

libcom3.h

[Go to the documentation of this file.](#)

```
1  #pragma once      2      3  #if defined __GNUC__ && defined VFI_COM_SHARED
<sys/types.h>    10  #include <ipc/jsobject.h>    11  #include <exception>
12      13  namespace vficom {    14      15  namespace cmd {    16      17  COM
<! String, can be empty    210  COM3_DllSpec extern const char bssid[];    211
  COM3_DllSpec extern const char 212  COM3_DllSpec extern const
char pairw[];    213  COM3_DllSpec extern const char group[];    214
COM3_DllSpec extern const char km[];    215  COM3_DllSpec extern const
char perc[];    216  COM3_DllSpec extern const char dbm[];    217
COM3_DllSpec extern const char freq[];    218  COM3_DllSpec extern const
char proto[];    219  COM3_DllSpec extern const char std[];    220  221
  } // namespace json_keys    222  223  } // namespace wlan_scan    224
225  } // namespace out    226  227  } // namespace cmdparam    228  229
  namespace event {    230  enum EventType {    231      EVT_URC_STK = 0,    232
  EVT_URC_USSD = 1,    233      EVT_MAX_ENUM = 0x7FFFFFFF    234  };    235
  236  namespace stk {    237  enum STKURCProCommand {    238
SET_UP_EVENT_LIST    = 5,    239      SELECT_ITEM    = 36,    240
MENU_SELECTION    = 37,    241      STK_URC_PRO_MAX_ENUM = 0x7FFFFFFF    242
  };    243  244  COM3_DllSpec extern const char urc[];
245  COM3_DllSpec extern const char proactive_command[];    246
COM3_DllSpec extern const char parameters[];    247  COM3_DllSpec
extern const char timeout[];    248  249  namespace
set_up_event_list {    250  251  COM3_DllSpec extern const char event_list
[];    252  253  } // namespace set_up_event_list    254  255  namespace
select_item {    256  257  COM3_DllSpec extern const char cmd_qualifier
[];    258  COM3_DllSpec extern const char alpha[];    259
COM3_DllSpec extern const char item_id[];    260  COM3_DllSpec
extern const char total_items[];    261  COM3_DllSpec extern const
char item_text[];    262  COM3_DllSpec extern const char next_action
[];    263  COM3_DllSpec extern const char default_item[];    264
  COM3_DllSpec extern const char icon_id[];    265  COM3_DllSpec
extern const char icon_id_list_elem[];    266  267  } // namespace
select_item    268  269  namespace menu_selection {    270  271
COM3_DllSpec extern const char cmd_qualifier[];    272  COM3_DllSpec
extern const char alpha[];    273  COM3_DllSpec extern const
char item_id[];    274  COM3_DllSpec extern const char total_items
[];    275  COM3_DllSpec extern const char item_text[];    276
  COM3_DllSpec extern const char next_action[];    277  COM3_DllSpec
extern const char icon_id[];    278  COM3_DllSpec extern const
char icon_id_list_elem[];    279  280  } // namespace menu_selection    281
  282  } // namespace stk    283  284  namespace ussd {    285  286
  // Names correspond to 3GPP TS 27.00, Chapter 7.15, Unstructured
```

```

supplementary service data +CUSD 287Â COM3_DllSpec extern const char m
[]; 288Â COM3_DllSpec extern const char str[]; 289Â COM3_DllSpec
extern const char dcs[]; 290Â 291Â } // namespace ussd 292Â 293
Â } // namespace event 294Â 295Â namespace status { 296Â 297Â enum
PipeStatus { 298Â OK = 0, 299Â INVALID = 1,
300Â NOSUPPORT = 2, 301Â NOTREADY = 3, 302Â
FAILED = 5, 303Â TIMEOUT = 6, 304Â STARVATION
= 7, 305Â USER_INTERRUPT = 8, 306Â MAX_ENUM =
0x7FFFFFFF 307Â }; 308Â 309Â } // namespace status 310Â 315Â class
COM3_DllSpec Exception : std::exception { 316Â char description[64
]; 317Â status::PipeStatus error; 318Â 319Â public: 320Â
Exception(enum status::PipeStatus error, const char *description); 321
Â 322Â virtual const char *what() const throw(); 323Â
status::PipeStatus getError() const; 324Â }; 325Â 326Â class
COM3_DllSpec Milliseconds { 327Â public: 328Â u_int64_t millis; 329
Â 330Â Milliseconds(u_int32_t millis); 331Â ~Milliseconds();
332Â }; 333Â 334Â class COM3_DllSpec CommandPipe { 335Â class
ComdaemonPipe *impl; 336Â 337Â // Copy not valid 338Â
CommandPipe(const CommandPipe &commandPipe); 339Â void operator=(const
CommandPipe &commandPipe); 340Â 341Â public: 342Â CommandPipe
(); 343Â virtual ~CommandPipe(); 344Â 354Â enum
status::PipeStatus sendRecv(const vfiipc::JSObject &in, vfiipc::JSObject &
out, Milliseconds millis); 355Â 367Â enum status::PipeStatus
sendRecv(const vfiipc::JSObject &in, vfiipc::JSObject &out, Milliseconds
millis, int interruptFD); 368Â }; 369Â 370Â class COM3_DllSpec
EventPipe { 371Â class ComdaemonPipe *impl; 372Â 373Â public: 374
Â EventPipe(event::EventType type); 375Â ~EventPipe(); 376Â
377Â // Copy not valid 378Â EventPipe(const EventPipe &eventPipe
); 379Â void operator=(const EventPipe &eventPipe); 380Â 389Â
enum status::PipeStatus recv(vfiipc::JSObject &out, Milliseconds millis);
390Â 401Â enum status::PipeStatus recv(vfiipc::JSObject &out,
Milliseconds millis, int interruptFD); 402Â }; 403Â 404Â } //
namespace vfiicom 405Â 406Â #undef COM3_DllSpec

```