.

Example

```
QCAP_GET_VIDEO_FILE_MIRROR( pFile, &bHorizontalMirror, &bVerticalMirror );
```

10.13 QCAP_SET_VIDEO_FILE_MIRROR

```
QRESULT QCAP_SET_VIDEO_FILE_MIRROR(  PVOID  pFile,
                                     BOOL   bHorizontalMirror,
                                     BOOL   bVerticalMirror )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
bHorizontalMirror	**IN**, Enable/Disable horizontal of mirror
bVerticalMirror	**IN**, Enable/Disable vertical of mirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set the status of mirror of file video.

Example

```
QCAP_SET_VIDEO_FILE_MIRROR( pFile, TRUE, TRUE );
```

10.14 QCAP_GET_AUDIO _FILE _VOLUME

```
QRESULT QCAP_GET_AUDIO_FILE_VOLUME( PVOID pFile, ULONG * pVolume )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pVolume	**OUT**, Pointer to the storage of audio volume

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get audio volume of file.

Example

```
QCAP_GET_AUDIO_FILE_VOLUME( pFile, nVolume );
```

10.15 QCAP_SET_AUDIO _FILE _VOLUME

```
QRESULT QCAP_SET_AUDIO_FILE_VOLUME( PVOID pFile, ULONG  nVolume )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
nVolume	**IN**, Specify the audio volume value is from 0~100

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio volume of file.

Example

```
QCAP_SET_AUDIO_FILE_VOLUME( pFile, 50 );
```

10.16 QCAP_SNAPSHOT_FILE_BMP

```
QRESULT QCAP_SNAPSHOT_FILE_BMP(   PVOID     pFile,
                                  CHAR *    pszFilePathName,
                                  BOOL      bIsAsync = TRUE,
                                  ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in millisecond User can use this function to snapshot a picture with BMP format on share records. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with BMP format, when video file playing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_FILE_BMP( pFile, "C:\\PICTURE1.BMP24" ); // PICTURE1.BMP24 ->PICTURE1.BMP
```

10.17 QCAP_SNAPSHOT_FILE_JPG

```
QRESULT QCAP_SNAPSHOT_FILE_JPG(   PVOID     pFile,
                                  CHAR *    pszFilePathName,
                                  ULONG     nQuality,
                                  BOOL      bIsAsync = TRUE,
                                  ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pszFilePathName	**IN**, Specify the file name to store JPG data
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshoot a picture with JPG format, when video file playing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_FILE_JPG( pFile, "C:\\PICTURE1.JPG", 100 );
```

10.18 QCAP_SNAPSHOT_FILE_BMP_EX

```
QRESULT QCAP_SNAPSHOT_FILE_BMP_EX(   PVOID      pFile,
                                     CHAR *     pszFilePathName,
                                     ULONG      nCropX,
                                     ULONG      nCropY,
                                     ULONG      nCropW,
                                     ULONG      nCropH,
                                     ULONG      nDstW,
                                     ULONG      nDstH,
                                     BOOL       bIsAsync = TRUE,
                                     ULONG      nMilliseconds = 0 )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with BMP format, when video file playing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

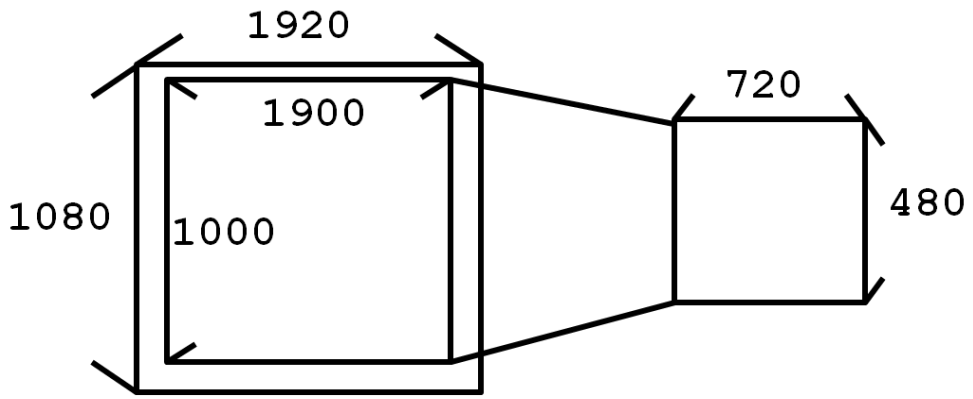
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_FILE_BMP_EX( pFile, "C:\\PICTURE1.BMP", 10, 40, 1900, 1000, 720, 480 );
```



10.19 QCAP_SNAPSHOT_FILE_JPG_EX

```
QRESULT QCAP_SNAPSHOT_FILE_JPG_EX(   PVOID     pFile,
                                     CHAR *    pszFilePathName,
                                     ULONG     nCropX,
                                     ULONG     nCropY,
                                     ULONG     nCropW,
                                     ULONG     nCropH,
                                     ULONG     nDstW,
                                     ULONG     nDstH,
                                     ULONG     nQuality,
                                     BOOL      bIsAsync = TRUE,
                                     ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pszFilePathName	**IN**, Specify the file name to store JPG data
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with JPG format, when video file playing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

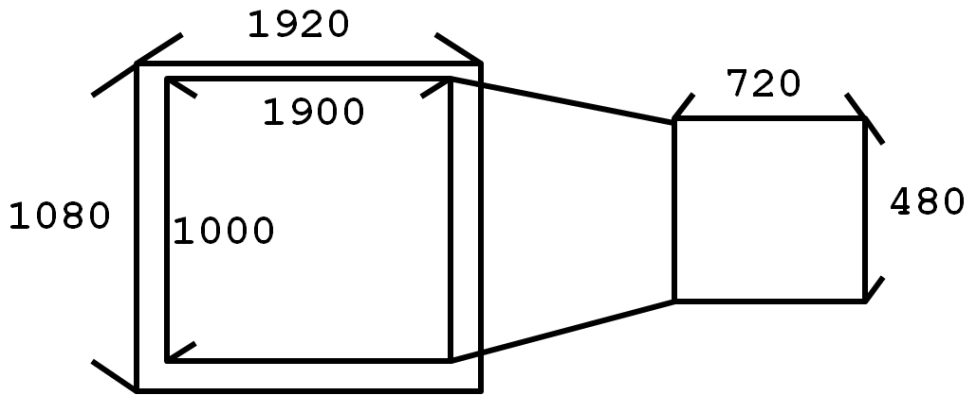
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_FILE_JPG_EX( pFile, "C:\\PICTURE1.JPG", 10, 40, 1900, 1000, 720, 480, 100 );
```



10.20 QCAP_GET_OSD_FILE_TEXT_BOUNDARY

```
QRESULT QCAP_GET_OSD_FILE_TEXT_BOUNDARY(    PVOID      pFile,
                                            UINT       iOsdNum,
                                            CHAR   *   pszString,
                                            CHAR   *   pszFontFamilyName,
                                            ULONG      nFontStyle,
                                            ULONG      nFontSize,
                                            ULONG  *   pBoundaryWidth,
                                            ULONG  *   pBoundaryHeight )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output
nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are QCAP_FONT_STYLE_REGULAR = 0x00000000, QCAP_FONT_STYLE_BOLD = 0x00000001, QCAP_FONT_STYLE_ITALIC = 0x00000002, QCAP_FONT_STYLE_BOLDITALIC = 0x00000003, QCAP_FONT_STYLE_UNDERLINE = 0x00000004, QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,
nFontSize	**IN**, Specify the font size used to display the text of OSD output
pBoundaryWidth	**OUT**, Pointer to the storage of boundary width
pBoundaryHeight	**OUT**, Pointer to the storage of boundary height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get size of OSD string, when the users use the file text function.

For example, user can use boundary information to set OSD string location on display window.

Example

```
QCAP_GET_OSD_FILE_TEXT_BOUNDARY( pFile, 0,“CH01”, “Arial”,
                                 QCAP_FONT_STYLE_BOLD, 12, pBoundaryWidth, pBoundaryHeight );
```

10.21 QCAP_SET_OSD_FILE_TEXT

```
QRESULT QCAP_SET_OSD_FILE_TEXT(  PVOID      pFile,
                                 UINT       iOsdNum,
                                 INT        x,
                                 INT        y,
                                 INT        width,
                                 INT        height,
                                 CHAR   *   pszString,
                                 CHAR   *   pszFontFamilyName,
                                 ULONG      nFontStyle,
                                 ULONG      nFontSize,
                                 DWORD      dwFontColor,
                                 DWORD      dwBackgroundColor,
                                 ULONG      nTransparent,
                                 INT        nTextStartPosX = 0,
                                 INT        nTextStartPosY = 0,
                                 ULONG      nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify they-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output, if width is -1, QCAP will auto calculate width.
height	**IN**, Specify the vertical height of OSD output, if height is -1, QCAP will auto calculate height.
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output

nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are
	`QCAP_FONT_STYLE_REGULAR = 0x00000000,`
	`QCAP_FONT_STYLE_BOLD = 0x00000001,`
	`QCAP_FONT_STYLE_ITALIC = 0x00000002,`
	`QCAP_FONT_STYLE_BOLDITALIC = 0x00000003,`
	`QCAP_FONT_STYLE_UNDERLINE = 0x00000004,`
	`QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,`

nFontSize	**IN**, Specify the font size used to display the text of OSD output
dwFontColor	**IN**, Specify the font color used to display the text of OSD output
dwBackgroundColor	**IN**, Specify the background color used to display the text of OSD output

nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nTextStartPosX	**IN**, Specify the text scrolling start position X of the upper-left corner of OSD text
nTextStartPosY	**IN**, Specify the text scrolling start position Y of the upper-left corner of OSD text
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected

QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

When the users create a file object,

If you want to get a text field object auto calculated actual width and height value;

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_FILE_TEXT( pFile, 0,0, 0,-1,-1,"CH01","Arial",
                        QCAP_FONT_STYLE_BOLD, 12, 0xFF000000,
                       0xFFFFFFFF, 128, 0, 0, QCAP_SEQUENCE_STYLE_FOREMOST );
```

10.22 QCAP_SET_OSD_FILE_PICTURE

```
QRESULT QCAP_SET_OSD_FILE_PICTURE(   PVOID      pFile,
                                     UINT       iOsdNum,
                                     INT        x,
                                     INT        y,
                                     INT        width,
                                     INT        height,
                                     CHAR *     pszFilePathName,
                                     ULONG      nTransparent,
                                     ULONG      nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify they-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output If width is -1, QCAP will use original picture width as default.
height	**IN**, Specify the vertical height of OSD output. If height is -1, QCAP will use original picture height as default.
pszFilePathName	**IN**, Specify the file name to display image OSD output, support a file extension of "BMP" as 24 or 32-bit,“JPG” and “PNG”
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

When the users create a file object,

If you want to get a text field object auto calculated actual width and height value;

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_FILE_PICTURE( pFile, 0, 0, 0, -1, -1,
                           “C:\\SAMPLE.PNG”,
                           128,
                           QCAP_SEQUENCE_STYLE_FOREMOST );
```

10.23 QCAP_SET_OSD_FILE_BUFFER

```
QRESULT QCAP_SET_OSD_FILE_BUFFER( PVOID     pFile,
                                  UINT      iOsdNum,
                                  INT       x,
                                  INT       y,
                                  INT       width,
                                  INT       height,
                                  ULONG     nColorSpaceType,
                                  BYTE *    pFrameBuffer,
                                  ULONG     nFrameWidth,
                                  ULONG     nFrameHeight,
                                  ULONG     nFramePitch,
                                  ULONG     nTransparent,
                                  DWORD     dwKeyColor = 0xFFFFFFFF,
                                  ULONG     nKeyColorThreshold = 25,
                                  ULONGnSequenceStyle =  QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output
height	**IN**, Specify the vertical height of OSD output
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameWidth	**IN**, Specify the horizontal width of frame contained in a buffer
nFrameHeight	**IN**, Specify the vertical height of frame contained in a buffer
nFramePitch	**IN**, Specify the distance in bytes between the start of one scan line to the next. If it is 0, we will auto calculate it by width and color space format.

nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
dwKeyColor	**IN**, Specify the key color of broadcast server OSD, default is 0xFFFFFFFF
nKeyColorThreshold	**IN**, Specify the threshold of key color, the range of 0 to 128
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

When the users create a file object,

The function’s purpose is used to point directly to a valid frame buffer whose entire pixel data,

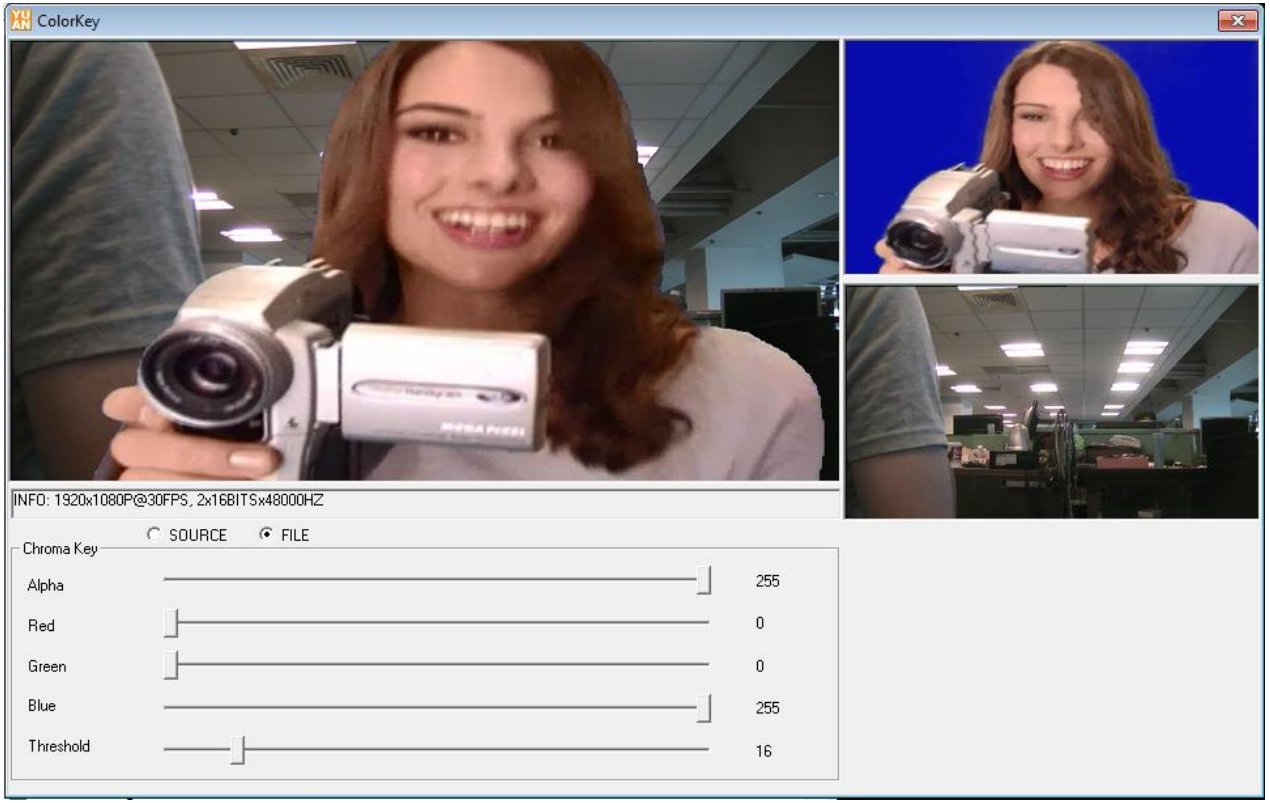
which is stored in available color space, and display it.

Beside, you also can adjust the width and height parameters to scale the OSD output.

Note

User can use this dwKeyColor parament to select colorkey mode Green&Blue mode and RGB mode.

For instance the Blue mode way as show in below



Example

```
QCAP_SET_OSD_FILE_BUFFER( pFile, 0, 0, 0, 1920, 1080, QCAP_COLORSPACE_TYEP_YUY2, pFrameBuffer,

                          800, 600, 0, 128, 0xFFFFFFFF, 25, QCAP_SEQUENCE_STYLE_FOREMOST );
```

10.24 QCAP_SCF_FILE_ENUMERATION

```
QRESULT QCAP_SCF_FILE_ENUMERATION( UINT        iChNum,
                                   double      dStartSearchTime,
                                   double      dStopSearchTime,
                                   ULONG    *  pFileSizeHigh,
                                   ULONG    *  pFileSizeLow,
                                   Double   *  pFileStartTime,
                                   Double   *  pFileStopTime,
                                   Double   *  pVideoStartTime,
                                   Double   *  pVideoStopTime,
                                   Double   *  pAudioStartTime,
                                   Double   *  pAudioStopTime,
                                   ULONG    *  pVideoEncoderFormat,
                                   ULONG    *  pVideoWidth,
                                   ULONG    *  pVideoHeight,
                                   Double   *  pVideoFrameRate,
                                   ULONG    *  pAudioEncoderFormat,
                                   ULONG    *  pAudioChannels,
                                   ULONG    *  pAudioBitsPerSample,
                                   ULONG    *  pAudioSampleFrequency,
                                   Double   *  pTotalDurationTimes,
                                   ULONG    *  pTotalVideoFrames,
                                   ULONG    *  pTotalAudioFrames,
```

```
BOOL        bNext = FALSE )
```

Parameter

Name	Description
iChNum	**IN**, Specify channel number to get SCF file parameters, start from 0
dStartSearchTime	**IN**, Specify the start search time of SCF file
dStopSearchTime	**IN**, Specify the stop search time of SCF file
pFileSizeHigh	**OUT**, Pointer to the storage of file size high
pFileSizeLow	**OUT**, Pointer to the storage of file size low
pFileStartTime	**OUT**, Pointer to the storage of file start time
pFileStopTime	**OUT**, Pointer to the storage of file stop time
pVideoStartTime	**OUT**, Pointer to the storage of video start time
pVideoStopTime	**OUT**, Pointer to the storage of video stop time
pAudioStartTime	**OUT**, Pointer to the storage of audio start time
pAudioStopTime	**OUT**, Pointer to the storage of audio stop time
pVideoEncoderFormat	**OUT**, Pointer to the storage of video encoder format
pVideoWidth	**OUT**, Pointer to the storage of video width
pVideoHeight	**OUT**, Pointer to the storage of video height
pVideoFrameRate	**OUT**, Pointer to the storage of video frame rate
pAudioEncoderFormat	**OUT**, Pointer to the storage of audio encoder format

pAudioChannels	**OUT**, Pointer to the storage of audio channels
pAudioBitsPerSample	**OUT**, Pointer to the storage of audio bits per sample
pAudioSampleFrequency	**OUT**, Pointer to the storage of audio sample frequency
pTotalDurationTimes	**OUT**, Pointer to the storage of total duration times
pTotalVideoFrames	**OUT**, Pointer to the storage of total video frames
pTotalAudioFrames	**OUT**, Pointer to the storage of total audio frames
bNext	**IN**, Specifies whether to use number n of the SCF file on the folder. The value FALSE is to use the first SCF file on the folder.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail



Introduction

User can use this function to get information enumeration of format from SCF file, such as video EncoderFormat, FramRate and audio sample frequency …

The pFileSizeHigh and pFileSizeLow are similar HighPart and LowPart.

The MSDN library provides reference material about HighPart and LowPart.

Example

```
QCAP_SCF_FILE_ENUMERATION( 0, 0, 0, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, &m_nVideoFormat,
                           &m_nVideoWidth, &m_nVideoHeight, &m_dVideoFrameRate, &m_nAudioFormat,
                           &m_nAudioChannels, &m_nAudioBitsPerSample, &m_nAudioSampleFrequency,
                           &m_dFileTotalDuationTimes, &m_nFileTotalVideoFrames, &m_nFileTotalAudioFrames,
                           TRUE );
```

10.25 QCAP_OPEN_SCF_FILE

```
QRESULT QCAP_OPEN_SCF_FILE( UINT        iChNum,
                            PVOID *     ppFile,
                            ULONG       nDecoderType,
                            HWND        hAttachedWindow,
                            BOOL        bThumbDraw = FALSE,
                            BOOL        bMaintainAspectRatio = FALSE,
                            double      dStartSampleTime = 0.0,
                            double      dStopSampleTime = 0.0 )
```

Parameter

Name	Description
iChNum	**IN**, Specify channel number to get SCF file, start from 0
ppFile	**OUT**, Handle of the file object
nDecoderType	**IN**, Specify decoder type. Available value are QCAP_DECODER_TYPE_SOFTWARE = 0 QCAP_DECODER_TYPE_INTEL_MEDIA_SDK = 2
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer
bMaintainAspectRatio	**IN**, Enable/Disable the maintain aspect ratio
dStartSampleTime	**IN**, Specify the start sample time, in sec unit
dStopSampleTime	**IN**, Specify the stop sample time, in sec unit

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

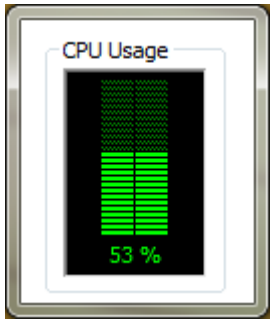
Introduction

User can use this function to create one video SCF file used for onscreen playback.

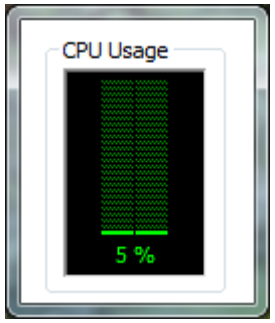
If you want to use editing mode to access this file, you can set the hAttachedWindowas NULL.

The thumb draw function can reduce the loading of video display, when display window size is small than the input video size.

For multiple channels developer, we suggest to turn on it in order to obtain the best rendering performance.



Normal Display



Thumb Draw Mode

Example

```
QCAP_OPEN_SCF_FILE( 0, &m_pFile, QCAP_DECODER_TYPE_SOFTWARE, m_statWindow.m_hWnd, TRUE, FALSE, 0, 0 );
```

10.26 QCAP_SET_SCF_FILE_TIMER

```
QRESULT QCAP_SET_SCF_FILE_TIMER( UINT      iChNum,
                                 BOOL      bEnableGlobalTimer )
```

Parameter

Name	Description
iChNum	**IN**, Specify channel number to get SCF file, start from 0
bEnableGlobalTimer	**IN**, Enable/Disable the global time

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to enable SCF time.

Example

```
QCAP_SET_SCF_FILE_TIMER( 0, FALSE );
```

10.27 QCAP_GET_GLOBAL_SCF_FILE_POSITION

```
QRESULT QCAP_GET_GLOBAL_SCF_FILE_POSITION( double *  pSampleTime )
```

Parameter

Name	Description
pSampleTime	**OUT**, Pointer to the storage of sample time

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get global SCF file position sample time.

Example

```
QCAP_GET_GLOBAL_SCF_FILE_POSITION( &nSampleTime );
```

10.28 QCAP_SET_GLOBAL_SCF_FILE_POSITION

```
QRESULT QCAP_SET_GLOBAL_SCF_FILE_POSITION( double dSampleTime )
```

Parameter

Name	Description
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set global SCF file position sample time.

Example

```
QCAP_SET_GLOBAL_SCF_FILE_POSITION( nSampleTime );
```

10.29 QCAP_OPEN_3D_FILE

```
QRESULT QCAP_OPEN_3D_FILE( CHAR    *  pszFileName,
                           PVOID   *  ppFile,
                           BOOL    *  pIs3D,
                           ULONG      nDecoderType,
                           ULONG   *  pVideoEncoderFormat,
                           ULONG   *  pVideoWidth,
                           ULONG   *  pVideoHeight,
                           double  *  pVideoFrameRate,
                           ULONG   *  pAudioEncoderFormat,
                           ULONG   *  pAudioChannels,
                           ULONG   *  pAudioBitsPerSample,
                           ULONG   *  pAudioSampleFrequency,
                           Double  *  pTotalDurationTimes,
                           ULONG   *  pTotalVideoFrames,
                           ULONG   *  pTotalAudioFrames,
                           HWND       hAttachedWindowL,
                           BOOL       bThumbDrawL = FALSE,
                           BOOL       bMaintainAspectRatioL = FALSE,
                           HWND       hAttachedWindowR = NULL,
                           BOOL       bThumbDrawR = FALSE,
                           BOOL       bMaintainAspectRatioR = FALSE )
```

Parameter

Name	Description
pszFileName	**IN**, Specify the video file name to display video output, support a file extension of "TS"
ppFile	**OUT**, Handle of the file object
pIs3D	**OUT**, If true, input file is 3D file. If false, input file is 2D file.

nDecoderType	**IN**, Specify decoder type. Available value are
	`QCAP_DECODER_TYPE_SOFTWARE = 0`
	`QCAP_DECODER_TYPE_INTEL_MEDIA_SDK = 2`

pVideoFormat	**OUT**, Pointer to the storage of the video format
pVideoWidth	**OUT**, Pointer to the storage of the video width
pVideoHeight	**OUT**, Pointer to the storage of the video height
pVideoFrameRate	**OUT**, Pointer to the storage of the video frame rate
pAudioEncoderFormat	**OUT**, Pointer to the storage of the audio format
pAudioChannels	**OUT**, Pointer to the storage of the audio channel
pAudioBitsPerSample	**OUT**, Pointer to the storage of the audio bit per sample
pAudioSampleFrequency	**OUT**, Pointer to the storage of the audio sample frequency
pTotalDurationTimes	**OUT**, Pointer to the storage of the total duration times
pTotalVideoFrames	**OUT**, Pointer to the storage of the total video frames
pTotalAudioFrames	**OUT**, Pointer to the storage of the audio frames
hAttachedWindowL	**IN**, Handle of the window to show the preview video of left side. If it is NULL, the file will be used at Editing mode only.

bThumbDrawL	**IN**, Enable/Disable the left side thumb draw renderer
bMaintainAspectRatioL	**IN**, Enable/Disable the left side maintain aspect ratio
hAttachedWindowR	**IN**, Handle of the window to show the preview video of right side. If it is NULL, the file will be used at Editing mode only.
bThumbDrawR	**IN**, Enable/Disable the right side thumb draw renderer
bMaintainAspectRatioR	**IN**, Enable/Disable the right side maintain aspect ratio

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

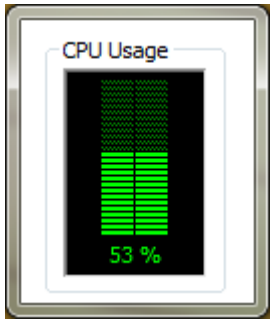
Introduction

User can use this function to create 3D video file or 2D video file used for on screen playback.

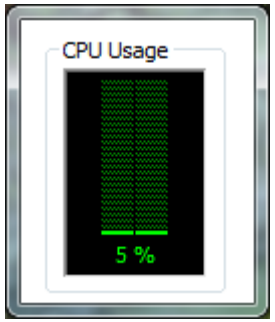
If you want to use editing mode to access this file, you can set the hAttachedWindowas NULL.

The thumb draw function can reduce the loading of video display, when display window size is small than the input video size.

For multiple channels developer, we suggest to turn on it in order to obtain the best rendering performance.



Normal Display



Thumb Draw Mode

Example

```
QCAP_OPEN_3D_FILE( "D:\\DEMO3D.TS", &pFile, &pIs3D, QCAP_DECODER_TYPE_SOFTWARE, &nVideoFormat,
                   &nVideoWidth, &nVideoHeight, &nVideoFrameRate, &nAudioFormat, &nAudioChannels,
                   &nAudioBitsPerSample, &nAudioSampleFrequency, &nTotalDurationTimes, &nTotalVideoFrames,
                   &nTotalAudioFrames, HAttachedWindowL, TRUE, FALSE, HAttachedWindowR, TRUE, FALSE  );
```

10.30 QCAP_GET_VIDEO_3D_FILE_DISPLAY_MODE

```
QRESULT QCAP_GET_VIDEO_3D_FILE_DISPLAY_MODE(   PVOID     pFile,
                                               ULONG  *   pStereoDisplayMode,
                                               BOOL   *   pLeftRightSwap )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pStereoDisplayMode	**OUT**, Pointer to the storage of current Stereo Display Mode
pLeftRightSwap	**OUT**, Pointer to the storage of current Left Right Swap

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current 3D file display mode.

Example

```
QCAP_GET_VIDEO_3D_FILE_DISPLAY_MODE( pFile, &nStereoDisplayMode, &nLeftRightSwap );
```

10.31 QCAP_SET_VIDEO_3D_FILE_DISPLAY_MODE

```
QRESULT QCAP_SET_VIDEO_3D_FILE_DISPLAY_MODE(   PVOID     pFile,
                                               ULONG     nStereoDisplayMode,
                                               BOOL      bLeftRightSwap )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
nStereoDisplayMode	**IN**, Specify 3D display mode. Available values are QCAP_3D_STEREO_DISPLAY_MODE_SIDE_BY_SIDE = 0 QCAP_3D_STEREO_DISPLAY_MODE_TOP_BOTTOM = 1 QCAP_3D_STEREO_DISPLAY_MODE_LINE_BY_LINE = 2 QCAP_3D_STEREO_DISPLAY_MODE_LEFT_ONLY = 3 QCAP_3D_STEREO_DISPLAY_MODE_RIGHT_ONLY = 4
bLeftRightSwap	**IN**, Change the input camera source's (right to left, left to right), the default is FALSE

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set current 3D file display mode.

Example

```
QCAP_SET_VIDEO_3D_FILE_DISPLAY_MODE( pFile, 0, FALSE );
```

10.32 QCAP_GET_VIDEO _FILE _STREAM_BUFFER

```
QRESULT QCAP_GET_VIDEO_FILE_STREAM_BUFFER( PVOID      pFile,
                                           UINT       iFrameNum,
                                           BYTE   *   pStreamBuffer,
                                           ULONG  *   pStreamBufferLen,
                                           BOOL   *   pIsKeyFrame,
                                           double *   pSampleTime )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
iFrameNum	**IN**, Specify the number of frame
pStreamBuffer	**OUT**, Pointer to the storage of the input source buffer
pStreamBufferLen	**IN**, Specify the max input source buffer's size **OUT**, Pointer to the storage of actual input source buffer's size
pIsKeyFrame	**OUT**, Pointer to the storage of the key frame
pSampleTime	**OUT**, Pointer to the storage of sample time

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get information from video stream buffer.

Example

```
QCAP_GET_VIDEO_FILE_STREAM_BUFFER( pFile, 100, &nStreamBuffer, &nStreamBufferLen, &nIsKeyFrame,
                                   &nSampleTime );
```

10.33 QCAP_GET_ AUDIO_FILE_ STREAM_BUFFER

```
QRESULT QCAP_GET_AUDIO_FILE_STREAM_BUFFER( PVOID      pFile,
                                           UINT       iFrameNum,
                                           BYTE   *   pStreamBuffer,
                                           ULONG  *   pStreamBufferLen,
                                           double *   pSampleTime )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
iFrameNum	**IN**, Specify the number of frame
pStreamBuffer	**OUT**, Pointer to the storage of the input source buffer
pStreamBufferLen	**IN**, Specify the max input source buffer's size **OUT**, Pointer to the storage of actual input source buffer's size
pSampleTime	**OUT**, Pointer to the storage of sample time

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get information from audio stream buffer.

Example

```
QCAP_GET_AUDIO_FILE_STREAM_BUFFER( pFile, 100, &nStreamBuffer, &nStreamBufferLen, &nSampleTime );
```

10.34 QCAP_EXPORT_FILE

```
QRESULT QCAP_EXPORT_FILE(  PVOID     pFile,
                           double    dStartSampleTime,
                           double    dStopSampleTime,
                           CHAR *    pszExportedFileName )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
dStartSampleTime	**IN**, Specify the start sample time, in sec unit
dStopSampleTime	**IN**, Specify the stop sample time, in sec unit
pszExportedFileName	**IN**, Specify the file name to store data, support a file extension of "MP4"

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

It is one helper function.

User can use this function to export one file segment from start sample time to stop start sample time.

Example

```
QCAP_EXPORT_FILE( pFile, 5.000, 10.000, “C:\\EXPORT.MP4” );
```

10.35 QCAP_MERGE_FILE

```
QRESULT QCAP_MERGE_FILE(   PVOID      pFrontEndFile,
                           PVOID      pBackEndFile,
                           CHAR *     pszMergedFileName )
```

Parameter

Name	Description
pFrontEndFile	**IN**, Specify the front end file
pBackEndFile	**IN**, Specify the back end file
pszMergedFileName	**IN**, Specify the merged file name, support a file extension of "MP4"

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

It is one helper function.

User can use this function to merge two files into one file.

Note

The recording format of two files must to be same.

You cannot combine one 1920x1080 file and one 1280x720 file together.

Example

```
QCAP_MERGE_FILE( pFrontEndFile, pBackEndFile, “C:\\MERGE.MP4” );
```

10.36 QCAP_MERGE_FILES_EX

```
QRESULT QCAP_MERGE_FILES_EX(  CHAR *    pszMergedFileName,
                              ULONG     nFileArgs,
                              CHAR *    pszSourceFileName1,
                              CHAR *    pszSourceFileName2, … )
```

Parameter

Name	Description
pszMergedFileName	**IN**, Specify the merged file name, support a file extension of "MP4"
nFileArgs	**IN**, Specify the merage files number
pszSourceFileName1	**IN**, Specify the first merged file name, support a file extension of "MP4"
pszSourceFileName2	**IN**, Specify the second merged file name, support a file extension of "MP4"

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

It is one helper function. User can use this function to merge many files into one file.

Note

The recording format of two files must to be same.

You cannot combine one 1920x1080 file and one 1280x720 file together.

Example

```
QCAP_MERGE_FILES_EX( "MERGE.MP4", 4, "1.MP4", "2.MP4", "3.MP4", "4.MP4"  );
```

10.37 QCAP_BUILD_VIDEO_ON_DEMAND_FILE

```
QRESULT QCAP_BUILD_VIDEO_ONDEMAND_FILE( CHAR *    pszOriginalFileName,
                                        CHAR *    pszVideoOnDemandFileName )
```

Parameter

Name	Description
pszOriginalFileName	**IN**, Specify the orignal video file name
pszVideoOnDemandFileName	**IN**, Specify the on demand video file name

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to convert the recording mp4 to one vod (video on demand) file.

Example

```
QCAP_BUILD_VIDEO_ONDEMAND_FILE( "File.MP4", "Other.MP4" );
```

10.38 QCAP_DIAGNOSE_FILE

```
QRESULT QCAP_DIAGNOSE_FILE(    CHAR *     pszBadFileName,
                               BOOL *     pIsHealthy )
```

Parameter

Name	Description
pszBadFileName	**IN**, Specify the video file name to display video output
pIsHealthy	**OUT**, Pointer to the storage of the file healthy state

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

It is import helper function. User can use this function to get file healthy state.

Example

```
QCAP_DIAGNOSE_FILE( "File.MP4", &nIshealthy );
```

10.39 QCAP_REPAIR_FILE

```
QRESULT QCAP_REPAIR_FILE(  CHAR *    pszBadFileName,
                           CHAR *    pszRepairedFileName )
```

Parameter

Name	Description
pszBadFileName	**IN**, Specify the bad file name, support a file extension of "MP4"
pszRepairedFileName	**IN**, Specify the first repaired file name, support a file extension of "MP4"

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

It is import helper function. User can use this function to fix mp4 file.

Note

Developer need set bEnableFileRepairFunction parameter of QCAP_SET_SYSTEM_CONFIGURATION is true, then the recording file will

Developer need set bEnableFileRepairFunction parameter is true the QCAP_SET_SYSTEM_CONFIGURATION

It is only support our capture card record file.

Example

```
QCAP_REPAIR_FILE( "BadFile.MP4", "FixFile.MP4" );
```

11 Broadcasting Server& Client Function API

11.1 Overview

In this section of the document, you will understand how to design a streaming server or RTSP, RTMP and HLS.

They all can be designed into your software and are enabled at the same time.

For different streaming protocols, only their created function is different in QCAP.

For example, if you had already used `QCAP_CREATE_BROADCAST_RTSP_SERVER` to create one RTSP server,

You can change it to `QCAP_CREATE_BROADCAST_HLS_SERVER` to enable HLS server.

Other programming steps are same as RTSP.

In document as below, we will use RTSP as one basic sample to describe how to design one streaming server.

Beside of streaming server, in our SDK, we also support the RTSP client function to allow you to obtain the video and audio streams from the server side application, such as VLC Player and QuickTime Player and so on.

For RTMP, we support RTMP Web Portal too. RTMP portal allows you to send stream to RTMP server, such as Adobe Flash Media Server.

All protocols, you can find its sample from our SDK samples folder.

Programming Workflow of BroadcastServer Object

Programming Step	Related API Functions
Create Broadcast Server Object	`QCAP_CREATE_BROADCAST_RTSP_SERVER( …, &pServer )`
Register Callback Function	`QCAP_REGISTER_VIDEO_BROADCAST_SERVER_CALLBACK(pServer, …);` `QCAP_REGISTER_AUDIO_BROADCAST_SERVER_CALLBACK(pServer, …);`
Set Video / Audio Property	`QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY( pServer, … )` `QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY( pServer, … )`
StartVideo / Audio Broadcast Server	`QCAP_START_BROADCAST_SERVER(pServer )`

Push Uncompression or Compression Buffer	`QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(pServer, … )`
	`QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(pServer, … )`
	`QCAP_SET_VIDEO_BROADCAST_SERVER_COMPRESSION_BUFFER(pServer, … )`
	`QCAP_SET_AUDIO_BROADCAST_SERVER_COMPRESSION_BUFFER(pServer, … )`

StopVideo / Audio Broadcast Server	`QCAP_STOP_BROADCAST_SERVER(pServer )`
Delete BroadcastServer Object	`QCAP_DESTROY_BROADCAST_SERVER(pServer )`

Programming Workflow of BroadcastClient Object

Programming Step	Related API Functions
Create Broadcast Client Object	`QCAP_CREATE_BROADCAST_CLIENT ( …, &pClient, … )`
Register Callback Function	`QCAP_REGISTER_VIDEO_BROADCAST_CLIENT_CALLBACK(pClient, … );` `QCAP_REGISTER_AUDIO_BROADCAST_CLIENT_CALLBACK(pClient, … );`
StartVideo / Audio Broadcast Client	`QCAP_START_BROADCAST_CLIENT(pClient )`
StopVideo / Audio Broadcast Client	`QCAP_STOP_BROADCAST_CLIENT(pClient )`
Delete Broadcast Client Object	`QCAP_DESTROY_BROADCAST_CLIENT(pClient )`

About RTMP of Adobe License

This software development kit contains software programming code and libraries related to RTMP of Adobeconnections that is subject to yourdevelopment use only. All the intellectual properties are held by individual license groups and/or the property holders. License and royalty fee should be charged by the license groups and/or property holders and paid by you if any of them is implemented into your commercial product. We will not be responsible for any illegal usage and/or any infringement of the rights of the individual owners of these intellectual properties.

Details of RTMP of Adobe could be found in the website:

http://www.adobe.com/products/eulas/

11.2 QCAP_CREATE_BROADCAST_RTSP_SERVER

```
QRESULT QCAP_CREATE_BROADCAST_RTSP_SERVER( UINT       iSvrNum,
                                           ULONG      nTotalSessions,
                                           PVOID  *   ppServer,
                                           CHAR   *   pszAccount = NULL,
                                           CHAR   *   pszPassword = NULL,
                                           ULONG      nNetworkPort_RTSP = 554,
                                           ULONG      nNetworkPort_RTSPOverHTTP = 0,
                                           BOOL       bEnableMulticasting = FALSE )
```

Parameter

Name	Description
iSvrNum	**IN**, Specify broadcast server number to get server parameters, start from 0, max is 0~63
nTotalSessions	**IN**, Specify the total number of session connections, max is 8
ppServer	**OUT**, Handle of the broadcast server object
pszAccount	**IN**, Specify the server side account information to login, the default string is NULL
pszPassword	**IN**, Specify the server side password information to login, the default string is NULL
NetworkPort_RTSP	**IN**, Specify the port number that the server listens on for RTSP, the default port is 554
NetworkPort_RTSPOverHTTP	**IN**, Specify the port number that the server listens on for RTSP overHTTP requests, the default port is 8080. 0 is disable this function.
bEnableMulticasting	**IN**, Enable/Disable the multicasting

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a broadcast server object.

iSvrNum can allow you to create multiple broadcast servers with different port.

nTotalSessionsdepends on the total number of channels of user's motherboard.

Every session includes one video stream and one audio stream.

In the client side, user can use the URL, rtsp://account:password@ip:port/session#num.mpg to access it.

For example, rtsp://root:root@10.10.10.123:554/session0.mpg.

Example

```
QCAP_CREATE_BROADCAST_RTSP_SERVER( 0, 4, &pServer, “root”, “root”, 554, 0, FALSE, FALSE )
```

11.3 QCAP_CREATE_BROADCAST_RTMP_SERVER

```
QRESULT QCAP_CREATE_BROADCAST_RTMP_SERVER( UINT       iSvrNum,
                                           ULONG      nTotalSessions,
                                           PVOID  *   ppServer,
                                           CHAR   *   pszAccount = NULL,
                                           CHAR   *   pszPassword = NULL,
                                           ULONG      nNetworkPort_RTMP = 1935,
                                           ULONG      nNetworkPort_RTMPOverHTTP = 0,
                                           CHAR   *   pszServerName = NULL )
```

Parameter

Name	Description
iSvrNum	**IN**, Specify broadcast RTMP server number to get server parameters, start from 0, max is 0~63
nTotalSessions	**IN**, Specify the total number of session connections, max is 8
ppServer	**OUT**, Handle of the broadcast server object
pszAccount	**IN**, Specify the server side account information to login, the default string is NULL
pszPassword	**IN**, Specify the server side password information to login, the default string is NULL
NetworkPort_RTMP	**IN**, Specify the port number that the server listens on for RTMP, the default port is 1935
NetworkPort_RTMPOverHTTP	**IN**, Specify the port number that the server listens on for RTMP overHTTP requests, the default port is 8080. 0 is disable this function.
pszServerName	**IN**, Specify the server name, the default name is flvplayback

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a broadcast RTMP server object.

iSvrNum can allow you to create multiple broadcast RTMP servers with different port.

nTotalSessions can create multiple sessions from one server.

Every session includes one video stream and one audio stream.

In the client side, user can use the URL, http://localhost/NetReceiver.Example/ to access it.

Note

1) If user change port number for example 1402 port, the URL of RTMP also need to add 1402 port as show in below

   rtmp://127.0.0.1:1402/Live/session0.mpg

2) User can use pszServerName to set server name.And if server name is “LiveRTMPServer” and the URL of RTMP is set as below”.

   rtmp://127.0.0.1/LiveRTMPServer/session0.mpg

3) If users want to use RTMP Server function, please read Yuan's QCAP SDK RTMP Server Installation Guide ENG firstly.

   Theencoderaudioformatshould set QCAP_ENCODER_FORMAT_AAC_ADTS.

Example

```
QCAP_CREATE_BROADCAST_RTMP_SERVER( 0, 1, &pServer, “root”, “root”, 1935, 0, NULL );
```

11.4 QCAP_CREATE_BROADCAST_RTMP_SERVER_EX

```
QRESULT QCAP_CREATE_BROADCAST_RTMP_SERVER_EX(  UINT       iSvrNum,
                                               ULONG      nTotalSessions,
                                               PVOID  *   ppServer,
                                               CHAR   *   pszAccount = NULL,
                                               CHAR   *   pszPassword = NULL,
                                               ULONG      nNetworkPort_RTMP = 1935,
                                               ULONG      nNetworkPort_RTMPOverHTTP = 0,
                                               CHAR   *   pszServerName = NULL )
```

Parameter

Name	Description
iSvrNum	**IN**, Specify broadcast RTMP server number to get server parameters, start from 0, max is 0~63
nTotalSessions	**IN**, Specify the total number of session connections, max is 8
ppServer	**OUT**, Handle of the broadcast server object
pszAccount	**IN**, Specify the server side account information to login, the default string is NULL
pszPassword	**IN**, Specify the server side password information to login, the default string is NULL
NetworkPort_RTMP	**IN**, Specify the port number that the server listens on for RTMP, the default port is 1935
NetworkPort_RTMPOverHTTP	**IN**, Specify the port number that the server listens on for RTMP overHTTP requests, the default port is 8080. 0 is disable this function.
pszServerName	**IN**, Specify the server name, the default name is flvplayback

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a broadcast RTMP server object.

iSvrNum can allow you to create multiple broadcast RTMP servers with different port.

nTotalSessions can create multiple sessions from one server.

Every session includes one video stream and one audio stream.

In the client side, user can use the URL, http://localhost/NetReceiver.Example/ to access it.

Note

4) If user change port number for example 1402 port, the URL of RTMP also need to add 1402 port as show in below

   rtmp://127.0.0.1:1402/Live/session0.mpg

5) User can use pszServerName to set server name.And if server name is “LiveRTMPServer” and the URL of RTMP is set as below”.

   rtmp://127.0.0.1/LiveRTMPServer/session0.mpg

6) If users want to use RTMP Server function, please readYuan's QCAP SDK RTMP Server Installation Guide ENG firstly.

   Theencoderaudioformatshould set QCAP_ENCODER_FORMAT_AAC_ADTS.

Example

```
QCAP_CREATE_BROADCAST_RTMP_SERVER_EX( 0, 1, &pServer, “root”, “root”, 1935, 0, NULL );
```

11.5 QCAP_CREATE_BROADCAST_RTMP_WEB_PORTAL_SERVER

```
QRESULT QCAP_CREATE_BROADCAST_RTMP_WEB_PORTAL_SERVER( UINT      iSvrNum,
                                                      CHAR  *   pszURL,
                                                      PVOID *   ppServer,
                                                      CHAR  *   pszAccount = NULL,
                                                      CHAR  *   pszPassword = NULL )
```

Parameter

Name	Description
iSvrNum	**IN**, Specify broadcast serve rnumber to get server parameters, start from 0, max is 0~63
pszURL	**IN**, In the client side,user can use the URL to accessit. For example, "`rtmp://10.10.0.80/live/session0.mpg`"
ppServer	**OUT**, Handle of the broadcast serverobject
pszAccount	**IN**, Specify the server side account information to login, the default string is NULL
pszPassword	**IN**, Specify the server side password information to login, the default string is NULL

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a broadcast RTMP web portal servers object.

iSvrNum can allow you to create multiple broadcast RTMP web portals.

One portal can send one video stream and one audio stream to one RTMP server’s URL mapping.

User can use the same URL, "rtmp://10.10.0.80/live/session0.mpg", to access live streams from RTMP server.

Note!! If user want to set and create QCAP rtmp web portal server function, please read

Yuan's QCAP SDK RTMP Web Portal with Adobe FMS Security Account Installation Guide ENG 1.1.0.130.2

Example

```
QCAP_CREATE_BROADCAST_RTSP_RTMP_WEB_PORTAL_SERVER( 0, "rtmp://10.10.0.80/live/session0.mpg", &pServer,
                                                   “root”, “root” )
```

11.6 QCAP_CREATE_BROADCAST_RTMP_WEB_PORTAL_SERVER_EX

```
QRESULT QCAP_CREATE_BROADCAST_RTMP_WEB_PORTAL_SERVER_EX( UINT      iSvrNum,
                                                         CHAR   *  pszURL,
                                                         PVOID  *  ppServer,
                                                         CHAR   *  pszAccount = NULL,
                                                         CHAR   *  pszPassword = NULL )
```

Parameter

Name	Description
iSvrNum	**IN**, Specify broadcast serve rnumber to get server parameters, start from 0, max is 0~63
pszURL	**IN**, In the client side, user can use the URL to accessit. For example, "`rtmp://10.10.0.80/live/session0.mpg`"
ppServer	**OUT**, Handle of the broadcast server object
pszAccount	**IN**, Specify the server side account information to login, the default string is NULL
pszPassword	**IN**, Specify the server side password information to login, the default string is NULL

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a broadcast RTMP web portal servers object.

iSvrNum can allow you to create multiple broadcast RTMP web portals.

One portal can send one video stream and one audio stream to one RTMP server’s URL mapping.

User can use the same URL, "rtmp://10.10.0.80/live/session0.mpg", to access live streams from RTMP server.

Note!! If user want to set and create QCAP rtmp web portal server function, please read

Yuan's QCAP SDK RTMP Web Portal with Adobe FMS Security Account Installation Guide ENG 1.1.0.130.2

Example

```
QCAP_CREATE_BROADCAST_RTSP_RTMP_WEB_PORTAL_SERVER_EX( 0, "rtmp://10.10.0.80/live/session0.mpg",
                                                      &pServer, “root”, “root” )
```

11.7 QCAP_CREATE_BROADCAST_MMS_WEB_PORTAL_SERVER

```
QRESULT QCAP_CREATE_BROADCAST_MMS_WEB_PORTAL_SERVER( UINT       iSvrNum,
                                                     CHAR   *   pszURL,
                                                     PVOID  *   ppServer,
                                                     CHAR   *   pszAccount = NULL,
                                                     CHAR   *   pszPassword = NULL )
```

Parameter

Name	Description
iSvrNum	**IN**, Specify broadcast serve rnumber to get server parameters, start from 0, max is 0~63
pszURL	**IN**, In the client side,user can use the URL to accessit. For example, "mms://127.0.0.1:41794/MMS"
ppServer	**OUT**, Handle of the broadcast serverobject
pszAccount	**IN**, Specify the server side account information to login, the default string is NULL
pszPassword	**IN**, Specify the server side password information to login, the default string is NULL

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a broadcast mms web portal servers object.

iSvrNum can allow you to create multiple broadcast mms web portals.

One portal can send one video stream and one audio stream to one mms server’s URL mapping.

User can use the same URL, "mms://127.0.0.1:41794/MMS", to access live streams from VLC Player.

Example

```
QCAP_CREATE_BROADCAST_RTSP_MMS_WEB_PORTAL_SERVER( 0, "mms://127.0.0.1:41794/MMS", &pServer,
                                                  “root”, “root” )
```

11.8 QCAP_CREATE_BROADCAST_HLS_SERVER

```
QRESULT QCAP_CREATE_BROADCAST_HLS_SERVER(   UINT       iSvrNum,
                                            ULONG      nTotalSessions,
                                            PVOID  *   ppServer,
                                            CHAR   *   pszWebServerRootFolderPath,
                                            CHAR   *   pszSubFolderPath,
                                            ULONG      nSegmentDuration )
```

Parameter

Name	Description
iSvrNum	**IN**, Specify broadcast server number to get server parameters, start from 0, max is 0~63
nTotalSessions	**IN**, Specify the total number of session connections, max is 8
ppServer	**OUT**, Handle of the broadcast server object
pszWebServerRootFolderPath	**IN**, Specify the web server's root folder path
pszSubFolderPath	**IN**, Specify the sub folder path to save HLS files.
nSegmentDuration	**IN**, Set the segment duration, the default number is 3000 msec.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a broadcast HLS servers object.

iSvrNum can allow you to create multiple broadcastHLSservers.

Every session includes one video stream and one audio stream and user can use nSegmentDuration to control the create session time.

If nSegmentDuration set large, the player video will more stable.

User can use pszWebServerRootFolderPath and pszSubFolderPath parameters to set Web server root folder path with sub folder path.

Note

When user use below API, we suggest user must to set dSampleTime. Due to HLS need accurate Timestamp.

```
QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER,

QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER,

QCAP_SET_VIDEO_BROADCAST_SERVER_COMPRESSION_BUFFER,

QCAP_SET_AUDIO_BROADCAST_SERVER_COMPRESSION_BUFFER
```

Example

```
QCAP_CREATE_BROADCAST_HLS_SERVER( 0, 4 /*4 CHANNELS*/, &pServer, "C:\\AppServ\\www\\" "hls\\" );

User can use these URL to access different HLS session as below:

CH01 URL -> http://xxx.xxx.xxx.xxx/hls/session0.m3u8 (HERE, xxx.xxx.xxx.xxx IS YOUR WEB SERVER ADDRESS)

CH02 URL -> http://xxx.xxx.xxx.xxx/hls/session1.m3u8

CH03 URL -> http://xxx.xxx.xxx.xxx/hls/session2.m3u8

 CH04 URL -> http://xxx.xxx.xxx.xxx/hls/session3.m3u8
```

11.9 QCAP_DESTROY_BROADCAST_SERVER

```
QRESULT QCAP_DESTROY_BROADCAST_SERVER( PVOID pServer )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Destroy broadcast server object to release its resource.

Example

```
QCAP_DESTROY_BROADCAST_SERVER( pServer );
```

11.10 QCAP_START_BROADCAST_SERVER

```
QRESULT QCAP_START_BROADCAST_SERVER( PVOID pServer )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start broadcast server.

Note

User must set QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY and QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY first, when user set QCAP_START_BROADCAST_SERVER.

Example

```
QCAP_START_BROADCAST_SERVER( pServer );
```

11.11 QCAP_STOP_BROADCAST_SERVER

```
QRESULT QCAP_STOP_BROADCAST_SERVER( PVOID pServer )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop broadcast server.

Example

```
QCAP_STOP_BROADCAST_SERVER( pServer );
```

11.12 QCAP_GET_VIDEO_BROADCAST_SERVER_PROPERTY

```
QRESULT QCAP_GET_VIDEO_BROADCAST_SERVER_PROPERTY( PVOID     pServer,
                                                  UINT      iSessionNum,
                                                  ULONG  *  pEncoderType,
                                                  ULONG  *  pEncoderFormat,
                                                  ULONG  *  pColorSpaceType,
                                                  ULONG  *  pWidth,
                                                  ULONG  *  pHeight,
                                                  double *  pFrameRate,
                                                  ULONG  *  pRecordMode,
                                                  ULONG  *  pQuality,
                                                  ULONG  *  pBitRate,
                                                  ULONG  *  pGOP,
                                                  ULONG  *  pAspectRatioX,
                                                  ULONG  *  pAspectRatioY,
                                                  HWND   *  pAttachedWindow = NULL,
                                                  BOOL   *  pThumbDraw = NULL,
                                                  BOOL   *  pMaintainAspectRatio = NULL,
                                                  DWORD  *  pFlags = NULL )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pEncoderType	**OUT**, Pointer to the storage of the encoder type for the i-th video streamsession
pEncoderFormat	**OUT**, Pointer to the storage of the encoder format for the i-th video streamsession
pColorSpaceType	**OUT**, Pointer to the storage of the color space type for the i-th video streamsession
pWidth	**OUT**, Pointer to the storage of the horizontal width for the i-th video streamsession
pHeight	**OUT**, Pointer to the storage of the vertical height for the i-th video streamsession
pFrameRate	**OUT**, Pointer to the storage of the frame rate for the i-th video streamsession
pRecordMode	**OUT**, Pointer to the storage of the record mode for the i-th video streamsession
pQuality	**OUT**, Pointer to the storage of the quality for the i-th video streamsession
pBitRate	**OUT**, Pointer to the storage of the bit rate for the i-th video streamsession
pGOP	**OUT**, Pointer to the storage of the GOP size for the i-th video streamsession
pAspectRatioX	**OUT**, Pointer to the storage of the aspect ratio X axle for the i-th video streamsession
pAspectRatioY	**OUT**, Pointer to the storage of the aspect ratio Y axle for the i-th video streamsession

pAttachedWindow	**OUT**, Pointer to the storage of window to show the i-th video streamsession
pThumbDraw	**OUT**, Pointer to the enable flag of thumb draw renderer
pMaintainAspectRatio	**OUT**, Pointer to the enable flag of maintain aspect ratio
pFlags	**OUT**, Pointer to the enable flag of the i-th video streamsession

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get parameters of the i-th video stream session in broadcast server.

Example

```
QCAP_GET_VIDEO_BROADCAST_SERVER_PROPERTY(  pServer, 0, &nEncoderType,&nEncoderFormat, &nColorSpaceType,
                                           &nWidth, &nHeight, &nFrameRate, &nRecordMode, &nQuality,
                                           &nBitRate, &nGOP, &nAspectRatioX, &nAspectRatioY,
                                           &nAttachedWindow, &nThumbDraw, &nMaintainAspectRatio,
                                           &nFlags );
```

11.13 QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY

```
QRESULT QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY( PVOID     pServer,
                                                  UINT      iSessionNum,
                                                  ULONG     nEncoderType,
                                                  ULONG     nEncoderFormat,
                                                  ULONG     nColorSpaceType,
                                                  ULONG     nWidth,
                                                  ULONG     nHeight,
                                                  double    dFrameRate,
                                                  ULONG     nRecordMode,
                                                  ULONG     nQuality,
                                                  ULONG     nBitRate,
                                                  ULONG     nGOP,
                                                  ULONG     nAspectRatioX,
                                                  ULONG     nAspectRatioY,
                                                  HWND      hAttachedWindow = NULL,
                                                  BOOL      bThumbDraw = FALSE,
                                                  BOOL      bMaintainAspectRatio = FLASE,
                                                  DWORD     dwFlags = QCAP_BROADCAST_FLAG_FULL );
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_RGB24 CAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_I420
nWidth	**IN**, Specify the horizontal width for the i-th video streamsession

nHeight	**IN**, Specify the vertical height for the i-th video streamsession
dFrameRate	**IN**, Specify the frame rate for the i-th video streamsession

nRecordMode	**IN**, Specify recording mode. Available values are
	`QCAP_RECORD_MODE_VBR = 0,`
	`QCAP_RECORD_MODE_CBR = 1,`
	`QCAP_RECORD_MODE_ABR = 2,`

nQuality	**IN**, Specify video quality. It is used by VBR and ABR.
nBitRate	**IN**, Specify video bit rate. It is used by CBR and ABR. (Ex. 6Mbps is 6 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0 ~ 255
nAspectRatioX	**IN**, Specify the aspect ratio X axle, 0 is turned off
nAspectRatioY	**IN**, Specify the aspect ratio Y axle, 0 is turned off
hAttachedWindow	**IN**, Specify the handle of the window to show the i-th video stream session
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer
bMaintainAspectRatio	**IN**, Enable/Disable the maintain aspect ratio

nFlags	**IN**, Specify broadcasting mode. Available values are
	`QCAP_BROADCAST_FLAG_FULL = 0x00000007`
	`QCAP_BROADCAST_FLAG_NETWORK = 0x00000001`
	`QCAP_BROADCAST_FLAG_ENCODE = 0x00000002`
	`QCAP_BROADCAST_FLAG_DISPLAY = 0x00000004`
	`QCAP_BROADCAST_FLAG_DECODE = 0x00000008`
	`QCAP_BROADCAST_FLAG_VIDEO_ONLY = 0x00000010`
	`QCAP_BROADCAST_FLAG_AUDIO_ONLY = 0x00000020`

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set parameters of the i-th video stream session in broadcast server.

Example

```
QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY(  pServer, 0, QCAP_ENCODER_TYPE_SOFTWARE,
                                           QCAP_ENCODER_FORMAT_H264,
                                           QCAP_COLORSPACE_TYEP_YUY2, 320, 240,60,
                                           QCAP_RECORD_MODE_CBR, 8000, 12 * 1024 * 1024,
                                           30, 4, 3, hAttachedWindow, FALSE, QCAP_BROADCAST_FLAG_FULL );
```

11.14 QCAP_GET_VIDEO_BROADCAST_SERVER_PROPERTY_EX

```
QRESULT QCAP_GET_VIDEO_BROADCAST_SERVER_PROPERTY_EX( PVOID     pServer,
                                                     UINT      iSessionNum,
                                                     ULONG  *  pEncoderType,
                                                     ULONG  *  pEncoderFormat,
                                                     ULONG  *  pColorSpaceType,
                                                     ULONG  *  pWidth,
                                                     ULONG  *  pHeight,
                                                     double *  pFrameRate,
                                                     ULONG  *  pRecordProfile,
                                                     ULONG  *  pRecordLevel,
                                                     ULONG  *  pRecordEntropy,
                                                     ULONG  *  pRecordComplexity,
                                                     ULONG  *  pRecordMode,
                                                     ULONG  *  pQuality,
                                                     ULONG  *  pBitRate,
                                                     ULONG  *  pGOP,
                                                     ULONG  *  pBFrames,
                                                     BOOL   *  pIsInterleaved,
                                                     ULONG  *  pSlices,
                                                     ULONG  *  pLayers,
                                                     ULONG  *  pSceneCut,
                                                     ULONG  *  pAspectRatioX,
```

```
ULONG   *   pAspectRatioY,
HWND    *   pAttachedWindow = NULL,
BOOL    *   pThumbDraw = NULL,
BOOL    *   pMaintainAspectRatio = NULL,
DWORD   *   pFlags = NULL )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pEncoderType	**OUT**, Pointer to the storage of the encoder type for the i-th video streamsession
pEncoderFormat	**OUT**, Pointer to the storage of the encoder format for the i-th video streamsession
pColorSpaceType	**OUT**, Pointer to the storage of the color space type for the i-th video streamsession
pWidth	**OUT**, Pointer to the storage of the horizontal width for the i-th video streamsession
pHeight	**OUT**, Pointer to the storage of the vertical heightfor the i-th video streamsession
pFrameRate	**OUT**, Pointer to the storage of the frame rate for the i-th video streamsession
pRecordProfile	**OUT**, Pointer to the storage of the record profile for the i-th video streamsession
pRecordLevel	**OUT**, Pointer to the storage of the record level for the i-th video streamsession

pRecordEntropy	**OUT**, Pointer to the storage of the record entropy for the i-th video streamsession
pRecordComplexity	**OUT**, Pointer to the storage of the record complexity for the i-th video streamsession
pRecordMode	**OUT**, Pointer to the storage of the record mode for the i-th video streamsession
pQuality	**OUT**, Pointer to the storage of the quality for the i-th video streamsession
pBitRate	**OUT**, Pointer to the storage of the bit rate for the i-th video streamsession
pGOP	**OUT**, Pointer to the storage of the GOP size for the i-th video streamsession
pBFrames	**OUT**, Pointer to the storage of the B Frames
pIsInterleaved	**OUT**, Pointer to the storage of current Interleaved
pSlices	**OUT**, Pointer to the storage of current slices
pLayers	**OUT**, Pointer to the storage of current layers
pSceneCut	**OUT**, Pointer to the storage of current recording screen cut
pAspectRatioX	**OUT**, Pointer to the storage of the aspect ratio X axle for the i-th video streamsession
pAspectRatioY	**OUT**, Pointer to the storage of the aspect ratio Y axle for the i-th video streamsession
pAttachedWindow	**OUT**, Pointer to the storage of window to show the i-th video streamsession
pThumbDraw	**OUT**, Pointer to the enable flag of thumb draw renderer
pMaintainAspectRatio	**OUT**, Pointer to the enable flag of maintain aspect ratio
pFlags	**OUT**, Pointer to the enable flag of the i-th video streamsession

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get additional parameters of the i-th video stream session in broadcast server.

Example

```
QCAP_GET_VIDEO_BROADCAST_SERVER_PROPERTY_EX(   pServer, 0, &nEncoderType, &nEncoderFormat,
                                               &nColorSpaceType, &nWidth, &nHeight, &nFrameRate,
                                               &nRecordMode, &nQuality, &nBitRate, &nGOP, &nBFrames,
                                               &bIsInterleaved, &nSlices, &nLayers, &nSceneCut,
                                               &nAspectRatioX, &nAspectRatioY, &nAttachedWindow,
                                               &nThumbDraw, &nMaintainAspectRatio, nFlags );
```

11.15 QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY_EX

```
QRESULT QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY_EX( PVOID     pServer,
                                                     UINT      iSessionNum,
                                                     ULONG     nEncoderType,
                                                     ULONG     nEncoderFormat,
                                                     ULONG     nColorSpaceType,
                                                     ULONG     nWidth,
                                                     ULONG     nHeight,
                                                     double    dFrameRate,
                                                     ULONG     nRecordProfile,
                                                     ULONG     nRecordLevel,
                                                     ULONG     nRecordEntropy,
                                                     ULONG     nRecordComplexity,
                                                     ULONG     nRecordMode,
                                                     ULONG     nQuality,
                                                     ULONG     nBitRate,
                                                     ULONG     nGOP,
                                                     ULONG     nBFrames,
                                                     BOOL      bIsInterleaved,
                                                     ULONG     nSlices,
                                                     ULONG     nLayers,
                                                     ULONG     nSceneCut,
                                                     ULONG     nAspectRatioX,
```

```
ULONG       nAspectRatioY,
HWND        hAttachedWindow = NULL,
BOOL        bThumbDraw = FALSE,
BOOL        bMaintainAspectRatio = FLASE,
DWORD       dwFlags = QCAP_BROADCAST_FLAG_FULL )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
nEncoderType	**IN**, Specify encoder type. Available value are QCAP_ENCODER_TYPE_SOFTWARE = 0 QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590 QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK= 2 QCAP_ENCODER_TYPE_NVIDIA_CUDA= 4
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_H264 = 0, QCAP_ENCODER_FORMAT_MPEG2 = 1, QCAP_ENCODER_FORMAT_H264_3D = 2, QCAP_ENCODER_FORMAT_H264_VC = 3, QCAP_ENCODER_FORMAT_RAW = 4,
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32

	QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_I420
nWidth	**IN**, Specify the horizontal width for the i-th video streamsession
nHeight	**IN**, Specify the vertical height for the i-th video streamsession
dFrameRate	**IN**, Specify the frame rate for the i-th video streamsession

nRecordProfile	**IN**, Specify recording profile
	QCAP_RECORD_PROFILE_BASELINE = 0, // DEFAULT
	QCAP_RECORD_PROFILE_MAIN = 1,
	QCAP_RECORD_PROFILE_HIGH = 2,

nRecordLevel	**IN**,Specify recording level QCAP_RECORD_LEVEL_1 = 10, QCAP_RECORD_LEVEL_1B = 9, QCAP_RECORD_LEVEL_11 = 11, QCAP_RECORD_LEVEL_12 = 12, QCAP_RECORD_LEVEL_13 = 13, QCAP_RECORD_LEVEL_2 = 20, QCAP_RECORD_LEVEL_21 = 21, QCAP_RECORD_LEVEL_22 = 22, QCAP_RECORD_LEVEL_3 = 30, QCAP_RECORD_LEVEL_31 = 31, QCAP_RECORD_LEVEL_32 = 32, QCAP_RECORD_LEVEL_4 = 40,

	QCAP_RECORD_LEVEL_41 = 41, // DEFAULT
	QCAP_RECORD_LEVEL_42 = 42,
	QCAP_RECORD_LEVEL_50 = 50,
	QCAP_RECORD_LEVEL_51 = 51,
	QCAP_RECORD_LEVEL_52 = 52,

nRecordEntropy	**IN**, Specify recording entropy
	QCAP_RECORD_ENTROPY_CAVLC = 0, // DEFAULT
	QCAP_RECORD_ENTROPY_CABAC = 1,

nRecordComplexity	**IN**, Specify recording complexity
	QCAP_RECORD_COMPLEXITY_0 = 0, // BEST SPEED, DEFAULT
	QCAP_RECORD_COMPLEXITY_1 = 1,
	QCAP_RECORD_COMPLEXITY_2 = 2,
	QCAP_RECORD_COMPLEXITY_3 = 3,
	QCAP_RECORD_COMPLEXITY_4 = 4,
	QCAP_RECORD_COMPLEXITY_5 = 5,
	QCAP_RECORD_COMPLEXITY_6 = 6, // BEST QUALITY

nRecordMode	**IN**, Specify recording mode. Available values are QCAP_RECORD_MODE_VBR = 0 QCAP_RECORD_MODE_CBR = 1 QCAP_RECORD_MODE_ABR = 2
nQuality	**IN**, Specify video quality. It is used by VBR and ABR.
nBitRate	**IN**, Specify video bit rate. It is used by CBR and ABR. (Ex. 6Mbps is 6 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0 ~ 255
nBFrames	**IN**, Specify recordingBFrames

bIsInterleaved	**IN**, Specify broadcast server Interleaved
nSlices	Specify broadcast server slices
nLayers	Specify broadcast server layers
nSceneCut	**IN**, Specify recording Scene Cut, recommedned value is 40， 0 is function turned off
nAspectRatioX	**IN**, Specify the aspect ratio X axle, 0 is turned off
nAspectRatioY	**IN**, Specify the aspect ratio Y axle, 0 is turned off
hAttachedWindow	**IN**, Specify the handle of the window to show the i-th video stream session
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer
bMaintainAspectRatio	**IN**, Enable/Disable the maintain aspect ratio
nFlags	**IN**, Specify broadcasting mode. Available values are QCAP_BROADCAST_FLAG_FULL = 0x00000007 QCAP_BROADCAST_FLAG_NETWORK = 0x00000001 QCAP_BROADCAST_FLAG_ENCODE = 0x00000002 QCAP_BROADCAST_FLAG_DISPLAY = 0x00000004 QCAP_BROADCAST_FLAG_DECODE = 0x00000008 QCAP_BROADCAST_FLAG_VIDEO_ONLY = 0x00000010 QCAP_BROADCAST_FLAG_AUDIO_ONLY = 0x00000020

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set additional parameters of the i-th video stream session in broadcast server.

Example

```
QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY_EX(   pServer, 0, QCAP_ENCODER_TYPE_SOFTWARE,
                                               QCAP_ENCODER_FORMAT_H264, QCAP_COLORSPACE_TYEP_YUY2, 320,
                                               240, 60, QCAP_RECORD_MODE_CBR, 8000, 12 * 1024 * 1024,
                                               nBFrames, FALUSE, 0, 0, 40, 4, 3, hAttachedWindow, FALSE,
                                               QCAP_BROADCAST_FLAG_FULL );
```

11.16 QCAP_GET_AUDIO_BROADCAST_SERVER_PROPERTY

```
QRESULT QCAP_GET_AUDIO_BROADCAST_SERVER_PROPERTY( PVOID     pServer,
                                                  UINT      iSessionNum,
                                                  ULONG  *  pEncoderType,
                                                  ULONG  *  pEncoderFormat,
                                                  ULONG  *  pChannels,
                                                  ULONG  *  pBitsPerSample,
                                                  ULONG  *  pSampleFrequency,
                                                  ULONG  *  pVolume )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pEncoderType	**OUT**, Pointer to the storage of the encoder type for the i-th audio stream session
pEncoderFormat	**OUT**, Pointer to the storage of the encoder format for the i-th video streamsession
pChannels	**OUT**, Pointer to the storage of encoder channels for the i-th video streamsession
pBitsPerSample	**OUT**, Pointer to the storage of encoder bits per sample for the i-th video stream session
pSampleFrequency	**OUT**, Pointer to the storage of encoder sample frequency for the i-th video stream session
pVolume	**OUT**, Pointer to the storage of audio renderer's volume for the i-th video stream session

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get parameters of the i-th audio stream session in broadcast server.

Example

```
QCAP_GET_AUDIO_BROADCAST_SERVER_PROPERTY( pServer, 0, &nEncoderType, &nEncoderFormat,
                                          &nChannels, &nBitsPerSample, &nSampleFrequency, &nVolume );
```

11.17 QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY

```
QRESULT QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY( PVOID     pServer,
                                                  UINT      iSessionNum,
                                                  ULONG     nEncoderType,
                                                  ULONG     nEncoderFormat,
                                                  ULONG     nChannels,
                                                  ULONG     nBitsPerSample,
                                                  ULONG     nSampleFrequency,
                                                  ULONG     nVolume )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0

nEncoderType	**IN**, Specify encoder type. Available value are
	`QCAP_ENCODER_TYPE_SOFTWARE = 0`
	`QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590`
	`QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2`
	`QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4`

nEncoderFormat	**IN**, Specify encoder format. Available value are
	`QCAP_ENCODER_FORMAT_PCM = 0`
	`QCAP_ENCODER_FORMAT_AAC = 1`
	`QCAP_ENCODER_FORMAT_AAC_ADTS = 2`

nChannels	**IN**, Specify encoder channels for the i-th audio stream session
nBitsPerSample	**IN**, Specify encoder bits per sample for the i-th audio stream session
nSampleFrequency	**IN**, Specify encoder sample frequency for the i-th audio stream session
nVolume	**IN**, Specify the volume of audio renderer for the i-th audio stream session, from 0 ~ 100

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set parameters of the i-th audio stream session in broadcast server.

Example

```
QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY(  pServer, 0, QCAP_ENCODER_TYPE_SOFTWARE,

                                           QCAP_ENCODER_FORMAT_PCM, 2, 16, 48000, 50 );
```

11.18 QCAP_GET_AUDIO_BROADCAST_SERVER_PROPERTY_EX

```
QRESULT QCAP_GET_AUDIO_BROADCAST_SERVER_PROPERTY_EX( PVOID      pServer,
                                                     UINT       iSessionNum,
                                                     ULONG  *   pEncoderType,
                                                     ULONG  *   pEncoderFormat,
                                                     ULONG  *   pChannels,
                                                     ULONG  *   pBitsPerSample,
                                                     ULONG  *   pSampleFrequency,
                                                     ULONG  *   pBitRate,
                                                     ULONG  *   pVolume )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pEncoderType	**OUT**, Pointer to the storage of the encoder type for the i-th audio stream session
pEncoderFormat	**OUT**, Pointer to the storage of the encoder format for the i-th video streamsession
pChannels	**OUT**, Pointer to the storage of encoder channels for the i-th video streamsession
pBitsPerSample	**OUT**, Pointer to the storage of encoder bits per sample for the i-th video stream session
pSampleFrequency	**OUT**, Pointer to the storage of encoder sample frequency for the i-th video stream session
pBitRate	**OUT**, Pointer to the storage of audio renderer's bitrate for the i-th video stream session
pVolume	**OUT**, Pointer to the storage of audio renderer's volume for the i-th video stream session

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get extra parameters of the i-th audio stream session in broadcast server.

Example

```
QCAP_GET_AUDIO_BROADCAST_SERVER_PROPERTY_EX( pServer, 0, &nEncoderType, &nEncoderFormat,
                                             &nChannels, &nBitsPerSample, &nSampleFrequency,
                                             &nBitRate, &nVolume );
```

11.19 QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY_EX

```
QRESULT QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY_EX( PVOID     pServer,
                                                     UINT      iSessionNum,
                                                     ULONG     nEncoderType,
                                                     ULONG     nEncoderFormat,
                                                     ULONG     nChannels,
                                                     ULONG     nBitsPerSample,
                                                     ULONG     nSampleFrequency,
                                                     ULONG     nBitRate,
                                                     ULONG     nVolume )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0

nEncoderType	**IN**, Specify encoder type. Available value are
	`QCAP_ENCODER_TYPE_SOFTWARE = 0`
	`QCAP_ENCODER_TYPE_HARDWARE = 1 for SC580 & SC590`
	`QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK = 2`
	`QCAP_ENCODER_TYPE_NVIDIA_CUDA = 4`

nEncoderFormat	**IN**, Specify encoder format. Available value are
	`QCAP_ENCODER_FORMAT_PCM = 0`
	`QCAP_ENCODER_FORMAT_AAC = 1`
	`QCAP_ENCODER_FORMAT_AAC_ADTS = 2`

nChannels	**IN**, Specify encoder channels for the i-th audio stream session
nBitsPerSample	**IN**, Specify encoder bits per sample for the i-th audio stream session
nSampleFrequency	**IN**, Specify encoder sample frequency for the i-th audio stream session
nBitRate	**IN**, Specify audio bit rate. Default is 128k and max is 496k
nVolume	**IN**, Specify the volume of audio renderer for the i-th audio stream session, from 0 ~ 100

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set extra parameters of the i-th audio stream session in broadcast server.

Example

```
QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY_EX(   pServer, 0, QCAP_ENCODER_TYPE_SOFTWARE,

                                               QCAP_ENCODER_FORMAT_PCM, 2, 16, 48000, 128*512, 50 );
```

11.20 QCAP_GET_VIDEO_BROADCAST_SERVER_SOFTWARE_ENCODER_PROPERTY

```
QRESULT QCAP_GET_VIDEO_BROADCAST_SERVER_SOFTWARE_ENCODER_PROPERTY(    PVOID     pServer,
                                                                      UINT      iSessionNum,
                                                                      ULONG  *  pRecordMode,
                                                                      ULONG  *  pQuality,
                                                                      ULONG  *  pBitRate,
                                                                      ULONG  *  pGOP )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pRecordMode	**OUT**, Pointer to the storage of the record mode for the i-th video streamsession
pQuality	**OUT**, Pointer to the storage of the quality for the i-th video streamsession
pBitRate	**OUT**, Pointer to the storage of the bit rate for the i-th video streamsession
pGOP	**OUT**, Pointer to the storage of the GOP size for the i-th video streamsession

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get additional software encoder parameters of the i-th video stream session in broadcast server.

User can use this function to get dynamic bitrate & gop reconfiguration for all software encoders.

Example

```
QCAP_GET_VIDEO_BROADCAST_SERVER_SOFTWARE_ENCODER_PROPERTY(  pServer, 0,

                                                            &nRecordMode, &nQuality, &nBitRate, &nGOP );
```

11.21 QCAP_SET_VIDEO_BROADCAST_SERVER_SOFTWARE_ENCODER_PROPERTY

```
QRESULT QCAP_SET_VIDEO_BROADCAST_SERVER_SOFTWARE_ENCODER_PROPERTY(
                                                    PVOID     pServer,
                                                    UINT      iSessionNum,
                                                    ULONG     nRecordMode,
                                                    ULONG     nQuality,
                                                    ULONG     nBitRate,
                                                    ULONG     Ngop,
                                                    BOOL      bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
nRecordMode	**IN**, Specify recording mode. Available values are QCAP_RECORD_MODE_VBR = 0 QCAP_RECORD_MODE_CBR = 1 QCAP_RECORD_MODE_ABR = 2
nQuality	**IN**, Specify video quality. It is used by VBR and ABR.
nBitRate	**IN**, Specify video bit rate. It is used by CBR and ABR. (Ex. 6Mbps is 6 x 1024 x 1024 bps)
nGOP	**IN**, Specify recording GOP size, from 0 ~ 255
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set additional parameters of the i-th video stream session in broadcast server.

User can use this function to set dynamic bitrate & gop reconfiguration for all software encoders.

Example

```
QCAP_SET_VIDEO_BROADCAST_SERVER_SOFTWARE_ENCODER_PROPERTY(  pServer, 0, QCAP_RECORD_MODE_CBR,

                                                            8000, 12 * 1024 * 1024, 30, FALSE );
```

11.22 QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(   PVOID     pServer,
                                                                UINT      iSessionNum,
                                                                ULONG     nColorSpaceType,
                                                                ULONG     nWidth,
                                                                ULONG     nHeight,
                                                                BYTE   *  pFrameBuffer,
                                                                ULONG     nFrameBufferLen,
                                                                double    dSampleTime = 0.0,
                                                                BOOL      bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
nColorSpaceType	**IN**, Specify the input source buffer's color space type, Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify the input source buffer's width
nHeight	**IN**, Specify the input source buffer's height
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the time stamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

* The performance of RGB type is slower than BGR in our color converter algorithm.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression video buffer into broadcasting recorder.

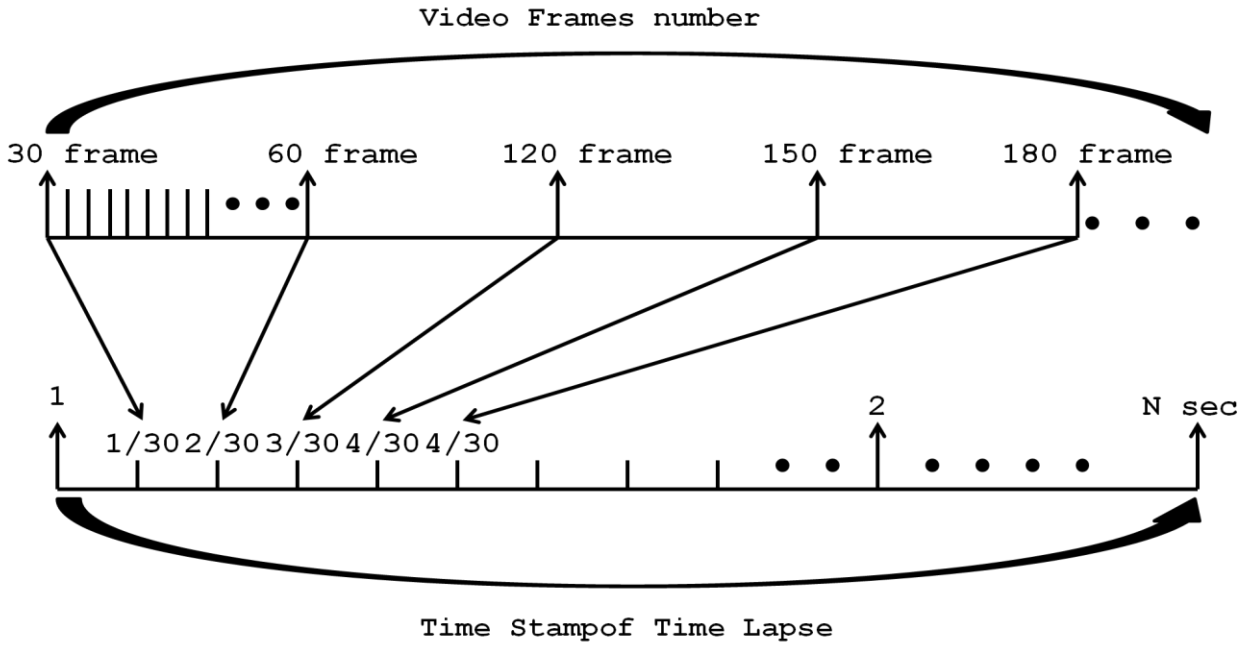
Server will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be send out by internet.

If the width or the height of source frame are different from the target frame size.

QCAP will enable one auto video scaler to resize it before encoding.

If users want to get Time Lapse record result, users can useSampleTime to controltimestamp of video. For instance the way as show in below.



Example

```
QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 0, QCAP_COLORSPACE_TYEP_YUY2,

                                                      1280, 720, pFrameBuffer, 1280 * 720 * 2, 0, FALSE );
```

11.23 QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER_EX(   PVOID      pServer,
                                                                   UINT       iSessionNum,
                                                                   ULONG      nColorSpaceType,
                                                                   ULONG      nWidth,
                                                                   ULONG      nHeight,
                                                                   BYTE   *   pFrameBuffer,
                                                                   ULONG      nFrameBufferLen,
                                                                   ULONG      nCropX,
                                                                   ULONG      nCropY,
                                                                   ULONG      nCropW,
                                                                   ULONG      nCropH,
                                                                   BOOL       bForceKeyFrame = FALSE,
                                                                   double     dSampleTime = 0.0,
                                                                   BOOL       bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
nColorSpaceType	**IN**, Specify the input source buffer's color space type, Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify the input source buffer's width
nHeight	**IN**, Specify the input source buffer's height
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
nCropX	**IN**, Specify the input source buffer's width
nCropY	**IN**, Specify the input source buffer's height
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
bForceKeyFrame	**IN**, Enforce the input source's frame is key-frame, the default is

	FALSE
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

* The performance of RGB type is slower than BGR in our color converter algorithm.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression video buffer and set crop parameters into broadcasting recorder.

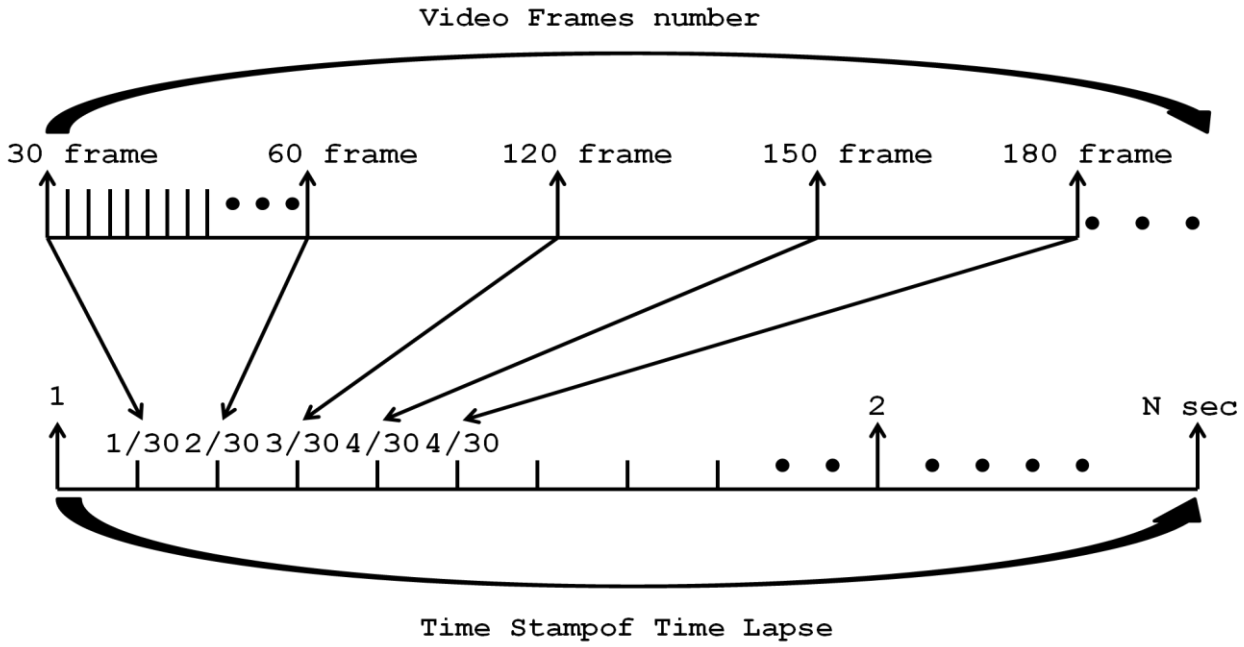
User can use bForceKeyFrame parameter to push key-frame immediately.

Server will send the buffer into the encoder and generate its bit-stream. Finally, the bit-stream can be send out by internet.

If the width or the height of source frame are different from the target frame size.

QCAP will enable one auto video scaler to resize it before encoding.

If users want to get Time Lapse record result, user can use SampleTime to control timestamp of video. For instance the way as show in below.



Example

```
QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER_EX( pServer, 0, QCAP_COLORSPACE_TYEP_YUY2,
                                                         1280, 720, pFrameBuffer, 1280 * 720 * 2, 0, 0, 720,
                                                         480, FALSE, 0, FALSE );
```

11.24 QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(   PVOID      pServer,
                                                                UINT       iSessionNum,
                                                                BYTE   *   pFrameBuffer,
                                                                ULONG      nFrameBufferLen,
                                                                double     dSampleTime = 0.0,
                                                                BOOL       bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression audio buffer into broadcasting recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be send out by internet.

Note

The SampleTime of audio must same with SampleTime of video.

Example

```
QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 0, pFrameBuffer, 4096, 0, FALSE );
```

11.25 QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER_EX(   PVOID      pServer,
                                                                   UINT       iSessionNum,
                                                                   ULONG      nChannels,
                                                                   ULONG      nBitsPerSample,
                                                                   ULONG      nSampleFrequency,
                                                                   BYTE   *   pFrameBuffer,
                                                                   ULONG      nFrameBufferLen,
                                                                   double     dSampleTime = 0.0,
                                                                   BOOL       bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
nChannels	**IN**, Specify encoder channels for the i-th audio stream session
nBitsPerSample	**IN**, Specify encoder bits per sample for the i-th audio stream session
nSampleFrequency	**IN**, Specify encoder sample frequency for the i-th audio stream session
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This is audio help function user can use this to resamle audio format.

For example, 2Ch 16Bit 48000HZ transform to 1Ch 8Bit 48000HZ.

This function is setting catcher input audio sample format, not resample audio format.

Example

```
QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER_EX( pServer, 0, 2, 16, 48000,
                                                         pFrameBuffer, nFrameBufferLen, FALSE );
```

11.26 QCAP_SET_VIDEO_BROADCAST_SERVER_COMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_BROADCAST_SERVER_COMPRESSION_BUFFER( PVOID     pServer,
                                                            UINT      iSessionNum,
                                                            BYTE   *  pStreamBuffer,
                                                            ULONG     nStreamBufferLen,
                                                            ULONG     bIsKeyFrame,
                                                            double    dSampleTime = 0.0,
                                                            BOOL      bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pStreamBuffer	**IN**, Specify the input source buffer
nStreamBufferLen	**IN**, Specify the input source buffer's size
bIsKeyFrame	**IN**, Specify the input source's frame whether it is key-frame or not
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

Returned Value

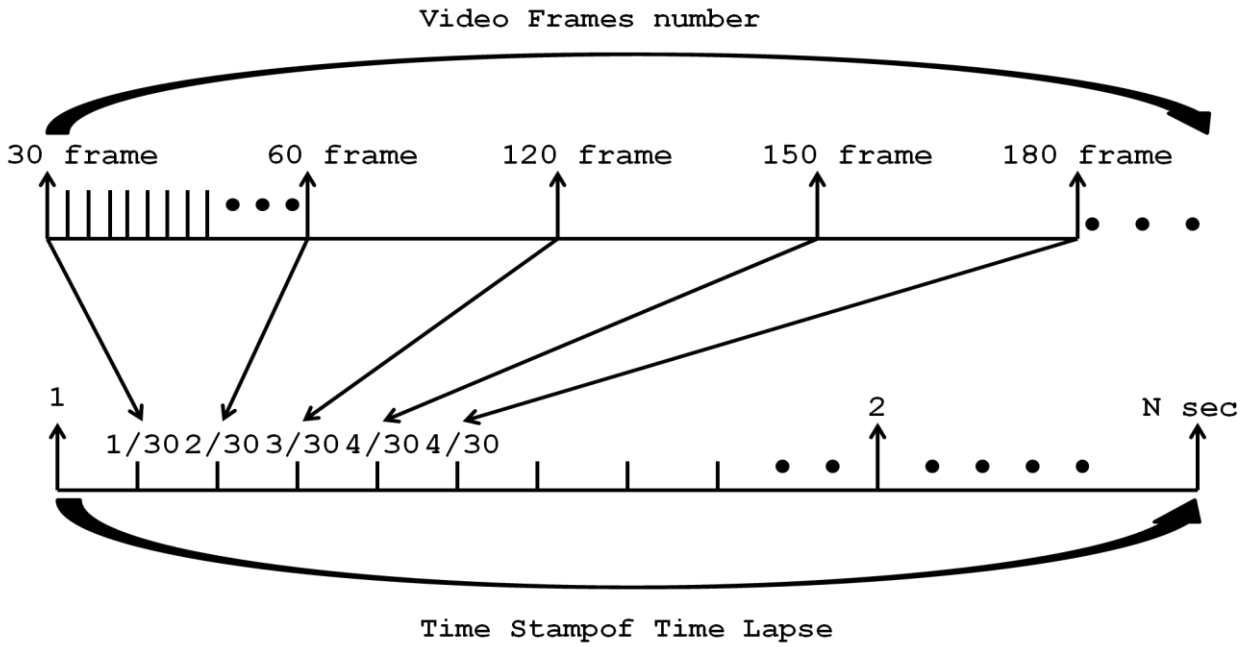
Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push compression stream buffer directly into broadcasting recorder.

The function will not enable encoder.

If users want to get Time Lapse result, user can use SampleTime to controltimestamp of video. For instance the way as shown in below.



Example

```
QCAP_SET_VIDEO_BROADCAST_SERVER_COMPRESSION_BUFFER( pServer, 0, pStreamBuffer, 3072, TRUE, 0, FALSE );
```

11.27 QCAP_SET_AUDIO_BROADCAST_SERVER_COMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_BROADCAST_SERVER_COMPRESSION_BUFFER( PVOID     pServer,
                                                            UINT      iSessionNum,
                                                            BYTE   *  pStreamBuffer,
                                                            ULONG     nStreamBufferLen,
                                                            double    dSampleTime = 0.0,
                                                            BOOL      bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pStreamBuffer	**IN**, Specify the input source buffer
nStreamBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push compression stream buffer directly into broadcasting recorder.

The function will not enable encoder.

Note

The SampleTime of audio must same with SampleTime of video.

Example

```
QCAP_SET_AUDIO_BROADCAST_SERVER_COMPRESSION_BUFFER( pServer, 0, pStreamBuffer, 4096, 0 );
```

11.28 QCAP_GET_SESSION_BROADCAST_SERVER_PROPERTY

```
QRESULT QCAP_GET_SESSION_BROADCAST_SERVER_PROPERTY(  PVOID      pServer,
                                                     UINT       iSessionNum,
                                                     CHAR  * *  ppSessionName,
                                                     ULONG *    pVideoDelayLiveDuration,
                                                     ULONG *    pAudioDelayLiveDuration )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
ppSessionName	**OUT**, Pointer to the address of session name
pVideoDelayLiveDuration	**OUT**, Pointer to the storage of current video delay live duration time
pAudioDelayLiveDuration	**OUT**, Pointer to the storage of current audio delay live duration time

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current session broadcast property.

Example

```
QCAP_GET_SESSION_BROADCAST_SERVER_PROPERTY( pServer, 0, &nSessionName, &nVideoDelayLiveDuration,
                                            &nAudioDelayLiveDuration );
```

11.29 QCAP_SET_SESSION_BROADCAST_SERVER_PROPERTY

```
QRESULT QCAP_SET_SESSION_BROADCAST_SERVER_PROPERTY(   PVOID     pServer,
                                                      UINT      iSessionNum,
                                                      CHAR  *   ppSessionName,
                                                      ULONG     nVideoDelayLiveDuration,
                                                      ULONG     nAudioDelayLiveDuration  )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pSessionName	**IN**, Specify the server name, the default name is "session%d.mpg"
nVideoDelayLiveDuration	**IN**, Specify video delay live dutation time, uint is millisecond
nAudioDelayLiveDuration	**IN**, Specify audio delay live dutation time, uint is millisecond

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set current session broadcast property.

Note

User can use ppSessionName to set session name. And if session name is “LiveRTMPSession” and the URL of RTMP is set as below”.

rtmp://127.0.0.1/live/LiveRTMPSession

If user use this function at HLS. And session name still need add .m3u8 as show in below.

http://hls/LiveHLSSession.m3u8

Example

```
QCAP_SET_SESSION_BROADCAST_SERVER_PROPERTY( pServer, 0, nSessionName, 0, 0 );
```

11.30 QCAP_SNAPSHOT_BROADCAST_SERVER_BMP

```
QRESULT QCAP_SNAPSHOT_BROADCAST_SERVER_BMP( PVOID     pServer,
                                            UINT      iSessionNum,
                                            CHAR *    pszFilePathName,
                                            BOOL      bIsAsync = TRUE,
                                            ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in millisecond User can use this function to snapshot a picture with BMP format on share records. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with BMP format, when broadcast server processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BROADCAST_SERVER_BMP( pServer, 0, “C:\\PICTURE1.BMP24” ); // PICTURE1.BMP24 ->PICTURE1.BMP
```

11.31 QCAP_SNAPSHOT_BROADCAST_SERVER_JPG

```
QRESULT QCAP_SNAPSHOT_BROADCAST_SERVER_JPG( PVOID     pServer,
                                            UINT      iSessionNum,
                                            CHAR *    pszFilePathName,
                                            ULONG     nQuality,
                                            BOOL      bIsAsync = TRUE,
                                            ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pszFilePathName	**IN**, Specify the file name to store JPG data
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in millisecondUser can use this function to snapshot a picture with BMP format on share records. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with JPG format, when broadcast server processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BROADCAST_SERVER_JPG( pServer,0, "C:\\PICTURE1.JPG", 100 );
```

11.32 QCAP_SNAPSHOT_BROADCAST_SERVER_BMP_EX

```
QRESULT QCAP_SNAPSHOT_BROADCAST_SERVER_BMP_EX( PVOID     pServer,
                                               UINT      iSessionNum,
                                               CHAR *    pszFilePathName,
                                               ULONG     nCropX,
                                               ULONG     nCropY,
                                               ULONG     nCropW,
                                               ULONG     nCropH,
                                               ULONG     nDstW,
                                               ULONG     nDstH,
                                               BOOL      bIsAsync = TRUE,
                                               ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with BMP format, when broadcast server processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

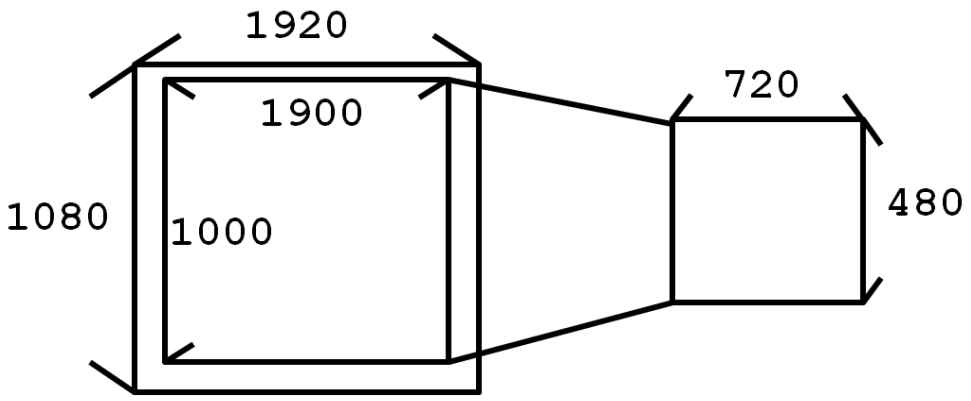
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BMP_EX( pServer, 0, "C:\\PICTURE1.BMP", 10, 40, 1900, 1000, 720, 480 );
```



11.33 QCAP_SNAPSHOT_BROADCAST_SERVER_JPG_EX

```
QRESULT QCAP_SNAPSHOT_BROADCAST_SERVER_JPG_EX( PVOID     pServer,
                                               UINT      iSessionNum,
                                               CHAR *    pszFilePathName,
                                               ULONG     nCropX,
                                               ULONG     nCropY,
                                               ULONG     nCropW,
                                               ULONG     nCropH,
                                               ULONG     nDstW,
                                               ULONG     nDstH,
                                               ULONG     nQuality,
                                               BOOL      bIsAsync = TRUE,
                                               ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pszFilePathName	**IN**, Specify the file name to store JPG data
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with JPG format, when broadcast server processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

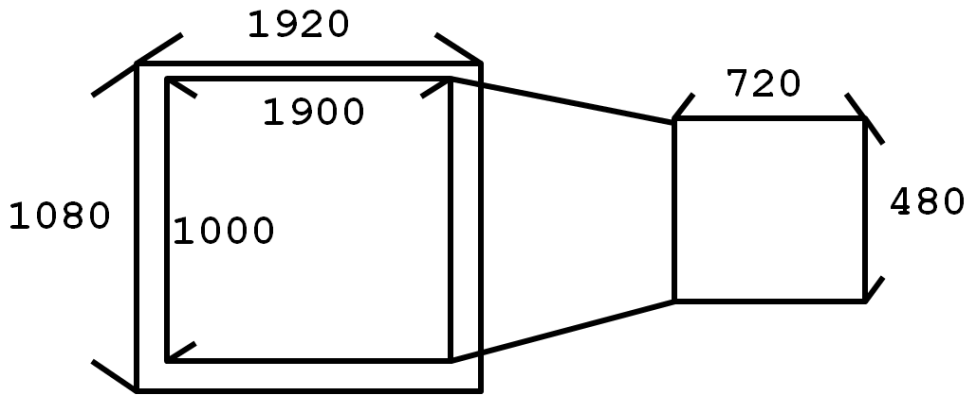
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BROADCAST_SERVER_JPG_EX( pServer, 0, "C:\\PICTURE1.JPG", 10, 40, 1900, 1000, 720, 480,100 );
```



11.34 QCAP_GET_OSD_BROADCAST_SERVER_TEXT_BOUNDARY

```
QRESULT QCAP_GET_OSD_BROADCAST_SERVER_TEXT_BOUNDARY(  PVOID       pServer
                                                      UINT        iSessionNum,
                                                      UINT        iOsdNum,
                                                      CHAR    *   pszString,
                                                      CHAR    *   pszFontFamilyName,
                                                      ULONG       nFontStyle,
                                                      ULONG       nFontSize,
                                                      ULONG   *   pBoundaryWidth,
                                                      ULONG   *   pBoundaryHeight )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output
nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are QCAP_FONT_STYLE_REGULAR = 0x00000000, QCAP_FONT_STYLE_BOLD = 0x00000001, QCAP_FONT_STYLE_ITALIC = 0x00000002, QCAP_FONT_STYLE_BOLDITALIC = 0x00000003, QCAP_FONT_STYLE_UNDERLINE = 0x00000004, QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,
nFontSize	**IN**, Specify the font size used to display the text of OSD output
pBoundaryWidth	**OUT**, Pointer to the storage of boundary width
pBoundaryHeight	**OUT**, Pointer to the storage of boundary height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get size of OSD string, when the users use the broadcast server function.

For example, user can use boundary information to set OSD string location on display window.

Example

```
QCAP_GET_OSD_BROADCAST_SERVER_TEXT_BOUNDARY( 0, 0, “CH01”, “Arial”,
                                QCAP_FONT_STYLE_BOLD, 12, pBoundaryWidth, pBoundaryHeight );
```

11.35 QCAP_SET_OSD_BROADCAST_SERVER_TEXT

```
QRESULT QCAP_SET_OSD_BROADCAST_SERVER_TEXT( PVOID      pServer,
                                            UINT       iSessionNum,
                                            UINT       iOsdNum,
                                            INT        x,
                                            INT        y,
                                            INT        width,
                                            INT        height,
                                            CHAR *     pszString,
                                            CHAR *     pszFontFamilyName,
                                            ULONG      nFontStyle,
                                            ULONG      nFontSize,
                                            DWORD      dwFontColor,
                                            DWORD      dwBackgroundColor,
                                            ULONG      nTransparent,
                                            INT        nTextStartPosX = 0,
                                            INT        nTextStartPosY = 0,
                                            ULONG      nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output, if width is -1, QCAP will auto calculate width.
height	**IN**, Specify the vertical height of OSD output, if height is -1, QCAP will auto calculate height.
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output
nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are QCAP_FONT_STYLE_REGULAR = 0x00000000, QCAP_FONT_STYLE_BOLD = 0x00000001, QCAP_FONT_STYLE_ITALIC = 0x00000002, QCAP_FONT_STYLE_BOLDITALIC = 0x00000003, QCAP_FONT_STYLE_UNDERLINE = 0x00000004, QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,
nFontSize	**IN**, Specify the font size used to display the text of OSD output
dwFontColor	**IN**, Specify the font color used to display the text of OSD output

dwBackgroundColor	**IN**, Specify the background color used to display the text of OSD output
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nTextStartPosX	**IN**, Specify the text scrolling start position X of the upper-left corner of OSD text
nTextStartPosY	**IN**, Specify the text scrolling start position Y of the upper-left corner of OSD text
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

When the users create a broadcast server object,

In the i-th video stream session we also support to show a text field objects used for onscreen display of information.

If you want to get a text field object auto calculated actual width and height value;

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_BROADCAST_SERVER_TEXT(  pServer, 0, 0, 0, 0, -1, -1, “CH01”, “Arial”,

                                     QCAP_FONT_STYLE_BOLD, 12, 0xFF000000,

                                     0xFFFFFFFF, 128, 0, 0, QCAP_SEQUENCE_STYLE_FOREMOST );
```

11.36 QCAP_SET_OSD_BROADCAST_SERVER_PICTURE

```
QRESULT QCAP_SET_OSD_BROADCAST_SERVER_PICTURE(    PVOID      pServer,
                                                  UINT       iSessionNum,
                                                  UINT       iOsdNum,
                                                  INT        x,
                                                  INT        y,
                                                  INT        width,
                                                  INT        height,
                                                  CHAR *     pszFilePathName,
                                                  ULONG      nTransparent,
                                                  ULONG      nSequenceStyle =
                                                             QCAP_SEQUENCE_STYLE_FOREMOST)
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output If width is -1, QCAP will use original picture width as default.
height	**IN**, Specify the vertical height of OSD output. If height is -1, QCAP will use original picture height as default.
pszFilePathName	**IN**, Specify the file name to display image OSD output, support a file extension of "BMP" as 24 or 32-bit,“JPG” and “PNG”
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

When the users create a broadcast server object,

In the i-th video stream session we also support to show a static image objects used for onscreen display of information.

If you want to get a text field object auto calculated actual width and height value;

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_BROADCAST_SERVER_PICTURE( pServer, 0, 0, 0, 0, -1, -1,
                                       “C:\\SAMPLE.PNG”,
                                       128,
                                       QCAP_SEQUENCE_STYLE_FOREMOST );
```

11.37 QCAP_SET_OSD_BROADCAST_SERVER_BUFFER

```
QRESULT QCAP_SET_OSD_BROADCAST_SERVER_BUFFER(  PVOID      pServer,
                                               UINT       iSessionNum,
                                               UINT       iOsdNum,
                                               INT        x,
                                               INT        y,
                                               INT        width,
                                               INT        height,
                                               ULONG      nColorSpaceType,
                                               BYTE *     pFrameBuffer,
                                               ULONG      nFrameWidth,
                                               ULONG      nFrameHeight,
                                               ULONG      nFramePitch,
                                               ULONG      nTransparent,
                                               DWORD      dwKeyColor = 0xFFFFFFFF,
                                               ULONG      nKeyColorThreshold = 25,
                                               ULONG      nSequenceStyle =
                                                          QCAP_SEQUENCE_STYLE_FOREMOST );
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output
height	**IN**, Specify the vertical height of OSD output
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameWidth	**IN**, Specify the horizontal width of frame contained in a buffer
nFrameHeight	**IN**, Specify the vertical height of frame contained in a buffer
nFramePitch	**IN**, Specify the distance in bytes between the start of one scan line to the next. If it is 0, we will auto calculate it by width and color space

	format.
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
dwKeyColor	**IN**, Specify the key color of broadcast server OSD, default is 0xFFFFFFFF
nKeyColorThreshold	**IN**, Specify the threshold of key color, the range of 0 to 128
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

When the users create a broadcast server object,

In the i-th video stream session we also support to show a frame buffer objects used for onscreen display of information.

The function’s purpose is used to point directly to a valid frame buffer whose entire pixel data,

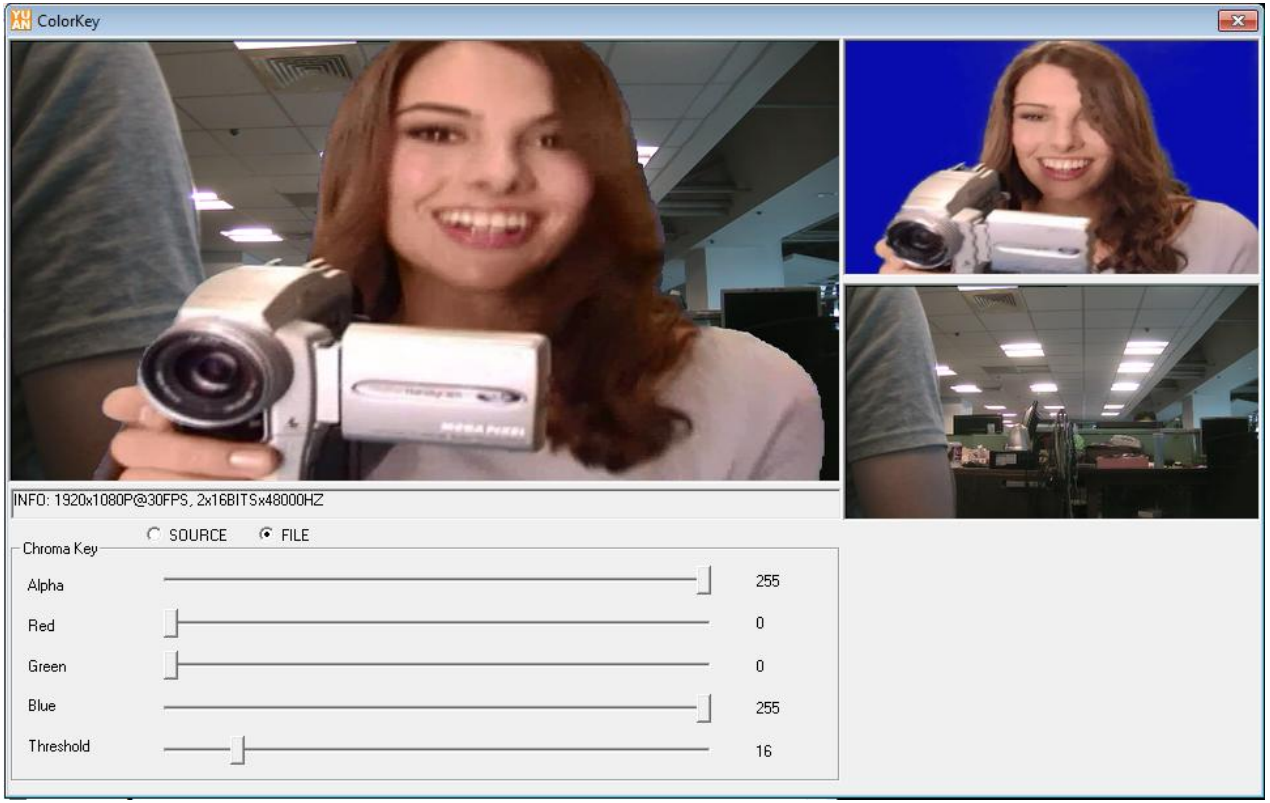
which is stored in available color space, and display it.

Beside, you also can adjust the width and height parameters to scale the OSD output.

Note

User can use this dwKeyColor parament to select colorkey mode Green&Blue mode and RGB mode.

For instance the Blue mode way as show in below



Example

```
QCAP_SET_OSD_BROADCAST_SERVER_BUFFER( pServer, 0, 0, 0, 0, 1920, 1080,
                                      QCAP_COLORSPACE_TYEP_YUY2, pFrameBuffer,
                                      800, 600, 0, 0xFFFFFFFF, 128, QCAP_SEQUENCE_STYLE_FOREMOST );
```

11.38 QCAP_SET_VIDEO_3D_BRODCAST_SERVER_STEREO_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_3D_BROADCAST_SERVER_STEREO_UNCOMPRESSION_BUFFER(   PVOID  pServer,
                                                                          UINT   iSessionNum,
                                                                          ULONG  nColorSpaceType,
                                                                          ULONG  nWidth,
                                                                          ULONG  nHeight,
                                                                          BYTE * pFrameBuffer,
                                                                          ULONG  nFrameBufferLen,
                                                                          ULONG  nStereoBufferType )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0, max is 8
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify the 3D input source buffer's width
nHeight	**IN**, Specify the 3D input source buffer's height
pFrameBuffer	**IN**, Specify the 3D input source buffer
nFrameBufferLen	**IN**, Specify the 3D input source buffer's size
nStereoBufferType	**IN**, Specify 3D stereo mode. Available values are QCAP_3D_STEREO_BUFFER_SIDE_BY_SIDE = 0 QCAP_3D_STEREO_BUFFER_TOP_BOTTOM = 0 QCAP_3D_STEREO_BUFFER_LINE_BY_LINE = 0

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use one camera to broadcast 3D video stream.

User can use this function to push 3D un-compression video buffer into broadcasting recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

If the width or the height of source frame are different from the target frame size.

QCAP will enable one auto video scaler to resize it before encoding.

Example

```
QCAP_SET_VIDEO_3D_SHARE_RECORD_STEREO_UNCOMPRESSION_BUFFER( pServer, 0, QCAP_COLORSPACE_TYEP_YUY2,
                                                            1280, 720, pFrameBuffer, nFrameBufferLen,
                                                            QCAP_3D_STEREO_BUFFER_SIDE_BY_SIDE )
```

11.39QCAP_SET_VIDEO_3D_BROADCAST_SERVER_STEREO_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_3D_BROADCAST_SERVER_STEREO_UNCOMPRESSION_BUFFER_EX(
                                                                    PVOID  pServer,
                                                                    UINT   iSessionNum,
                                                                    ULONG  nColorSpaceType,
                                                                    ULONG  nWidth,
                                                                    ULONG  nHeight,
                                                                    BYTE * pFrameBuffer,
                                                                    ULONG  nFrameBufferLen,
                                                                    ULONG  nCropX,
                                                                    ULONG  nCropY,
                                                                    ULONG  nCropW,
                                                                    ULONG  nCropH,
                                                                    ULONG  nStereoBufferType )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0, max is 0 ~ 8
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D input source buffer's width
height	**IN**, Specify the 3D input source buffer's height
pFrameBuffer	**IN**, Specify the 3D input source buffer
nFrameBufferLen	**IN**, Specify the 3D input source buffer's size
nCropX	**IN**, Specify the x-coordinate of the crop of 3D video
nCropY	**IN**, Specify the y-coordinate of the crop of 3D video
nCropW	**IN**, Specify the horizontal width of crop of 3D video
nCropH	**IN**, Specify the vertical height of crop of 3D video
nStereoBufferType	**IN**, Specify 3D stereo mode. Available values are QCAP_3D_STEREO_BUFFER_SIDE_BY_SIDE = 0

	QCAP_3D_STEREO_BUFFER_TOP_BOTTOM = 0 QCAP_3D_STEREO_BUFFER_LINE_BY_LINE = 0

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

If user is use one camera to recording 3D video stream.

User can use this function to push 3D un-compression video buffer into share and set crop parameters into broadcasting recorder.

And User can use crop parameters to set any part of a video screen at will.

Example

```
QCAP_SET_VIDEO_3D_BROADCAST_SERVER_STEREO_UNCOMPRESSION_BUFFER_EX(pServer, 0, QCAP_COLORSPACE_TYEP_YUY2,
                                                                  1280, 720, pFrameBuffer, nFrameBufferLen,
                                                                  100, 150, 720, 480
                                                                  QCAP_3D_STEREO_BUFFER_SIDE_BY_SIDE);
```

11.40 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_L_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_3D_BROADCAST_SERVER_L_UNCOMPRESSION_BUFFER(    PVOID     pServer,
                                                                      UINT      iSessionNum,
                                                                      ULONG     nColorSpaceType,
                                                                      ULONG     nWidth,
                                                                      ULONG     nHeight,
                                                                      BYTE *    pFrameBuffer,
                                                                      ULONG     nFrameBufferLen )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0, max is 8
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D of left input source buffer's width
height	**IN**, Specify the 3D of left input source buffer's height
pFrameBuffer	**IN**, Specify the 3D of left input source buffer
nFrameBufferLen	**IN**, Specify the 3D of left input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D of left un-compression video buffer into broadcasting recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

Example

```
QCAP_SET_VIDEO_3D_BROADCAST_SERVER_L_UNCOMPRESSION_BUFFER( pServer, 0, QCAP_COLORSPACE_TYEP_YUY2,
                                                           1280, 720, pFrameBuffer, nFrameBufferLen);
```

11.41 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_L_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_3D_BROADCAST_SERVER_L_UNCOMPRESSION_BUFFER_EX(  PVOID     pServer,
                                                                       UINT      iSessionNum,
                                                                       ULONG     nColorSpaceType,
                                                                       ULONG     nWidth,
                                                                       ULONG     nHeight,
                                                                       BYTE *    pFrameBuffer,
                                                                       ULONG     nFrameBufferLen,
                                                                       ULONG     nCropX,
                                                                       ULONG     nCropY,
                                                                       ULONG     nCropW,
                                                                       ULONG     nCropH )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0, max is 8
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D of left input source buffer's width
height	**IN**, Specify the 3D of left input source buffer's height
pFrameBuffer	**IN**, Specify the 3D of left input source buffer
nFrameBufferLen	**IN**, Specify the 3D of left input source buffer's size
nCropX	**IN**, Specify the x-coordinate of the crop of 3D video
nCropY	**IN**, Specify the y-coordinate of the crop of 3D video
nCropW	**IN**, Specify the horizontal width of crop of 3D video
nCropH	**IN**, Specify the vertical height of crop of 3D video

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D of left un-compression video buffer into share recorder and set crop parameters into broadcast recorder.

And User can use crop parameters to set any part of a video screen at will.

Example

```
QCAP_SET_VIDEO_3D_BROADCAST_SERVER_L_UNCOMPRESSION_BUFFER_EX(  pServer, 0, QCAP_COLORSPACE_TYEP_YUY2,
                                                               1280, 720, pFrameBuffer,
                                                               nFrameBufferLen, 100, 150, 720, 480 );
```

11.42 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_R_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_3D_BROADCAST_SERVER_R_UNCOMPRESSION_BUFFER ( PVOID    pServer,
                                                                    UINT     iSessionNum,
                                                                    ULONG    nColorSpaceType,
                                                                    ULONG    nWidth,
                                                                    ULONG    nHeight,
                                                                    BYTE *   pFrameBuffer,
                                                                    ULONG    nFrameBufferLen )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0, max is 8
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D of right input source buffer's width
height	**IN**, Specify the 3D of right input source buffer's height
pFrameBuffer	**IN**, Specify the 3D of right input source buffer
nFrameBufferLen	**IN**, Specify the 3D of right input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D of right un-compression video buffer into broadcast recorder.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

Example

```
QCAP_SET_VIDEO_3D_BROADCAST_SERVER_R_UNCOMPRESSION_BUFFER( pServer, 0, QCAP_COLORSPACE_TYEP_YUY2,
                                                           1280, 720, pFrameBuffer, nFrameBufferLen );
```

11.43 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_R_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_3D_BROADCAST_SERVER_R_UNCOMPRESSION_BUFFER_EX(    PVOID     pServer,
                                                                         UINT      iSessionNum,
                                                                         ULONG     nColorSpaceType,
                                                                         ULONG     nWidth,
                                                                         ULONG     nHeight,
                                                                         BYTE *    pFrameBuffer,
                                                                         ULONG     nFrameBufferLen,
                                                                         ULONG     nCropX,
                                                                         ULONG     nCropY,
                                                                         ULONG     nCropW,
                                                                         ULONG     nCropH )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0, max is 8
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
width	**IN**, Specify the 3D of right input source buffer's width
height	**IN**, Specify the 3D of right input source buffer's height
pFrameBuffer	**IN**, Specify the 3D of right input source buffer
nFrameBufferLen	**IN**, Specify the 3D of right input source buffer's size
nCropX	**IN**, Specify the x-coordinate of the crop of 3D video
nCropY	**IN**, Specify the y-coordinate of the crop of 3D video
nCropW	**IN**, Specify the horizontal width of crop of 3D video
nCropH	**IN**, Specify the vertical height of crop of 3D video

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D of right un-compression video buffer into share recorder and set crop parameters into broadcast recorder.

And User can use crop parameters to set any part of a video screen at will.

Example

```
QCAP_SET_VIDEO_3D_BROADCAST_SERVER_R_UNCOMPRESSION_BUFFER_EX( pServer, 0, QCAP_COLORSPACE_TYEP_YUY2,
                                                              1280, 720, pFrameBuffer, nFrameBufferLen,
                                                              100, 150, 720, 480 );
```

11.44 QCAP_SET_VIDEO_3D_BROADCAST_SERVER_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_3D_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(   PVOID     pServer,
                                                                   UINT      iSessionNum,
                                                                   ULONG     nStereoDisplayMode,
                                                                   BOOL      bLeftRightSwap = FALSE,
                                                                   BOOL      bForceKeyFrame = FALSE,
                                                                   double    dSampleTime = 0.0,
                                                                   BOOL      bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specifythe number of channel session connections, start from 0, max is 8
nStereoDisplayMode	**IN**, Specify 3D display mode. Available values are QCAP_3D_STEREO_DISPLAY_MODE_SIDE_BY_SIDE = 0 QCAP_3D_STEREO_DISPLAY_MODE_TOP_BOTTOM = 1 QCAP_3D_STEREO_DISPLAY_MODE_LINE_BY_LINE = 2 QCAP_3D_STEREO_DISPLAY_MODE_LEFT_ONLY = 3 QCAP_3D_STEREO_DISPLAY_MODE_RIGHT_ONLY = 4
bLeftRightSwap	**IN**, Change the input camera source's (right to left, left to right), the default is FALSE
bForceKeyFrame	**IN**, Enforce the input source's frame is key-frame, the default is FALSE

dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push 3D un-compression video buffer into broadcast recorder.

User can use bForceKeyFrame parameter to push key-frame immediately.

Recorder will send the buffer into the encoder and generate its bit-stream.

Finally, the bit-stream can be saved to disk by the file writer.

If the width or the height of source frame are different from the target frame size.

QCAP will enable one auto video scaler to resize it before encoding.

Example

```
QCAP_SET_VIDEO_3D_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 0, 0, FALSE, FALSE, 0, FALSE );
```

11.45 QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER(   PVOID     pServer,
                                                                          UINT      iSessionNum,
                                                                          BYTE *    pFrameBuffer,
                                                                          ULONG     nFrameBufferLen )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0, max is 8
pFrameBuffer	**IN**, Specify the audio input source buffer
nFrameBufferLen	**IN**, Specify the audio input source buffer's size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to mixing audio share record uncompression buffer on audio broadcast server callback.

User must set `QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER` to mixer audio record first then call `QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER`.

Example

```
if( nChannelNum == 0 ){
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pServer, 0,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pServer, 1,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );

    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 0 );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 1 );
}
if( nChannelNum == 1 ){
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pServer, 0,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pServer, 2,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );

    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 2 );
}
```

11.46 QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(   PVOID     pServer,
                                                                   UINT      iSessionNum,
                                                                   double    dSampleTime = 0.0,
                                                                   BOOL      bClearDelayLive = FALSE )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0, max is 8
dSampleTime	**IN**, Specify the timestamp of this frame.
bClearDelayLive	**IN**, If true, qcap will clear current delay live buffer dtae

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio broadcast server uncompression buffer on audio broadcast server callback.

Example

```
if( nChannelNum == 0 ){
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pServer, 0,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pServer, 1,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );

    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 0 );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 1 );
}
if( nChannelNum == 1 ){
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pServer, 0,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pServer, 2,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );

    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 2 );
}
```

11.47 QCAP_CREATE_BROADCAST_CLIENT

```
QRESULT QCAP_CREATE_BROADCAST_CLIENT(   UINT       iCliNum,
                                        CHAR   *   pszURL,
                                        PVOID  *   ppClient,
                                        ULONG      nDecoderType,
                                        HWND       hAttacheWindow = NULL,
                                        BOOL       bThumbDraw = FALSE,
                                        BOOL       bMaintainAspectRatio = FALSE )
```

Parameter

Name	Description
iCliNum	**IN**, Specify broadcast client number to get client parameters, start from 0 max is 0~63
pszURL	**IN**, In the client side, user can use the URL to access it. For example rtsp://account:password@ip:port/session#num.mpg
ppClient	**OUT**, Handle of the client object
nDecoderType	**IN**, Specify query decodertype. Available values are QCAP_DECODER_TYPE_SOFTWARE QCAP_DECODER_TYPE_INTEL_MEDIA_SDK
hAttacheWindow	**IN**, Handle of the window to show the preview video, the default string is NULL
bThumbDraw	**IN**, Enable/Disable the thumb draw renderer
bMaintainAspectRatio	**IN**, Enable/Disable the maintain aspect ratio

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a broadcast client object.

Example

```
QCAP_CREATE_BROADCAST_CLIENT( 0, "rtsp://root:root@127.0.0.1:554/session0.mpg", &pClient,

                              hWindow, TRUE, FALSE );
```

11.48 QCAP_DESTROY_BROADCAST_CLIENT

```
QRESULT QCAP_DESTROY_BROADCAST_CLIENT( PVOID pClient )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Destroy broadcast client object to release its resource.

Example

```
QCAP_DESTROY_BROADCAST_CLIENT( pClient );
```

11.49 QCAP_START_BROADCAST_CLIENT

```
QRESULT QCAP_START_BROADCAST_CLIENT( PVOID  pClient,
                                     ULONG  nProtocol = QCAP_BROADCAST_PROTOCOL_UDP,
                                     ULONG  nReconnectionTimeout = 3000,
                                     ULONG  nDelayQueueDuration = 0 )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
nProtocol	**IN,** BROADCAST PROTOCOL, QCAP_BROADCAST_PROTOCOL_UDP = 0x00000000, QCAP_BROADCAST_PROTOCOL_TCP = 0x00000001, QCAP_BROADCAST_PROTOCOL_HTTP = 0x00000002, The default protocol is QCAP_BROADCAST_PROTOCOL_UDP.
nReconnectionTimeout	**IN**, The parameter allows you to specify the timeout duration of the reconnection, this function is 3000. Uint is millisecond
nDelayQueueDuration	**IN**, Specify delay queue dutation time, default is 1000 millisecond

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start broadcast client.

Beside, user also can adjust the broadcast protocol to connect the video and audio stream.

If user set bitrate of video broadcast property is over 12*1024*1024. We recommend user use TCP protocol to streaming.

Note

If DelayQueueDuration is set to 0, we offer .TS file to One Vod for hls streaming.

Example

```
QCAP_START_BROADCAST_CLIENT( pClient, QCAP_BROADCAST_PROTOCOL_UDP );
```

11.50 QCAP_STOP_BROADCAST_CLIENT

```
QRESULT QCAP_STOP_BROADCAST_CLIENT( PVOID pClient )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop broadcast client.

Example

```
QCAP_STOP_BROADCAST_CLIENT( pClient );
```

11.51 QCAP_GET_VIDEO_BROADCAST_CLIENT_REGION_DISPLAY

```
QRESULT QCAP_GET_VIDEO_BROADCAST_CLIENT_REGION_DISPLAY( PVOID      pClient,
                                                        ULONG *    pCropX,
                                                        ULONG *    pCropY,
                                                        ULONG *    pCropW,
                                                        ULONG *    pCropH )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCropX	**OUT**, Pointer to the storage of x-coordinate of the crop region display
pCropY	**OUT**, Pointer to the storage of y-coordinate of the crop region display
pCropW	**OUT**, Pointer to the storage of horizontal width of crop region display
pCropH	**OUT**, Pointer to the storage of vertical height of crop region display

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get the current region display information of video broadcast client.

Example

```
QCAP_GET_VIDEO_BROADCAST_CLIENT_REGION_DISPLAY( pClient, &nCropX, &nCropY, &nCropW, &nCropW  );
```

11.52 QCAP_SET_VIDEO_BROADCAST_CLIENT_REGION_DISPLAY

```
QRESULT QCAP_SET_VIDEO_BROADCAST_CLIENT_REGION_DISPLAY( PVOID  pClient,
                                                        ULONG  nCropX,
                                                        ULONG  nCropY,
                                                        ULONG  nCropW,
                                                        ULONG  nCropH )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
nCropX	**IN**, Specify the x-coordinate of the crop region display
nCropY	**IN**, Specify the y-coordinate of the crop region display
nCropW	**IN**, Specify the horizontal width of crop region display
nCropH	**IN**, Specify the vertical height of crop region display

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set the current region display of video broadcast client.

Example

```
QCAP_SET_VIDEO_BROADCAST_CLIENT_MIRROR( pClient, 0, 0, 1280, 720 );
```

11.53 QCAP_GET_VIDEO_BROADCAST_CLIENT_MIRROR

```
QRESULT QCAP_GET_VIDEO_BROADCAST_CLIENT_MIRROR( PVOID    pClient,
                                                BOOL *   pHorizontalMirror,
                                                BOOL *   pVerticalMirror )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pHorizontalMirror	**OUT**, Returned the current horizontal of mirror
pVerticalMirror	**OUT**, Returned the current vertical of mirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get the current status of mirror of video broadcast client.

Example

```
QCAP_GET_VIDEO_BROADCAST_CLIENT_MIRROR( pClient, &bHorizontalMirror, &bVerticalMirror );
```

11.54 QCAP_SET_VIDEO_BROADCAST_CLIENT_MIRROR

```
QRESULT QCAP_SET_VIDEO_BROADCAST_CLIENT_MIRROR( PVOID pClient,
                                                BOOL  bHorizontalMirror,
                                                BOOL  bVerticalMirror )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
bHorizontalMirror	**IN**, Enable/Disable horizontal of mirror
bVerticalMirror	**IN**, Enable/Disable vertical of mirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set the status of mirror of video broadcast client.

Example

```
QCAP_SET_VIDEO_BROADCAST_CLIENT_MIRROR( pClient, TRUE, TRUE );
```

11.55 QCAP_GET_ AUDIO_BROADCAST_CLIENT_ VOLUME

```
QRESULT QCAP_GET_AUDIO_BROADCAST_CLIENT_VOLUME( PVOID pClient, ULONG * pVolume )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pVolume	**OUT**, Pointer to the storage of audio volume

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get audio volume of broadcast client.

Example

```
QCAP_GET_AUDIO_BROADCAST_CLIENT_VOLUME( pClient, &nVolume );
```

11.56 QCAP_SET_ AUDIO_BROADCAST_CLIENT_ VOLUME

```
QRESULT QCAP_SET_AUDIO_BROADCAST_CLIENT_VOLUME( PVOID pClient, ULONG  nVolume )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
nVolume	**IN**, Audio volume, from 0~100, 0 is mute

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio volume of broadcast client.

Example

```
QCAP_SET_AUDIO_BROADCAST_CLIENT_VOLUME( pClient, 50 );
```

11.57 QCAP_START_BROADCAST_CLIENT_RECORD

```
QRESULT QCAP_START_BROADCAST_CLIENT_RECORD( PVOID     pClient,
                                            CHAR *    pszFilePathName,
                                            double    dVideoDelayTime = 0.0,
                                            double    dAudioDelayTime = 0.0,
                                            double    dSegmentDurationTime = 0.0,
                                            ULONG     nSegmentDurationSizeKB = 0 )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pszFilePathName	**IN**, Specify the file name of storage for recording. Recording file format is AVI, if extension name is AVI. Recording file format is MP4, if extension name is MP4. Recording file format is ASF, if extension name is ASF. Recording file format is FLV, if extension name is FLV. Recording file format is TS, if extension name is TS. Recording file format is SCF, if extension name is SCF. Recording file format is WMV, if extension name is WMV. Recording file format is WAV, if extension name is WAV.
dVideoDelayTime	**IN**, Specify video delay time, default is 0.0
dAudioDelayTime	**IN**, Specify audio delay time, default is 0.0
dSegmentDurationTime	**IN**, Specify segment duration time
nSegmentDurationSizeKB	**IN**, Specify segment duration size, the uint is K Byte

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start recording of broadcast client.

Users can use dVideoDelayTime& dAudioDelayTime parameters to set video or audio delay time for control play speed.

User can use dSegmentDurationTime parameter to set segment duration time of record file.

The dSegmentDurationTime parameter offer more variables for recording file path name as show in below:

$Y, $M, $D, $h, $m, $h, $i for start system time

%Y, %M, %D, %h, %m, %h, %i for stop system time

The dSegmentDurationTime and nSegmentDurationSizeKB can't be set at same time.

Example

```
QCAP_START_BROADCAST_CLIENT_RECORD( pClient, "D:\\XXX.AVI" );

str1.Format( "%sMAIN_CH%02d.$Y.$M.$D.$h.$m.$ss.$i.mp4", g_scmProtypeEx.pszRecordFilePathName, i );

   QCAP_START_BROADCAST_CLIENT_RECORD ( pClient,
                                        (CHAR*)(LPCSTR)str1 ,
                                        QCAP_RECORD_FLAG_FULL,
                                        0.0 /*dVideoDelayTime = 0.0*/,
                                        0.0/*dAudioDelayTime = 0.0*/,
                                        10 );
```

11.58 QCAP_STOP_BROADCAST_CLIENT_RECORD

```
QRESULT QCAP_STOP_BROADCAST_CLIENT_RECORD( PVOID pClient, BOOLbIsAsync = TRUE, ULONG nMilliseconds = 0 )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
bIsAsync	**IN**, Synchronized stop recording flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop recording of broadcast client.

Example

```
QCAP_STOP_BROADCAST_CLIENT_RECORD( pClient, 0 );
```

11.59 QCAP_RESUME_BROADCAST_CLIENT_RECORD

QRESULT QCAP_RESUME_BROADCAST_CLIENT_RECORD(PVOID pClient)

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

User can use this function to resume recording of broadcast client.

Example

```
QCAP_RESUME_BROADCAST_CLIENT_RECORD( pClient );
```

11.60 QCAP_PAUSE_BROADCAST_CLIENT_RECORD

```
QRESULT QCAP_PAUSE_BROADCAST_CLIENT_RECORD( PVOID pClient )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to pause recording of broadcast client.

Example

```
QCAP_PAUSE_BROADCAST_CLIENT_RECORD( pClient );
```

11.61 QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP

```
QRESULT QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP(PVOID     pClient,
                                           CHAR *    pszFilePathName,
                                           BOOL      bIsAsync = TRUE,
                                           ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in millisecondUser can use this function to snapshot a picture with BMP format on share records. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snap shot a picture with BMP format, when broadcast client processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP( pClient, "C:\\PICTURE1.BMP24" ); // PICTURE1.BMP24 ->PICTURE1.BMP
```

11.62 QCAP_SNAPSHOT_BROADCAST_CLIENT_JPG

```
QRESULT QCAP_SNAPSHOT_BROADCAST_CLIENT_JPG( PVOID     pClient,
                                            CHAR *    pszFilePathName,
                                            ULONG     nQuality,
                                            BOOL      bIsAsync = TRUE,
                                            ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pszFilePathName	**IN**, Specify the file name to store JPG data
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snap shot a picture with JPG format, when broadcast client processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BROADCAST_CLIENT_JPG ( pClient, "C:\\PICTURE1.JPG", 100 );
```

11.63 QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP_EX

```
QRESULT QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP_EX(  PVOID    pClient,
                                                CHAR *   pszFilePathName,
                                                ULONG    nCropX,
                                                ULONG    nCropY,
                                                ULONG    nCropW,
                                                ULONG    nCropH,
                                                ULONG    nDstW,
                                                ULONG    nDstH,
                                                BOOL     bIsAsync = TRUE,
                                                ULONG    nMilliseconds = 0 )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with BMP format, when broadcast client processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

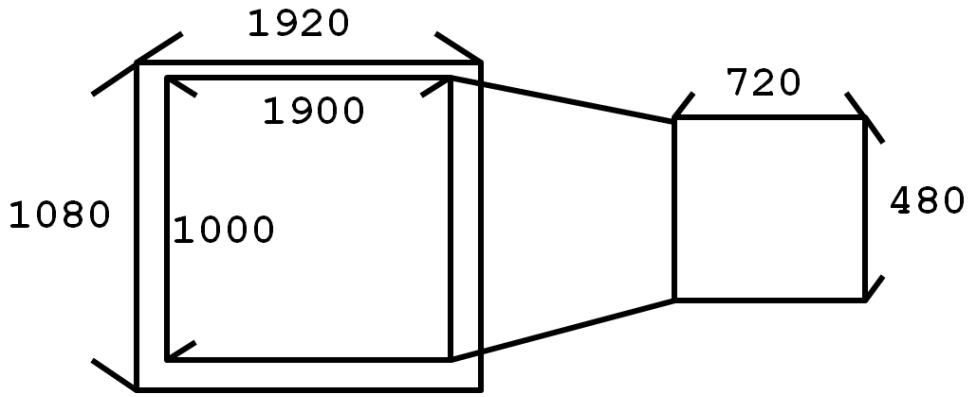
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP_EX( pClient, "C:\\PICTURE1.BMP", 10, 40, 1900, 1000, 720, 480 );
```



11.64 QCAP_SNAPSHOT_BROADCAST_CLIENT_JPG_EX

```
QRESULT QCAP_SNAPSHOT_BROADCAST_CLIENT_JPG_EX(  PVOID       pClient,
                                                CHAR *      pszFilePathName,
                                                ULONG       nCropX,
                                                ULONG       nCropY,
                                                ULONG       nCropW,
                                                ULONG       nCropH,
                                                ULONG       nDstW,
                                                ULONG       nDstH,
                                                ULONG       nQuality,
                                                BOOL        bIsAsync = TRUE,
                                                ULONG       nMilliseconds = 0 )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pszFilePathName	**IN**, Specify the file name to store JPG data
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of snapshot a picture with JPG format, when broadcast client processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

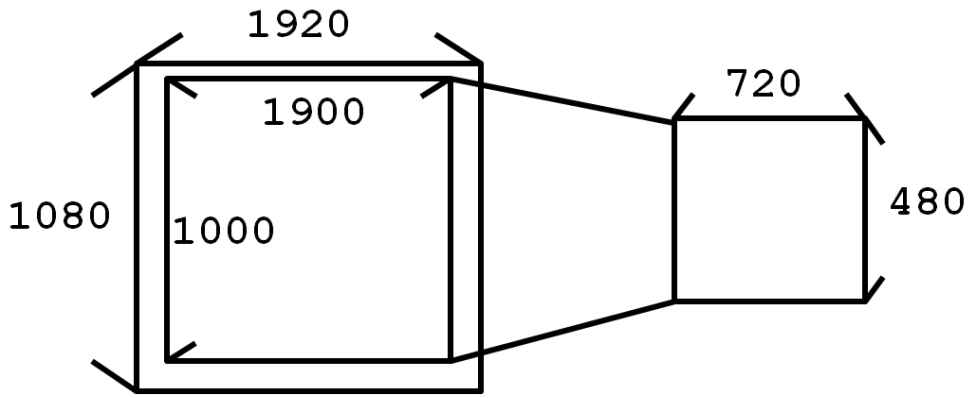
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_BROADCAST_CLIENT_JPG_EX( pClient, "C:\\PICTURE1.JPG", 10, 40, 1900, 1000, 720, 480,100 );
```



11.65 QCAP_GET_OSD_BROADCAST_CLIENT_TEXT_BOUNDARY

```
QRESULT QCAP_GET_OSD_BROADCAST_CLIENT_TEXT_BOUNDARY( PVOID      pClient,
                                                     UINT       iOsdNum,
                                                     CHAR   *   pszString,
                                                     CHAR   *   pszFontFamilyName,
                                                     ULONG      nFontStyle,
                                                     ULONG      nFontSize,
                                                     ULONG  *   pBoundaryWidth,
                                                     ULONG  *   pBoundaryHeight )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output
nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are QCAP_FONT_STYLE_REGULAR = 0x00000000, QCAP_FONT_STYLE_BOLD = 0x00000001, QCAP_FONT_STYLE_ITALIC = 0x00000002, QCAP_FONT_STYLE_BOLDITALIC = 0x00000003, QCAP_FONT_STYLE_UNDERLINE = 0x00000004, QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,
nFontSize	**IN**, Specify the font size used to display the text of OSD output
pBoundaryWidth	**OUT**, Pointer to the storage of boundary width
pBoundaryHeight	**OUT**, Pointer to the storage of boundary height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get size of OSD string, when the users use the broadcast client function.

For example, user can use boundary information to set OSD string location on display window.

Example

```
QCAP_GET_OSD_BROADCAST_CLIENT_TEXT_BOUNDARY( pClient, 0, “CH01”, “Arial”,
                                             QCAP_FONT_STYLE_BOLD, 12,
                                             pBoundaryWidth, pBoundaryHeight );
```

11.66 QCAP_SET_OSD_BROADCAST_CLIENT_TEXT

```
QRESULT QCAP_SET_OSD_BROADCAST_CLIENT_TEXT(  PVOID       pClient,
                                             UINT        iOsdNum,
                                             INT         x,
                                             INT         y,
                                             INT         w,
                                             INT         h,
                                             CHAR *      pszString,
                                             CHAR *      pszFontFamilyName,
                                             ULONG       nFontStyle,
                                             ULONG       nFontSize,
                                             DWORD       dwFontColor,
                                             DWORD       dwBackgroundColor,
                                             ULONG       nTransparent,
                                             INT         nTextStartPosX = 0,
                                             INT         nTextStartPosY = 0,
                                             ULONG       nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify they-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output, if width is -1, QCAP will auto calculate width.
height	**IN**, Specify the vertical height of OSD output, if height is -1, QCAP will auto calculate height.
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output

nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are
	`QCAP_FONT_STYLE_REGULAR = 0x00000000,`
	`QCAP_FONT_STYLE_BOLD = 0x00000001,`
	`QCAP_FONT_STYLE_ITALIC = 0x00000002,`
	`QCAP_FONT_STYLE_BOLDITALIC = 0x00000003,`
	`QCAP_FONT_STYLE_UNDERLINE = 0x00000004,`
	`QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,`

nFontSize	**IN**, Specify the font size used to display the text of OSD output
dwFontColor	**IN**, Specify the font color used to display the text of OSD output
dwBackgroundColor	**IN**, Specify the background color used to display the text of OSD output
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0

	to 255
nTextStartPosX	**IN**, Specify the text scrolling start position X of the upper-left corner of OSD text
nTextStartPosY	**IN**, Specify the text scrolling start position Y of the upper-left corner of OSD text
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a text field objects used for on broadcast client of information.

If you want to get a text field object auto calculated actual width and height value; you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Note

Color space of FontColor is ABGR, don’t forgot to set its alpha value.

Color space of BackgroundColor is ABGR, don’t forgot to set its alpha value.

Transparent is one global parameter for all colors.

Example

```
QCAP_SET_OSD_BROADCAST_CLIENT_TEXT(  pClient, 0, 0, 0, -1, -1, “broadcast_client”, “Arial”,
                                     QCAP_FONT_STYLE_BOLD, 12, 0xFF000000,
                                     0xFFFFFFFF, 128, 0, 0, QCAP_SEQUENCE_STYLE_FOREMOST );
```

11.67 QCAP_SET_OSD_BROADCAST_CLIENT_PICTURE

```
QRESULT QCAP_SET_OSD_BROADCAST_CLIENT_PICTURE( PVOID     pClient,
                                               UINT      iOsdNum,
                                               INT       x,
                                               INT       y,
                                               INT       width,
                                               INT       height,
                                               CHAR *    pszFilePathName,
                                               ULONG     nTransparent
                                               ULONG     nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output If width is -1, QCAP will use original picture width as default.
height	**IN**, Specify the vertical height of OSD output. If height is -1, QCAP will use original picture height as default.
pszFilePathName	**IN**, Specify the file name to display image OSD output, support a file extension of "BMP" as 24 or 32-bit,“JPG” and “PNG”
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to createa static image objects used for on broadcast client of information.

If you want to get a static image object auto calculated actual width and height value;

you should set height and widthparameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_BROADCAST_CLIENT_PICTURE(  pClient, 0, 0, 0, -1, -1, "C:\\SAMPLE.PNG", 128,
                                        QCAP_SEQUENCE_STYLE_FOREMOST );
```

11.68 QCAP_SET_OSD_BROADCAST_CLIENT_BUFFER

```
QRESULT QCAP_SET_OSD_BROADCAST_CLIENT_BUFFER(  PVOID      pClient,
                                               UINT       iOsdNum,
                                               INT        x,
                                               INT        y,
                                               INT        width,
                                               INT        height,
                                               ULONG      nColorSpaceType,
                                               BYTE *     pFrameBuffer,
                                               ULONG      nFrameWidth,
                                               ULONG      nFrameHeight,
                                               ULONG      nFramePitch,
                                               ULONG      nTransparent,
                                               DWORD      dwKeyColor = 0xFFFFFFFF,
                                               ULONG      nKeyColorThreshold = 25,
                                               ULONG      nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output
height	**IN**, Specify the vertical height of OSD output
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameWidth	**IN**, Specify the horizontal width of frame contained in a buffer
nFrameHeight	**IN**, Specify the vertical height of frame contained in a buffer
nFramePitch	**IN**, Specify the distance in bytes between the start of one scan line to the next. If it is 0, we will auto calculate it by width and color space format.

nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
dwKeyColor	**IN**, Specify the key color of OSD, default is 0xFFFFFFFF
nKeyColorThreshold	**IN**, Specify the threshold of key color, the range of 0 to 128
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

* The performance of BGR type is slower than RGB in our alpha blending algorithm.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to createa frame buffer objects used for on broadcast client of information.

The function’s purpose is used to point directly to a valid frame buffer whose entire pixel data,

which is stored in available color space, and display it.

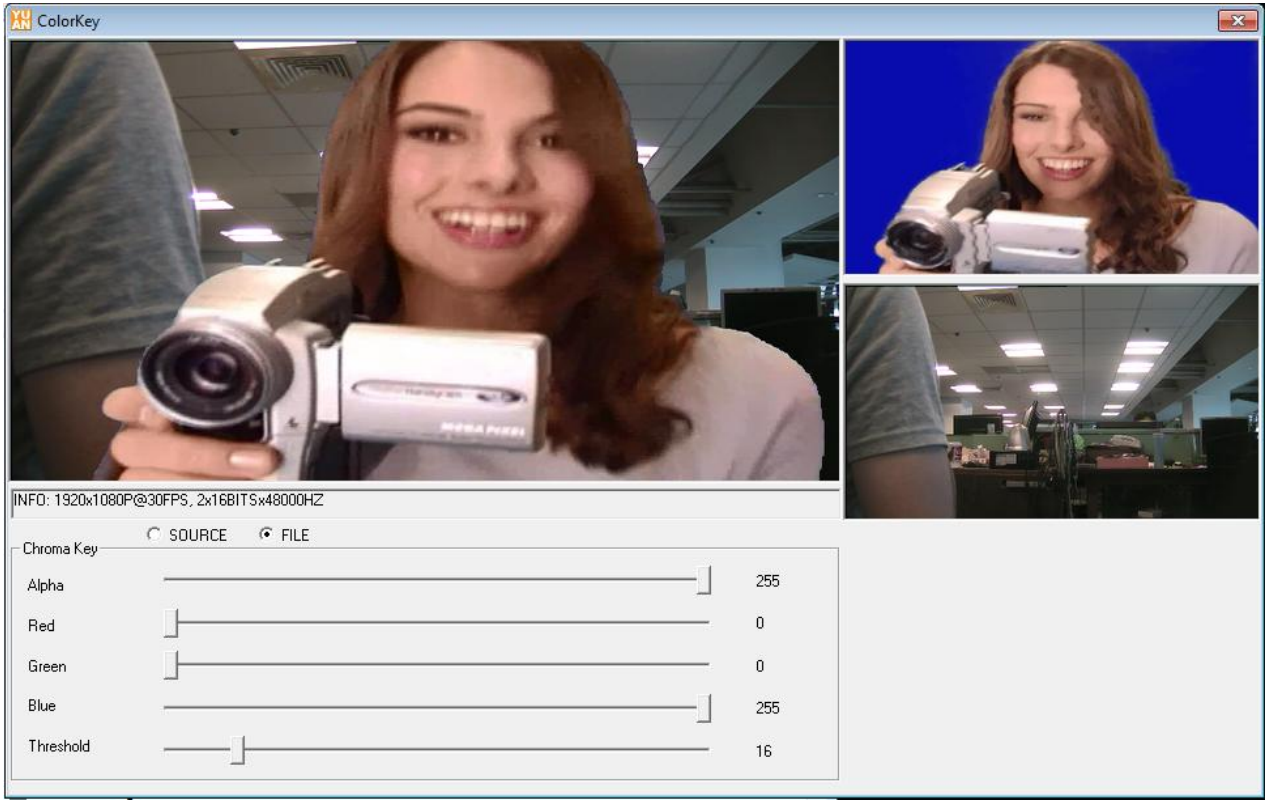
Beside, you also can adjust the width and height parameters to scale the OSD output

And OSD sequence style to video-streams video-recording the OSD output.

Note

User can use this dwKeyColor parament to select colorkey mode Green&Blue mode and RGB mode.

For instance the Blue mode way as show in below



Example

```
QCAP_SET_OSD_BROADCAST_CLIENT_BUFFER( pClient, 0, 0, 0, 1920, 1080, QCAP_COLORSPACE_TYEP_YUY2,
                                      pFrameBuffer, 800, 600, 0, 0xFFFFFFFF
                                      128, QCAP_SEQUENCE_STYLE_FOREMOST );
```

11.69 QCAP_GET_BROADCAST_CLIENT_CUSTOM_PROPERTY

```
QRESULT QCAP_GET_BROADCAST_CLIENT_CUSTOM_PROPERTY(    PVOID     pClient,
                                                      CHAR   *  pszProperty,
                                                      CHAR   *  pszValue )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pszProperty	**IN**, Specify get client custom propery interface
pszValue	**OUT**, Pointer to a variable that stores the specify property value. It cannot be NULL.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is used to get broadcast client value of custom properties.

Example

```
QCAP_GET_BROADCAST_CLIENT_CUSTOM_PROPERTY(  pClient, “Record_Start”, “Start111” );
```

11.70 QCAP_SET_BROADCAST_CLIENT_CUSTOM_PROPERTY

```
QRESULT QCAP_SET_BROADCAST_CLIENT_CUSTOM_PROPERTY(    PVOID     pClient,
                                                      CHAR   *  pszProperty,
                                                      CHAR   *  pszValue )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pszProperty	**IN**, Specify client custom propery interface
pszValue	**IN**, Specifies the property value

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is used to set/change broadcast client value of custom properties.

Example

```
QCAP_SET_BROADCAST_CLIENT_CUSTOM_PROPERTY(  pClient, “Record_Start”, “Start111” );
```

11.71 QCAP_CREATE_3D_BROADCAST_CLIENT

```
QRESULT QCAP_CREATE_3D_BROADCAST_CLIENT(    UINT      iCliNum,
                                            CHAR *    pszURL,
                                            PVOID *   ppClient,
                                            HWND      hAttacheWindowL = NULL,
                                            BOOL      bThumbDrawL = FALSE,
                                            BOOL      bMaintainAspectRatioL = FALSE,
                                            HWND      hAttacheWindowR = NULL,
                                            BOOL      bThumbDrawR = FALSE,
                                            BOOL      bMaintainAspectRatioR = FALSE )
```

Parameter

Name	Description
iCliNum	**IN**, Specify broadcast client number to get client parameters, start from 0 max is 0~63
pszURL	**IN**, In the client side, user can use the URL to access it. For example rtsp://account:password@ip:port/session#num.mpg
ppClient	**OUT**, Handle of the client object
hAttacheWindowL	**IN**, Handle of the window to show the preview video of 3D of left, the default string isNULL
BThumbDrawL	**IN**, Enable/Disable the input thumb draw renderer of 3D of left
bMaintainAspectRatioL	**IN**, Enable/Disable the maintain aspect ratio of 3D of left
HAttacheWindowR	**IN**, Handle of the window to show the preview video of 3D of right,the default string is NULL
BThumbDrawR	**IN**, Enable/Disable the input thumb draw renderer of 3D of right
bMaintainAspectRatioR	**IN**, Enable/Disable the maintain aspect ratio of 3D of right

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create a 3D broadcast client object.

Example

```
QCAP_CREATE_3D_BROADCAST_CLIENT( 0,"rtsp://root:root@127.0.0.1:554/session0.mpg", &pClient,

                                 hWindowL, TRUE, FALSE,

                                 hWindowR, TRUE, FALSE );
```

11.72 QCAP_GET_VIDEO_3D_BROADCAST_CLIENT_DISPLAY_MODE

```
QRESULT QCAP_GET_VIDEO_3D_BROADCAST_CLIENT_DISPLAY_MODE( PVOID     pClient,
                                                         ULONG  *  pStereoDisplayMode,
                                                         BOOL   *  pLeftRightSwap )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pStereoDisplayMode	**OUT**, Pointer to the storage of current Stereo Display Mode
pLeftRightSwap	**OUT**, Pointer to the storage of current Left Right Swap

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get current 3D broadcast client display mode.

Example

```
QCAP_GET_VIDEO_3D_BROADCAST_CLIENT_DISPLAY_MODE( pClient, &nStereoDisplayMode, &nLeftRightSwap );
```

11.73 QCAP_SET_VIDEO_3D_BROADCAST_CLIENT_DISPLAY_MODE

```
QRESULT QCAP_SET_VIDEO_3D_BROADCAST_CLIENT_DISPLAY_MODE( PVOID     pClient,
                                                         ULONG     nStereoDisplayMode,
                                                         BOOL      bLeftRightSwap )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
nStereoDisplayMode	**IN**, Specify 3D display mode. Available values are QCAP_3D_STEREO_DISPLAY_MODE_SIDE_BY_SIDE = 0 QCAP_3D_STEREO_DISPLAY_MODE_TOP_BOTTOM = 1 QCAP_3D_STEREO_DISPLAY_MODE_LINE_BY_LINE = 2 QCAP_3D_STEREO_DISPLAY_MODE_LEFT_ONLY = 3 QCAP_3D_STEREO_DISPLAY_MODE_RIGHT_ONLY = 4
bLeftRightSwap	**IN**, Change the input camera source's (right to left, left to right), the default is FALSE

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set current 3D broadcast client display mode.

Example

```
QCAP_SET_VIDEO_3D_BROADCAST_CLIENT_DISPLAY_MODE( pClient, 0, FALSE );
```

11.74 QCAP_CREATE_COMMUNICATION_ONVIF_SERVER

```
QRESULT QCAP_CREATE_COMMUNICATION_ONVIF_SERVER(   ULONG     nTotalVideoSources,
                                                  ULONG     nTotalAudioSources,
                                                  ULONG     nTotalVideoEncoders,
                                                  ULONG     nTotalAudioEncoders,
                                                  ULONG     nTotalProfiles,
                                                  PVOID  *  ppServer,
                                                  CHAR   *  pszManufacturer = NULL,
                                                  CHAR   *  pszModel = NULL,
                                                  CHAR   *  pszFirmwareVersion = NULL,
                                                  CHAR   *  pszSerialNumber = NULL,
                                                  CHAR   *  pszHardwareID = NULL )
```

Parameter

Name	Description
nTotalVideoSources	**IN**, Specify the total communication video sources
nTotalAudioSources	**IN**, Specify the total communication audio sources
nTotalVideoEncoders	**IN**, Specify the total video encoders
nTotalAudioEncoders	**IN**, Specify the total audio encoders
nTotalProfiles	**IN**, Specify the total communication profile
ppServer	**OUT**, Handle of the communication server object
pszManufacturer	**IN**, Specify the manufacturer name
pszModel	**IN**, Specify the model name
pszFirmwareVersion	**IN**, Specify the firmware version
pszSerialNumber	**IN**, Specify the serial number
pszHardwareID	**IN**, Specify the hardware ID

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create and set communication onvif server information such as Manufacturer name and Model name.

Example

```
QCAP_CREATE_COMMUNICATION_ONVIF_SERVER( 4, 4, 4, 4, 4, &m_pServer );
```

11.75 QCAP_DESTROY_COMMUNICATION_SERVER

```
QRESULT QCAP_DESTROY_COMMUNICATION_SERVER( PVOID pServer )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Destroy communication onvif server object to release its resource.

Example

```
QCAP_DESTROY_COMMUNICATION_SERVER( pServer );
```

11.76 QCAP_START_COMMUNICATION_SERVER

```
QRESULT QCAP_START_COMMUNICATION_SERVER(    PVOID      pServer,
                                            ULONG      nNetworkPort_ONVIF = 8001 )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
nNetworkPort_ONVIF	**IN**, Specify the port number that the server listens on for onvif, the default port is 8001

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start communication onvif server server.

Example

```
QCAP_START_COMMUNICATION_SERVER( pServer, 8001 );
```

11.77 QCAP_STOP_COMMUNICATION_SERVER

```
QRESULT QCAP_STOP_COMMUNICATION_SERVER( PVOID  pServe )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop communication onvif server.

Example

```
QCAP_STOP_COMMUNICATION_SERVER( pServer );
```

11.78 QCAP_GET_COMMUNICATION_SERVER_VIDEO_SOURCE

```
QRESULT QCAP_GET_COMMUNICATION_SERVER_VIDEO_SOURCE(   PVOID     pServer,
                                                      UINT      iVidSrcNum,
                                                      ULONG *   pWidth,
                                                      ULONG *   pHeight,
                                                      ULONG *   pFrameRate )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iVidSrcNum	**IN**, Specify the number of video source
pWidth	**OUT**, Pointer to the storage of the video width for the i-th video streamsession
pHeight	**OUT**, Pointer to the storage of the video height for the i-th video streamsession
pFrameRate	**OUT**, Pointer to the storage of the video framerate for the i-th video streamsession

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get parameters of the i-th video stream session in communication onvif server.

Example

```
QCAP_GET_COMMUNICATION_SERVER_VIDEO_SOURCE( pServer, 0, &nWidth, &nHeight, &nFrameRate );
```

11.79 QCAP_SET_COMMUNICATION_SERVER_VIDEO_SOURCE

```
QRESULT QCAP_SET_COMMUNICATION_SERVER_VIDEO_SOURCE(   PVOID     pServer,
                                                      UINT      iVidSrcNum,
                                                      ULONG     nWidth,
                                                      ULONG     nHeight,
                                                      ULONG     nFrameRate )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iVidSrcNum	**IN**, Specify the number of video source
nWidth	**IN**, Specify the width for the i-th video streamsession
nWidth	**IN**, Specify the height for the i-th video streamsession
nFrameRate	**IN**, Specify the framerate for the i-th video streamsession

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set parameters of the i-th video stream session in communication onvif server.

Example

```
QCAP_SET_COMMUNICATION_SERVER_VIDEO_SOURCE( pServer, 0, 1280, 720, 30 );
```

11.80 QCAP_GET_COMMUNICATION_SERVER_AUDIO_SOURCE

```
QRESULT QCAP_GET_COMMUNICATION_SERVER_AUDIO_SOURCE(   PVOID      pServer,
                                                      UINT       iAudSrcNum,
                                                      ULONG *    pChannels,
                                                      ULONG *    pBitsPerSample,
                                                      ULONG *    pSampleFrequency )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iAudSrcNum	**IN**, Specify the number of audio source
pChannels	**OUT**, Pointer to the storage of the audio channel for the i-th audio streamsession
pBitsPerSample	**OUT**, Pointer to the storage of the audio bitpersample for the i-th audio streamsession
pSampleFrequency	**OUT**, Pointer to the storage of the audio samplefrequency for the i-th audio streamsession

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get parameters of the i-th audio stream session in communication onvif server.

Example

```
QCAP_GET_COMMUNICATION_SERVER_AUDIO_SOURCE( pServer, 0, &nChannels, &nBitsPerSample, &nSampleFrequency );
```

11.81 QCAP_SET_COMMUNICATION_SERVER_AUDIO_SOURCE

```
QRESULT QCAP_SET_COMMUNICATION_SERVER_AUDIO_SOURCE(  PVOID     pServer,
                                                     UINT      iAudSrcNum,
                                                     ULONG     nChannels,
                                                     ULONG     nBitsPerSample,
                                                     ULONG     nSampleFrequency )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iAudSrcNum	**IN**, Specify the number of audio source
nChannels	**IN**, Specify the Channels for the i-th audio streamsession
nBitsPerSample	**IN**, Specify the BitsPerSample for the i-th audio streamsession
nSampleFrequency	**IN**, Specify the SampleFrequency for the i-th audio streamsession

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set parameters of the i-th audio stream session in communication onvif server.

Example

```
QCAP_SET_COMMUNICATION_SERVER_AUDIO_SOURCE( pServer, 0, 2, 16, 48000 );
```

11.82 QCAP_GET_COMMUNICATION_SERVER_VIDEO_ENCODER

```
QRESULT QCAP_GET_COMMUNICATION_SERVER_VIDEO_ENCODER( PVOID     pServer,
                                                     UINT      iVidEncNum,
                                                     UINT      iVidSrcNum,
                                                     ULONG *   pWidth,
                                                     ULONG *   pHeight,
                                                     ULONG *   pEncoderFormat,
                                                     ULONG *   pFrameRate_Min,
                                                     ULONG *   pFrameRate_Max,
                                                     ULONG *   pFrameRate_Default,
                                                     ULONG *   pRecordMode_Min,
                                                     ULONG *   pRecordMode_Max,
                                                     ULONG *   pRecordMode_Default,
                                                     ULONG *   pQuality_Min,
                                                     ULONG *   pQuality_Max,
                                                     ULONG *   pQuality_Default,
                                                     ULONG *   pBitRate_Min,
                                                     ULONG *   pBitRate_Max,
                                                     ULONG *   pBitRate_Default,
                                                     ULONG *   pGOP_Min,
                                                     ULONG *   pGOP_Max,
                                                     ULONG *   pGOP_Default )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iVidEncNum	**IN**, Specify the number of video encoder
iVidSrcNum	**IN**, Specify the number of video source
pWidth	**OUT**, Pointer to the storage of the video width for the i-th video streamsession
pHeight	**OUT**, Pointer to the storage of the video height for the i-th video streamsession
pEncoderFormat	**OUT**, Pointer to the storage of current communication encoder format
pFrameRate_Min	**OUT**, Pointer to the min storage of communication frame rate
pFrameRate_Max	**OUT**, Pointer to the max storage of communication frame rate
pFrameRate_Default	**OUT**, Pointer to the default storage of communication frame rate
pRecordMode_Min	**OUT**, Pointer to the min storage of communication record mode
pRecordMode_Max	**OUT**, Pointer to the max storage of communication record mode
pRecordMode_Default	**OUT**, Pointer to the default storage of communication record mode
pQuality_Min	**OUT**, Pointer to the min storage of communication quality
pQuality_Max	**OUT**, Pointer to the max storage of communication quality
pQuality_Default	**OUT**, Pointer to the default storage of communication quality
pBitRate_Min	**OUT**, Pointer to the min storage of communication bitrate
pBitRate_Max	**OUT**, Pointer to the max storage of communication bitrate

pBitRate_Default	**OUT**, Pointer to the default storage of communication bitrate
pGOP_Min	**OUT**, Pointer to the min storage of communication GOP
pGOP_Max	**OUT**, Pointer to the max storage of communication GOP
pGOP_Default	**OUT**, Pointer to the default storage of communication GOP

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get additional parameters of the i-th video stream session in communication onvif server.

Example

```
QCAP_GET_COMMUNICATION_SERVER_VIDEO_ENCODER( pServer, 0, 0, &nWidth, &nHeight, &nEncoderFormat,
                                             &nFrameRate_Min, &nFrameRate_Max, &nFrameRate_Default,
                                             &nRecordMode_Min, &nRecordMode_Max, &nRecordMode_Default,
                                             &nQuality_Min, &nQuality_Max, &nQuality_Default,
                                             &nBitRate_Min, &nBitRate_Max, &nBitRate_Default,
                                             &nGOP_Min, &nGOP_Max, &nGOP_Default );
```

11.83 QCAP_SET_COMMUNICATION_SERVER_VIDEO_ENCODER

```
QRESULT QCAP_SET_COMMUNICATION_SERVER_VIDEO_ENCODER( PVOID     pServer,
                                                     UINT      iVidEncNum,
                                                     UINT      iVidSrcNum,
                                                     ULONG     nWidth,
                                                     ULONG     nHeight,
                                                     ULONG     nEncoderFormat,
                                                     ULONG     nFrameRate_Min,
                                                     ULONG     nFrameRate_Max,
                                                     ULONG     nFrameRate_Default,
                                                     ULONG     nRecordMode_Min,
                                                     ULONG     nRecordMode_Max,
                                                     ULONG     nRecordMode_Default,
                                                     ULONG     nQuality_Min,
                                                     ULONG     nQuality_Max,
                                                     ULONG     nQuality_Default,
                                                     ULONG     nBitRate_Min,
                                                     ULONG     nBitRate_Max,
                                                     ULONG     nBitRate_Default,
                                                     ULONG     nGOP_Min,
                                                     ULONG     nGOP_Max,
                                                     ULONG     nGOP_Default )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iVidEncNum	**IN**, Specify the number of video encoder
iVidSrcNum	**IN**, Specify the number of video source
nWidth	**IN**, Specify the width for the i-th video streamsession
nWidth	**IN**, Specify the height for the i-th video streamsession
nEncoderFormat	**IN**, Specify encoder format. Available value are `QCAP_ENCODER_FORMAT_H264 = 0,`
nFrameRate_Min	**IN**, Specify the minimum frame rate
nFrameRate_Max	**IN**, Specify the maximum frame rate
nFrameRate_Default	**IN**, Specify the default frame rate

nRecordMode_Min	**IN**, Specify the minimum record mode. Available values are
	`QCAP_RECORD_MODE_VBR = 0`
	`QCAP_RECORD_MODE_CBR = 1`
	`QCAP_RECORD_MODE_ABR = 2`

nRecordMode_Max	**IN**, Specify the maximum frame rate. Available values are
	`QCAP_RECORD_MODE_VBR = 0`
	`QCAP_RECORD_MODE_CBR = 1`
	`QCAP_RECORD_MODE_ABR = 2`

nRecordMode_Default	**IN**, Specify the default frame rate. Available values are
	`QCAP_RECORD_MODE_VBR = 0`
	`QCAP_RECORD_MODE_CBR = 1`
	`QCAP_RECORD_MODE_ABR = 2`

nQuality_Min	**IN**, Specify the minimum quality is 0. It is used for VBR and ABR.

nQuality_Max	**IN**, Specify the maximum quality is 10000. It is used for VBR and ABR.
nQuality_Default	**IN**, Specify the default quality value. It is used for VBR and ABR.
nBitRate_Min	**IN**, Specify the minimum bitrate value.
nBitRate_Max	**IN**, Specify the maximum bitrate value.
nBitRate_Default	**IN**, Specify the default bitrate value.
nGOP_Min	**IN**, Specify the minimum GOP is 0.
nGOP_Max	**IN**, Specify the maximum GOP is 255.
nGOP_Default	**IN**, Specify the default GOP value.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set additional parameters of the i-th video stream session in communication onvif server.

Example

```
QCAP_SET_COMMUNICATION_SERVER_VIDEO_ENCODER( m_pServer, 0, 0, 1920, 1080, QCAP_ENCODER_FORMAT_H264,
                                             0, 30, 30, 0, 2, 1, 0, 10000, 8000,
                                             1*1024*1024, 16*1024*1024, 4*1024*1024,
                                             0, 255, 30  );
```

11.84 QCAP_GET_COMMUNICATION_SERVER_AUDIO_ENCODER

```
QRESULT QCAP_GET_COMMUNICATION_SERVER_AUDIO_ENCODER( PVOID     pServer,
                                                     UINT      iAudEncNum,
                                                     UINT      iAudSrcNum,
                                                     ULONG *   pEncoderFormat,
                                                     ULONG *   pBitRate_Min,
                                                     ULONG *   pBitRate_Max,
                                                     ULONG *   pBitRate_Default )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iAudEncNum	**IN**, Specify the number of audio encoder
iAudSrcNum	**IN**, Specify the number of audio source
pEncoderFormat	**OUT**, Pointer to the storage of current communication encoder format
pBitRate_Min	**OUT**, Pointer to the min storage of communication bit rate
pBitRate_Max	**OUT**, Pointer to the max storage of communication bit rate
pBitRate_Default	**OUT**, Pointer to the default storage of communication bit rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get additional parameters of the i-th audio stream session in communication onvif server.

Example

```
QCAP_GET_COMMUNICATION_SERVER_AUDIO_ENCODER( pServer, 0, 0, &nEncoderFormat, &nBitRate_Min,
                                             &nBitRate_Max, &nBitRate_Default );
```

11.85 QCAP_SET_COMMUNICATION_SERVER_AUDIO_ENCODER

```
QRESULT QCAP_SET_COMMUNICATION_SERVER_AUDIO_ENCODER( PVOID     pServer,
                                                     UINT      iAudEncNum,
                                                     UINT      iAudSrcNum,
                                                     ULONG     nEncoderFormat,
                                                     ULONG     nBitRate_Min,
                                                     ULONG     nBitRate_Max,
                                                     ULONG     nBitRate_Default )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iAudEncNum	**IN**, Specify the number of audio encoder
iAudSrcNum	**IN**, Specify the number of audio source
nEncoderFormat	**IN**, Specify encoder format. Available value are QCAP_ENCODER_FORMAT_PCM = 0 QCAP_ENCODER_FORMAT_AAC = 1 QCAP_ENCODER_FORMAT_AAC_RAW = 1 QCAP_ENCODER_FORMAT_AAC_ADTS = 2
nBitRate_Min	**IN**, Specify the minimum bit rate
nBitRate_Max	**IN**, Specify the maximum bit rate
nBitRate_Default	**IN**, Specify the default bit rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set additional parameters of the i-th audio stream session in communication onvif server.

Example

```
QCAP_SET_COMMUNICATION_SERVER_AUDIO_ENCODER( m_pServer, 0, 0, QCAP_ENCODER_FORMAT_AAC,
                                             0, 12*1024*1024, 4*1024*1024 );
```

11.86 QCAP_GET_COMMUNICATION_SERVER_PROFILE

```
QRESULT QCAP_GET_COMMUNICATION_SERVER_PROFILE( PVOID     pServer,
                                               UINT      iProNum,
                                               CHAR * *  ppszURL,
                                               UINT *    pVidEncNum,
                                               UINT *    pAudEncNum )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iProNum	**IN**, Specify the number of portal
ppszURL	**OUT**, Pointer to the storage of URL
pVidEncNum	**OUT**, Pointer to the storage of video encoder number
pAudEncNum	**OUT**, Pointer to the storage of audio encoder number

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get additional parameters of the i-th profile in communication onvif server.

Example

```
QCAP_GET_COMMUNICATION_SERVER_PROFILE( m_pServer, 0, &pszURL, &nVidEncNum, &nAudEncNum );
```

11.87 QCAP_SET_COMMUNICATION_SERVER_PROFILE

```
QRESULT QCAP_SET_COMMUNICATION_SERVER_PROFILE( PVOID     pServer,
                                               UINT      iProNum,
                                               CHAR *    pszURL,
                                               UINT      iVidEncNum,
                                               UINT      iAudEncNum )
```

Parameter

Name	Description
pServer	**IN**, Handle of the communication server object
iProNum	**IN**, Specify the number of porfile
ppszURL	**IN**, user can use the URL to access it. For example, "rtsp://root:root@127.0.0.1:800/session0.mpg"
iVidEncNum	**IN**, Specify the video encoder number
iAudEncNum	**IN**, Specify the audio encoder number

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set additional parameters of the i-th profile in communication onvif server.

Example

```
QCAP_SET_COMMUNICATION_SERVER_PROFILE( m_pServer, 0, "rtsp://root:root@10.10.80.24:800/session0.mpg",

                                                0, 0 );
```

11.88 An Broadcasting Software Programming Guides

```
//BROADCASTING SERVER EXAMPLE:
//
// STEP#01
//
QCAP_CREATE_BROADCAST_RTSP_SERVER( 0, 4 /*4 CHANNELS*/, &pServer, "root", "root", 554 );

//        CH01 URL -> rtsp://root:root@127.0.0.1:554/session0.mpg
//
//        CH02 URL -> rtsp://root:root@127.0.0.1:554/session1.mpg
//
//        CH03 URL -> rtsp://root:root@127.0.0.1:554/session2.mpg
//
//        CH04 URL -> rtsp://root:root@127.0.0.1:554/session3.mpg

// STEP#02
//
QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY(   pServer,
                                            0 /*CH01*/,
                                            QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK,
                                            QCAP_ENCODER_FORMAT_H264,
                                            QCAP_COLORSPACE_TYEP_YUY2,
                                            480,
                                            270,
                                            30,
                                            QCAP_RECORD_MODE_CBR,
                                            8000,
                                            1000000,
                                            30,
                                            0,
                                            0,
```



```
                                        NULL,
                                        FALSE,
                                        FALSE,
                                        QCAP_BROADCAST_FLAG_NETWORK | QCAP_BROADCAST_FLAG_ENCODE );

QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY(   pServer,
                                        1 /*CH02*/,
                                        QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK,
                                        QCAP_ENCODER_FORMAT_H264,
                                        QCAP_COLORSPACE_TYEP_YUY2,
                                        480,
                                        270,
                                        30,
                                        QCAP_RECORD_MODE_CBR,
                                        8000,
                                        1000000,
                                        30,
                                        0,
                                        0,
                                        NULL,
                                        FALSE,
                                        FALSE,
                                        QCAP_BROADCAST_FLAG_NETWORK | QCAP_BROADCAST_FLAG_ENCODE );

QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY(   pServer,
                                        2 /*CH03*/,
                                        QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK,
                                        QCAP_ENCODER_FORMAT_H264,
                                        QCAP_COLORSPACE_TYEP_YUY2,
                                        480,
                                        270,
                                        30,
```



```
                                        QCAP_RECORD_MODE_CBR,
                                        8000,
                                        1000000,
                                        30,
                                        0,
                                        0,
                                        NULL,
                                        FALSE,
                                        FALSE,
                                        QCAP_BROADCAST_FLAG_NETWORK | QCAP_BROADCAST_FLAG_ENCODE );

QCAP_SET_VIDEO_BROADCAST_SERVER_PROPERTY(   pServer,
                                        3 /*CH04*/,
                                        QCAP_ENCODER_TYPE_INTEL_MEDIA_SDK,
                                        QCAP_ENCODER_FORMAT_H264,
                                        QCAP_COLORSPACE_TYEP_YUY2,
                                        480,
                                        270,
                                        30,
                                        QCAP_RECORD_MODE_CBR,
                                        8000,
                                        1000000,
                                        30,
                                        0,
                                        0,
                                        NULL,
                                        FALSE,
                                        FALSE,
                                        QCAP_BROADCAST_FLAG_NETWORK | QCAP_BROADCAST_FLAG_ENCODE );

QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY(   pServer,
                                        0 /*CH01*/,
```

```
                                        QCAP_ENCODER_TYPE_SOFTWARE,
                                        QCAP_ENCODER_FORMAT_AAC,
                                        2,
                                        16,
                                        48000,
                                        0 );

QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY(   pServer,
                                        1 /*CH02*/,
                                        QCAP_ENCODER_TYPE_SOFTWARE,
                                        QCAP_ENCODER_FORMAT_AAC,
                                        2,
                                        16,
                                        48000,
                                        0 );

QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY(   pServer,
                                        2 /*CH03*/,
                                        QCAP_ENCODER_TYPE_SOFTWARE,
                                        QCAP_ENCODER_FORMAT_AAC,
                                        2,
                                        16,
                                        48000,
                                        0 );

QCAP_SET_AUDIO_BROADCAST_SERVER_PROPERTY(   pServer,
                                        3 /*CH04*/,
                                        QCAP_ENCODER_TYPE_SOFTWARE,
                                        QCAP_ENCODER_FORMAT_AAC,
                                        2,
                                        16,
                                        48000,
```



```
                                                0 );
// STEP#03
//
QCAP_START_BROADCAST_SERVER( pServer );


// STEP#04
//
QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(pServer,
                                                     0 /*CH01*/,
                                                     QCAP_COLORSPACE_TYEP_YUY2,
                                                     1920,
                                                     1080,
                                                     pFrameBuffer,
                                                     nFrameBufferLen );

QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(pServer,
                                                     1 /*CH02*/,
                                                     QCAP_COLORSPACE_TYEP_YUY2,
                                                     1920,
                                                     1080,
                                                     pFrameBuffer,
                                                     nFrameBufferLen );

QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(pServer,
                                                     2 /*CH03*/,
                                                     QCAP_COLORSPACE_TYEP_YUY2,
                                                     1920,
                                                     1080,
                                                     pFrameBuffer,
                                                     nFrameBufferLen );
```



```
QCAP_SET_VIDEO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(pServer,
                                                     3 /*CH04*/,
                                                     QCAP_COLORSPACE_TYEP_YUY2,
                                                     1920,
                                                     1080,
                                                     pFrameBuffer,
                                                     nFrameBufferLen );

QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER(pServer, 0 /*CH01*/,pFrameBuffer, nFrameBufferLen );

QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 1 /*CH02*/, pFrameBuffer, nFrameBufferLen );

QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 2 /*CH03*/, pFrameBuffer, nFrameBufferLen );

QCAP_SET_AUDIO_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pServer, 3 /*CH04*/, pFrameBuffer, nFrameBufferLen );

// STEP#05
//
QCAP_STOP_BROADCAST_SERVER( pServer );

// STEP#06
//
QCAP_DRSTROY_BROADCAST_SERVER( pServer );
```

12 Animation Function API

12.1 QCAP_CREATE_ANIMATION_CLIP

```
QRESULT QCAP_CREATE_ANIMATION_CLIP ( CHAR  *   pszAniFileName,
                                     PVOID *   ppClip,
                                     ULONG *   pTotalFrames,
                                     ULONG *   pTotalSprites,
                                     ULONG *   pColorSpaceType,
                                     ULONG *   pWidth,
                                     ULONG *   pHeight )
```

Parameter

Name	Description
pszAniFileName	**IN**, Specify file name of animation
ppClip	**OUT**, Pointer to the storage of the Clip
pTotalFrames	**OUT**, Pointer to the storage of the total frames
pColorSpaceType	**OUT**, Pointer to the storage of animation clip color space type
pWidth	**OUT**, Pointer to the storage of animation clip width
pHeight	**OUT**, Pointer to the storage of animation clip height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create animation clip, the parameters, pTotalFrames, pColorSpaceType, pWidth and pHeight are definded in XML File.

Example

```
QCAP_CREATE_ANIMATION_CLIP( "EXAMPLE.COOL.QUAD.XML", &pClip, &nFrames, &nSprites,
                            &nColorSpaceType, &nWidth, &nHeight );
```

12.2 QCAP_DESTROY_ANIMATION_CLIP

```
QRESULT QCAP_DESTROY_ANIMATION_CLIP( PVOID pClip )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Destroy animation clip to release its resource.

Note

If you have use `QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE`,

you must to destroy the Animation clip before device destroy or to set source to NULL.

Example

```
QCAP_DESTROY_ANIMATION_CLIP( pClip );
```

12.3 QCAP_START_ANIMATION_CLIP

```
QRESULT QCAP_START_ANIMATION_CLIP( PVOID pClip )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start animation clip.

Example

```
QCAP_START_ANIMATION_CLIP( pClip );
```

12.4 QCAP_STOP_ANIMATION_CLIP

```
QRESULT QCAP_STOP_ANIMATION_CLIP( PVOID pClip )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop animation clip.

Example

```
QCAP_STOP_ANIMATION_CLIP( pClip );
```

12.5 QCAP_STEP_ANIMATION_CLIP

```
QRESULT QCAP_STEP_ANIMATION_CLIP(  PVOID       pClip,
                                   UINT  *     pFrameNum,
                                   BYTE  * *   ppFrameBuffer,
                                   ULONG *     pFrameBufferLen,
                                   BOOL        bClearBackground = TRUE )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
pFrameNum	**OUT**, Pointer to the storage of the frame num
ppFrameBuffer	**OUT**, Pointer to the storage of the frame buffer
pFrameBufferLen	**OUT**, Pointer to the storage of the frame buffer len
bClearBackground	**IN**, Enable/Disable the Clear Background

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to execution animation clip by continuous step by step.

Every time, the function is called, the Animation clip will go to next frame and output current frame result as show in Fig.2.

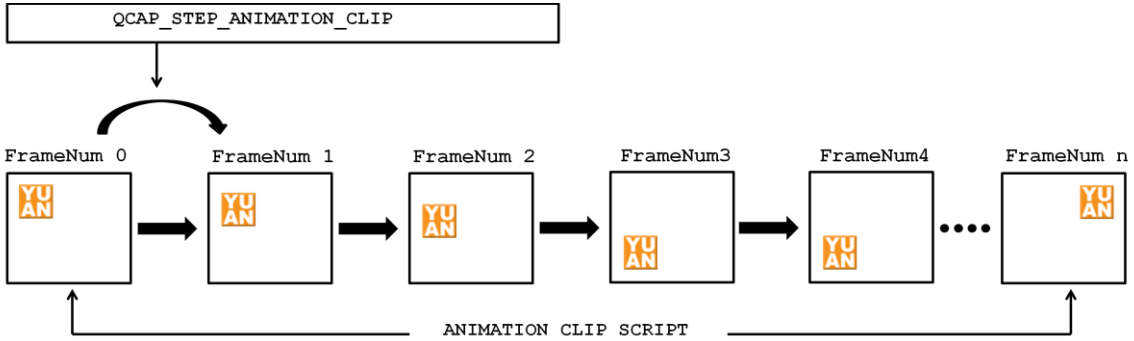


Fig.2

Example

```
QCAP_STEP_ANIMATION_CLIP( pClip, &pFrameNum, &pFrameBuffer, &nFrameBufferLen, TRUE );
```

12.6 QCAP_SEEK_ANIMATION_CLIP

```
QRESULT QCAP_SEEK_ANIMATION_CLIP(  PVOID     pClip,
                                   UINT      iFrameNum,
                                   BYTE * * ppFrameBuffer,
                                   ULONG *   pFrameBufferLen,
                                   BOOL      bClearBackground = TRUE )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iFrameNum	**IN**, Specifythe number of frame
ppFrameBuffer	**OUT**, Pointer to the input source buffer
pFrameBufferLen	**OUT**, Pointer to the input source buffer's size
bClearBackground	**IN**, Enable/Disable the Clear Background

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to execution animation clip by seek.

Every time, the function is called, the Animation clip can jump any frame and output current frame result as show in Fig.3.

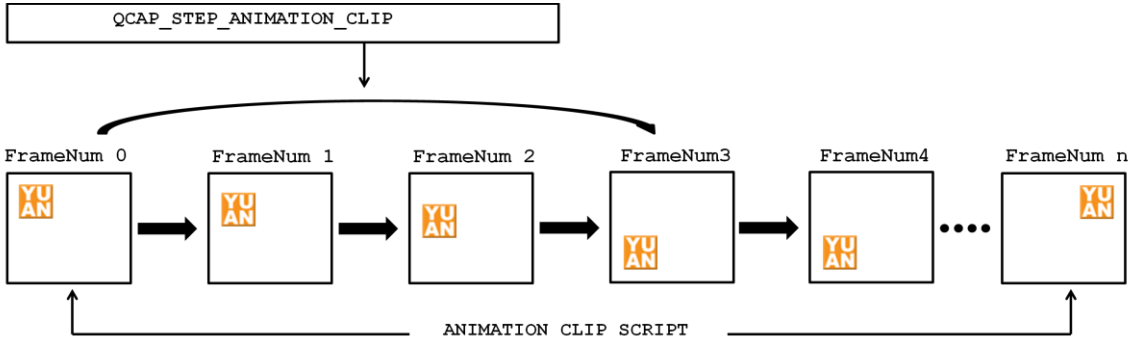


Fig.3

Example

```
QCAP_SEEK_ANIMATION_CLIP( pClip, iFrameNum, &pFrameBuffer, &nFrameBufferLen, TRUE );
```

12.7 QCAP_GET_ANIMATION_CLIP_SPRITE_TRANSFORM_PROPERTY

```
QRESULT QCAP_GET_ANIMATION_CLIP_SPRITE_TRANSFORM_PROPERTY( PVOID   pClip,
                                                           UINT    iSpriteNum,
                                                           UINT    iFrameNum,
                                                           INT *   pPositionX,
                                                           INT *   pPositionY,
                                                           INT *   pScaleW,
                                                           INT *   pScaleH )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iSpriteNum	**IN**, Specify the number of channel sprite, start from 0
iFrameNum	**IN**, Specify the number of frame
pPositionX	**OUT**, Pointer to the storage of current horizontal width of position X
pPositionY	**OUT**, Pointer to the storage of current vertical height of position Y
pScaleW	**OUT**, Pointer to the storage of current transform down scale width
pScaleH	**OUT**, Pointer to the storage of current transform down scale height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get animation clip sprite transform property.

Example

```
QCAP_GET_ANIMATION_CLIP_SPRITE_TRANSFORM_PROPERTY( pClip, 0, 100, &nPositionX, &nPositionY,

                                                   &nScaleW, &nScaleH );
```

12.8 QCAP_SET_ANIMATION_CLIP_SPRITE_TRANSFORM_PROPERTY

```
QRESULT QCAP_SET_ANIMATION_CLIP_SPRITE_TRANSFORM_PROPERTY( PVOID   pClip,
                                                           UINT    iSpriteNum,
                                                           UINT    iFrameNum,
                                                           INT     nPositionX,
                                                           INT     nPositionY,
                                                           INT     nScaleW,
                                                           INT     nScaleH )
```

Parameter

Name	Description
pClip	**IN**,Handle of the clip object
iSpriteNum	**IN**,Specify the number of channel sprite, start from 0
iFrameNum	**IN**,Specify the number of frame
pPositionX	**IN**,Specify the horizontal width of positionX
pPositionY	**IN**,Specify the vertical height of position Y
pScaleW	**IN**,Specify the transform down scale width
pScaleH	**IN**,Specify the transform down scale height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set animation clip sprite transform property.

Example

```
QCAP_SET_ANIMATION_CLIP_SPRITE_TRANSFORM_PROPERTY( pClip, 0, 100, 150, 150, 100, 100 );
```

12.9 QCAP_GET_ANIMATION_CLIP_SPRITE_ALPHA_PROPERTY

```
QRESULT QCAP_GET_ANIMATION_CLIP_SPRITE_ALPHA_PROPERTY (  PVOID       pClip,
                                                         UINT        iSpriteNum,
                                                         UINT        iFrameNum,
                                                         double *    pAlpha )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iSpriteNum	**IN**, Specify the number of channel sprite, start from 0
iFrameNum	**IN**, Specify the number of frame
pAlpha	**OUT**, Pointer to the storage of current alpha value

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get animation clip sprite alpha property.

Example

```
QCAP_GET_ANIMATION_CLIP_SPRITE_ALPHA_PROPERTY( pClip, 0, 100, &nAlpha );
```

12.10 QCAP_SET_ANIMATION_CLIP_SPRITE_ALPHA_PROPERTY

```
QRESULT QCAP_SET_ANIMATION_CLIP_ALPHA_PROPERTY(  PVOID      pClip,
                                                 UINT       iSpriteNum,
                                                 UINT       iFrameNum,
                                                 double     nAlpha )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iSpriteNum	**IN**, Specify the number of channel sprite, start from 0
iFrameNum	**IN**, Specify the number of frame
nAlpha	**IN**, Specify the alpha value

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get animation clip sprite alpha property.

Example

```
QCAP_SET_ANIMATION_CLIP_SPRITE_ALPHA_PROPERTY( pClip, 0, 100, 50 );
```

12.11 QCAP_SET_ANIMATION_CLIP_SPRITE_PICTURE

```
QRESULT QCAP_SET_ANIMATION_CLIP_SPRITE_PICTURE( PVOID    pClip,
                                                UINT     iSpriteNum,
                                                CHAR *   pszFilePathName )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iSpriteNum	**IN**, Specify the number of channel sprite, start from 0
pszFilePathName	**IN**, Specify the file name to store BMP, JPG, or PNG data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set a picture on animation clip’s sprite.

Example

```
QCAP_SET_ANIMATION_CLIP_SPRITE_PICTURE( pClip, 0, “C:\\PICTURE1.BMP” );
```

12.12 QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER

```
QRESULT QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER ( PVOID    pClip,
                                                UINT     iSpriteNum,
                                                ULONG    nColorSpaceType,
                                                ULONG    nWidth,
                                                ULONG    nHight,
                                                BYTE *   pFrameBuffer,
                                                ULONG    nFrameBufferLen,
                                                BOOL     bCloneCopy = FALSE )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iSpriteNum	**IN**, Specify the number of channel sprite, start from 0
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify the horizontal width of frame contained in a buffer
nHeight	**IN**, Specify the vertical height of frame contained in a buffer
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
bCloneCopy	**IN**, Enable/Disable the Clone copy

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set each channels buffer of animation clip.

Developer can call it to update sprites buffer.

Example

```
QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER( pClip, 0, QCAP_COLORSPACE_TYEP_YUY2, FALSE );
```

12.13 QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER_EX

```
QRESULT QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER_EX( PVOID     pClip,
                                                  UINT      iSpriteNum,
                                                  ULONG     nColorSpaceType,
                                                  ULONG     nWidth,
                                                  ULONG     nHight,
                                                  BYTE *    pFrameBuffer,
                                                  ULONG     nFrameBufferLen,
                                                  ULONG     nCropX,
                                                  ULONG     nCropY,
                                                  ULONG     nCropW,
                                                  ULONG     nCropH,
                                                  BOOL      bCloneCopy = FALSE )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iSpriteNum	**IN**, Specify the number of channel sprite, start from 0
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify the horizontal width of frame contained in a buffer
nHeight	**IN**, Specify the vertical height of frame contained in a buffer
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
bCloneCopy	**IN**, Enable/Disable the Clone copy

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of each channels buffer of animation clip.

Developer can call it to update sprites buffer.

Example

```
QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER_EX( pClip, 0, QCAP_COLORSPACE_TYEP_YUY2, 10, 40, 1900, 1000, FALSE );
```

12.14 QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE

```
QRESULT QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE( PVOID     pClip,
                                               UINT      iSpriteNum,
                                               PVOID     pDevice,
                                               ULONG     nSequenceStyle,
                                               BOOL      bCloneCopy = FALSE )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iSpriteNum	**IN**, Specifythe number of channel sprite, start from 0
pDevice	**IN**, Handle of the capture card object
nSequenceStyle	**IN**, Specify sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002
bCloneCopy	**IN**, Enable/Disable the Clone copy

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set each channels source of animation clip.

It is one helper function, it can help user to use complex QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER function then to one attach live device channel on it, when device's preview date coming, the QCAP_SET_ANIMATION_CLIP_SPRITE_BUFFER will be called AUTO.

Example

```
QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE( pClip, 0, QCAP_COLORSPACE_TYEP_YUY2, FALSE );
```

12.15 QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE_EX

```
QRESULT QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE_EX (    PVOID  pClip,
                                                      UINT   iSpriteNum,
                                                      PVOID  pDevice,
                                                      ULONG  nCropX,
                                                      ULONG  nCropY,
                                                      ULONG  nCropW,
                                                      ULONG  nCropH,
                                                      ULONG  nSequenceStyle =
                                                             QCAP_SEQUENCE_STYLE_FOREMOST,
                                                      BOOL   bCloneCopy = FALSE )
```

Parameter

Name	Description
pClip	**IN**, Handle of the clip object
iSpriteNum	**IN**, Specify the number of channel sprite, start from 0
pDevice	**IN**, Handle of the capture card object
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
nSequenceStyle	**IN**, Specify sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002
bCloneCopy	**IN**, Enable/Disable the Clone copy

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set crop parameters of each channels source of animation clip.

Example

```
QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE_EX( pClip, 0, 10, 40, 1900, 1000, QCAP_COLORSPACE_TYEP_YUY2, FALSE );
```

12.16 An Animation Clip Software Programming Guides

```
//ANIMATIONCLIP EXAMPLE:
// STEP#01
//
// QCAP_CREATE_ANIMATION_CLIP( "EXAMPLE.FADEOUT.XML",
                               &m_pClip,
                               &m_nClipFrames,
                               &m_nClipSprites,
                               &m_nClipColorSpaceType,
                               &m_nClipWidth,
                               &m_nClipHeight );

// STEP#02
//
// QCAP_SET_ANIMATION_CLIP_SPRITE_SOURCE(m_pClip,
                                         0,
                                         m_pDevices,
                                         QCAP_COLORSPACE_TYEP_YUY2,
                                         FALSE );

// STEP#03
//
// QCAP_SRART_ANIMATION_CLIP( m_pClip )
```



```
// STEP#04
//
// QCAP_STEP_ANIMATION_CLIP( m_pClip,
                             &iFrameNum,
                             &pClipFrameBuffer,
                             &nClipFrameBufferLen,
                             TRUE );

// STEP#05
//
// QCAP_STOP_ANIMATION_CLIP( m_pClip );

// STEP#06
//
// QCAP_DESTROY_ANIMATION_CLIP( m_pClip );
```

13 Virtual Camera Function API

13.1 QCAP_CREATE_VIRTUAL_CAMERA

```
QRESULT QCAP_CREATE_VIRTUAL_CAMERA(  UINT       iCamNum,
                                     PVOID *    ppCamera )
```

Parameter

Name	Description
iCamNum	**IN**, Specify the camera num, start from 0 max is 0~63
ppCamera	**OUT**,Handle of the camera object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

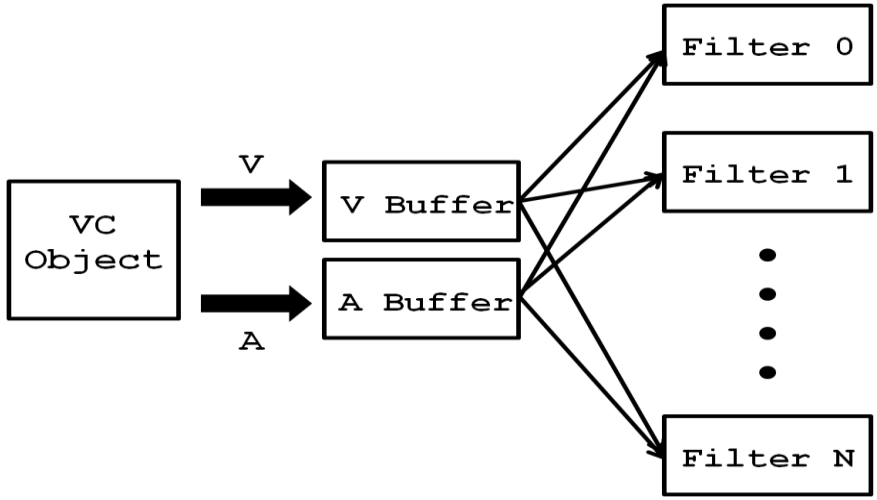
User can use this function to create one virtual camera object.

Virtual camera can allow you to create one share mapping of video frame buffer to another 3rd party software, such as QQ or Skype.

Customer wants to use virtual camera need register the VIRCAM.X86.AX filter into your system,

we call it as Custom Virtual Camera Filter. You can find it from our QCAP\TOOLS folder.

The virtual camera object likes one streaming sender and the Custom Virtual Camera Filter likes one streaming receiver.



Example

```
QCAP_CREATE_VIRTUAL_CAMERA( 0, &pCamera );
```

13.2 QCAP_DESTROY_VIRTUAL_CAMERA

```
QRESULT QCAP_DESTROY_VIRTUAL_CAMERA( PVOID pCamera )
```

Parameter

Name	Description
pCamera	**IN**,Handle of the camera object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Destroy camera object to release its resource.

Example

```
QCAP_DESTROY_VIRTUAL_CAMERA( pCamera );
```

13.3 QCAP_GET_VIDEO_VIRTUAL_CAMERA_PROPERTY

```
QRESULT QCAP_GET_VIDEO_VIRTUAL_CAMERA_PROPERTY (  PVOID     pCamera,
                                                  ULONG *   pColorSpaceType,
                                                  ULONG *   pWidth,
                                                  ULONG *   pHeight,
                                                  double *  pFrameRate )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pColorSpaceType	**OUT**, Pointer to the storage of current color space type
pWidth	**OUT**, Pointer to the storage of current video width
pHeight	**OUT**, Pointer to the storage of current video height
pFrameRate	**OUT**, Pointer to the storage of current frame rate

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get the video format of virtual camera.

Example

```
QCAP_GET_VIDEO_VIRTUAL_CAMERA_PROPERTY( pCamera, &nColorSpaceType, &nWidth, &nHeight, &nFrameRate );
```

13.4 QCAP_SET_VIDEO_VIRTUAL_CAMERA_PROPERTY

```
QRESULT QCAP_SET_VIDEO_VIRTUAL_CAMERA_PROPERTY (  PVOID     pCamera,
                                                  ULONG     nColorSpaceType,
                                                  ULONG     nWidth,
                                                  ULONG     nHeight,
                                                  double    dFrameRate )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_H264
nWidth	**IN**, Specify encoder width.
nHeight	**IN**, Specify encoder height.
dFrameRate	**IN**, Specify encoder frame rate.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set video format of virtual camera.

It will change the output format of Virtual Camera Filter’s video output pin.

Example

```
QCAP_SET_VIDEO_VIRTUAL_CAMERA_PROPERTY( pCamera, QCAP_COLORSPACE_TYEP_YV12,
                                        nWidth, nHeight,
                                        dFrameRate );
```

13.5 QCAP_GET_AUDIO_VIRTUAL_CAMERA_PROPERTY

```
QRESULT QCAP_GET_AUDIO_VIRTUAL_CAMERA_PROPERTY (  PVOID     pCamera,
                                                  ULONG *   pChannels,
                                                  ULONG *   pBitPerSample,
                                                  ULONG *   pSampleFrequency )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pChannels	**OUT**, Pointer to the storage of current audio encoder channels
pBitsPerSample	**OUT**, Pointer to the storage of current audio encoder bits per sample
pSampleFrequency	**OUT**, Pointer to the storage of current audio encoder sample frequency

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get audio format of virtual camera.

Example

```
QCAP_GET_AUDIO_VIRTUAL_CAMERA_PROPERTY( pCamera, &nChannels, &nBitsPerSample, &nSampleFrequency );
```

13.6 QCAP_SET_AUDIO_VIRTUAL_CAMERA_PROPERTY

```
QRESULT QCAP_SET_AUDIO_VIRTUAL_CAMERA_PROPERTY(   PVOID  pCamera,
                                                  ULONG  nChannels,
                                                  ULONG  nBitsPerSample,
                                                  ULONG  nSampleFrequency )
```

Parameter

Name	Description
pCamera	**IN**,Handle of the camera object
nChannels	**IN**, Specify encoder channels.
nBitsPerSample	**IN**, Specify encoder bits per sample.
nSampleFrequency	**IN**, Specify encoder sample frequency.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio format of virtual camera.

It will change the output format of Virtual Camera Filter’s audio output pin.

Example

```
QCAP_SET_AUDIO_VIRTUAL_CAMERA_PROPERTY( pCamera, nChannels, nBitsPerSample, nSampleFrequency );
```

13.7 QCAP_SET_VIDEO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_VIDEO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER( PVOID     pCamera,
                                                            ULONG     nColorSpaceType,
                                                            ULONG     nWidth,
                                                            ULONG     nHeight,
                                                            BYTE *    pFrameBuffer,
                                                            ULONG     nFrameBufferLen,
                                                            double    dSampleTime = 0.0 )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32
nWidth	**IN**, Specify video width
nHeight	**IN**, Specify video height
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

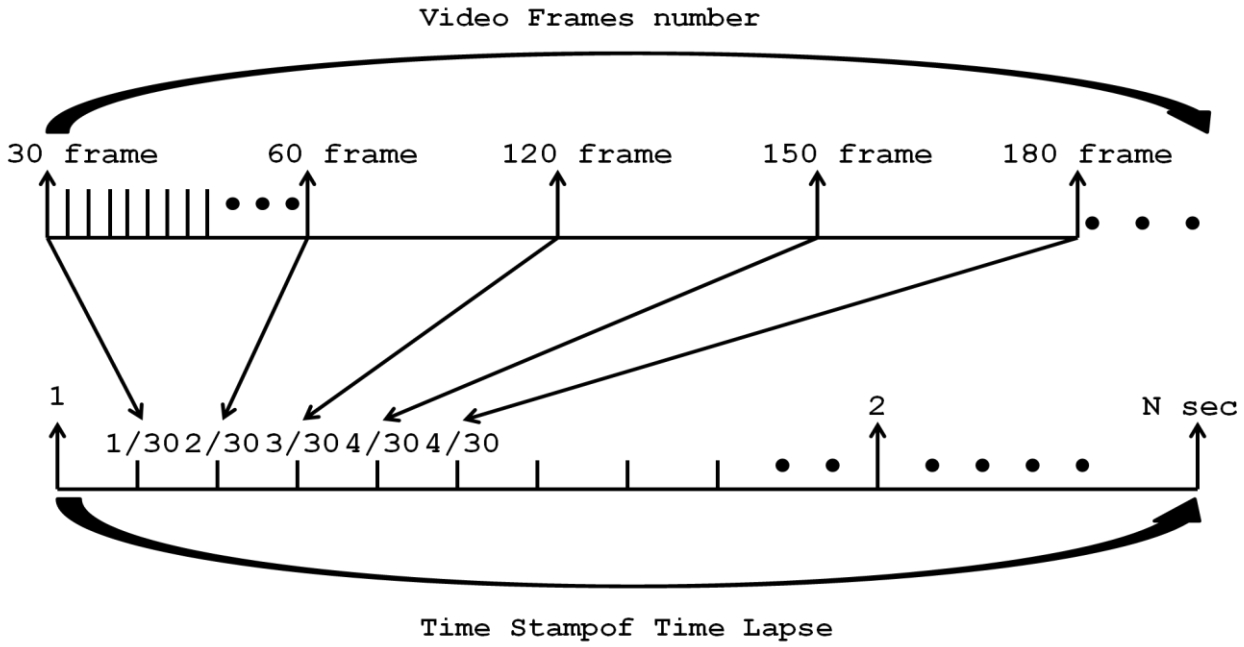
Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression video buffer into virtual camera’s share memory.

If users want to get Time Lapse result, user can use SampleTime to controltimestamp of video. For instance the way as show in below.



Example

```
QCAP_SET_VIDEO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER( pCamera, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,

                                                    &nFrameBuffer, 1280 * 720 * 2, 0 );
```

13.8 QCAP_SET_VIDEO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_VIDEO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER_EX( PVOID     pCamera,
                                                               ULONG     nColorSpaceType,
                                                               ULONG     nWidth,
                                                               ULONG     nHeight,
                                                               BYTE *    pFrameBuffer,
                                                               ULONG     nFrameBufferLen,
                                                               ULONG     nCropX,
                                                               ULONG     nCropY,
                                                               ULONG     nCropW,
                                                               ULONG     nCropH,
                                                               double    dSampleTime = 0.0 )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32
nWidth	**IN**, Specify video width
nHeight	**IN**, Specify video height
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

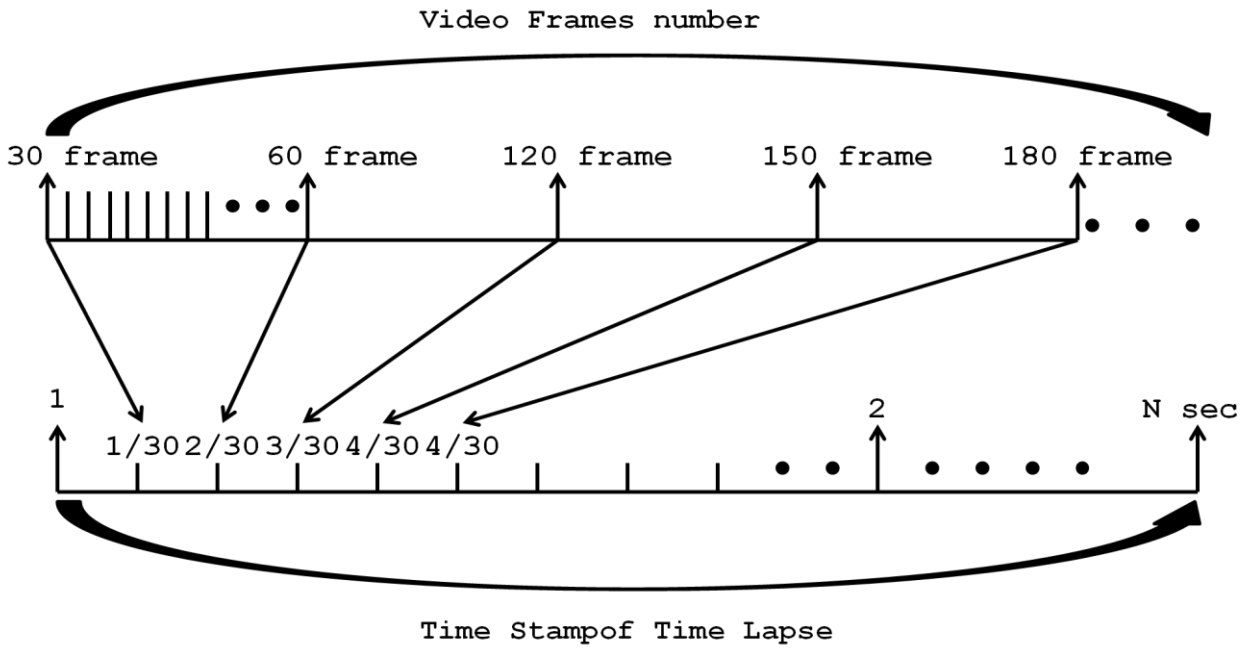
Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression video buffer and set crop parameters into virtual camera’s share memory.

If users want to get Time Lapse result, user can useSampleTime to controltimestamp of video.

For instance the way as show in below.



Example

```
QCAP_SET_VIDEO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER_EX( pCamera, QCAP_COLORSPACE_TYEP_YUY2, 1280, 720,
                                                       &nFrameBuffer, 1280 * 720 * 2, 0, 0, 720, 480, 0 );
```

13.9 QCAP_SET_AUDIO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER( PVOID     pCamera,
                                                            BYTE *    pFrameBuffer,
                                                            ULONG     nFrameBufferLen,
                                                            double    dSampleTime = 0.0)
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to push un-compression audio buffer into virtual camera’s share memory.

Note

The SampleTime of audio must same with SampleTime of video.

Example

```
QCAP_SET_AUDIO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER( pCamera, &nFrameBuffer, 1280 * 720 * 2, 0 );
```

13.10 QCAP_SET_AUDIO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER_EX

```
QRESULT QCAP_SET_AUDIO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER_EX( PVOID      pCamera,
                                                               ULONG      nChannels,
                                                               ULONG      nBitsPerSample,
                                                               ULONG      nSampleFrequency,
                                                               BYTE *     pFrameBuffer,
                                                               ULONG      nFrameBufferLen,
                                                               double     dSampleTime = 0.0 )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
nChannels	**IN**, Specify encoder channels for the i-th audio stream session
nBitsPerSample	**IN**, Specify encoder bits per sample for the i-th audio stream session
nSampleFrequency	**IN**, Specify encoder sample frequency for the i-th audio stream session
pFrameBuffer	**IN**, Specify the input source buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
dSampleTime	**IN**, Specify the timestamp of this frame. If the value is 0, QCAP will auto generate it by system clock time.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This is audio help function user can use this to resamle audio format.

For example, 2Ch 16Bit 48000HZ transform to 1Ch 8Bit 48000HZ.

This function is setting catcher input audio sample format, not resample audio format.

Example

```
QCAP_SET_AUDIO_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER_EX( pCamera, 2, 16, 48000,
                                                       pFrameBuffer, nFrameBufferLen );
```

13.11 QCAP_START_VIRTUAL_CAMERA

```
QRESULT QCAP_START_VIRTUAL_CAMERA( PVOID pCamera )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start virtual camera.

Note

User must set QCAP_SET_VIDEO_VIRTUAL_CAMERA_PROPERTY and QCAP_SET_AUDIO_VIRTUAL_CAMERA_PROPERTY first, when user use QCAP_START_VIRTUAL_CAMERA api.

Example

```
QCAP_START_VIRTUAL_CAMERA( pCamera );
```

13.12 QCAP_STOP_VIRTUAL_CAMERA

```
QRESULT QCAP_STOP_VIRTUAL_CAMERA( PVOID pCamera )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop virtual camera.

Example

```
QCAP_STOP_VIRTUAL_CAMERA( pCamera );
```

13.13 QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP

```
QRESULT QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP(   PVOID      pCamera,
                                            CHAR *     pszFilePathName,
                                            BOOL       bIsAsync = TRUE,
                                            ULONG      nMilliseconds = 0 )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in millisecond. User can use this function to snapshot a picture with BMP format on share records. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with BMP format, when virtual camera processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP( pCamera, "C:\\PICTURE1.BMP24" ); //PICTURE1.BMP24 ->PICTURE1.BMP
```

13.14 QCAP_SNAPSHOT_VIRTUAL_CAMERA_JPG

```
QRESULT QCAP_SNAPSHOT_VIRTUAL_CAMERA_JPG(   PVOID     pCamera,
                                            CHAR *    pszFilePathName,
                                            ULONG     nQuality,
                                            BOOL      bIsAsync = TRUE,
                                            ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pszFilePathName	**IN**, Specify the file name to store JPG data
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to snapshot a picture with JPG format, when virtual camera processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_VIRTUAL_CAMERA_JPG( pCamera, "C:\\PICTURE1.JPG", 100 );
```

13.15 QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP_EX

```
QRESULT QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP_EX(   PVOID     pCamera,
                                               CHAR *    pszFilePathName,
                                               ULONG     nCropX,
                                               ULONG     nCropY,
                                               ULONG     nCropW,
                                               ULONG     nCropH,
                                               ULONG     nDstW,
                                               ULONG     nDstH,
                                               BOOL      bIsAsync = TRUE,
                                               ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pszFilePathName	**IN**, Specify the file name to store BMP data "*.BMP24" -> 24bit BMP "*.BMP32" -> 32bit BMP "*.BMP" -> 32bit BMP
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to setcrop parameters of snapshot a picture with BMP format, when virtual camera processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

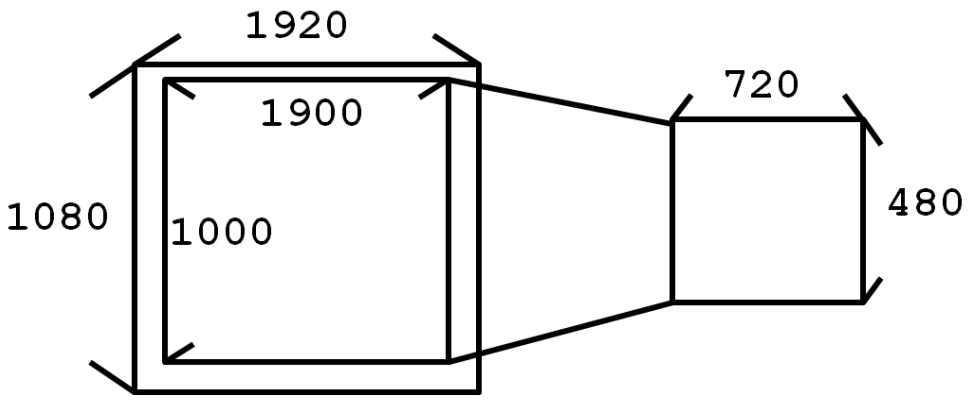
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP_EX( pCamera, "C:\\PICTURE1.BMP", 10, 40, 1900, 1000, 720, 480 );
```



13.16 QCAP_SNAPSHOT_VIRTUAL_CAMERA_JPG_EX

```
QRESULT QCAP_SNAPSHOT_VIRTUAL_CAMERA_JPG_EX(   PVOID     pCamera,
                                               CHAR *    pszFilePathName,
                                               ULONG     nCropX,
                                               ULONG     nCropY,
                                               ULONG     nCropW,
                                               ULONG     nCropH,
                                               ULONG     nDstW,
                                               ULONG     nDstH,
                                               ULONG     nQuality,
                                               BOOL      bIsAsync = TRUE,
                                               ULONG     nMilliseconds = 0 )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pszFilePathName	**IN**, Specify the file name to store JPG data
nCropX	**IN**, Specify the x-coordinate of the crop
nCropY	**IN**, Specify the y-coordinate of the crop
nCropW	**IN**, Specify the horizontal width of crop
nCropH	**IN**, Specify the vertical height of crop
nDstW	**IN**, Specify the horizontal width of current recording down scale
nDstH	**IN**, Specify the vertical height of current recording down scale
nQuality	**IN**, Specify the quality of JPG file, from 0~100
bIsAsync	**IN**, Synchronized snapshot flag
nMilliseconds	**IN**, Time-out interval, in milliseconds. The parameter is used at sync mode only. If nMilliseconds is zero, the function returns immediately. If it is INFINITE, the function's time-out interval never elapses.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to setcrop parameters of snapshot a picture with JPG format, when virtual camera processing.

At async mode, QCAP will use one degressive counter to snapshot image.

For example, if you call the function 100 times, you will generate 100 different and continuous images in sequence.

If the bIsAsync is FALSE, and the nMillisecond is INFINITE, the function will be blocked and returned until the snapshot is completed.

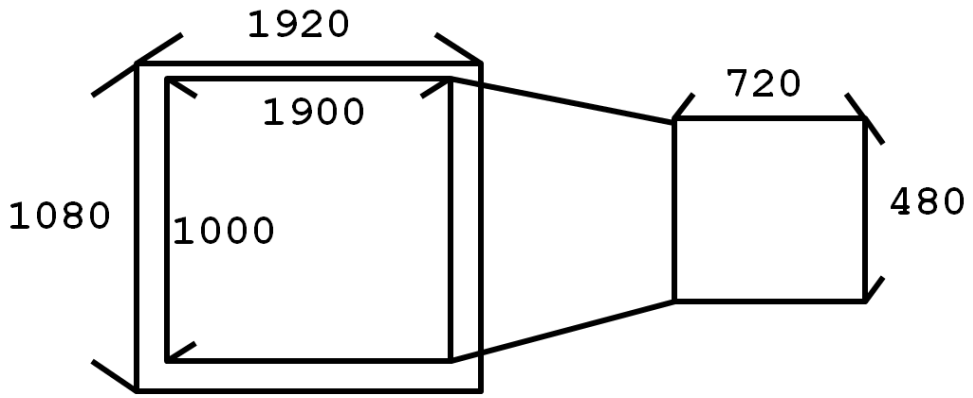
If the bIsAsync is FALSE, and the nMillisecond is 0, the function will return immediately and generate image file when next frame is coming.

Note

For .Net developer, the INFINITE is defined as 0xFFFFFFFF.

Example

```
QCAP_SNAPSHOT_VIRTUAL_CAMERA_JPG_EX( pCamera, "C:\\PICTURE1.JPG", 10, 40, 1900, 1000, 720, 480,100 );
```



13.17 QCAP_GET_OSD_VIRTUAL_CAMERA_TEXT_BOUNDARY

```
QRESULT QCAP_GET_OSD_VIRTUAL_CAMERA_TEXT_BOUNDARY(    PVOID     pCamera,
                                                      UINT      iOsdNum,
                                                      CHAR   *  pszString,
                                                      CHAR   *  pszFontFamilyName,
                                                      ULONG     nFontStyle,
                                                      ULONG     nFontSize,
                                                      ULONG  *  pBoundaryWidth,
                                                      ULONG  *  pBoundaryHeight )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output
nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are QCAP_FONT_STYLE_REGULAR = 0x00000000, QCAP_FONT_STYLE_BOLD = 0x00000001, QCAP_FONT_STYLE_ITALIC = 0x00000002, QCAP_FONT_STYLE_BOLDITALIC = 0x00000003, QCAP_FONT_STYLE_UNDERLINE = 0x00000004, QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,
nFontSize	**IN**, Specify the font size used to display the text of OSD output
pBoundaryWidth	**OUT**, Pointer to the storage of boundary width
pBoundaryHeight	**OUT**, Pointer to the storage of boundary height

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get size of OSD string, when the users use the virtual camera function.

For example, user can use boundary information to set OSD string location on display window.

Example

```
QCAP_GET_OSD_VIRTUAL_CAMERA_TEXT_BOUNDARY( pCamera, 0, “CH01”, “Arial”,
                                           QCAP_FONT_STYLE_BOLD, 12,
                                           pBoundaryWidth, pBoundaryHeight );
```

13.18 QCAP_SET_OSD_VIRTUAL_CAMERA_TEXT

```
QRESULT QCAP_SET_OSD_VIRTUAL_CAMERA_TEXT(   PVOID       pCamera,
                                            UINT        iOsdNum,
                                            INT         x,
                                            INT         y,
                                            INT         width,
                                            INT         height,
                                            CHAR*       pszString,
                                            CHAR *      pszFontFamilyName,
                                            ULONG       nFontStyle,
                                            ULONG       nFontSize,
                                            DWORD       dwFontColor,
                                            DWORD       dwBackgroundColor,
                                            ULONG       nTransparent,
                                            INT         nTextStartPosX = 0,
                                            INT         nTextStartPosY = 0,
                                            ULONG       nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63

x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output, if width is -1, QCAP will auto calculate width.
height	**IN**, Specify the vertical height of OSD output, if height is -1, QCAP will auto calculate height.
pszString	**IN**, Specify to display the text of OSD output
pszFontFamilyName	**IN**, Specify the font name used to display the text of OSD output

nFontStyle	**IN**, Specify the font style used to display the text of OSD output Available values are
	`QCAP_FONT_STYLE_REGULAR = 0x00000000,`
	`QCAP_FONT_STYLE_BOLD = 0x00000001,`
	`QCAP_FONT_STYLE_ITALIC = 0x00000002,`
	`QCAP_FONT_STYLE_BOLDITALIC = 0x00000003,`
	`QCAP_FONT_STYLE_UNDERLINE = 0x00000004,`
	`QCAP_FONT_STYLE_STRIKEOUT = 0x00000008,`

nFontSize	**IN**, Specify the font size used to display the text of OSD output
dwFontColor	**IN**, Specify the font color used to display the text of OSD output
dwBackgroundColor	**IN**, Specify the background color used to display the text of OSD output
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nTextStartPosX	**IN**, Specify the text scrolling start position X of the upper-left corner of OSD text
nTextStartPosY	**IN**, Specify the text scrolling start position Y of the upper-left corner of OSD text

nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

We also support to create a text field objects used for onscreen display of information when the users use the virtual camera function.

If you want to get a text field object auto calculated actual width and height value;

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_VIRTUAL_CAMERA_TEXT( pCamera, 0, 0, 0, -1, -1, “CH01”, “Arial”, QCAP_FONT_STYLE_BOLD, 12,
                                  0xFF000000, 0xFFFFFFFF, 128, 0, 0, QCAP_SEQUENCE_STYLE_FOREMOST );
```

13.19 QCAP_SET_OSD_VIRTUAL_CAMERA_PICTURE

```
QRESULT QCAP_SET_OSD_VIRTUAL_CAMERA_PICTURE(  PVOID      pCamera,
                                              UINT       iOsdNum,
                                              INT        x,
                                              INT        y,
                                              INT        width,
                                              INT        height,
                                              CHAR *     pszFilePathName,
                                              ULONG      nTransparent,
                                              ULONG      nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63
x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**, Specify the horizontal width of OSD output If width is -1, QCAP will use original picture width as default.
height	**IN**, Specify the vertical height of OSD output. If height is -1, QCAP will use original picture height as default.
pszFilePathName	**IN**, Specify the file name to display image OSD output, support a file extension of "BMP" as 24 or 32-bit,“JPG” and “PNG”
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000 QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

We also support to createa static image objects used for onscreen display of information when the users use the virtual camera function.

If you want to get atext field object auto calculated actual width and height value;

you should set height and width parameter as -1.

Beside, you also can adjust the OSD sequence style to video-streams recording the OSD output.

Example

```
QCAP_SET_OSD_VIRTUAL_CAMERA_PICTURE(pCamera, 0, 0, 0,-1,-1,“C:\\SAMPLE.PNG”, 128,
                                    QCAP_SEQUENCE_STYLE_FOREMOST );
```

13.20 QCAP_SET_OSD_VIRTUAL_CAMERA_BUFFER

```
QRESULT QCAP_SET_OSD_VIRTUAL_CAMERA_BUFFER( PVOID     pCamera,
                                            UINT      iOsdNum,
                                            INT       x,
                                            INT       y,
                                            INT       width,
                                            INT       height,
                                            ULONG     nColorSpaceType,
                                            BYTE *    pFrameBuffer,
                                            ULONG     nFrameWidth,
                                            ULONG     nFrameHeight,
                                            ULONG     nFramePitch,
                                            ULONG     nTransparent,
                                            DWORD     dwKeyColor = 0xFFFFFFFF,
                                            ULONG     nKeyColorThreshold = 25,
                                            ULONG     nSequenceStyle = QCAP_SEQUENCE_STYLE_FOREMOST )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
iOsdNum	**IN**, Specify the i-th layer number of OSD output OSD NUM: 0 ~ 63

x	**IN**, Specify the x-coordinate of the upper-left corner of OSD output
y	**IN**, Specify the y-coordinate of the upper-left corner of OSD output
width	**IN**,Specify the horizontal width of OSD output
height	**IN**,Specify the vertical height of OSD output
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameWidth	**IN**, Specify the horizontal width of frame contained in a buffer
nFrameHeight	**IN**, Specify the vertical height of frame contained in a buffer
nFramePitch	**IN**, Specify the distance in bytes between the start of one scan line to the next. If it is 0, we will auto calculate it by width and color space format.
nTransparent	**IN**, Specify the transparent ratio of whole OSD output, the range of 0 to 255
dwKeyColor	**IN**, Specify the key color of OSD, default is 0xFFFFFFFF
nKeyColorThreshold	**IN**, Specify the threshold of key color, the range of 0 to 128
nSequenceStyle	**IN**, Specify OSD sequence style type. Available values are QCAP_SEQUENCE_STYLE_FOREMOST = 0x00000000

	QCAP_SEQUENCE_STYLE_BEFORE_ENCODE = 0x00000001 QCAP_SEQUENCE_STYLE_AFTERMOST = 0x00000002

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

We also support to createa frame buffer objects used for onscreen display of information when the users use the virtual camera function.

The function’s purpose is used to point directly to a valid frame buffer whose entire pixel data,

which is stored in available color space, and display it.

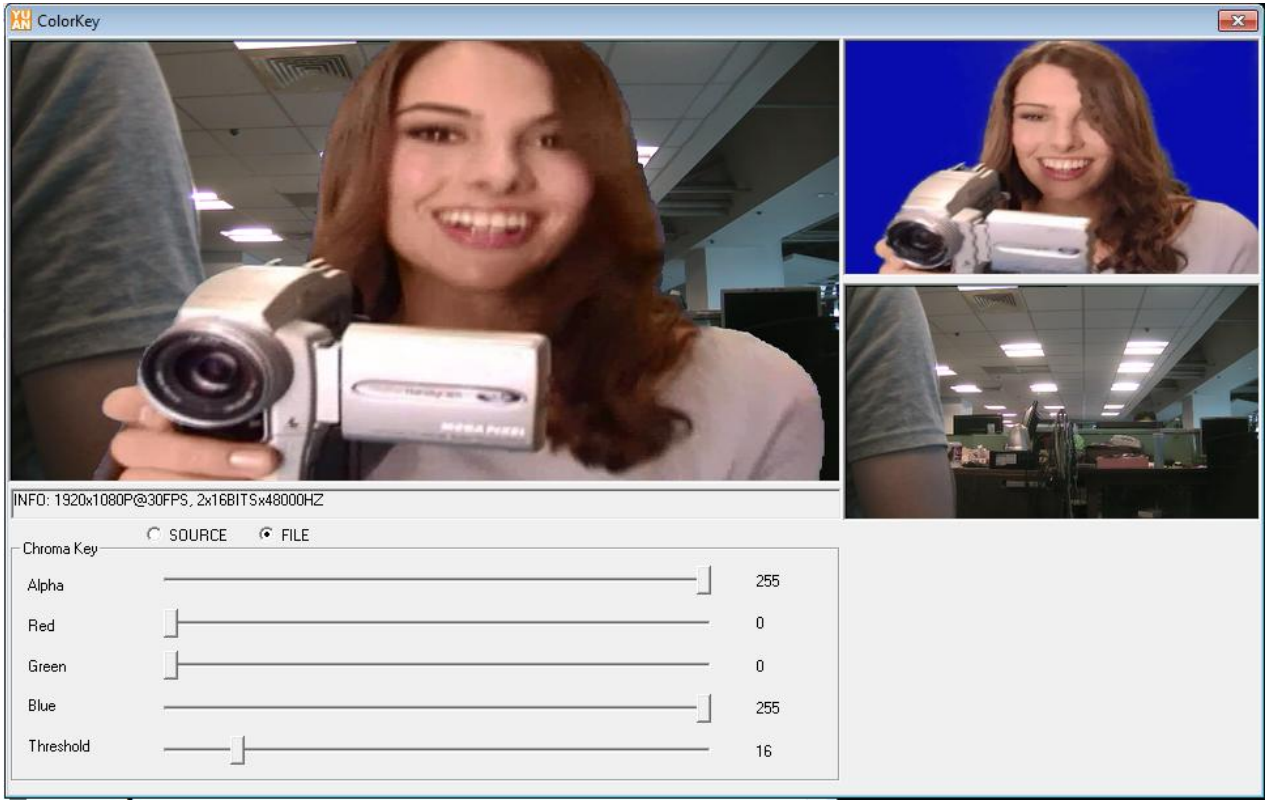
Beside, you also can adjust the width and height parameters to scale the OSD output

And OSD sequence style to video-streams recording the OSD output.

Note

User can use this dwKeyColor parament to select colorkey mode Green&Blue mode and RGB mode.

For instance the Blue mode way as show in below



Example

```
QCAP_SET_OSD_VIRTUAL_CAMERA_BUFFER( pCamera, 0, 0, 0, 1920, 1080,
                                    QCAP_COLORSPACE_TYEP_YUY2, pFrameBuffer,
                                    800, 600, 800*2, 128, 0xFFFFFFFF, QCAP_SEQUENCE_STYLE_FOREMOST );
```

13.21 QCAP_SET_AUDIO_MX_VIRTUAL_CAMERA_MIXING_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_MX_VIRTUAL_CAMERA_MIXING_UNCOMPRESSION_BUFFER(    PVOID     pCamera,
                                                                         BYTE *    pFrameBuffer,
                                                                         ULONG     nFrameBufferLen )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pFrameBuffer	**IN**, Specify the audio input source buffer
nFrameBufferLen	**IN**, Specify the audio input source buffer’s size

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to mixing audio virtual uncompression buffer on audio preview callback.

User must set `QCAP_SET_AUDIO_MX_VIRTUAL_CAMERA_MIXING_UNCOMPRESSION_BUFFER` to mixer audio record first then call

`QCAP_SET_AUDIO_MX_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER`.

Example

```
if( nChannelNum == 0 ){
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pCamera,
                                                                    pFrameBuffer,
                                                                    nFrameBufferLen );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pCamera );
}
if( nChannelNum == 1 ){
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pCamera,
                                                                    pFrameBuffer,
                                                                    nFrameBufferLen );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pCamera );
}
```

13.22 QCAP_SET_AUDIO_MX_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER

```
QRESULT QCAP_SET_AUDIO_MX_VIRTUAL_CAMERA_UNCOMPRESSION_BUFFER (    PVOID     pCamera,
                                                                   double    dSampleTime = 0.0 )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
dSampleTime	**IN**, Specify the timestamp of this frame.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to set audio virtual uncompression buffer on audio preview callback.

Example

```
if( nChannelNum == 0 ){
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pCamera,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pCamera );
}
if( nChannelNum == 1 ){
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_MIXING_UNCOMPRESSION_BUFFER( pCamera,
                                                                   pFrameBuffer,
                                                                   nFrameBufferLen );
    QCAP_SET_AUDIO_MX_BROADCAST_SERVER_UNCOMPRESSION_BUFFER( pCamera );
}
```

14 Callback Function API

14.1 QCAP_REGISTER_NO_SIGNAL_DETECTED_CALLBACK

```
QRETURN QCAP_REGISTER_NO_SIGNAL_DETECTED_CALLBACK( PVOID                              pDevice,
                                                   PF_NO_SIGNAL_DETECTED_CALLBACK     pCB,
                                                   PVOID                              pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_NO_SIGNAL_DETECTED_CALLBACK)( PVOID pDevice /*IN*/,
                                                                ULONG nVideoInput /*IN*/,
                                                                ULONG nAudioInput /*IN*/,
                                                                PVOID pUserData /*IN*/ );
```

User can use this function to register above callback function.

When capture device cannot detect the signal from current input, the callback function will be called.

User can implement its own function to handle these information.

Note

This function is started after `QCAP_CREATE` right now `(` not after `QCAP_RUN )`.

Example

```
QCAP_REGISTER_NO_SIGNAL_DETECTED_CALLBACK( pDevice, pCB, this );
```

14.2 QCAP_REGISTER_SIGNAL_REMOVED_CALLBACK

```
QRETURN QCAP_REGISTER_SIGNAL_REMOVED_CALLBACK(  PVOID                          pDevice,
                                                PF_SIGNAL_REMOVED_CALLBACK     pCB,
                                                PVOID                          pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_SIGNAL_REMOVED_CALLBACK)( PVOID pDevice /*IN*/,
                                                            ULONG nVideoInput /*IN*/,
                                                            ULONG nAudioInput /*IN*/,
                                                            PVOID pUserData /*IN*/ );
```

User can use this function to register above callback function.

When the input signal is loss, the callback function will be called.

The callback function is different from `NO_SIGNAL_DETECTED_CALLBACK`, and it is only called when input is loss from exist state.

User can implement its own functions to handle these information.

Note

This function is started after `QCAP_CREATE` right now `(` not after `QCAP_RUN )`.

Example

```
QCAP_REGISTER_SIGNAL_REMOVED_CALLBACK( pDevice, pCB, this );
```

14.3 QCAP_REGISTER_FORMAT_CHANGED_CALLBACK

```
QRETURN QCAP_REGISTER_FORMAT_CHANGED_CALLBACK(  PVOID                          pDevice,
                                                PF_FORMAT_CHANGED_CALLBACK     pCB,
                                                PVOID                          pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_FORMAT_CHANGED_CALLBACK)( PVOID  pDevice /*IN*/,
                                                            ULONG  nVideoInput /*IN*/,
                                                            ULONG  nAudioInput /*IN*/,
                                                            ULONG  nVideoWidth /*IN*/,
                                                            ULONG  nVideoHeight /*IN*/,
                                                            BOOL   bVideoIsInterleaved /*IN*/,
                                                            double dVideoFrameRate /*IN*/,
                                                            ULONG  nAudioChannels /*IN*/,
                                                            ULONG  nAudioBitsPerSample /*IN*/,
                                                            ULONG  nAudioSampleFrequency /*IN*/,
                                                            PVOID  pUserData /*IN*/ );
```

User can use this function to register above callback function.

When the input signal is detected or input format is changed, the callback function will be called.

The callback function can provide video/audio information, such as video input, audio input, resolution, video interleaved,

frame per second, audio channel number, audio sample width and audio sampling frequency.

User can implement its own function to handle these information.

Note

This function is started after `QCAP_CREATE` right now `(` not after `QCAP_RUN )`.

Example

```
QCAP_REGISTER_FORMAT_CHANGED_CALLBACK( pDevice, pCB, this );
```

14.4 QCAP_REGISTER_SNAPSHOT_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_SNAPSHOT_DONE_CALLBACK(  PVOID                          pDevice,
                                               PF_SNAPSHOT_DONE_CALLBACK      pCB,
                                               PVOID                          pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_SNAPSHOT_DONE_CALLBACK)(  PVOID pDevice /*IN*/,
                                                            CHAR *  pszFilePathName /*IN*/,
                                                            PVOID   pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the snapshot file of `QCAP_SNAPSHOT` is complete.

The developer, who uses async snapshot, can use the function to know when snapshot file is ready.

Example

```
QCAP_REGISTER_SNAPSHOT_DONE_CALLBACK( pDevice, pCB, this );
```

14.5 QCAP_REGISTER_RECORD_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_RECORD_DONE_CALLBACK( PVOID                       pDevice,
                                            UINT                        iRecNum,
                                            PF_RECORD_DONE_CALLBACK     pCB,
                                            PVOID                       pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_RECORD_DONE_CALLBACK)( PVOID   pDevice /*IN*/,
                                                         UINT    iRecNum /*IN*/,
                                                         CHAR *  pszFilePathName /*IN*/,
                                                         PVOID   pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the video record file of `QCAP_START_RECORD`is complete.

And developer, who uses async record, can use the function to know when video record file is ready.

Note

Some recording format, such as MP4 the file closing need cost several seconds to finish, developer who use asyc record stop flag, can use this callback to know when file is ready.

Example

```
QCAP_REGISTER_RECORD_DONE_CALLBACK( pDevice, pCB, this );
```

14.6 QCAP_REGISTER_SHARE_RECORD_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_SHARE_RECORD_DONE_CALLBACK( UINT                                iRecNum,
                                                  PF_SHARE_RECORD_DONE_CALLBACK       pCB,
                                                  PVOID                               pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_SHARE_RECORD_DONE_CALLBACK)( UINT    iRecNum /*IN*/,
                                                               CHAR *  pszFilePathName /*IN*/,
                                                               PVOID   pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the video record file of `QCAP_START_SHARE_RECORD` is complete.

And developer, who uses async record, can use the function to know when video record file is ready.

Note

Some recording format, such as MP4 the file closing need cost several seconds to finish, developer who use asyc record stop flag, can use this callback to know when file is ready.

Example

```
QCAP_REGISTER_SHARE_RECORD_DONE_CALLBACK( pDevice, pCB, this );
```

14.7 QCAP_REGISTER_SHARE_RECORD_SNAPSHOT_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_SHARE_RECORD_SNAPSHOT_DONE_CALLBACK(
                                                    UINT                                        iRecNum,
                                                    PF_SHARE_RECORD_SNAPSHOT_DONE_CALLBACK      pCB,
                                                    PVOID                                       pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_SHARE_RECORD_SNAPSHOT_DONE_CALLBACK)(
                                                        UINT    iRecNum /*IN*/,
                                                        CHAR *  pszFilePathName /*IN*/,
                                                        PVOID   pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the snapshot file of `QCAP_SNAPSHOT_SHARE_RECORD_BMP(JPG)` is complete.

And developer, who uses async, can use the function to know when snapshot file is ready.

Example

```
QCAP_REGISTER_SHARE_RECORD_SNAPSHOT_DONE_CALLBACK( iRecNum, pCB, this );
```

14.8 QCAP_REGISTER_BROADCAST_SERVER_GET_CUSTOM_PROPERTY_CALLBACK

```
QRETURN QCAP_REGISTER_BROADCAST_SERVER_GET_CUSTOM_PROPERTY_CALLBACK(
                                                    PVOID                                           pServer,
                                                    PF_BROADCAST_SERVER_GET_CUSTOM_PROPERTY_CALLBACK  pCB,
                                                    PVOID                                           pUserData )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_BROADCAST_SERVER_GET_CUSTOM_PROPERTY_CALLBACK)(
                                                        PVOID     pServer /*IN*/,
                                                        UINT      iSessionNum /*IN*/,
                                                        CHAR *    pszProperty /*IN*/,
                                                        CHAR *    pszValue /*OUT*/,
                                                        PVOID     pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, When get custom property command from client, the callback function will be called.

Example

```
QCAP_REGISTER_BROADCAST_SERVER_GET_CUSTOM_PROPERTY_CALLBACK( pServer, pCB, pUserData );
```

14.9 QCAP_REGISTER_BROADCAST_SERVER_SET_CUSTOM_PROPERTY_CALLBACK

```
QRETURN QCAP_REGISTER_BROADCAST_SERVER_SET_CUSTOM_PROPERTY_CALLBACK(
                                                    PVOID                                           pServer,
                                                    PF_BROADCAST_SERVER_SET_CUSTOM_PROPERTY_CALLBACK  pCB,
                                                    PVOID                                           pUserData )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_BROADCAST_SERVER_SET_CUSTOM_PROPERTY_CALLBACK)(
                                                        PVOID     pServer /*IN*/,
                                                        UINT      iSessionNum /*IN*/,
                                                        CHAR *    pszProperty /*IN*/,
                                                        CHAR *    pszValue /*IN*/,
                                                        PVOID     pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, When set custom property command from client, the callback function will be called.

Example

```
QCAP_REGISTER_BROADCAST_SERVER_SET_CUSTOM_PROPERTY_CALLBACK( pServer, pCB, pUserData );
```

14.10 QCAP_REGISTER_BROADCAST_SERVER_SNAPSHOT_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_BROADCAST_SERVER_SNAPSHOT_DONE_CALLBACK(
                                                    PVOID                                       pServer,
                                                    UINT                                        iSessionNum,
                                                    PF_BORADCAST_SERVER_SNAPSHOT_DONE_CALLBACK  pCB,
                                                    PVOID                                       pUserData )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**,Specify the number of channel session connections, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_BROADCAST_SERVER_SNAPSHOT_DONE_CALLBACK)(
                                                                PVOID    pServer /*IN*/,
                                                                UINT     iSessionNum /*IN*/,
                                                                CHAR *   pszFilePathName /*IN*/,
                                                                PVOID    pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the snpashot file of `QCAP_SNAPSHOT_BROADCAST_SERVER_BMP(JPG)` is complete.

And developer, who uses async, can use the function to know when snapshot file is ready.

Example

```
QCAP_REGISTER_BROADCAST_SERVER_SNAPSHOT_DONE_CALLBACK( pServer, 0, pCB, pUserData );
```

14.11 QCAP_REGISTER_BROADCAST_CLIENT_RECORD_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_BROADCAST_CLIENT_RECORD_DONE_CALLBACK(
                                                    PVOID                                   pClient,
                                                    PF_BROADCAST_CLIENT_RECORD_DONE_CALLBACK    pCB,
                                                    PVOID                                   pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_BROADCAST_CLIENT_RECORD_DONE_CALLBACK )(
                                                                    PVOID   pClient /*IN*/,
                                                                    CHAR *  pszFilePathName /*IN*/,
                                                                    PVOID   pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the video record file of `QCAP_START_BROADCAST_CLIENT_RECORD` is complete.

And developer, who uses async record, can use the function to know when video record file is ready.

Note

Some recording format, such as MP4 the file closing need cost several seconds to finish, developer who use asyc record stop flag, can use this callback to know when file is ready.

Example

```
QCAP_REGISTER_BROADCAST_CLIENT_RECORD_DONE_CALLBACK( pClient, pCB, this );
```

14.12 QCAP_REGISTER_BROADCAST_CLIENT_CONNECTED_CALLBACK

```
QRETURN QCAP_REGISTER_BROADCAST_CLIENT_CONNECTED_CALLBACK(
                                                    PVOID                                   pClient,
                                                    PF_BROADCAST_CLIENT_CONNECTED_CALLBACK  pCB,
                                                    PVOID                                   pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_BROADCAST_CLIENT_CONNECTED_CALLBACK)(
                                                                      PVOID   pClient /*IN*/,
                                                                      ULONG   nVideoWidth /*IN*/,
                                                                      ULONG   nVideoHeight /*IN*/,
                                                                      BOOL    bVideoIsInterleaved /*IN*/,
                                                                      double  dVideoFrameRate /*IN*/,
                                                                      ULONG   nAudioChannels /*IN*/,
                                                                      ULONG   nAudioBitsPerSample /*IN*/,
                                                                      ULONG   nAudioSampleFrequency/*IN*/,
                                                                      PVOID   pUserData /*IN*/ );
```

User can use this function to register above callback function.

User can use this function to get format from broadcast client, when the broad cleint connected was successful, the callback function will be called.The callback function can get video/audio information, such as video width, video height, video interleaved, frame per second, audio channel number, audio sample width and audio sampling frequency. User can implement its own function to handle these information.

Example

```
QCAP_REGISTER_BROADCAST_CLIENT_CONNECTED_CALLBACK( pClient, pCB, this );
```

14.13 QCAP_REGISTER_BROADCAST_CLIENT_SNAPSHOT_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_BROADCAST_CLIENT_SNAPSHOT_DONE_CALLBACK(
                                                    PVOID                                   pClient,
                                                    PF_BROADCAST_CLIENT_SNAPSHOT_DONE_CALLBACK pCB,
                                                    PVOID                                   pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_BROADCAST_CLIENT_SNAPSHOT_DONE_CALLBACK )(
                                                                    PVOID   pClient /*IN*/,
                                                                    CHAR *  pszFilePathName /*IN*/,
                                                                    PVOID   pUserDate /*IN*/);
```

User can use this function to register above callback function.

The callback function will be called, when the snapshot file of `QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP(JPG)` is complete.

And developer, who uses async, can use the function to know when snapshot file is ready.

Example

```
QCAP_REGISTER_BROADCAST_CLIENT_SNAPSHOT_DONE_CALLBACK( pClient, pCB, this );
```

14.14 QCAP_REGISTER_VIRTUAL_CAMERA_SNAPSHOT_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_VIRTUAL_CAMERA_SNAPSHOT_DONE_CALLBACK(
                                                    PVOID                                   pCamera,
                                                    PF_VIRTUAL_CAMERA_SNAPSHOT_DONE_CALLBACKpCB,
                                                    PVOID                                   pUserData )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_VIRTUAL_CAMERA_SNAPSHOT_DONE_CALLBACK)(
                                                                    PVOID   pCamera /*IN*/,
                                                                    CHAR *  pszFilePathName /*IN*/,
                                                                    PVOID   pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the snapshot file of `QCAP_SNAPSHOT_VIRTUAL_CAMERA_BMP(JPG)` is complete.

And developer, who uses async, can use the function to know when snapshot file is ready.

Example

```
QCAP_REGISTER_VIRTUAL_CAMERA_SNAPSHOT_DONE_CALLBACK( pCamera, pCB, this );
```

14.15 QCAP_REGISTER_FILE_SNAPSHOT_DONE_CALLBACK

```
QRETURN QCAP_REGISTER_FILE_SNAPSHOT_DONE_CALLBACK(
                                                  PVOID                              pFile,
                                                  PF_FILE_SNAPSHOT_DONE_CALLBACK     pCB,
                                                  PVOID                              pUserData )
```

Parameter

Name	Description
pFile	**IN**, Handle of the filer object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT * PF_VIRTUAL_CAMERA_SNAPSHOT_DONE_CALLBACK)(
                                                                    PVOID   pFile/*IN*/,
                                                                    CHAR *  pszFilePathName /*IN*/,
                                                                    PVOID   pUserDate /*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the snapshot file of `QCAP_SNAPSHOT_FILE_BMP(JPG)` is complete.

And developer, who uses async, can use the function to know when snapshot file is ready.

Example

```
QCAP_REGISTER_FILE_SNAPSHOT_DONE_CALLBACK( pFile, pCB, this );
```

14.16 QCAP_REGISTER_VIDEO_PREVIEW_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_PREVIEW_CALLBACK(  PVOID                        pDevice,
                                               PF_VIDEO_PREVIEW_CALLBACK    pCB,
                                               PVOID                        pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_PREVIEW_CALLBACK)( PVOID    pDevice /*IN*/,
                                                          double   dSampleTime /*IN*/,
                                                          BYTE *   pFrameBuffer /*IN*/,
                                                          ULONG    nFrameBufferLen /*IN*/,
                                                          PVOID    pUserData /*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide video preview data (YUY2 or YV12) for developer to use.

Developer can directly modify the pixel data whith in the callback function, such as alpha blending.

Then the modifcation will directly geneate one filter effect in recording file and display.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and will not be displayed and not sent into recording engine on the window.

#2 User can use callback return `QCAP_RT_SKIP_RECORD_XXX` to `QCAP` for set skip frame buffer and select which channel will be skip frame buffer by himself, specify callback return type. Available values are

```
QCAP_RT_SKIP_RECORD_NUM_00   = 0x00010002
QCAP_RT_SKIP_RECORD_NUM_01   = 0x00020002
QCAP_RT_SKIP_RECORD_NUM_02   = 0x00040002
QCAP_RT_SKIP_RECORD_NUM_03   = 0x00080002
```

#3 If callback return `QCAP_RT_SKIP_DISPLAY (CLONE_DISPLAY)` to `QCAP`, the video preview data (YUY2 or YV12) and clone video data will not bedisplayed on the window.

```
QCAP_RT_SKIP_DISPLAY         = 0x00000004
QCAP_RT_SKIP_CLONE_DISPLAY   = 0x00000008
```

#4 User can use callback return `QCAP_RT_RECORD_XXX` to `QCAP` for set rest frame bufferand select which channel will be rest frame buffer by himself, specify callback return type. Available values are

```
QCAP_RT_RESET_RECORD_NUM_00 = 0x01000010

QCAP_RT_RESET_RECORD_NUM_01 = 0x02000010

QCAP_RT_RESET_RECORD_NUM_02 = 0x04000010

QCAP_RT_RESET_RECORD_NUM_03 = 0x08000010
```

#5 dSampleTime is in second unit.

#6 The video format of device is

"DC1150 USB"is YUY2

"UB658G USB"is YV12

"CY3014 USB"is YUY2

"QP0204 USB"is YV12

"TW6802 PCI"is YUY2

"CX2581 PCI"is YUY2

"AH8400 PCI"is YUY2

"FH8735 PCI"is YV12

"TW5864 PCI"is YV12

"SA7160 PCI"is YUY2

"FH8735 PCI"is YV12

"TW2809 PCI"is YV12

"QP0203 PCI"is YV12

Example

```
QCAP_REGISTER_VIDEO_PREVIEW_CALLBACK( pDevice, pCB, pUserData );
```

14.17 QCAP_REGISTER_AUDIO_PREVIEW_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_PREVIEW_CALLBACK(  PVOID                        pDevice,
                                               PF_AUDIO_PREVIEW_CALLBACK    pCB,
                                               PVOID                        pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_PREVIEW_CALLBACK)( PVOID    pDevice /*IN*/,
                                                          double   dSampleTime /*IN*/,
                                                          BYTE *   pFrameBuffer /*IN*/,
                                                          ULONG    nFrameBufferLen /*IN*/,
                                                          PVOID    pUserData /*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide audio uncompressed data (PCM) for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 User can use callback return `QCAP_RT_SKIP_RECORD_XXX` to `QCAP` for set skip frame buffer, specify callback return type. Available values are

```
QCAP_RT_SKIP_RECORD_NUM_00   = 0x00010002
QCAP_RT_SKIP_RECORD_NUM_01   = 0x00020002
QCAP_RT_SKIP_RECORD_NUM_02   = 0x00040002
QCAP_RT_SKIP_RECORD_NUM_03   = 0x00080002
```

#3If callback return `QCAP_RT_SKIP_ XXX ( CLONE_DISPLAY )` to `QCAP`, the audio uncompressed data (PCM) and clone data will not be played on the window.

```
QCAP_RT_SKIP_DISPLAY         = 0x00000004
QCAP_RT_SKIP_CLONE_DISPLAY   = 0x00000008
```

#4 User can use callback return `QCAP_RT_RECORD_XXX` to `QCAP` for set rest frame buffer and select which channel will berest frame buffer by himself, specify callback return type. Available values are

```
QCAP_RT_RESET_RECORD_NUM_00 = 0x01000010
QCAP_RT_RESET_RECORD_NUM_01 = 0x02000010
QCAP_RT_RESET_RECORD_NUM_02 = 0x04000010
QCAP_RT_RESET_RECORD_NUM_03 = 0x08000010
```

#5 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_PREVIEW_CALLBACK( pDevice, pCB, pUserData );
```

14.18 QCAP_REGISTER_VIDEO_HARDWARE_ENCODER_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_HARDWARE_ENCODER_CALLBACK( PVOID                                pDevice,
                                                       ULONG                                iRecNum,
                                                       PF_VIDEO_HARDWARE_ENCODER_CALLBACK pCB,
                                                       PVOID                                pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index of hardware encoder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_HARDWARE_ENCODER_CALLBACK)( PVOID     pDevice /*IN*/,
                                                                   UINT      iRecNum /*IN*/,
                                                                   double    dSampleTime /*IN*/,
                                                                   BYTE *    pStreamBuffer /*IN*/,
                                                                   ULONG     nStreamBufferLen /*IN*/,
                                                                   BOOL      bIsKeyFrame /*IN*/,
                                                                   PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide video hardware compressed data (H.264) for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be send into the channel recorder.

#2 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_HARDWARE_ENCODER_CALLBACK( pDevice, pCB, pUserData );
```

14.19 QCAP_REGISTER_SNAPSHOT_STREAM_CALLBACK

```
QRETURN QCAP_REGISTER_SNAPSHOT_STREAM_CALLBACK(   PVOID                          pDevice,
                                                  PF_SNAPSHOT_STREAM_CALLBACK    pCB,
                                                  PVOID                          pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_SNAPSHOT_STREAM_CALLBACK)( PVOID     pDevice /*IN*/,
                                                            CHAR *    pszFilePathName /*IN*/,
                                                            BYTE *    pStreamBuffer /*IN*/,
                                                            ULONG     nStreamBufferLen /*IN*/,
                                                            PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the cheannel snapshot file of `QCAP_SNAPSHOT_BMP(JPG)` is complete.

And then, user can get snapshot file buffer immediately.

Example

```
QCAP_REGISTER_SNAPSHOT_STREAM_CALLBACK( pDevice, pCB, this );
```

14.20 QCAP_REGISTER_SHARE_RECORD_SNAPSHOT_STREAM_CALLBACK

```
QRETURN QCAP_REGISTER_SHARE_RECORD_SNAPSHOT_STREAM_CALLBACK(
                                                    UINT                                    iRecNum,
                                                    PF_SHARE_RECORD_SNAPSHOT_STREAM_CALLBACK pCB,
                                                    PVOID                                   pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_SHARE_RECORD_SNAPSHOT_STREAM_CALLBACK)(UINT    iRecNum /*IN*/,
                                                                        CHAR *  pszFilePathName /*IN*/,
                                                                        BYTE *  pStreamBuffer /*IN*/,
                                                                        ULONG   nStreamBufferLen /*IN*/,
                                                                        PVOID   pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the share record snapshot file of `QCAP_SNAPSHOT_SHARE_RECORD_BMP(JPG)` is complete.

And then, user can get snapshot file buffer immediately.

Example

```
QCAP_REGISTER_SHARE_RECORD_SNAPSHOT_STREAM_CALLBACK( 0, pCB, this );
```

14.21 QCAP_REGISTER_BROADCAST_SERVER_SNAPSHOT_STREAM_CALLBACK

```
QRETURN QCAP_REGISTER_BROADCAST_SERVER_SNAPSHOT_STREAM_CALLBACK(
                                        PVOID                                   pServer,
                                        UINT                                    iSessionNum,
                                        PF_BROADCAST_SERVER_SNAPSHOT_STREAM_CALLBACK   pCB,
                                        PVOID                                   pUserData )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_BROADCAST_SERVER_SNAPSHOT_STREAM_CALLBACK)(
                                                    PVOID     pServer /*IN*/,
                                                    UINT      iSessionNum /*IN*/,
                                                    CHAR *    pszFilePathName /*IN*/,
                                                    BYTE *    pStreamBuffer /*IN*/,
                                                    ULONG     nStreamBufferLen /*IN*/,
                                                    PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the broadcast server snapshot file of

`QCAP_SNAPSHOT_BROADCAST_SERVER_BMP(JPG)` is complete. And then, user can get snapshot file buffer immediately.

Example

```
QCAP_REGISTER_BROADCAST_SERVER_SNAPSHOT_STREAM_CALLBACK( pServer, 0, pCB, this );
```

14.22 QCAP_REGISTER_BROADCAST_CLIENT_SNAPSHOT_STREAM_CALLBACK

```
QRETURN QCAP_REGISTER_BROADCAST_CLIENT_SNAPSHOT_STREAM_CALLBACK(
                                        PVOID                                       pClient,
                                        PF_BROADCAST_CLIENT_SNAPSHOT_STREAM_CALLBACK pCB,
                                        PVOID                                       pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_BROADCAST_CLIENT_SNAPSHOT_STREAM_CALLBACK)(
                                                        PVOID     pClient /*IN*/,
                                                        CHAR *    pszFilePathName /*IN*/,
                                                        BYTE *    pStreamBuffer /*IN*/,
                                                        ULONG     nStreamBufferLen /*IN*/,
                                                        PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the broadcast client snapshot file of

`QCAP_SNAPSHOT_BROADCAST_CLIENT_BMP(JPG)` is complete. And then, user can get snapshot file buffer immediately.

Example

```
QCAP_REGISTER_BROADCAST_CLIENT_SNAPSHOT_STREAM_CALLBACK( pClient, pCB, this );
```

14.23 QCAP_REGISTER_VIRTUAL_CAMERA_SNAPSHOT_STREAM_CALLBACK

```
QRETURN QCAP_REGISTER_VIRTUAL_CAMERA_SNAPSHOT_STREAM_CALLBACK(
                                        PVOID                                       pCamera,
                                        PF_VIRTUAL_CAMERA_SNAPSHOT_STREAM_CALLBACK  pCB,
                                        PVOID                                       pUserData )
```

Parameter

Name	Description
pCamera	**IN**, Handle of the camera object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIRTUL_CAMERA_SNAPSHOT_STREAM_CALLBACK)(
                                                        PVOID     pCamera /*IN*/,
                                                        CHAR *    pszFilePathName /*IN*/,
                                                        BYTE *    pStreamBuffer /*IN*/,
                                                        ULONG     nStreamBufferLen /*IN*/,
                                                        PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the broadcast client snapshot file of

`QCAP_SNAPSHOT_VIRTUAL_SNAPSHOT_BMP(JPG)` is complete. And then, user can get snapshot file buffer immediately.

Example

```
QCAP_REGISTER_VIRTUAL_CAMERA_SNAPSHOT_STREAM_CALLBACK( pCamera, pCB, this );
```

14.24 QCAP_REGISTER_FILE_SNAPSHOT_STREAM_CALLBACK

```
QRETURN QCAP_REGISTER_FILE_SNAPSHOT_STREAM_CALLBACK( PVOID                                pFile,
                                                     PF_FILE_SNAPSHOT_STREAM_CALLBACK     pCB,
                                                     PVOID                                pUserData )
```

Parameter

Name	Description
pFile	**IN**, Handle of the file object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_FILE_SNAPSHOT_STREAM_CALLBACK)(
                                                        PVOID     pFile /*IN*/,
                                                        CHAR *    pszFilePathName /*IN*/,
                                                        BYTE *    pStreamBuffer /*IN*/,
                                                        ULONG     nStreamBufferLen /*IN*/,
                                                        PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will be called, when the broadcast client snapshot file of

`QCAP_SNAPSHOT_FILE_BMP(JPG)` is complete. And then, user can get snapshot file buffer immediately.

Example

```
QCAP_REGISTER_FILE_SNAPSHOT_STREAM_CALLBACK( pFile, pCB, this );
```

14.25 QCAP_REGISTER_VIDEO_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_RECORD_CALLBACK( PVOID                    pDevice,
                                             UINT                     iRecNum,
                                             PF_VIDEO_RECORD_CALLBACK pCB,
                                             PVOID                    pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_RECORD_CALLBACK)( PVOID     pDevice /*IN*/,
                                                         UINT      iRecNum /*IN*/,
                                                         double    dSampleTime /*IN*/,
                                                         BYTE *    pStreamBuffer /*IN*/,
                                                         ULONG     nStreamBufferLen /*IN*/,
                                                         BOOL      bIsKeyFrame /*IN*/,
                                                         PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide video recording's compressed data (H.264) for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be saved on the file.

#2 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_RECORD_CALLBACK( pDevice, pCB, pUserData );
```

14.26 QCAP_REGISTER_AUDIO_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_RECORD_CALLBACK( PVOID                    pDevice,
                                             UINT                     iRecNum,
                                             PF_AUDIO_RECORD_CALLBACK pCB,
                                             PVOID                    pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_RECORD_CALLBACK)( PVOID     pDevice /*IN*/,
                                                         UINT      iRecNum /*IN*/,
                                                         double    dSampleTime /*IN*/,
                                                         BYTE *    pStreamBuffer /*IN*/,
                                                         ULONG     nStreamBufferLen /*IN*/,
                                                         PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide audio recording's compressed data (PCM or AAC) for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be saved on the file.

#2 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_RECORD_CALLBACK( pDevice, pCB, pUserData );
```

14.27 QCAP_REGISTER_MEDIA_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_MEDIA_RECORD_CALLBACK( PVOID                    pDevice,
                                             UINT                     iRecNum,
                                             PF_MEDIA_RECORD_CALLBACK pCB,
                                             PVOID                    pUserData )
```

Parameter

Name	Description
pDevice	**IN**, Handle of the capture card object
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_MEDIA_RECORD_CALLBACK)( PVOID     pDevice /*IN*/,
                                                         UINT      iRecNum /*IN*/,
                                                         double    dSampleTime /*IN*/,
                                                         BYTE *    pStreamBuffer /*IN*/,
                                                         ULONG     nStreamBufferLen /*IN*/,
                                                         PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide media recording's data (video + audio) for developer to use.

Note

#1 this callback function provide data only support for TS and FLV format.

#2 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be saved on the file.

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_MEDIA_RECORD_CALLBACK( pDevice, pCB, pUserData );
```

14.28 QCAP_REGISTER_VIDEO_SHARE_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_SHARE_RECORD_CALLBACK( UINT                              iRecNum,
                                                   PF_VIDEO_RECORD_SHARE_CALLBACK    pCB,
                                                   PVOID                             pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_SHARE_RECORD_CALLBACK)( UINT      iRecNum /*IN*/,
                                                               double    dSampleTime /*IN*/,
                                                               BYTE *    pStreamBuffer /*IN*/,
                                                               ULONG     nStreamBufferLen /*IN*/,
                                                               BOOL      bIsKeyFrame /*IN*/,
                                                               PVOID     pUserData/*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide video recording's compressed data (H.264) for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be saved on the file.

#2 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_SHARE_RECORD_CALLBACK( pDevice, pCB, pUserData );
```

14.29 QCAP_REGISTER_AUDIO_SHARE_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_SHARE_RECORD_CALLBACK( UINT                              iRecNum,
                                                   PF_AUDIO_SHARE_RECORD_CALLBACK    pCB,
                                                   PVOID                             pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_SHARE_RECORD_CALLBACK)( UINT      iRecNum /*IN*/,
                                                               double    dSampleTime /*IN*/,
                                                               BYTE *    pStreamBuffer /*IN*/,
                                                               ULONG     nStreamBufferLen /*IN*/,
                                                               PVOID     pUserData/*IN*/);
```

User can use this function to register above callback function.

The callback function will provide audio recording's compressed data (PCM or AAC) for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be saved on the file.

#2 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_SHARE_RECORD_CALLBACK( pDevice, pCB, pUserData );
```

14.30 QCAP_REGISTER_AUDIO_MX_SHARE_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_MX_SHARE_RECORD_CALLBACK( UINT                          iRecNum,
                                                      PF_MX_SHARE_RECORD_CALLBACK   pCB,
                                                      PVOID                         pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_MX_SHARE_RECORD_CALLBACK)( UINT       iRecNum /*IN*/,
                                                                  UINT       iTrackNum /*IN*/,
                                                                  double     dSampleTime /*IN*/,
                                                                  BYTE *     pStreamBuffer /*IN*/,
                                                                  ULONG      nStreamBufferLen /*IN*/,
                                                                  PVOID      pUserData/*IN*/);
```

User can use this function to register above callback function.

The callback function will provide mix audio share data for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be saved on the file.

#2 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_MX_SHARE_RECORD_CALLBACK( iRecNum, pCB, pUserData );
```

14.31 QCAP_REGISTER_MEDIA_SHARE_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_MEDIA_SHARE_RECORD_CALLBACK( UINT                              iRecNum,
                                                   PF_MEDIA_SHARE_RECORD_CALLBACK    pCB,
                                                   PVOID                             pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_MEDIA_SHARE_RECORD_CALLBACK)( UINT      iRecNum /*IN*/,
                                                               double    dSampleTime /*IN*/,
                                                               BYTE *    pStreamBuffer /*IN*/,
                                                               ULONG     nStreamBufferLen /*IN*/,
                                                               PVOID     pUserData/*IN*/);
```

User can use this function to register above callback function.

The callback function will provide media recording's data (video or audio) for developer to use.

Note

#1 this callback function provide dtaa only support for TS and FLV format.

#2 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be saved on the file.

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_MEDIA_SHARE_RECORD_CALLBACK( pDevice, pCB, pUserData );
```

14.32 QCAP_REGISTER_VIDEO_DECODER_SHARE_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_DECODER_SHARE_RECORD_CALLBACK(
                                                    UINT                                    iRecNum,
                                                    PF_VIDEO_DECODER_SHARE_RECORD_CALLBACK  pCB,
                                                    PVOID                                   pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_DECODER_SHARE_RECORD_CALLBACK)(    UINT      iRecNum /*IN*/,
                                                                          double    dSampleTime /*IN*/,
                                                                          BYTE *    pFrameBuffer /*IN*/,
                                                                          ULONG     nFrameBufferLen /*IN*/,
                                                                          PVOID     pUserData/*IN*/);
```

User can use this function to register above callback function.

If user register `QCAP_REGISTER_VIDEO_DECODER_SHARE_RECORD_CALLBACK`, this callback function will providevideo

recording's uncompressed data after video decoder fordeveloperto use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame buffer will be dropped and will not be displayed on the window.

#2 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`,

the video preview data will not be displayed on the window.

`QCAP_RT_SKIP_DISPLAY= 0x00000004`

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_DECODER_SHARE_RECORD_CALLBACK( iRecNum, pCB, pUserData );
```

14.33 QCAP_REGISTER_AUDIO_DECODER_SHARE_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_DECODER_SHARE_RECORD_CALLBACK(
                                                    UINT                                    iRecNum,
                                                    PF_AUDIO_DECODER_SHARE_RECORD_CALLBACK  pCB,
                                                    PVOID                                   pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_DECODER_SHARE_RECORD_CALLBACK)(    UINT      iRecNum /*IN*/,
                                                                          double    dSampleTime /*IN*/,
                                                                          BYTE *    pFrameBuffer /*IN*/,
                                                                          ULONG     nFrameBufferLen /*IN*/,
                                                                          PVOID     pUserData/*IN*/);
```

User can use this function to register above callback function.

If users register `QCAP_REGISTER_AUDIO_DECODER_SHARE_RECORD_CALLBACK`, this callback function will provide audio uncompressed data after audio decoder for developerto use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`,

the audio uncompressed data will not beplayed on the window.

```
QCAP_RT_SKIP_DISPLAY = 0x00000004
```

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_DECODER_SHARE_RECORD_CALLBACK( iRecNum, pCB, pUserData );
```

14.34 QCAP_REGISTER_AUDIO_DECODER_MX_SHARE_RECORD_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_DECODER_MX_SHARE_RECORD_CALLBACK(
                                                    UINT                                    iRecNum,
                                                    PF_AUDIO_DECODER_MX_SHARE_RECORD_CALLBACK  pCB,
                                                    PVOID                                   pUserData )
```

Parameter

Name	Description
iRecNum	**IN**, Specify recorder index of recorder, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_DECODER_MX_SHARE_RECORD_CALLBACK)(UINT      iRecNum /*IN*/,
                                                                         UINT      iTrackNum /*IN*/,
                                                                         double    dSampleTime /*IN*/,
                                                                         BYTE *    pFrameBuffer /*IN*/,
                                                                         ULONG     nFrameBufferLen /*IN*/,
                                                                         PVOID     pUserData/*IN*/);
```

User can use this function to register above callback function.

If users register `QCAP_REGISTER_AUDIO_DECODER_MX_SHARE_RECORD_CALLBACK`, this callback function will provide audio

Uncompressed mix share data after audio decoder for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`,

the audio uncompressed data will not beplayed on the window.

```
QCAP_RT_SKIP_DISPLAY = 0x00000004
```

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_DECODER_MX_SHARE_RECORD_CALLBACK( iRecNum, pCB, pUserData );
```

14.35 QCAP_REGISTER_VIDEO_BROADCAST_SERVER_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_BROADCAST_SERVER_CALLBACK(   PVOID                              pServer,
                                                         UINT                               iSessionNum,
                                                         PF_VIDEO_BROADCAST_SERVER_CALLBACK  pCB,
                                                         PVOID                              pUserData)
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_BROADCAST_SERVER_CALLBACK)( PVOID     pServer /*IN*/,
                                                                   UINT      iSessionNum /*IN*/,
                                                                   double    dSampleTime /*IN*/,
                                                                   BYTE   *  pStreamBuffer /*IN*/,
                                                                   ULONG     nStreamBufferLen /*IN*/,
                                                                   BOOL      bIsKeyFrame /*IN*/,
                                                                   PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide video recording's compressed data (H.264) for developer of broadcast server to use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame stream buffer will be dropped and will not be saved on the file.

#2 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_BROADCAST_SERVER_CALLBACK( pDevice, pCB, pUserData );
```

14.36 QCAP_REGISTER_AUDIO_BROADCAST_SERVER_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_BROADCAST_SERVER_CALLBACK(   PVOID                              pServer,
                                                         UINT                               iSessionNum,
                                                         PF_AUDIO_BROADCAST_SERVER_CALLBACK  pCB,
                                                         PVOID                              pUserData )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_BROADCAST_SERVER_CALLBACK)( PVOID     pServer /*IN*/,
                                                                   UINT      iSessionNum /*IN*/,
                                                                   double    dSampleTime /*IN*/,
                                                                   BYTE   *  pStreamBuffer /*IN*/,
                                                                   ULONG     nStreamBufferLen /*IN*/,
                                                                   PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

The callback function will provide audio recording's compressed data for developer of broadcast server to use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_BROADCAST_SERVER_CALLBACK( pDevice, pCB, pUserData );
```

14.37 QCAP_REGISTER_VIDEO_DECODER_BROADCAST_SERVER_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_DECODER_BROADCAST_SERVER_CALLBACK(
                                        PVOID                                       pServer,
                                        UINT                                        iSessionNum,
                                        PF_VIDEO_DECODER_BROADCAST_SERVER_CALLBACK   pCB,
                                        PVOID                                       pUserData )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast serverobject
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_DECODER_BROADCAST_SERVER_CALLBACK)(
                                                                    PVOID     pServer /*IN*/,
                                                                    UINT      iSessionNum /*IN*/,
                                                                    double    dSampleTime /*IN*/,
                                                                    BYTE   *  pFrameBuffer /*IN*/,
                                                                    ULONG     nFrameBufferLen /*IN*/,
                                                                    PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If users register `QCAP_REGISTER_VIDEO_DECODER_BROADCAST_SERVER_CALLBACK`, this callback function will provide video recording's uncompressed data after video decoder for developer of broadcast server to use.

Note

#1 If callback return `QCAP_RT_FAIL` to `QCAP`, the frame buffer will be dropped and will not be displayed on the window.

#2 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`, the video preview data will not be displayed on the window of broadcast server.

```
QCAP_RT_SKIP_DISPLAY = 0x00000004
```

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_DECODER_BROADCAST_SERVER_CALLBACK( pDevice, pCB, pUserData );
```

14.38 QCAP_REGISTER_AUDIO_DECODER_BROADCAST_SERVER_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_DECODER_BROADCAST_SERVER_CALLBACK(
                                        PVOID                                       pServer,
                                        UINT                                        iSessionNum,
                                        PF_AUDIO_DEOCDER_BROADCAST_SERVER_CALLBACK  pCB,
                                        PVOID                                       pUserData )
```

Parameter

Name	Description
pServer	**IN**, Handle of the broadcast server object
iSessionNum	**IN**, Specify the number of channel session connections, start from 0
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_DECODER_BROADCAST_SERVER_CALLBACK)(
                                                                    PVOID     pServer /*IN*/,
                                                                    UINT      iSessionNum /*IN*/,
                                                                    double    dSampleTime /*IN*/,
                                                                    BYTE   *  pFrameBuffer /*IN*/,
                                                                    ULONG     nFrameBufferLen /*IN*/,
                                                                    PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If users register `QCAP_REGISTER_AUDIO_DECODER_BROADCAST_SERVER_CALLBACK`, this callback function will provide audio uncompressed data after audio decoder for developer ofbroadcast server to use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`,

the audio uncompressed data will not be played on the window of broadcast server.

```
QCAP_RT_SKIP_DISPLAY = 0x00000004
```

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_DECODER_BROADCAST_SERVER_CALLBACK( pDevice, pCB, pUserData );
```

14.39 QCAP_REGISTER_VIDEO_BROADCAST_CLIENT_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_BROADCAST_CLIENT_CALLBACK(
                                        PVOID                                   pClient,
                                        PF_VIDEO_BROADCAST_CLIENT_CALLBACK      pCB,
                                        PVOID                                   pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

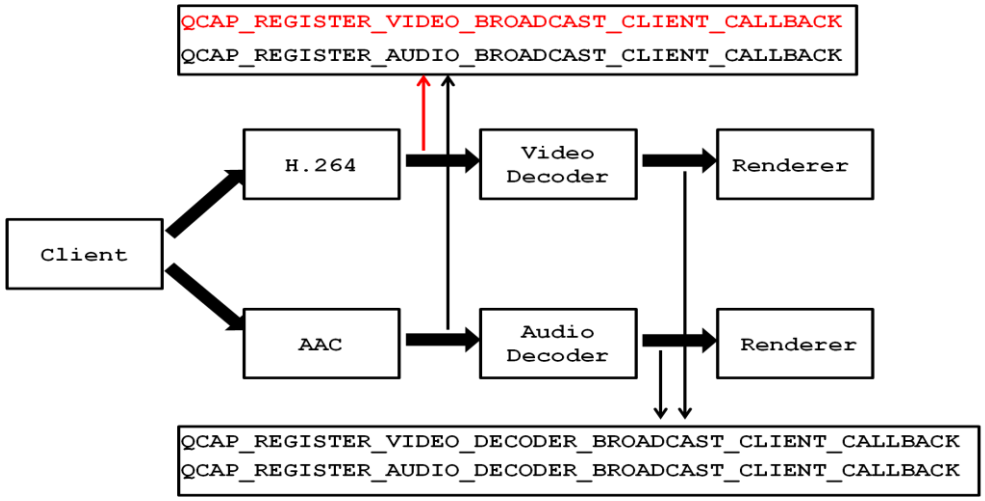
Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_BROADCAST_CLIENT_CALLBACK)( PVOID     pClient /*IN*/,
                                                                   double    dSampleTime /*IN*/,
                                                                   BYTE   *  pStreamBuffer /*IN*/,
                                                                   ULONG     nStreamBufferLen /*IN*/,
                                                                   BOOL      bIsKeyFrame /*IN*/,
                                                                   PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

The flowchart of the video & audio recording approach of broadcast client is shown in figure below.

If user register QCAP_REGISTER_VIDEO_BROADCAST_CLIENT_CALLBACK, this callback function will provide video compressed data (H.264) before video decoder for developer of broadcast client to use.





Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and will not be displayed and not sent into recording engine on the window.

#2 User can use callback return `QCAP_RT_SKIP_RECORD_NUM_00` to `QCAP` for set skip frame buffer, specify callback return type. Available values is

`QCAP_RT_SKIP_RECORD_NUM_00 = 0x00010002`

#3 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`, the video preview data (YUY2 or YV12) will not be displayed on the window of broadcast client.

`QCAP_RT_SKIP_DISPLAY = 0x00000004`

#4 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_BROADCAST_CLIENT_CALLBACK( pClient, pCB, pUserData );
```

14.40 QCAP_REGISTER_AUDIO_BROADCAST_CLIENT_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_BROADCAST_CLIENT_CALLBACK(
                                                PVOID                                   pClient,
                                                PF_AUDIO_BROADCAST_CLIENT_CALLBACK      pCB,
                                                PVOID                                   pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_BROADCAST_CLIENT_CALLBACK)( PVOID     pClient /*IN*/,
                                                                   double    dSampleTime /*IN*/,
                                                                   BYTE   *  pStreamBuffer /*IN*/,
                                                                   ULONG     nStreamBufferLen /*IN*/,
                                                                   PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If user register QCAP_REGISTER_AUDIO_BROADCAST_CLIENT_CALLBACK, this callback function will provide audio compressed data (PCM / AAC) before audio decoder for developer of broadcast client to use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 User can use callback return `QCAP_RT_SKIP_RECORD_NUM_00` to `QCAP` for set skip frame buffer, specify callback return type.

Available values is

`QCAP_RT_SKIP_RECORD_NUM_00 = 0x00010002`

#3 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`,

the audio uncompressed data will not be played on the window of broadcast client.

`QCAP_RT_SKIP_DISPLAY = 0x00000004`

#4 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_BROADCAST_CLIENT_CALLBACK( pClient, pCB, pUserData );
```

14.41 QCAP_REGISTER_VIDEO_DECODER_BROADCAST_CLIENT_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_DECODER_BROADCAST_CLIENT_CALLBACK(
                                                    PVOID                                          pClient,
                                                    PF_VIDEO_DECODER_BROADCAST_CLIENT_CALLBACK pCB,
                                                    PVOID                                          pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_DECODER_BROADCAST_CLIENT_CALLBACK)(
                                                                    PVOID     pClient /*IN*/,
                                                                    double    dSampleTime /*IN*/,
                                                                    BYTE   *  pFrameBuffer /*IN*/,
                                                                    ULONG     nFrameBufferLen /*IN*/,
                                                                    PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If user register QCAP_REGISTER_VIDEO_DECODER_BROADCAST_CLIENT_CALLBACK, this callback function will provide video recording's uncompressed data after video decoder for developer of broadcast client to use.

Note

#1 If callback return QCAP_RT_FAIL to QCAP, the frame buffer will be dropped and will not be displayed on the window.

#2 If callback return QCAP_RT_SKIP_DISPLAY to QCAP,

the video preview data will not be displayed on the window of broadcast client.

```
QCAP_RT_SKIP_DISPLAY= 0x00000004
```

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_DECODER_BROADCAST_CLIENT_CALLBACK( pClient, pCB, pUserData );
```

14.42 QCAP_REGISTER_VIDEO_DECODER_3D_BROADCAST_CLIENT_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_DECODER_3D_BROADCAST_CLIENT_CALLBACK(
                                                    PVOID                                                pClient,
                                                    PF_VIDEO_DECODER_3D_BROADCAST_CLIENT_CALLBACK pCB,
                                                    PVOID                                                pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_DECODER_3D_BROADCAST_CLIENT_CALLBACK)(
                                                                    PVOID     pClient /*IN*/,
                                                                    UINT      iChNum /*IN*/,
                                                                    double    dSampleTime /*IN*/,
                                                                    BYTE *    pFrameBuffer /*IN*/,
                                                                    ULONG     nFrameBufferLen /*IN*/,
                                                                    PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If user register QCAP_REGISTER_VIDEO_DECODER_3D_BROADCAST_CLIENT_CALLBACK, this callback function will provide video recording's uncompressed data after video decoder fordeveloperto use.

Note

#1 If callback return QCAP_RT_FAIL to QCAP, the frame buffer will be dropped and will not be displayed on the window.

#2 If callback return QCAP_RT_SKIP_DISPLAY to QCAP,

the video preview data will not bedisplayed on the window of 3D player.

QCAP_RT_SKIP_DISPLAY= 0x00000004

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_DECODER_3D_BROADCAST_CLIENT_CALLBACK( pClient, pCB, pUserData );
```

14.43 QCAP_REGISTER_AUDIO_DECODER_BROADCAST_CLIENT_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_DECODER_BROADCAST_CLIENT_CALLBACK(
                                        PVOID                                       pClient,
                                        PF_AUDIO_DECODER_BROADCAST_CLIENT_CALLBACK pCB,
                                        PVOID                                       pUserData )
```

Parameter

Name	Description
pClient	**IN**, Handle of the broadcast client object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_DECODER_BROADCAST_CLIENT_CALLBACK)(
                                                                    PVOID     pClient /*IN*/,
                                                                    double    dSampleTime /*IN*/,
                                                                    BYTE   *  pFrameBuffer /*IN*/,
                                                                    ULONG     nFrameBufferLen /*IN*/,
                                                                    PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If user register `QCAP_REGISTER_AUDIO_DECODER_BROADCAST_CLIENT_CALLBACK`, this callback function will provide audio uncompressed data after audio decoder for developer of broadcast client to use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`,

the audio uncompressed data will not beplayed on the window of broadcast client.

`QCAP_RT_SKIP_DISPLAY= 0x00000004`

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_DECODER_BROADCAST_CLIENT_CALLBACK( pClient, pCB, pUserData );
```

14.44 QCAP_REGISTER_VIDEO_DECODER_FILE_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_DECODER_FILE_CALLBACK(    PVOID                              pFile,
                                                      PPF_VIDEO_DECODER_FILE_CALLBACK    pCB,
                                                      PVOID                              pUserData)
```

Parameter

Name	Description
pFile	**IN**, Handle of the file object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_DECODER_FILE_CALLBACK)( PVOID     pFile /*IN*/,
                                                               double    dSampleTime /*IN*/,
                                                               BYTE   *  pFrameBuffer /*IN*/,
                                                               ULONG     nFrameBufferLen /*IN*/,
                                                               PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If user register QCAP_REGISTER_VIDEO_DECODER_FILE_CALLBACK, this callback function will provide video uncompressed data after video decoder fordeveloperto use.

Note

#1 If callback return QCAP_RT_FAIL to QCAP, the frame buffer will be dropped and its sound will be muted.

#2 If callback return QCAP_RT_SKIP_DISPLAY to QCAP,

the audio uncompressed data will not be played on the window.

QCAP_RT_SKIP_DISPLAY= 0x00000004

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_DECODER_FILE_CALLBACK( pFile, pCB, pUserData );
```

14.45 QCAP_REGISTER_VIDEO_DECODER_3D_FILE_CALLBACK

```
QRETURN QCAP_REGISTER_VIDEO_DECODER_3D_FILE_CALLBACK(PVOID                                  pFile,
                                                     PPF_VIDEO_DECODER_3D_FILE_CALLBACK    pCB,
                                                     PVOID                                 pUserData )
```

Parameter

Name	Description
pFile	**IN**, Handle of the file object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_VIDEO_DECODER_3D_FILE_CALLBACK)(  PVOID     pFile /*IN*/,
                                                                   UINT      iChNum /*IN*/,
                                                                   double    dSampleTime /*IN*/
                                                                   BYTE   *  pFrameBuffer /*IN*/,
                                                                   ULONG     nFrameBufferLen /*IN*/,
                                                                   PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If user register `QCAP_REGISTER_VIDEO_DECODER_3D_FILE_CALLBACK`, this callback function will provide video uncompressed data after video decoder for developer to use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`,

the audio uncompressed data will not beplayed on the window.

```
QCAP_RT_SKIP_DISPLAY = 0x00000004
```

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_VIDEO_DECODER_3D_FILE_CALLBACK( pFile, pCB, pUserData );
```

14.46 QCAP_REGISTER_AUDIO_DECODER_FILE_CALLBACK

```
QRETURN QCAP_REGISTER_AUDIO_DECODER_FILE_CALLBACK(   PVOID                              pFile,
                                                     PPF_AUDIO_DECODER_FILE_CALLBACK    pCB,
                                                     PVOID                              pUserData )
```

Parameter

Name	Description
pFile	**IN**, Handle of the file object
pCB	**IN**, Callback function
pUserData	**IN**, Additional data

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_GENERAL	Fail due to some errors

Introduction

```
typedef QRETURN (QCAP_EXPORT *PF_AUDIO_DECODER_FILE_CALLBACK)( PVOID     pFile /*IN*/,
                                                               double    dSampleTime /*IN*/,
                                                               BYTE   *  pFrameBuffer /*IN*/,
                                                               ULONG     nFrameBufferLen /*IN*/,
                                                               PVOID     pUserData /*IN*/ );
```

User can use this function to register above callback function.

If user register `QCAP_REGISTER_AUDIO_DECODER_FILE_CALLBACK`, this callback function will provide audio uncompressed data after audio decoder for developerto use.

Note

#1 If callback return `QCAP_RT_FAIL to QCAP`, the frame buffer will be dropped and its sound will be muted.

#2 If callback return `QCAP_RT_SKIP_DISPLAY` to `QCAP`,

the audio uncompressed data will not be played on the window.

`QCAP_RT_SKIP_DISPLAY = 0x00000004`

#3 dSampleTime is in second unit.

Example

```
QCAP_REGISTER_AUDIO_DECODER_FILE_CALLBACK( pFile, pCB, pUserData );
```

15 CD&DVD Burning Function API

15.1 QCAP_CREATE_BURNING_DRIVE

```
QRESULT QCAP_CREATE_BURNING_DRIVE( CHAR cDriveName, CHAR * pVolumeName, PVOID * ppDrive )
```

Parameter

Name	Description
cDriveName	**IN**, Specify the driver name
pVolumeName	**IN**, Specify the file name
ppDrive	**OUT**, Pointer to the address of CD&DVD driver

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to create burning driver.

Note

User can find driver name of DVD RW drive on Devices with Removable Storage.

Example

```
QCAP_CREATE_BURNING_DRIVE( 'E', "TEST", &pDrive );
```

15.2 QCAP_DESTROY_BURNING_DRIVE

```
QRESULT QCAP_DESTROY_BURNING_DRIVE( PVOID pDrive )
```

Parameter

Name	Description
pDrive	**IN**, Handle of the CD&DVD RW driver

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to destroy burning driver.

Example

```
QCAP_DESTROY_BURNING_DRIVE( pDrive );
```

15.3 QCAP_START_BURNING_DRIVE

```
QRESULT QCAP_START_BURNING_DRIVE( PVOID pDrive )
```

Parameter

Name	Description
pDrive	**IN**, Handle of the driver

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected

QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to start burning driver.

NOTE

The video record files are only support TS and FLV format.

Example

```
QCAP_START_BURNING_DRIVE( pDrive );
```

15.4 QCAP_STOP_BURNING_DRIVE

```
QRESULT QCAP_STOP_BURNING_DRIVE( PVOID pDrive )
```

Parameter

Name	Description
pDrive	**IN**, Handle of the driver

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to stop burning driver.

Example

```
QCAP_STOP_BURNING_DRIVE( pDrive );
```

15.5 QCAP_UNLOAD_DISC

```
QRESULT QCAP_UNLOAD_DISC( PVOID pDrive )
```

Parameter

Name	Description
pDrive	**IN**, Handle of the driver

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected

QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to unload disc.

Example

```
QCAP_UNLOAD_DISC( pDrive );
```

15.6 QCAP_LOAD_DISC

```
QRESULT QCAP_LOAD_DISC( PVOID pDrive )
```

Parameter

Name	Description
pDrive	**IN**, Handle of the driver

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to load disc.

Example

```
QCAP_LOAD_DISC( pDrive );
```

15.7 QCAP_GET_DISC_INFO

```
QRESULT QCAP_GET_DISC_INFO(   PVOID          pDrive,
                              BOOL       *   pIsDiscBlank,
                              BOOL       *   pIsDiscWriteable,
                              DWORD      *   pDiscType,
                              ULONGLONG  *   pDiscRemainCapability )
```

Parameter

Name	Description
pDrive	**IN**, Handle of the driver
pIsDiscBlank	**OUT**, Pointer of storage of disc blank
pIsDiscWriteable	**OUT**, Pointer of storage of disc writeable
pDiscType	**OUT**, Pointer of storage of disc type
pDiscRemainCapability	**OUT**, Pointer of storage of disc remain capability

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to get burn disc type information, the disc type as show in below table.

INFO Value	DISC Type
0X0008	CD-ROM
0X000A	CD-RW
0X0012	DVD-RAM
0X1314	DVD-RW
0X0014	DVD-RW
0X0016	DVD-R DL
0X001A	DVD+RW
0X0020	DDCD-ROM
0X0022	DDCD-RW
0X002B	DVD+R DL
0X0041	BD-R
0X0043	BD-RE
0X0051	HD-R
0X0053	HD-RW
0X005A	HD-RW DL
0X0009	CD-R

0X0010	DVD-ROM
0X0011	DVD-R
0X0013	DVD-RW
0X0015	DVD-R DL
0X0017	DVD-RW DL
0X001B	DVD+R
0X0021	DDCD-R
0X002A	DVD+RW DL
0X0040	BD-ROM
0X0042	BD-R
0X0050	HD-ROM
0X0052	HD-RAM
0X0052	DVD-R DL
0X0000	Errors DISC

Example

```
QCAP_GET_DISC_INFO( pDrive, &nIsDiscBlank, &nIsDiscWriteable, &nDiscType, &nDiscRemainCapability );
```

15.8 QCAP_ERASE_DISC

```
QRESULT QCAP_ERASE_DISC( PVOID pDrive )
```

Parameter

Name	Description
pDrive	**IN**, Handle of the driver

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to erase disc.

Example

```
QCAP_ERASE_DISC( pDrive );
```

15.9 CD&DVD Burning Software Programming Guides

```
// BURNING EXAMPLE:
//
// STEP#01 QCAP_CREATE_BURNING_DRIVE( 'E', "TEST", &pDrive );

//
// STEP#02 QCAP_START_BURNING_DRIVE( pDrive );

//
// STEP#03 QCAP_START_RECORD( pDevice, 0, "E:\\CH01.TS" );

//
// STEP#04 QCAP_STOP_RECORD( pDevice, 0, FALSE, INFINE ); // NOTE!! WAITIING FILE CLOSE

//
// STEP#05 QCAP_STOP_BURNING_DRIVE( pDrive );

//
// STEP#06 QCAP_UNLOAD_DISC( pDrive );

//
// STEP#07 QCAP_DESTROY_BURNING_DRIVE( pDrive );
```

16 CameraFunction API

16.1 QCAP_CAMERA_ENUMERATION

```
QRESULT QCAP_CAMERA_ENUMERATION( CHAR * * ppszCameraDevName, BOOLbNext = FALSE )
```

Parameter

Name	Description
ppszCameraDevName	**OUT**, Pointer to the address of all camera devices
bNext	**IN**, Specifies whether to usenumber n of the camera device on the platform. The value FALSE is to use the first camera device on the platform.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is used to enumerate all camera devices on the platform.

Example

```
CHAR * psz_camera_device_name = NULL;

BOOL bNext = FALSE;

QRESULT returns = QCAP_CAMERA_ENUMERATION( &m_pMainDialog->psz_camera_device_name, FALSE );

while( returns == QCAP_RS_SUCCESSFUL ) {

returns = QCAP_CAMERA_ENUMERATION( &m_pMainDialog->psz_camera_device_name, TRUE );

}
```

16.2 QCAP_SET_DEFAULT_CAMERA

```
QRESULTQCAP_SET_DEFAULT_CAMERA(   CHAR * pszCameraDevName )
```

Parameter

Name	Description
pszCameraDevName	**IN**, Specify capture filter name of microphone, default is “Camera”

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to modify the default DirectShow Video Capture Filter's name.

In Win7, the usual friendly name of VideoCapture Filter is "XXX" in different language systems;

So QCAP use string, "Camera", as default.

Example

```
QCAP_SET_DEFAULT_CAMERA( "Camera1" ); // ENGLISH WIN7

QCAP_SET_DEFAULT_CAMERA( "攝影機" ); // CHT WIN7
```

17 Sound Card Function API

17.1 QCAP_SOUNDCARD_ENUMERATION

```
QRESULT QCAP_SOUNDCARD_ENUMERATION(  CHAR * *  ppszSoundCardDevName, BOOLbNext = FALSE )
```

Parameter

Name	Description
ppszSoundCardDevName	**OUT**, Pointer to the address of all sound card devices
bNext	**IN**, Specifies whether to use number n of the sound card device on the platform. The value FALSE is to use the first sound card device on the platform.

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This function is used to enumerate all sound card devices on the platform.

Example

```
CHAR * psz_soundcard_device_name = NULL;

BOOL bNext = FALSE;

QRESULT returns = QCAP_SOUNDCARD_ENUMERATION( &m_pMainDialog->psz_soundcard_device_name, FALSE );

while( returns == QCAP_RS_SUCCESSFUL ) {

returns = QCAP_SOUNDCARD_ENUMERATION( &m_pMainDialog->psz_soundcard_device_name, TRUE );

}
```

17.2 QCAP_SET_DEFAULT_SOUNDCARD

```
QRESULT QCAP_SET_DEFAULT_SOUNDCARD(  CHAR * pszMicrophoneDevName, CHAR * pszLineInDevName )
```

Parameter

Name	Description
pszMicrophoneDevName	**IN**, Specify capture filter name of microphone, default is “High”
pszLineInDevName	**IN**, Specify capture filter name of line in, default is “High”

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

User can use this function to modify the default DirectShow Audio Capture Filter's name.

In Win7, the usual friendly name of AudioCapture Filter is "XXX (XXX High Definition Audio)" in different language systems;

So QCAP use string, "High", as default.

Example

```
QCAP_SET_DEFAULT_SOUNDCARD( "Microphone", "Line In" ); // ENGLISH WIN7

QCAP_SET_DEFAULT_SOUNDCARD( "麥克風", "線路輸入" ); // CHT WIN7

QCAP_SET_DEFAULT_SOUNDCARD( "麦克风", "线路输入" ); // CHS WIN7

QCAP_SET_DEFAULT_SOUNDCARD( "High", "High" ); // DEFAULT
```

18 Helper Function API

18.1 QCAP_GET_H264_BUFFER_LAYER_ID

```
QRESULT QCAP_GET_H264_BUFFER_LAYER_ID(  BYTE   *   pStreamBuffer,
                                        ULONG      nStreamBufferLen,
                                        ULONG  *   pLayerID )
```

Parameter

Name	Description
pStreamBuffer	**IN**, Specify the Frame source buffer
nStreamBufferLen	**IN**, Specify the source width
pLayerID	**OUT**, Returned the current ID layer number of H.264

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This is function for user to get ID layer number of H.264.

Example

```
QCAP_GET_H264_BUFFER_LAYER_ID( pStreamBuffer, nStreamBufferLen, &nLayerID );
```

18.2 QCAP_COLORSPACE_YUY2_TO_ABGR32

```
QRESULT QCAP_COLORSPACE_YUY2_TO_ABGR32( BYTE * pSrcFrameBuffer,
                                        ULONG  nSrcWidth,
                                        ULONG  nSrcHeight,
                                        ULONG  nSrcPitch,
                                        BYTE * pDstFrameBuffer,
                                        ULONG  nDstWidth,
                                        ULONG  nDstHeight,
                                        ULONG  nDstPitch,
                                        BYTE   bAlpah = 0x00,
                                        BOOL   bHorizontalMirror   = FALSE,
                                        BOOL   bVerticalMirror     = FALSE )
```

Parameter

Name	Description
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
pDstFrameBuffer	**OUT**, Handle of the target frame buffer
nDstWidth	**IN**, Specify the current horizontal width of target frame
nDstHeight	**IN**, Specify the vertical height of target frame
nDstPitch	**IN**, Specify the pitch of target frame
bAlpah	**IN**, Specify the alpah of target frame, the default is 0x00
bHorizontalMirror	**IN**, Enable/Disable HorizontalMirror
bVerticalMirror	**IN**, Enable/Disable VerticalMirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Use can use this function to change color space from YUY2 to ABGR32.

Example

```
QCAP_COLORSPACE_YUY2_TO_ABGR32( pSrcFrameBuffer, nSrcWidth, nSrcHeight, nSrcPitch, &nDstFrameBuffer,
                                nDstWidth, nDstHeight, nDstPitch, 0X00,
                                FALSE, FALSE );
```

18.3 QCAP_COLORSPACE_YV12_TO_ABGR32

```
QRESULT QCAP_COLORSPACE_YV12_TO_ABGR32( BYTE * pSrcFrameBuffer,
                                        ULONG  nSrcWidth,
                                        ULONG  nSrcHeight,
                                        ULONG  nSrcPitch,
                                        BYTE * pDstFrameBuffer,
                                        ULONG  nDstWidth,
                                        ULONG  nDstHeight,
                                        ULONG  nDstPitch,
                                        BYTE   bAlpah = 0x00,
                                        BOOL   bHorizontalMirror   = FALSE,
                                        BOOL   bVerticalMirror     = FALSE )
```

Parameter

Name	Description
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
pDstFrameBuffer	**OUT**, Handle of the target frame buffer
nDstWidth	**IN**, Specify the current horizontal width of target frame
nDstHeight	**IN**, Specify the vertical height of target frame
nDstPitch	**IN**, Specify the pitch of target frame
bAlpah	**IN**, Specify the alpah of target frame, the default is 0x00
bHorizontalMirror	**IN**, Enable/Disable HorizontalMirror
bVerticalMirror	**IN**, Enable/Disable VerticalMirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Use can use this function to to change color space from YV12 to ABGR32.

Example

```
QCAP_COLORSPACE_YV12_TO_ABGR32( pSrcFrameBuffer, nSrcWidth, nSrcHeight, nSrcPitch, &nDstFrameBuffer,
                                nDstWidth, nDstHeight, nDstPitch, 0X00,
                                FALSE, FALSE );
```

18.4 QCAP_COLORSPACE_YUY2_TO_BGR24

```
QRESULT QCAP_COLORSPACE_YUY2_TO_BGR24(  BYTE * pSrcFrameBuffer,
                                        ULONG  nSrcWidth,
                                        ULONG  nSrcHeight,
                                        ULONG  nSrcPitch,
                                        BYTE * pDstFrameBuffer,
                                        ULONG  nDstWidth,
                                        ULONG  nDstHeight,
                                        ULONG  nDstPitch,
                                        BOOL   bHorizontalMirror   = FALSE,
                                        BOOL   bVerticalMirror     = FALSE )
```

Parameter

Name	Description
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
pDstFrameBuffer	**OUT**, Handle of the target frame buffer
nDstWidth	**IN**, Specify the current horizontal width of target frame
nDstHeight	**IN**, Specify the vertical height of target frame
nDstPitch	**IN**, Specify the pitch of target frame
bHorizontalMirror	**IN**, Enable/Disable HorizontalMirror
bVerticalMirror	**IN**, Enable/Disable VerticalMirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Use can use this function to to change color space from YUY2 to BGR24.

Example

```
QCAP_COLORSPACE_YUY2_TO_BGR24( pSrcFrameBuffer, nSrcWidth, nSrcHeight, nSrcPitch, &nDstFrameBuffer,
                               nDstWidth, nDstHeight, nDstPitch, FALSE, FALSE );
```

18.5 QCAP_COLORSPACE_YV12_TO_BGR24

```
QRESULT QCAP_COLORSPACE_YV12_TO_BGR24(  BYTE * pSrcFrameBuffer,
                                        ULONG  nSrcWidth,
                                        ULONG  nSrcHeight,
                                        ULONG  nSrcPitch,
                                        BYTE * pDstFrameBuffer,
                                        ULONG  nDstWidth,
                                        ULONG  nDstHeight,
                                        ULONG  nDstPitch,
                                        BOOL   bHorizontalMirror   = FALSE,
                                        BOOL   bVerticalMirror     = FALSE )
```

Parameter

Name	Description
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
pDstFrameBuffer	**OUT**, Handle of the another frame buffer
nDstWidth	**IN**, Specify the current horizontal width of target frame
nDstHeight	**IN**, Specify the vertical height of target frame
nDstPitch	**IN**, Specify the pitch of target frame
bHorizontalMirror	**IN**, Enable/Disable HorizontalMirror
bVerticalMirror	**IN**, Enable/Disable VerticalMirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Use can use this function to to change color space from YV12 to BGR24.

Example

```
QCAP_COLORSPACE_YV12_TO_BGR24( pSrcFrameBuffer, nSrcWidth, nSrcHeight, nSrcPitch, &nDstFrameBuffer,
                               nDstWidth, nDstHeight, nDstPitch, FALSE, FALSE );
```

18.6 QCAP_COLORSPACE_YUY2_TO_YV12

```
QRESULT QCAP_COLORSPACE_YUY2_TO_YV12(   BYTE * pSrcFrameBuffer,
                                        ULONG  nSrcWidth,
                                        ULONG  nSrcHeight,
                                        ULONG  nSrcPitch,
                                        BYTE * pDstFrameBuffer,
                                        ULONG  nDstWidth,
                                        ULONG  nDstHeight,
                                        ULONG  nDstPitch,
                                        BOOL   bHorizontalMirror   = FALSE,
                                        BOOL   bVerticalMirror     = FALSE )
```

Parameter

Name	Description
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
pDstFrameBuffer	**OUT**, Handle of the target frame buffer
nDstWidth	**IN**, Specify the current horizontal width of target frame
nDstHeight	**IN**, Specify the vertical height of target frame
nDstPitch	**IN**, Specify the pitch of target frame
bHorizontalMirror	**IN**, Enable/Disable HorizontalMirror
bVerticalMirror	**IN**, Enable/Disable VerticalMirror

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

Use can use this function to change color space from YUY2 to YV12.

Example

```
QCAP_COLORSPACE_YUY2_TO_YV12( pSrcFrameBuffer, nSrcWidth, nSrcHeight, nSrcPitch, &nDstFrameBuffer,
                              nDstWidth, nDstHeight, nDstPitch, FALSE, FALSE );
```

18.7 QCAP_CONVERT_3D_STEREO_BUFFER

```
QRESULT QCAP_CONVERT_3D_STEREO_BUFFER(  ULONG     nColorSpaceType,
                                        ULONG     nSrcStereoDisplayMode,
                                        BYTE *    pSrcFrameBuffer,
                                        ULONG     nSrcWidth,
                                        ULONG     nSrcHeight,
                                        ULONG     nSrcPitch,
                                        ULONG     nDstStereoDisplayMode,
                                        BYTE *    pDstFrameBuffer,
                                        ULONG     nDstWidth,
                                        ULONG     nDstHeight,
                                        ULONG     nDstPitch,
                                        BOOL      bLeftRightSwap = FALSE )
```

Parameter

Name	Description
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
nSrcStereoDisplayMode	**IN**, Specify source 3D stereo display mode. Available values are QCAP_3D_STEREO_DISPLAY_MODE_SIDE_BY_SIDE = 0, QCAP_3D_STEREO_DISPLAY_MODE_TOP_BOTTOM = 1, QCAP_3D_STEREO_DISPLAY_MODE_LINE_BY_LINE = 2,
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
nDstStereoDisplayMode	**IN**, Specify another 3D stereo display mode. Available values are QCAP_3D_STEREO_DISPLAY_MODE_SIDE_BY_SIDE = 0, QCAP_3D_STEREO_DISPLAY_MODE_TOP_BOTTOM = 1, QCAP_3D_STEREO_DISPLAY_MODE_LINE_BY_LINE = 2,
pDstFrameBuffer	**OUT**, Handle of the another frame buffer
nDstWidth	**IN**, Specify the horizontal width of another frame

nDstHeight	**IN**, Specify the vertical height of another frame
nDstPitch	**IN**, Specify the pitch of another frame
bLeftRightSwap	**IN**, Swap the another frame source's (right to left, left to right), the default is FALSE

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

The developer can use function to convert 3D stereo buffer.

Example

```
QCAP_CONVERT_3D_STEREO_BUFFER( QCAP_COLORSPACE_TYEP_YUY2, QCAP_3D_STEREO_DISPLAY_MODE_SIDE_BY_SIDE
                               &nSrcFrameBuffer, nSrcWidth, nSrcHeight, nSrcPitch,
                               QCAP_3D_STEREO_DISPLAY_MODE_LINE_BY_LINE, nDstWidth, nDstHeight,
                              nDstPitch, FALSE );
```

18.8 QCAP_RESIZE_VIDEO_BUFFER

```
QRESULT QCAP_RESIZE_VIDEO_BUFFER(ULONG     nColorSpaceType,
                                 BYTE *    pSrcFrameBuffer,
                                 ULONG     nSrcWidth,
                                 ULONG     nSrcHeight,
                                 ULONG     nSrcPitch,
                                 BYTE *    pDstFrameBuffer,
                                 ULONG     nDstWidth,
                                 ULONG     nDstHeight,
                                 ULONG     nDstPitch )
```

Parameter

Name	Description
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
pDstFrameBuffer	**OUT**, Handle of the another frame buffer
nDstWidth	**IN**, Specify the horizontal width of another frame
nDstHeight	**IN**, Specify the vertical height of another frame
nDstPitch	**IN**, Specify the pitch of another frame

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

The developer can use function to resize of video source.

Example

```
QCAP_RESIZE_VIDEO_BUFFER( QCAP_COLORSPACE_TYEP_YUY2, &nSrcFrameBuffer, nSrcWidth, nSrcHeight, nSrcPitch,
                          &nDstFrameBuffer,  nDstWidth,  nDstHeight, nDstPitch );
```

18.9 QCAP_GET_ROTATE_VIDEO_BUFFER_BOUNDARY

```
QRESULT QCAP_GET_ROTATE_VIDEO_BUFFER_BOUNDARY(    ULONG     nSrcWidth,
                                                  ULONG     nSrcHeight,
                                                  ULONG     nSrcPitch,
                                                  ULONG *   pDstWidth,
                                                  ULONG *   pDstHeight,
                                                  double    dAngle )
```

Parameter

Name	Description
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
pDstWidth	**OUT**, Pointer to the storage of horizontal width of another frame
pDstHeight	**OUT**, Pointer to the storage of vertical height of another frame
dAngle	**IN**, Specify the rotate angle of source, range is 0~360

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

The developer can use function to get boundary of rotate angle of source.

Example

```
QCAP_GET_ROTATE_BOUNDARY( nSrcWidth, nSrcHeight, nSrcPitch,
                          nDstWidth, nDstHeight, dAngle );
```

18.10 QCAP_ROTATE_VIDEO_BUFFER

```
QRESULT QCAP_ROTATE_VIDEO_BUFFER(ULONG     nColorSpaceType,
                                 BYTE *    pSrcFrameBuffer,
                                 ULONG     nSrcWidth,
                                 ULONG     nSrcHeight,
                                 ULONG     nSrcPitch,
                                 BYTE *    pDstFrameBuffer,
                                 ULONG     nDstWidth,
                                 ULONG     nDstHeight,
                                 ULONG     nDstPitch,
                                 double    dAngle,
                                 BYTE *    pSrcTempFrameBuffer = NULL,
                                 BYTE *    pDstTempFrameBuffer = NULL,
                                 BOOL      bClearBackground = TRUE )
```

Parameter

Name	Description
nColorSpaceType	**IN**, Specify encoder color space type. Available values are QCAP_COLORSPACE_TYEP_UYVY QCAP_COLORSPACE_TYEP_I420 QCAP_COLORSPACE_TYEP_YUY2 QCAP_COLORSPACE_TYEP_YV12 QCAP_COLORSPACE_TYEP_RGB24 QCAP_COLORSPACE_TYEP_BGR24 QCAP_COLORSPACE_TYEP_ARGB32 QCAP_COLORSPACE_TYEP_ABGR32 QCAP_COLORSPACE_TYEP_H264
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcWidth	**IN**, Specify the source width
nSrcHeight	**IN**, Specify the source height
nSrcPitch	**IN**, Specify the source pitch
pDstFrameBuffer	**OUT**, Handle of the another frame buffer
nDstWidth	**IN**, Specify the current horizontal width of another frame
nDstHeight	**IN**, Specify the vertical height of another frame
nDstPitch	**IN**, Specify the pitch of another frame
dAngle	**IN**, Specify the rotate angle of source, range is 0~360
pSrcTempFrameBuffer	**IN**, Specify the temp Frame source buffer
pDstTempFrameBuffer	**IN**, Specify the temp Frame another buffer

bClearBackground	**IN**, If true, the background color will be remove

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

The developer can use function to set rotate angle of source.

Example

```
QCAP_ROTATE_VIDEO_BUFFER( QCAP_COLORSPACE_TYEP_YUY2, &nSrcFrameBuffer, nSrcWidth, nSrcHeight, nSrcPitch,
                &nDstFrameBuffer,  nDstWidth,  nDstHeight, nDstPitch,  dAngle, NULL, NULL, TRUE );
```

18.11 QCAP_LOAD_PICTURE_BUFFER

```
QRESULT QCAP_LOAD_PICTURE_BUFFER(    CHAR   *   pszFileName,
                                     ULONG  *   pColorSpaceType,
                                     BYTE   *   pFrameBuffer,
                                     ULONG  *   pFrameBufferSize,
                                     ULONG  *   pFrameWidth,
                                     ULONG  *   pFrameHeight,
                                     ULONG  *   pFramePitch )
```

Parameter

Name	Description
pszFileName	**IN**, Specify the picture name to display picture output
pColorSpaceType	**OUT**, Pointer to the storage of the color space type for the picture
pFrameBuffer	**OUT**, Pointer to the storage of the picture buffer
pFrameBufferSize	**OUT**, Pointer to the input picture buffer's size **IN**, Specify the input picture buffer's size
pFrameWidth	**OUT**, Pointer to the storage of the picture width
pFrameHeight	**OUT**, Pointer to the storage of the picture Height
pFramePitch	**OUT**, Pointer to the storage of the picture pitch

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

It is a help function. User can use this function to load picture buffer.

Example

Step1: User can set pFrameBuffer of pitcure is NULL to get input pitcure buffer's size, and then use pitcure buffer's size information to malloc buffer.

```
QCAP_LOAD_PICTURE_BUFFER( "Pitcure.bmp", &nColorSpaceType, NULL, &nFrameBufferSize,
                          &nFrameWidth, &nFrameHeight, &nFramePitch );
```

Step2: call QCAP_LOAD_PITCURE_BUFFER again.

```
QCAP_LOAD_PITCURE_BUFFER( "Pitcure.bmp", &nColorSpaceType, &nFrameBuffer, nFrameBufferSize,
                          &nFrameWidth, &nFrameHeight, &nFramePitch );
```

18.12 QCAP_GET_AUDIO_BUFFER_VOLUME_DB

```
QRESULT QCAP_GET_AUDIO_BUFFER_VOIUME_DB(    BYTE   *   pFrameBuffer,
                                            ULONG      nFrameBufferLen,
                                            ULONG      nChannels,
                                            ULONG      nBitsPerSample,
                                            ULONG      nSampleFrequency,
                                            UINT       iChNum,
                                            double *   pVolumeDB )
```

Parameter

Name	Description
pFrameBuffer	**IN**, Specify a raw data of frame contained in a buffer
nFrameBufferLen	**IN**, Specify the input source buffer's size
nChannels	**IN**, Specify the source channels for the i-th audio stream session
nBitsPerSample	**IN**, Specify the source bits per sample for the i-th audio stream session
nSampleFrequency	**IN**, Specify the source sample frequency for the i-th audio stream session
iChNum	**IN**, Specify the number of channel, 0 is left channel 1 is right channel
pVolumeDB	**OUT**, Pointer to the storage of the number of channel volume

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This is audio help function user can use this to get channel volume.

Example

```
QCAP_GET_AUDIO_BUFFER_VOLUME_DB( pFrameBuffer, nFrameBufferLen, 2, 16, 48000,
                                 0, &VolumeDB_L );  //Left Channel
QCAP_GET_AUDIO_BUFFER_VOLUME_DB( pFrameBuffer, nFrameBufferLen, 2, 16, 48000,
                                 1, &VolumeDB_R );  //Right Channel
```

18.13 QCAP_RESAMPLE_AUDIO_BUFFER

```
QRESULT QCAP_RESAMPLE_AUDIO_BUFFER(  BYTE   *   pSrcFrameBuffer,
                                     ULONG      nSrcFrameBufferLen,
                                     ULONG      nSrcChannels,
                                     ULONG      nSrcBitsPerSample,
                                     ULONG      nSrcSampleFrequency,
                                     BYTE   *   pDstFrameBuffer,
                                     ULONG  *   pDstFrameBufferLen,
                                     ULONG      nDstChannels,
                                     ULONG      nDstBitsPerSample,
                                     ULONG      nDstSampleFrequency )
```

Parameter

Name	Description
pSrcFrameBuffer	**IN**, Specify the Frame source buffer
nSrcFrameBufferLen	**IN**, Specify the source buffer's size
nSrcChannels	**IN**, Specify the source channels for the i-th audio stream session
nSrcBitsPerSample	**IN**, Specify the source bits per sample for the i-th audio stream session
nSrcSampleFrequency	**IN**, Specify the source sample frequency for the i-th audio stream session
pDstFrameBuffer	**OUT**, Pointer to the storage of the target frame buffer
pDstFrameBufferLen	**IN**, Specify the target frame buffer's size **OUT**, Handle of the target frame buffer's size
nDstChannels	**IN**, Specify the target channels for the i-th audio stream session
nDstBitsPerSample	**IN**, Specify the target bits per sample for the i-th audio stream session
nDstSampleFrequency	**IN**, Specify the target sample frequency for the i-th audio stream session

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This is audio help function user can use this to resamle audio format.

For example, 2Ch 16Bit 48000HZ transform to 1Ch 8Bit 44100HZ.

Example

Step1: User can set pFrameBuffer of pitcure is NULL to get input source buffer's size, and then use input source buffer's size information to malloc buffer.

```
QCAP_RESAMPLE_AUDIO_BUFFER( pFrameBuffer, nFrameBufferLen, 2, 16, 48000,
                            NULL, &nDstAudioBufferLen, 2, 16, 44100 );
```

Step2: call QCAP_RESAMPLE_AUDIO_BUFFER again.

```
QCAP_RESAMPLE_AUDIO_BUFFER( pFrameBuffer, nFrameBufferLen, 2, 16, 48000,
                            NULL, &nDstAudioBufferLen, 2, 16, 44100 );
```

18.14 QCAP_HELPER_OBJPTR

```
QRESULT QCAP_HELPER_OBJPTR( PVOID pObj )
```

Parameter

Name	Description
pObj	**IN**, Handle of the helper object

Returned Value

Name	Description
QCAP_RS_SUCCESSFUL	Success
QCAP_RS_ERROR_GENERAL	Fail due to some errors
QCAP_RS_ERROR_OUT_OF_MEMORY	Fail due to system memory is not enough to allocate the capture object
QCAP_RS_ERROR_OUT_OF_RESOURCE	Fail due to out of resource
QCAP_RS_ERROR_INVALID_DEVICE	Cannot open video or audio capture device
QCAP_RS_ERROR_INVALID_PARAMETER	Fail due to invalid input parameter
QCAP_RS_ERROR_NON_SUPPORT	Fail due to encoder typenon support
QCAP_RS_ERROR_TIMEOUT	Fail due to time out
QCAP_RS_ERROR_INVALID_ANIMATION_SCRIPT	Fail due to invalid animation script
QCAP_RS_ERROR_NO_SIGNAL_DETECTED	Fail due to no signal detected
QCAP_RS_ERROR_NEED_MORE_DATA	Fail due to need more data information
QCAP_RS_ERROR_CONNECT_FAIL	Fail due to connect fail

Introduction

This is help function for .NET developer to get a handle helper object.

Example

```
QCAP_HELPER_OBJPTR( pObj );
```