# Vehicle Service

# API specification

**Note/**
**CWORDXX is mask word for confidential information.**

2019.10.17
TOYOTA MOTOR CORPRATION

# Module Documentation

## BaseSystem

Vehicle_service

---

**Detailed Description**

## Vehicle_service

Positioning
Positioning_base_library

---

**Detailed Description**

## Positioning

struct POS_POSDATA
struct SENSORLOCATION_MSG_LONLATINFO
struct SENSORLOCATION_MSG_ALTITUDEINFO
struct SENSORMOTION_MSG_HEADINGINFO
struct SENSORMOTION_MSG_SPEEDINFO
struct POS_LOCATIONINFO_NMEA
struct SENSOR_MSG_SEND_DAT
struct SENSOR_PKG_MSG_VSINFO
struct SENSOR_MSG_VSINFO

**Macros**

#define SENSORLOCATION_STATUS_DISABLE (0)
*Not available.*

#define SENSORLOCATION_STATUS_ENABLE (1)
*Available.*

#define SENSORMOTION_STATUS_DISABLE (0)
*Not available.*

#define SENSORMOTION_STATUS_ENABLE (1)
*Available.*

#define SENSOR_GET_METHOD_AUTO (0)
*Not specified.*

#define SENSOR_GET_METHOD_GPS (1)

*GPS.*

#define SENSOR_GET_METHOD_NAVI (2)

*Navigation.*

#define SENSOR_GET_METHOD_DR (3)

*Dead Reckoning.*

#define SENSOR_GET_METHOD_POS (4)

*positioning (Internal)*

#define LOCATIONINFO_NMEA_MAX (1020)

*Max length of 'Location Information'.*

#define CID_POSIF_REGISTER_LISTENER_SPEED (0x0203)

*Delivery speed.*

#define CID_POSIF_REGISTER_LISTENER_LONLAT 0x0781

*Delivery longitude and latitude.*

#define CID_POSIF_REGISTER_LISTENER_ALTITUDE 0x0782

*Delivery altitude.*

#define CID_POSIF_REGISTER_LISTENER_HEADING 0x0783

*Delivery heading.*

#define POS_AVAILABILITY "Positioning/Availability"

*Availability.*

#define POS_NTFY_SEND_THREAD "POS_Main"

*POS_Main thread.*

#define POS_NTFY_SEND_THREAD_GPS "POS_Gps"

*POS_Gps thread.*

#define SENSOR_RET_NORMAL 0

*normal finish*

#define SENSOR_RET_ERROR_PID (-1)

*thread ID error*

#define SENSOR_RET_ERROR_DID (-2)

*data ID error*

#define SENSOR_RET_ERROR_DID_DIS (-3)

*data ID not CAN ID*

#define SENSOR_RET_ERROR_PARAM (-4)

*parameter error*

#define SENSOR_RET_ERROR_BUFFULL (-5)

*buffer full error*

#define SENSOR_RET_ERROR_CREATE_EVENT (-6)

*create event error*

#define SENSOR_RET_ERROR_TIMER (-7)

*create timer error*

#define SENSOR_RET_ERROR_OUTOF_MEMORY (-8)

*share memory allocation size error*

#define SENSOR_RET_ERROR_SIZE (-9)

*memory size error*

#define SENSOR_RET_ERROR (-10)

*error occured*

#define SENSOR_RET_ERROR_NOSUPPORT (-11)

*no support*

#define SENSOR_RET_ERROR_INNER (-12)

*Internal error.*

#define SENSOR_RET_ERROR_RESOURCE (-13)

*lack of resources*

#define SENSOR_RET_ERROR_MIN POS_RET_ERROR_MIN

*min value of error*

#define POS_LOC_INFO_LAT 0x01

*current position latitude(bit0) 1:valid 0:invalid*

#define POS_LOC_INFO_LON 0x02

*current position longitude(bit1) 1:valid 0:invalid*

#define POS_LOC_INFO_ALT 0x04

*current position altitude(bit2) 1:valid 0:invalid*

#define POS_LOC_INFO_HEAD 0x08

*current position heading(bit3) 1:valid 0:invalid*

#define POS_LOC_INFO_USE_GSP 0x01

*GPS data used result(bit0) 1:valid 0:invalid.*

#define POS_LOC_INFO_USE_DGPS 0x02

*DGPS data used result(bit1) 1:valid 0:invalid.*

#define POS_LOC_INFO_USE_DR 0x04

*Dead Reckoning used result(bit2) 1:valid 0:invalid.*

#define POS_LOC_INFO_USE_MAPMATCHING 0x08

*MapMatching result(bit3) 1:valid 0:invalid.*

#define SENSOR_DELIVERY_REGIST 0x01

*register delivery*

#define SENSOR_DELIVERY_TIMING_UPDATE 0x01

*delivery update timing*

#define SENSOR_DELIVERY_TIMING_CHANGE 0x02

*delivery change timing*

#define CID_POSIF_REGISTER_LISTENER_GPS_TIME_SET_REQ 0x0780

*GPS time setting result delivery command ID.*

#define POS_DID_SPEED_PULSE 0x80000012

*Data ID of speed pulse.*

#define POS_DID_SPEED_KMPH 0x80000013

*Data ID of KMPH speed.*

#define POS_DID_SNS_COUNTER ***

*Data ID of sensor counter.*

#define POS_DID_GYRO ***

*Data ID of gyro.*

#define POS_DID_GSNS_X ***

*Data ID of x axis gsensor.*

#define POS_DID_GSNS_Y ***

*Data ID of Y axis gsensor.*

#define POS_DID_REV ***

*Data ID of reverse signal.*

#define POS_DID_GPS_ANTENNA ***

*Data ID of GPS antenna status.*

#define POS_DID_SPEED_PULSE_FST ***

*Data ID of first time speed pulse.*

#define POS_DID_GYRO_FST ***

*Data ID of first time gyro.*

#define POS_DID_REV_FST ***

*Data ID of first time reverse signal.*

#define POS_DID_GYRO_TEMP ***

*Data ID of gyro temperature.*

#define POS_DID_GYRO_TEMP_FST ***

*Data ID of first time gyro temperature.*

#define POS_DID_GSNS_X_FST ***

*Data ID of first time x axis gsensor.*

#define POS_DID_GSNS_Y_FST ***

*Data ID of first time Y axis gsensor.*

#define POS_DID_PULSE_TIME ***

*Data ID of pulse time.*

#define POS_DID_GPS_CWORD44_NMEA ***

*Data ID of CWORD44 GPS NMEA sentence.*

#define POS_DID_GPS_CWORD44_P_CWORD44_DGP4 ***

*Data ID of CWORD44 GPS P_CWORD44_DGP4 data.*

#define POS_DID_GPS_CWORD44_FULLBINARY ***

*Data ID of CWORD44 GPS full binary data.*

#define POS_DID_GPS_NMEA ***

*Data ID of GPS NMEA sentence.*

#define POS_DID_GPS_CLOCK_DRIFT ***

*Data ID of GPS time drift data.*

#define POS_DID_GPS_CLOCK_FREQ ***

*Data ID of GPS time frequency data.*

#define CID_POSIF_REGISTER_LISTENER_PKG_SENSOR_DATA 0x0700

*sensor extra package register command ID*

#define CID_POSIF_REGISTER_LISTENER_SENSOR_DATA 0x0200

*sensor information register command ID*

#define SENSOR_MSGBUF_DSIZE 2264

*message body maximum size*

#define SENSOR_VSHEAD_DSIZE 36

*vehicle sensor header size(1+3+16*2)*

#define SENSOR_VSINFO_DSIZE (SENSOR_MSGBUF_DSIZE - SENSOR_VSHEAD_DSIZE)

*vehicle sensor data size*

#define SENSOR_MSG_VSINFO_DSIZE 1272

*vehicle sensor message body maximum size*

#define SENSOR_PKG_DELIVERY_MAX 16

*number of data ID per a package*

**Typedefs**

typedef int32_t SENSOR_RET_API

*API return value.*

typedef int32_t POS_RET_API

*API return value.*

typedef uint32_t DID

*data ID*

typedef int32_t RET_API

*CWORD64 API return value*

typedef int32_t NAVIINFO_RET_API

*define return value of Get/Set GPS information API*

**Functions**

POS_RET_API POS_RegisterListenerLonLat (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming, uint8_t ucGetMethod)

POS_RET_API POS_RegisterListenerAltitude (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming, uint8_t ucGetMethod)

POS_RET_API POS_RegisterListenerSpeed (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming, uint8_t ucGetMethod)

POS_RET_API POS_RegisterListenerHeading (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming, uint8_t ucGetMethod)

POS_RET_API POS_GetLonLat (HANDLE hApp, SENSORLOCATION_LONLATINFO_DAT *dat, uint8_t ucGetMethod)

POS_RET_API POS_GetAltitude (HANDLE hApp, SENSORLOCATION_ALTITUDEINFO_DAT *dat, uint8_t ucGetMethod)
POS_RET_API POS_GetSpeed (HANDLE hApp, SENSORMOTION_SPEEDINFO_DAT *dat, uint8_t ucGetMethod)
POS_RET_API POS_GetHeading (HANDLE hApp, SENSORMOTION_HEADINGINFO_DAT *dat, uint8_t ucGetMethod)
POS_RET_API POS_SetSpeedInfo (HANDLE hApp, uint16_t navispeed)
POS_RET_API POS_SetLocationInfo (HANDLE hApp, POS_POSDATA *pstPosData)
POS_RET_API POS_SetLocationInfoNmea (HANDLE hApp, POS_LOCATIONINFO_NMEA *locationInfo)
int32_t POS_ReqGPSSetting (HANDLE hApp, SENSOR_MSG_SEND_DAT *p_data)
NAVIINFO_RET_API POS_SetGPSInfo (HANDLE hApp, NAVIINFO_ALL *navilocinfo)
NAVIINFO_RET_API POS_GetGPSInfo (HANDLE hApp, NAVIINFO_DIAG_GPS *navidiaginfo)
POS_RET_API POS_ReqGPSReset (HANDLE hApp, PCSTR ResName, uint8_t mode)
POS_RET_API POS_GetGPSVersion (HANDLE hApp, uint8_t buf_size, int8_t *buf, uint8_t *size)
POS_RET_API POS_RegisterListenerGPSTimeSetReq (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg)
POS_RET_API POS_SetGPStime (HANDLE hApp, POS_DATETIME *pstDateTime)
SENSOR_RET_API POS_RegisterListenerGPStime (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming)
POS_RET_API POS_GetGPStime (HANDLE hApp, SENSOR_GPSTIME *dat)
SENSOR_RET_API POS_RegisterListenerPkgSensData (HANDLE hApp, PCSTR notifyName, uint8_t ucPkgNum, DID *pulDid, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming)
SENSOR_RET_API POS_RegisterListenerSensData (HANDLE hApp, PCSTR notifyName, DID ulDid, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming)
POS_RET_API POS_GetSensData (HANDLE hApp, DID ulDid, void *pDestData, uint16_t usDestSize)

---

**Detailed Description**

---

**Class Documentation**

**struct POS_POSDATA**

position information

**Class Members:**

| | | |
|---|---|---|
| int32_t | altitude | current altitude |
| int16_t | heading | current heading |
| int32_t | latitude | current latitude |
| int32_t | longitude | current longitude |
| uint16_t | posAcc | position accuracy |
| uint8_t | posSts | position status |

| | | |
|---|---|---|
| uint8_t | reserved[2] | reserve |
| int8_t | status | data status |

**struct SENSORLOCATION_MSG_LONLATINFO**

longitude and latitude information data delivery message

**Class Members:**

| | | |
|---|---|---|
| SENSORLOCATION_LONLATINFO_DAT | data | longitude and latitude information data |

**struct SENSORLOCATION_MSG_ALTITUDEINFO**

altitude information data delivery message

**Class Members:**

| | | |
|---|---|---|
| SENSORLOCATION_ALTITUDEINFO_DAT | data | altitude information data |

**struct SENSORMOTION_MSG_HEADINGINFO**

heading information data delivery message

**Class Members:**

| | | |
|---|---|---|
| SENSORMOTION_HEADINGINFO_DAT | data | heading information data |

**struct SENSORMOTION_MSG_SPEEDINFO**

speed information data delivery message

**Class Members:**

| | | |
|---|---|---|
| SENSORMOTION_SPEEDINFO_DAT | data | speed information data |

**struct POS_LOCATIONINFO_NMEA**

Structure of Location Information (NMEA)

**Class Members:**

| | | |
|---|---|---|
| uint8_t | data[LOCATIONINFO_NMEA_MAX] | data |
| uint32_t | length | length |

**struct SENSOR_MSG_SEND_DAT**

message for setting vehicle sensor data(to vehicle sensor)

**Class Members:**

| | | |
|---|---|---|
| uint8_t | data[502] | data body |
| DID | did | data ID |
| uint16_t | usSize | data size |

**struct SENSOR_PKG_MSG_VSINFO**

positioning sensor notification message (to User)

**Class Members:**

| | | |
|---|---|---|
| uint8_t | ucData[SENSOR_VSINFO_DSIZE] | data body |
| uint8_t | ucDataBreak | data lack infomation |
| uint8_t | ucDivideCnt | total partition |
| uint8_t | ucDivideSendCnt | partition transmit count |
| uint8_t | ucDNum | number of data |
| uint16_t | usOffset[SENSOR_PKG_DELIVERY_MAX] | offset |

**struct SENSOR_MSG_VSINFO**

message delivery positioning sensor information

**Class Members:**

| | | |
|---|---|---|
| uint8_t | data[SENSOR_MSG_VSINFO_DSIZE] | data body |
| DID | did | data ID |
| uint8_t | rcvFlag | reception flag |
| uint8_t | reserve | reserve |
| uint16_t | size | data size |

---

**Macro Definition Documentation**

**#define CID_POSIF_REGISTER_LISTENER_GPS_TIME_SET_REQ 0x0780**

GPS time setting result delivery command ID.

If you want to catch above envents, use NSFW like below.

```
1 l_eStatus = _CWORD33_AttachCallbackToDispatcher(test_params_->h_app,"POS_Main",
CID_POSIF_REGISTER_LISTENER_GPS_TIME_SET_REQ, CBCallbackA);
```

---

**Function Documentation**

**POS_RET_API POS_GetAltitude (HANDLE *hApp*, SENSORLOCATION_ALTITUDEINFO_DAT * *dat*, uint8_t *ucGetMethod*)**

**Brief**

Get altitude data

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| out | *dat* | SENSORLOCATION_ALTITUDEINFO_DAT* - output pointer to altitude data |
| in | *ucGetMethod* | uint8_t - Get method(GPS/Navi/Not specified) |

Get method(ucGetMethod)
SENSOR_GET_METHOD_GPS - GPS The altitude from GPS will be deliveried.
SENSOR_GET_METHOD_NAVI - Navi The altitude from Navi will be deliveried.
SENSOR_GET_METHOD_AUTO - Not specified The altitude will be deliveried according to the current environment
Avaliable method is descriped as following in each environment.
In the following environment, if the SENSOR_GET_METHOD_AUTO is set,treated as default
In the following environment, if the unsupported method has been specified, POS_RET_ERROR_PARAM will be returned.
*CWORD96*__*CWORD84*_ *SENSOR_GET_METHOD_GPS(default)*
_*CWORD95* /_CWORD101__CWORD84_ (unsupported)
*CWORD95* /_CWORD101__CWORD80_ SENSOR_GET_METHOD_GPS(default)
*CWORD95* /_CWORD101__CWORD84_ error(no *CWORD80* ) (unsupported)

SENSORLOCATION_ALTITUDEINFO_DAT structure

```
1 typedef struct {
2   uint8_t   getMethod;  /* get method */
3   uint8_t   SyncCnt;    /* sync count */
4   uint8_t   isEnable;   /* enable or not */
5   uint8_t   isExistDR;  /* DR exist or notnot used */
6   uint8_t   DRStatus    /* DR statusnot used */
7   uint8_t   Reserve[3]; /* Reserve */
8   int32_t   Altitude;   /* Altitude(unit0.01m) */
9 } SENSORLOCATION_ALTITUDEINFO_DAT;
```

Get method(getMethod)
SENSOR_GET_METHOD_GPS - altitude from GPS
SENSOR_GET_METHOD_NAVI - altitude from Navi
Synchrony count(SyncCnt)
Count for position data synchronous

When delivery altitude and heading data, position data can be synchronized by this count.
But the data of different method can not be synchronized by this count.
example 1: [longitude and latitude from GPS] and [heading from GPS] can be synchronized by the count.
example 2: [longitude and latitude from GPS] and [longitude and latitude from Navi] can not be
synchronized by the count.
Caution: The sensor count in sensor data delivery is another data.
Enable or not(isEnable)
To describe this data is whether enable or not.
0 - not avaliable
not 0 - avaliable
Altitude is invalid at following condition when GPS data specified, so [not avaliable] provieded
Immediately after system start, GPS unit has not received current location data and GPS unit
status is not positioning fix
If it is not initialization status, certainly provide [avaliable] when Navi data specified
If the status is [not avaliable], data following can not be guaranteed.
Altitude
altitude data(unit 0.01m)

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter dat is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* , and the parameter ucGetMethod is neither GPS
(SENSOR_GET_METHOD_GPS) nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD84* [POS_RET_ERROR_NOSUPPORT]

Hardware environment is *CWORD95__CWORD101___CWORD80* , and the parameter ucGetMethod is not GPS (SENSOR_GET_METHOD_GPS) or Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* or *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max [POS_RET_ERROR_INNER]
The event is created in system, but the count of reference is reach to max [POS_RET_ERROR_INNER]
Memory allocate falied during the event table creation for event registering. [POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]
The event table is full during event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed during event table creation. [POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, try to allocate the memory of the thread table creation, but failed. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed. [POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
The memory for storing semaphore control data allocate failed. [POS_RET_ERROR_INNER]
Specified semaphore ID has not been registered when lock semaphore. [POS_RET_ERROR_INNER]
Internal mutex HANDLE is NULL when lock mutex. [POS_RET_ERROR_INNER]
Internal mutex HANDLE has not been registered in mutex table when lock mutex. [POS_RET_ERROR_INNER]
The owner of specified mutex is not itself when lock mutex. [POS_RET_ERROR_INNER]
Mutex has been multiple locked [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address. [POS_RET_ERROR_INNER]
Can not get usable share memory address. [POS_RET_ERROR_INNER]
Memory allocate failed for share memory map. [POS_RET_ERROR_INNER]

Memory allocate failed for share memory management. [POS_RET_ERROR_INNER]
Can not open share memory. [POS_RET_ERROR_INNER]
Failed to mapping share memory. [POS_RET_ERROR_INNER]
No empty field in share memory. [POS_RET_ERROR_INNER]
Failed to guarantee share memory. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [POS_RET_ERROR_INNER]
The count of thread is reach to max in event management table during event waiting. [POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting. [POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address for accessing received data. [POS_RET_ERROR_INNER]
Can not get usable share memory address for accessing received data. [POS_RET_ERROR_INNER]
Memory for share memory map allocate failed for accessing received data. [POS_RET_ERROR_INNER]
Memory for share memory management allocate failed for accessing received data. [POS_RET_ERROR_INNER]
Can not open share memory for accessing received data. [POS_RET_ERROR_INNER]
Failed to mapping share memory for accessing received data. [POS_RET_ERROR_INNER]
The size of data stored in share memory is larger than the size of received data. [POS_RET_ERROR_INNER]
The result of getting sensor data is POS_RET_ERROR_RESOURCE. [POS_RET_ERROR_RESOURCE]
The result of getting sensor data is not POS_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Sync

**NAVIINFO_RET_API POS_GetGPSInfo (HANDLE *hApp*, NAVIINFO_DIAG_GPS * *navidiaginfo*)**

**Brief**

Get GPS data

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| out | *navidiaginfo* | NAVIINFO_DIAG_GPS* - pointer to get GPS data |

NAVIINFO_DIAG_GPS structure

```
1 typedef struct
2 {
3   NAVIINFO_DIAG_GPS_FIX stFix;  /* position fix information */
4   NAVIINFO_DIAG_GPS_SAT stSat;  /* all satellite information */
5 } NAVIINFO_DIAG_GPS;
```

NAVIINFO_DIAG_GPS_FIX structure

```
1 typedef struct
2 {
3   uint8_t               ucFixSts;     /* position fix status */
4   uint8_t               ucReserve[3]; /* reserve */
5   NAVIINFO_DIAG_GPS_FIX_CNT stCnt;       /* position fix count data */
6   NAVIINFO_DIAG_GPS_FIX_XYZ stWgs84;      /* lonlat data(WGS84 geodetic)  */
7 } NAVIINFO_DIAG_GPS_FIX;
```

position fix status(ucFixSts)
NAVIINFO_DIAG_GPS_FIX_STS_NONnot fixed
NAVIINFO_DIAG_GPS_FIX_STS_2D 2D fix
NAVIINFO_DIAG_GPS_FIX_STS_3D 3D fix

NAVIINFO_DIAG_GPS_FIX_CNT structure

```
1 typedef struct
2 {
3   uint32_t    ulCnt3d;    /* position fix count:3D(unitsec) */
4   uint32_t    ulCnt2d;    /* position fix count:2D(unitsec) */
5   uint32_t    ulCntElse;  /* position fix count:not fix(unitsec) */
```

```
6 } NAVIINFO_DIAG_GPS_FIX_CNT;
```

NAVIINFO_DIAG_GPS_FIX_XYZ structure

```
1 typedef struct
2 {
3   int32_t     lLat;    /* GPS latitude(unit1/256sec) (+: north latitude, -: south latitude) */
4   int32_t     lLon;    /* GPS longitude(unit1/256sec) (+: east longitude, -: west longitude) */
5 } NAVIINFO_DIAG_GPS_FIX_XYZ;
```

NAVIINFO_DIAG_GPS_SAT structure

```
1 typedef struct
2 {
3   NAVIINFO_DIAG_GPS_PRN  stPrn[12];  /* all satellite information */
4 } NAVIINFO_DIAG_GPS_SAT;
```

NAVIINFO_DIAG_GPS_PRN structure

```
1 typedef struct
2 {
3   uint8_t   ucRcvSts;     /* reception status */
4   uint8_t   ucPrn;        /* satellite No. */
5   uint8_t   ucelv;        /* satellite angle(unit1.0deg) */
6   uint8_t   ucLv;         /* satellite level */
7   uint16_t  usAzm;        /* satellite azimuth(unit1.0deg) (clockwise from north) */
8   uint8_t   ucReserve[2]; /* reserve */
9 } NAVIINFO_DIAG_GPS_PRN;
```

reception status(ucRcvSts)
NAVIINFO_DIAG_GPS_RCV_STS_NOTUSE : not used
NAVIINFO_DIAG_GPS_RCV_STS_SEARCHING : searching
NAVIINFO_DIAG_GPS_RCV_STS_TRACHING : tracking
NAVIINFO_DIAG_GPS_RCV_STS_NOTUSEFIX : not used for position fix
NAVIINFO_DIAG_GPS_RCV_STS_USEFIX : used for position fix

**Return values:**

| | |
|---|---|
| *NAVIINFO_RET_NORMAL* | normal finish |
| *NAVIINFO_RET_ERROR_PARAM* | parameter error |
| *NAVIINFO_RET_ERROR_INNER* | internal error |
| *NAVIINFO_RET_ERROR_NOSUPPORT* | unsupported |
| *NAVIINFO_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.)
of the dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter navidiaginfo is NULL [NAVIINFO_RET_ERROR_PARAM]
The parameter hApp is NULL [NAVIINFO_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD80* [NAVIINFO_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [NAVIINFO_RET_ERROR_RESOURCE]
The count of mutex is reach to max [NAVIINFO_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [NAVIINFO_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max [NAVIINFO_RET_ERROR_INNER]
The event is created in system, but the count of reference is reach to max [NAVIINFO_RET_ERROR_INNER]
Memory allocate falied during the event table creation for event registering. [NAVIINFO_RET_ERROR_INNER]
The thread can not register in the event table. [NAVIINFO_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation. [NAVIINFO_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [NAVIINFO_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [NAVIINFO_RET_ERROR_INNER]
The event table is full during event creation. [NAVIINFO_RET_ERROR_INNER]
The memory for event HANDLE allocate failed during event table creation. [NAVIINFO_RET_ERROR_INNER]
The thread can not be registered in event table. [NAVIINFO_RET_ERROR_INNER]
In event table, try to allocate the memory of the thread table creation, but failed. [NAVIINFO_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered. [NAVIINFO_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed. [NAVIINFO_RET_ERROR_INNER]
Specified event ID has not been registered in table. [NAVIINFO_RET_ERROR_INNER]
Initialize event object failed. [NAVIINFO_RET_ERROR_INNER]
The memory for storing semaphore control data allocate failed. [NAVIINFO_RET_ERROR_INNER]
Specified semaphore ID has not been registered when semaphore lock. [NAVIINFO_RET_ERROR_INNER]
Internal mutex HANDLE is NULL when mutex lock. [NAVIINFO_RET_ERROR_INNER]
Internal mutex HANDLE has not been registered in mutex table when mutex lock. [NAVIINFO_RET_ERROR_INNER]

The owner of specified mutex is not itself when mutex lock.
[NAVIINFO_RET_ERROR_INNER]
Mutex has been multiple locked [NAVIINFO_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address.
[NAVIINFO_RET_ERROR_INNER]
Can not get usable share memory address. [NAVIINFO_RET_ERROR_INNER]
Memory allocate failed for share memory map. [NAVIINFO_RET_ERROR_INNER]
Memory allocate failed for share memory management. [NAVIINFO_RET_ERROR_INNER]
Can not open share memory. [NAVIINFO_RET_ERROR_INNER]
Failed to mapping share memory. [NAVIINFO_RET_ERROR_INNER]
No empty field in share memory. [NAVIINFO_RET_ERROR_INNER]
Failed to guarantee share memory. [NAVIINFO_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message
transfered between processes. [NAVIINFO_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes.
[NAVIINFO_RET_ERROR_INNER]
Message transfer failed between processes. [NAVIINFO_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message
transfer between processes. [NAVIINFO_RET_ERROR_INNER]
The message queue name has not been registered in control table when message
transfer between processes. [NAVIINFO_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes.
[NAVIINFO_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered
between processes. [NAVIINFO_RET_ERROR_INNER]
Message transfer failed between processes. [NAVIINFO_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during
event waiting. [NAVIINFO_RET_ERROR_INNER]
The count of thread arrived max in event management table during event waiting.
[NAVIINFO_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [NAVIINFO_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting.
[NAVIINFO_RET_ERROR_INNER]
The flagID has not been registered during event getting. [NAVIINFO_RET_ERROR_INNER]
The interruption happened during event getting [NAVIINFO_RET_ERROR_INNER]
Whatever error happened during event getting. [NAVIINFO_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address for accessing
received data. [NAVIINFO_RET_ERROR_INNER]
Can not get usable share memory address for accessing received data.
[NAVIINFO_RET_ERROR_INNER]
Memory for share memory map allocate failed for accessing received data.
[NAVIINFO_RET_ERROR_INNER]
Memory for share memory management allocate failed for accessing received data.
[NAVIINFO_RET_ERROR_INNER]
Can not open share memory for accessing received data. [NAVIINFO_RET_ERROR_INNER]
Failed to mapping share memory for accessing received data.
[NAVIINFO_RET_ERROR_INNER]
The size of data stored in share memory is larger than the size of

received data. [NAVIINFO_RET_ERROR_INNER]
The result of sensor data get processing is POS_RET_ERROR_RESOURCE.
[NAVIINFO_RET_ERROR_RESOURCE]
The result of sensor data get processing is not POS_RET_ERROR_RESOURCE.
[NAVIINFO_RET_ERROR_INNER]

**Classification**

Public

**Type**

Sync

**POS_RET_API POS_GetGPStime (HANDLE *hApp*, SENSOR_GPSTIME * *dat*)**

**Brief**

Get GPS time

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| out | *dat* | SENSOR_GPSTIME* - output buffer pointer to store GPS time |

SENSOR_GPSTIME structure

```
1 typedef struct {
2   NAVIINFO_UTCTIME  utc;       /* UTC time */
3   uint8_t          tdsts;     /* date amd time status */
4   uint8_t          reserve[3]; /* reserve */
5 } SENSOR_GPSTIME;
```

date amd time status(tdsts)
0= time has not been adjusted after GPS receiver reset(time input or master reset or CSF start)
1= time output from RTC Backup(have time adjustment result)
2= time adjustment completed

NAVIINFO_UTCTIME structure

```
1 typedef struct {
2   uint16_t  year;     /* A.D.(1~) */
3   uint8_t   month;    /* month(1~12) */
4   uint8_t   date;     /* date(1~31) */
```

```
5   uint8_t   hour;     /* hour(0~23) */
6   uint8_t   minute;   /* minute(0~59) */
7   uint8_t   second;   /* second(0~59) */
8   uint8_t   reserved; /* not used */
9 } NAVIINFO_UTCTIME;
```

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter dat is NULL [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* and *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max [POS_RET_ERROR_INNER]
The event is created in system, but the count of reference is reach to max [POS_RET_ERROR_INNER]
Memory allocate falied during the event table creation for event registering. [POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]
The event table is full during event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed during event table creation. [POS_RET_ERROR_INNER]

The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, try to allocate the memory of the thread table creation, but failed.
[POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered.
[POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed.
[POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
The memory for storing semaphore control data allocate failed. [POS_RET_ERROR_INNER]
Specified semaphore ID has not been registered when semaphore lock.
[POS_RET_ERROR_INNER]
Internal mutex HANDLE is NULL when mutex lock. [POS_RET_ERROR_INNER]
Internal mutex HANDLE has not been registered in mutex table when mutex lock.
[POS_RET_ERROR_INNER]
The owner of specified mutex is not itself when mutex lock. [POS_RET_ERROR_INNER]
Mutex has been multiple locked [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address.
[POS_RET_ERROR_INNER]
Can not get usable share memory address. [POS_RET_ERROR_INNER]
Memory allocate failed for share memory map. [POS_RET_ERROR_INNER]
Memory allocate failed for share memory management. [POS_RET_ERROR_INNER]
Can not open share memory. [POS_RET_ERROR_INNER]
Failed to mapping share memory. [POS_RET_ERROR_INNER]
No empty field in share memory. [POS_RET_ERROR_INNER]
Failed to guarantee share memory. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message
transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message
transfer between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message
transfer between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered
between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during
event waiting. [POS_RET_ERROR_INNER]
The count of thread arrived max in event management table during event waiting.
[POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting.
[POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]

The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address for accessing received data. [POS_RET_ERROR_INNER]
Can not get usable share memory address for accessing received data. [POS_RET_ERROR_INNER]
Memory for share memory map allocate failed for accessing received data. [POS_RET_ERROR_INNER]
Memory for share memory management allocate failed for accessing received data. [POS_RET_ERROR_INNER]
Can not open share memory for accessing received data. [POS_RET_ERROR_INNER]
Failed to mapping share memory for accessing received data. [POS_RET_ERROR_INNER]
The size of data stored in share memory is larger than the size of received data. [POS_RET_ERROR_INNER]
The result of sensor data getting process is POS_RET_ERROR_RESOURCE. [POS_RET_ERROR_RESOURCE]
The result of sensor data getting process is not POS_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Sync

**POS_RET_API POS_GetGPSVersion (HANDLE *hApp*, uint8_t *buf_size*, int8_t * *buf*, uint8_t * *size*)**

**Brief**

Get GPS device version

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *buf_size* | uint8_t - size of the output buffer |
| out | *buf* | int8_t* - output buffer of storing GPS version |
| out | *size* | uint8_t* - the size of getting GPS version data |

**Return values:**

| | |
|---|---|
| *POS_RET_ERROR_NOSUPPORT* | unsupported |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.)
of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Classification**

Public

**Type**

Fire and Forget

**See also:**

None

**POS_RET_API POS_GetHeading (HANDLE *hApp*, SENSORMOTION_HEADINGINFO_DAT * *dat*, uint8_t *ucGetMethod*)**

**Brief**

Get heading data

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| out | *dat* | SENSORMOTION_HEADINGINFO_DAT* - output buffer pointer to store heading data |
| in | *ucGetMethod* | uint8_t - Get method(GPS/Navi/Not specified) |

Get method(ucGetMethod)
SENSOR_GET_METHOD_GPS - GPS The altitude from GPS will be deliveried.
SENSOR_GET_METHOD_NAVI - Navi The altitude from Navi will be deliveried.

SENSOR_GET_METHOD_AUTO - Not specified The altitude which *CWORD105* suitable in current environment
will be deliveried.
Avaliable method is descriped as following in each environment.
In corresponding environment, the SENSOR_GET_METHOD_AUTO is set as default.
In corresponding environment, if the unsupported method has been specified, POS_RET_ERROR_PARAM will be returned.
*CWORD96___CWORD84_* *SENSOR_GET_METHOD_GPS(default)*
*_CWORD95* /_CWORD101___CWORD84_ SENSOR_GET_METHOD_NAVI(default)
*CWORD95* /_CWORD101___CWORD80_ SENSOR_GET_METHOD_GPS / SENSOR_GET_METHOD_NAVI(default)
*CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ) SENSOR_GET_METHOD_NAVI(default)
Please note that although it is possible to register delivery, but the data will not be deliveried in environment *CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ) .

SENSORMOTION_HEADINGINFO_DAT structure

```
typedef struct {
uint8_t   getMethod;    /* get method */
uint8_t   SyncCnt;      /* sync count */
uint8_t   isEnable;     /* enable or not */
uint8_t   isExistDR;    /* DR exist or notnot used */
uint8_t   DRStatus      /* DR statusnot used */
uint8_t   posSts;       /* position status */
uint8_t   Reserve1[2];  /* Reserve */
uint16_t  Heading;      /* heading(unit0.01degree) */
uint8_t   Reserve2[2];  /* Reserve */
} SENSORMOTION_HEADINGINFO_DAT;
```

Get method(getMethod)
SENSOR_GET_METHOD_GPS - Heading from GPS
SENSOR_GET_METHOD_NAVI - Heading from Navi
Synchrony count(SyncCnt)
Count for position data synchronous
When delivery altitude and heading data, position data can be synchronized by this count.
But the data of different method can not be synchronized by this count.
example 1: [longitude and latitude from GPS] and [heading from GPS] can be synchronized by the count.
example 2: [longitude and latitude from GPS] and [longitude and latitude from Navi] can not be
synchronized by the count.
Caution: The sensor count in sensor data delivery is another data.
Enable or not(isEnable)
To describe this data is whether enable or not.
0 - not avaliable
not 0 - avaliable

Heading is invalid at following condition when GPS data specified, so [not avaliable] provieded
Immediately after system start, GPS unit has not received current location data and GPS unit
status is not positioning fix
If it is not initialization status, certainly provide [avaliable] when Navi data specified
If the status is [not avaliable], data following can not be guaranteed.
Position status(posSts)
It is valid only when "Get method is Navi" and "evironment is _CWORD80_".otherwise it will be set as 0
Bit0 : GPS data used result(1:used,0:not used, definition of maskbit:POS_LOC_INFO_USE_GSP)
Bit1 : DGPS data used result(1:used,0:not used, definition of maskbit:POS_LOC_INFO_USE_DGPS)
Bit2 : DR data used result(1:used,0:not used, definition of maskbit:POS_LOC_INFO_USE_DR)
Bit3 : MapMatching data use result(1:used,0:not used, definition of maskbit:POS_LOC_INFO_USE_MAPMATCHING)
Heading
Heading data(unit 0.01degree, based on north and clockwise count) But in environment *CWORD95* /_CWORD101___CWORD84_,the heading from Navi is a approximate that separate 360 degree into 16 part.
example: 0, 2300, 4500, ..., 31500, 33800

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher
for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter dat is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* , and the parameter ucGetMethod is neither GPS

(SENSOR_GET_METHOD_GPS) nor Auto(SENSOR_GET_METHOD_AUTO)
[POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD84* , and the parameter ucGetMethod is neither Navi
(SENSOR_GET_METHOD_NAVI) nor Auto(SENSOR_GET_METHOD_AUTO)
[POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD80* , and the parameter ucGetMethod is not GPS
(SENSOR_GET_METHOD_GPS) and Navi(SENSOR_GET_METHOD_NAVI) and Auto
(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* and *CWORD95__CWORD101___CWORD84* and *CWORD95__CWORD101___CWORD80*
[POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max
[POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max
[POS_RET_ERROR_INNER]
The event is created in system, but the count of reference is reach to max
[POS_RET_ERROR_INNER]
Memory allocate falied during the event table creation for event registering.
[POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation.
[POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed.
[POS_RET_ERROR_INNER]
The event table is full during event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed during event table creation.
[POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, try to allocate the memory of the thread table creation, but failed.
[POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered.
[POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed.
[POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
The memory for storing semaphore control data allocate failed. [POS_RET_ERROR_INNER]
Specified semaphore ID has not been registered when semaphore lock.
[POS_RET_ERROR_INNER]
Internal mutex HANDLE is NULL when mutex lock. [POS_RET_ERROR_INNER]
Internal mutex HANDLE has not been registered in mutex table when mutex lock.
[POS_RET_ERROR_INNER]

The owner of specified mutex is not itself when mutex lock. [POS_RET_ERROR_INNER]
Mutex has been multiple locked [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address.
[POS_RET_ERROR_INNER]
Can not get usable share memory address. [POS_RET_ERROR_INNER]
Memory allocate failed for share memory map. [POS_RET_ERROR_INNER]
Memory allocate failed for share memory management. [POS_RET_ERROR_INNER]
Can not open share memory. [POS_RET_ERROR_INNER]
Failed to mapping share memory. [POS_RET_ERROR_INNER]
No empty field in share memory. [POS_RET_ERROR_INNER]
Failed to guarantee share memory. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer
between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between
processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting[POS_RET_ERROR_INNER]
The count of thread is reach to max in event management table during event waiting.
[POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting.
[POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address for accessing received data. [POS_RET_ERROR_INNER]
Can not get usable share memory address for accessing received data.
[POS_RET_ERROR_INNER]
Memory for share memory map allocate failed for accessing received data.
[POS_RET_ERROR_INNER]
Memory for share memory management allocate failed for accessing received data.
[POS_RET_ERROR_INNER]
Can not open share memory for accessing received data. [POS_RET_ERROR_INNER]
Failed to mapping share memory for accessing received data. [POS_RET_ERROR_INNER]

The size of data stored in share memory is larger than the size of received data.
[POS_RET_ERROR_INNER]
The result of getting sensor data is POS_RET_ERROR_RESOURCE.
[POS_RET_ERROR_RESOURCE]
The result of getting sensor data is not POS_RET_ERROR_RESOURCE.
[POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Sync

**POS_RET_API POS_GetLonLat (HANDLE *hApp*, SENSORLOCATION_LONLATINFO_DAT * *dat*, uint8_t *ucGetMethod*)**

**Brief**

Get longitude and latitude information

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| out | *dat* | SENSORLOCATION_LONLATINFO_DAT* - output pointer to longitude and latitude informaiton |
| in | *ucGetMethod* | uint8_t - Get method(GPS/Navi/Not specified) |

Get method(ucGetMethod)
SENSOR_GET_METHOD_GPS - GPS The longitude and latitude from GPS will be deliveried.
SENSOR_GET_METHOD_NAVI - Navi The longitude and latitude from Navi will be deliveried.
SENSOR_GET_METHOD_AUTO - Not specified The longitude and latitude will be deliveried
according to the current environment.
Avaliable method is descriped as following in each environment.
In the following environment, if the SENSOR_GET_METHOD_AUTO is set,treated as default
In the following environment, if the unsupported method has been specified,
POS_RET_ERROR_PARAM will be returned.
*CWORD96___CWORD84_ SENSOR_GET_METHOD_GPS(default)*

*_CWORD95* /_CWORD101___CWORD84_ SENSOR_GET_METHOD_NAVI(default)
*CWORD95* /_CWORD101___CWORD80_ SENSOR_GET_METHOD_GPS /
SENSOR_GET_METHOD_NAVI(default)
*CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* )
SENSOR_GET_METHOD_NAVI(default)
Please note that although it is possible to get longitude and latitude, but the obtained longitude and latitude will be the initial value in *CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ).

SENSORLOCATION_LONLATINFO_DAT structure

```
typedef struct {
uint8_t   getMethod;  /* get method */
uint8_t   SyncCnt;    /* sync count */
uint8_t   isEnable;   /* enable or not */
uint8_t   isExistDR;  /* DR exist or notnot used */
uint8_t   DRStatus    /* DR statusnot used */
uint8_t   posSts;     /* position status */
uint16_t  posAcc;     /* Position accuracy */
int32_t   Longitude;  /* Longitude */
int32_t   Latitude;   /* Latitude */
} SENSORLOCATION_LONLATINFO_DAT;
```

Get method(getMethod)
SENSOR_GET_METHOD_GPS - longitude and latitude from GPS
SENSOR_GET_METHOD_NAVI - longitude and latitude from Navi
Synchrony count(SyncCnt)
Count for position data synchronous
When delivery altitude and heading data, position data can be synchronized by this count.
But the data of different method can not be synchronized by this count.
example 1: [longitude and latitude from GPS] and [heading from GPS] can be synchronized by this count.
example 2: [longitude and latitude from GPS] and [longitude and latitude from Navi] can not be
synchronized by this count.
Caution: The sensor count in sensor data delivery is another data.
Enable or not(isEnable)
To describe this data is whether enable or not.
0 - not avaliable
not 0 - avaliable
longitude and latitude is invalid at following condition when GPS data specified,
so [not avaliable] provieded
Immediately after system start, GPS unit has not received current location data and
GPS unit status is not positioning fix
If it is not initialization status, certainly provide [avaliable] when Navi data specified
If the status is [not avaliable], data following can not be guaranteed.
Position status(posSts)
It is valid only when "Get method is Navi" and "evironment is _CWORD80_".otherwise it will be set as 0

Bit0 : GPS data used result(1:used,0:not used, definition of
maskbit:POS_LOC_INFO_USE_GSP)
Bit1 : DGPS data used result(1:used,0:not used, definition of
maskbit:POS_LOC_INFO_USE_DGPS)
Bit2 : DR data used result(1:used,0:not used, definition of
maskbit:POS_LOC_INFO_USE_DR)
Bit3 : MapMatching data used result(1:used,0:not used, definition of
maskbit:POS_LOC_INFO_USE_MAPMATCHING)
Position accuracy(posAcc)
Detected accruray of current position1LSB=1m
It is valid only when "Get method is Navi" and "evironment is _CWORD80_".otherwise it
will be set as 0
0000H0m
0001H1m
FFFDH65533m
FFFEH65534m and larger than 65534m
FFFFHno data
Longitude : (WGS-84)(10-7degree as 1)
East longitude is positive value and west longitude is minus value.
Latitude : (WGS-84)(10-7degree as 1)
North latitudeis positive value and south latitude is minus value.

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the
Dispatcher
for App are completed.
Availability of positioning service is TRUE.

**change of internal status**

There is no change of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter dat is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* , and the parameter ucGetMethod is
neither GPS(SENSOR_GET_METHOD_GPS)
nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD84* , and the parameter
ucGetMethod is neither Navi(SENSOR_GET_METHOD_NAVI)
nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]

Hardware environment is *CWORD95__CWORD101___CWORD80* , and the parameter ucGetMethod is not GPS(SENSOR_GET_METHOD_GPS) or Navi(SENSOR_GET_METHOD_NAVI) or Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]

Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* or *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]

The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]

The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]

The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]

The event is created in same process, but the count of reference is reach to max [POS_RET_ERROR_INNER]

The event is created in system, but the count of reference is reach to max [POS_RET_ERROR_INNER]

Memory allocate failed during the event table creation for event registering. [POS_RET_ERROR_INNER]

The thread can not register in the event table. [POS_RET_ERROR_INNER]

Memory allocate failed in event table during the thread table creation. [POS_RET_ERROR_INNER]

After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]

After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]

The event table is full during event creation. [POS_RET_ERROR_INNER]

The memory for event HANDLE allocate failed during event table creation. [POS_RET_ERROR_INNER]

The thread can not be registered in event table. [POS_RET_ERROR_INNER]

In event table, try to allocate the memory of the thread table creation, but failed. [POS_RET_ERROR_INNER]

After register the thread in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]

After register the thread in event table, the event flag register failed. [POS_RET_ERROR_INNER]

Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]

Initialize event object failed. [POS_RET_ERROR_INNER]

The memory for storing semaphore control data allocate failed. [POS_RET_ERROR_INNER]

Specified semaphore ID has not been registered when lock semaphore. [POS_RET_ERROR_INNER]

Internal mutex HANDLE is NULL when lock mutex. [POS_RET_ERROR_INNER]

Internal mutex HANDLE has not been registered in mutex table when lock mutex. [POS_RET_ERROR_INNER]

The owner of specified mutex is not itself when lock mutex. [POS_RET_ERROR_INNER]

Mutex has been multiple locked [POS_RET_ERROR_INNER]

The HANDLE is NULL when getting usable share memory address. [POS_RET_ERROR_INNER]

Can not get usable share memory address. [POS_RET_ERROR_INNER]

Memory allocate failed for share memory map. [POS_RET_ERROR_INNER]

Memory allocate failed for share memory management. [POS_RET_ERROR_INNER]
Can not open share memory. [POS_RET_ERROR_INNER]
Failed to mapping share memory. [POS_RET_ERROR_INNER]
No empty field in share memory. [POS_RET_ERROR_INNER]
Failed to guarantee share memory. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer
between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer failed between process. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting.
[POS_RET_ERROR_INNER]
The count of thread is reach to max in event management table during event waiting.
[POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting.
[POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address for accessing received data.
[POS_RET_ERROR_INNER]
Can not get usable share memory address for accessing received data.
[POS_RET_ERROR_INNER]
Memory for share memory map allocate failed for accessing received data.
[POS_RET_ERROR_INNER]
Memory for share memory management allocate failed for accessing received data.
[POS_RET_ERROR_INNER]
Can not open share memory for accessing received data. [POS_RET_ERROR_INNER]
Failed to mapping share memory for accessing received data. [POS_RET_ERROR_INNER]
The size of data stored in share memory is larger than the size of received data.
[POS_RET_ERROR_INNER]
The result of getting sensor data get is POS_RET_ERROR_RESOURCE.
[POS_RET_ERROR_RESOURCE]

The result of getting sensor data get is not POS_RET_ERROR_RESOURCE.
[POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Sync

**POS_RET_API POS_GetSensData (HANDLE *hApp*, DID *ulDid*, void * *pDestData*, uint16_t *usDestSize*)**

**Brief**

Get vehicle sensor data.

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *ulDid* | DID - Data ID of vehicle info |
| out | *pDestData* | void* - pointer of buffer for storing vehicle sensor data |
| in | *usDestSize* | uint16_t - vehicle sensor data buffer size |

Data ID of vehicle info(ulDid)
POS_DID_SPEED_PULSE-speed pulse(count of pulse)
POS_DID_GYRO -gyro output16bit A/D value
POS_DID_GSNS_X -Gsensor output (X axis)12bit A/D value(left justified:154bit)
POS_DID_GSNS_Y -Gsensor output (Y axis)12bit A/D value(left justified:154bit)
POS_DID_GPS_ANTENNA -GPS antenna connection status
POS_DID_GPS__CWORD44__NMEA -GPS NMEA_CWORD44_
POS_DID_GPS__CWORD44__FULLBINARY -GPS *CWORD44* full binary_CWORD44_
POS_DID_GPS_NMEA -GPS NMEA
POS_DID_GYRO_TEMP -gyro temperature
POS_DID_GPS_CLOCK_DRIFT -GPS clock drift[ns/s]
POS_DID_GPS_CLOCK_FREQ -GPS clock frequency[Hz]
The avaliable data ID of each hardware type is as following.

- The Gsensor output is 0 in the environment without Gsensor hardware.

vehicle sensor data buffer size(usDestSize)

Please note it is the size of output buffer, not the size of data.

**Return values:**

| | |
|---|---|
| *more* | than 0 data size |
| *POS_RET_ERROR_CREATE_EVENT* | event create failed |
| *POS_RET_ERROR_OUTOF_MEMORY* | share memory guarantee failed |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_SIZE* | buffer size error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**change of internal status**

There is no change of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter pDestData is NULL [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* and *CWORD95__CWORD101___CWORD84* and *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The parameter ulDid is not avaliable value in current hardware environment [POS_RET_ERROR_PARAM]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max [POS_RET_ERROR_CREATE_EVENT]
The event is created in system, but the count of reference is reach to max [POS_RET_ERROR_CREATE_EVENT]
Memory allocate falied during the event table creation for event registering. [POS_RET_ERROR_CREATE_EVENT]
The thread can not register in the event table. [POS_RET_ERROR_CREATE_EVENT]
Memory allocate failed in event table during the thread table creation. [POS_RET_ERROR_CREATE_EVENT]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_CREATE_EVENT]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_CREATE_EVENT]
The event table is full during event creation. [POS_RET_ERROR_CREATE_EVENT]
The memory for event HANDLE allocate failed during event table creation. [POS_RET_ERROR_CREATE_EVENT]
The thread can not be registered in event table. [POS_RET_ERROR_CREATE_EVENT]

In event table, to allocate the memory of the thread table creation,
but failed. [POS_RET_ERROR_CREATE_EVENT]
After register the thread in event table, the event flag has already been
registered. [POS_RET_ERROR_CREATE_EVENT]
After register the thread in event table, the event flag register failed.
[POS_RET_ERROR_CREATE_EVENT]
Specified event ID has not been registered in table. [POS_RET_ERROR_CREATE_EVENT]
Initialize event object failed. [POS_RET_ERROR_CREATE_EVENT]
There is no empty field in semaphore table for semaphore creation
[POS_RET_ERROR_OUTOF_MEMORY]
The memory for storing semaphore control data allocate failed.
[POS_RET_ERROR_OUTOF_MEMORY]
Specified semaphore ID has not been registered when semaphore lock.
[POS_RET_ERROR_OUTOF_MEMORY]
Internal mutex HANDLE is NULL when mutex lock. [POS_RET_ERROR_OUTOF_MEMORY]
Internal mutex HANDLE has not been registered in mutex table when mutex lock.
[POS_RET_ERROR_OUTOF_MEMORY]
The owner of specified mutex is not itself when mutex lock.
[POS_RET_ERROR_OUTOF_MEMORY]
Mutex has been multiple locked [POS_RET_ERROR_OUTOF_MEMORY]
The HANDLE is NULL when getting usable share memory address.
[POS_RET_ERROR_OUTOF_MEMORY]
Can not get usable share memory address. [POS_RET_ERROR_OUTOF_MEMORY]
Memory allocate failed for share memory map. [POS_RET_ERROR_OUTOF_MEMORY]
Memory allocate failed for share memory management.
[POS_RET_ERROR_OUTOF_MEMORY]
Can not open share memory. [POS_RET_ERROR_OUTOF_MEMORY]
Failed to mapping share memory. [POS_RET_ERROR_OUTOF_MEMORY]
No empty field in share memory. [POS_RET_ERROR_OUTOF_MEMORY]
ProcessNo has not been registered in message control table when message
transfered between processes. [POS_RET_ERROR_CREATE_EVENT]
Message transfer HANDLE get failed when message transfered between processes.
[POS_RET_ERROR_CREATE_EVENT]
Message transfer failed between processes. [POS_RET_ERROR_CREATE_EVENT]
The destination process name size is larger than 20 characters when message
transfer between processes. [POS_RET_ERROR_CREATE_EVENT]
The message queue name has not been registered in control table when message
transfer between processes. [POS_RET_ERROR_CREATE_EVENT]
Message transfer HANDLE create failed when message transfered between processes.
[POS_RET_ERROR_CREATE_EVENT]
Message transfer HANDLE get failed from internal table when message transfered
between processes. [POS_RET_ERROR_CREATE_EVENT]
Message transfer failed between processes. [POS_RET_ERROR_CREATE_EVENT]
Specified event HANDLE has not been registered in event HANDLE table during
event waiting. [POS_RET_ERROR_INNER]
The count of thread is reach to max in event management table during event waiting.
[POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event waiting. [POS_RET_ERROR_INNER]

The ID of message event queue has not been created during event waiting. [POS_RET_ERROR_INNER]
The flagID has not been registered during event waiting. [POS_RET_ERROR_INNER]
The interruption happened during event waiting [POS_RET_ERROR_INNER]
Whatever error happened during event waiting. [POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during getting event. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during getting event. [POS_RET_ERROR_INNER]
The flagID has not been registered during getting event. [POS_RET_ERROR_INNER]
The interruption happened during getting event [POS_RET_ERROR_INNER]
Whatever error happened during getting event. [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address for accessing received data. [POS_RET_ERROR_OUTOF_MEMORY]
Can not get usable share memory address for accessing received data. [POS_RET_ERROR_OUTOF_MEMORY]
Memory for share memory map allocate failed for accessing received data. [POS_RET_ERROR_OUTOF_MEMORY]
Memory for share memory management allocate failed for accessing received data. [POS_RET_ERROR_OUTOF_MEMORY]
Can not open share memory for accessing received data. [POS_RET_ERROR_OUTOF_MEMORY]
Failed to mapping share memory for accessing received data. [POS_RET_ERROR_OUTOF_MEMORY]
Can not get share memory normally [POS_RET_ERROR_OUTOF_MEMORY]
The size of data stored in share memory is larger than the size of received data. [POS_RET_ERROR_SIZE]

**Detial**

Call this API to get vehicle sensor data.
This vehicle sensor data stored in the output buffer of the parameter, this API return.

**Classification**

Public

**Type**

Sync

**POS_RET_API POS_GetSpeed (HANDLE *hApp*, SENSORMOTION_SPEEDINFO_DAT * *dat*, uint8_t *ucGetMethod*)**

**Brief**

Get speed data

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | |

| | | |
|---|---|---|
| | | HANDLE - App Handle |
| out | *dat* | SENSORMOTION_SPEEDINFO_DAT* - output buffer pointer to store speed data |
| in | *ucGetMethod* | uint8_t - Get method(POS/Navi/Not specified) |

Get method(ucGetMethod)
SENSOR_GET_METHOD_POS - The speed calculated in positioning based on speed pulse will be deliveried.
SENSOR_GET_METHOD_NAVI - Navi The speed from Navi will be deliveried.
SENSOR_GET_METHOD_AUTO - Not specified The speed will be deliveried according to current
environment.
Avaliable method is descriped as following in each environment.
In the following environment, if the SENSOR_GET_METHOD_AUTO is set,treated as default
In the following environment, if the unsupported method has been specified,
POS_RET_ERROR_PARAM will be returned.
*CWORD96___CWORD84_ SENSOR_GET_METHOD_POS(default)*
*_CWORD95* /_CWORD101___CWORD84_ SENSOR_GET_METHOD_POS(default)
*CWORD95* /_CWORD101___CWORD80_ SENSOR_GET_METHOD_POS / SENSOR_GET_METHOD_NAVI(default)
*CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ) SENSOR_GET_METHOD_POS(default)

SENSORMOTION_SPEEDINFO_DAT structure

```
1 typedef struct {
2 uint8_t   getMethod;    /* get method */
3 uint8_t   SyncCnt;      /* sync count */
4 uint8_t   isEnable;     /* enable or not */
5 uint8_t   isExistDR;    /* DR exist or notnot used */
6 uint8_t   Reserve1[3];  /* Reserve */
7 uint8_t   DRStatus      /* DR statusnot used */
8 uint16_t  Speed;        /* speed(unit0.01m/sec) */
9 uint8_t   Reserve2[2];  /* Reserve */
10 } SENSORMOTION_SPEEDINFO_DAT;
```

Get method(getMethod)
SENSOR_GET_METHOD_POS - The speed calculated in positioning based on speed pulse will be deliveried.
SENSOR_GET_METHOD_NAVI - speed from Navi
Sync count(SyncCnt)
0.

Enable or not(isEnable)
To describe this data is whether enable or not
0 - not avaliable
not 0 - avaliable
Speed is invalid at following condition when speed pulse specified, so [not avaliable] provieded
Immediately after system start, the sensor data have not been received from SYS micon
If it is not initialization status, certainly provide [avaliable] when Navi data specified
If the status is [not avaliable], data following can not be guaranteed.
Speed
speed data(unit 0.01m/sec)

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter dat is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* , and the parameter ucGetMethod is neither POS (SENSOR_GET_METHOD_POS) nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD84* , and the parameter ucGetMethod is neither POS (SENSOR_GET_METHOD_POS) nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD80* , and the parameter ucGetMethod is not POS (SENSOR_GET_METHOD_POS) or Navi(SENSOR_GET_METHOD_NAVI) or Auto (SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* or *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]

The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max [POS_RET_ERROR_INNER]
The event is created in system, but the count of reference is reach to max [POS_RET_ERROR_INNER]
Memory allocate falied when the event table creation that for event registering. [POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table when the thread table creation. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]
The event table is full when event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed when event table creation. [POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, to allocate the memory of the thread table creation, but failed. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed. [POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
The memory for storing semaphore control data allocate failed. [POS_RET_ERROR_INNER]
Specified semaphore ID has not been registered when semaphore lock. [POS_RET_ERROR_INNER]
Internal mutex HANDLE is NULL when mutex lock. [POS_RET_ERROR_INNER]
Internal mutex HANDLE has not been registered in mutex table when mutex lock. [POS_RET_ERROR_INNER]
The owner of specified mutex is not itself when mutex lock. [POS_RET_ERROR_INNER]
Mutex has been multiple locked [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address. [POS_RET_ERROR_INNER]
Can not get usable share memory address. [POS_RET_ERROR_INNER]
Memory allocate failed for share memory map. [POS_RET_ERROR_INNER]
Memory allocate failed for share memory management. [POS_RET_ERROR_INNER]
Can not open share memory. [POS_RET_ERROR_INNER]
Failed to mapping share memory. [POS_RET_ERROR_INNER]
No empty field in share memory. [POS_RET_ERROR_INNER]
Failed to guarantee share memory. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes. [POS_RET_ERROR_INNER]

Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer between processes [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [POS_RET_ERROR_INNER]
The count of thread is reach to max in event management table during event waiting. [POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting. [POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
The HANDLE is NULL when getting usable share memory address for accessing received data. [POS_RET_ERROR_INNER]
Can not get usable share memory address for accessing received data. [POS_RET_ERROR_INNER]
Memory for share memory map allocate failed for accessing received data. [POS_RET_ERROR_INNER]
Memory for share memory management allocate failed for accessing received data. [POS_RET_ERROR_INNER]
Can not open share memory for accessing received data. [POS_RET_ERROR_INNER]
Failed to mapping share memory for accessing received data. [POS_RET_ERROR_INNER]
The size of data stored in share memory is larger than the size of received data. [POS_RET_ERROR_INNER]
The result of getting sensor data is POS_RET_ERROR_RESOURCE. [POS_RET_ERROR_RESOURCE]
The result of getting sensor data is not POS_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Sync

**POS_RET_API POS_RegisterListenerAltitude (HANDLE *hApp*, PCSTR *notifyName*, uint8_t *ucCtrlFlg*, uint8_t *ucDeliveryTiming*, uint8_t *ucGetMethod*)**

**Brief**

Register altitude delivery

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *notifyName* | PCSTR - Destination thread name |
| in | *ucCtrlFlg* | uint8_t - Delivery control flag(register) |
| in | *ucDeliveryTiming* | uint8_t - Delivery timing(change/update) |
| in | *ucGetMethod* | uint8_t - Get method(GPS/Navi/Not specified) |

Delivery control flag(ucCtrlFlg)
SENSOR_DELIVERY_REGIST - register
Register specified altitude delivery
Please note that if the same data delivery has been registered for multiple times, the data will
also be deliveried for registered multiple times.
The specified altitude will be deliveried at registered time first delivery.
Delivery timing(ucDeliveryTiming)
SENSOR_DELIVERY_TIMING_CHANGE - change. Specified altitude will be deliveried only when it is changed.
SENSOR_DELIVERY_TIMING_UPDATE - update. Specified altitude will be deliveried when it is updated by
vehicle sensor.
Get method(ucGetMethod)
SENSOR_GET_METHOD_GPS - GPS. The altitude from GPS will be deliveried.
SENSOR_GET_METHOD_NAVI - Navi. The altitude from Navigation will be deliveried.
SENSOR_GET_METHOD_AUTO - Not specified. The altitude will be deliveried according to the current environment
Avaliable method is descriped as following in each environment.
In the following environment, if the SENSOR_GET_METHOD_AUTO is set,treated as default

In the following environment, if the unsupported method has been specified, POS_RET_ERROR_PARAM will be returned.
*CWORD96___CWORD84_ SENSOR_GET_METHOD_GPS(default)*
*_CWORD95* /_CWORD101___CWORD84_ (unsupported)
*CWORD95* /_CWORD101___CWORD80_ SENSOR_GET_METHOD_GPS(default)
*CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ) (unsupported)

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_BUFFULL* | the register count is full |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter ucDeliveryTiming is neither update(SENSOR_DELIVERY_TIMING_UPDATE) nor change (SENSOR_DELIVERY_TIMING_CHANGE) [POS_RET_ERROR_PARAM]
The parameter ucCtrlFlg is not registered(SENSOR_DELIVERY_REGIST) [POS_RET_ERROR_PARAM]
The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter notifyName is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* , and the parameter ucGetMethod is neither GPS(SENSOR_GET_METHOD_GPS) nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD84* [POS_RET_ERROR_NOSUPPORT]
Hardware environment is *CWORD95__CWORD101___CWORD80* , and the parameter ucGetMethod is not GPS(SENSOR_GET_METHOD_GPS) or Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* or *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reached to the max [POS_RET_ERROR_RESOURCE]
The count of mutex is reached to the max [POS_RET_ERROR_RESOURCE]

The count of item in ProcessName-ProcessNo convert table is reached to the max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference to the event is reach to max [POS_RET_ERROR_INNER]
The event is created in system, but the count of reference to the event is reach to the max [POS_RET_ERROR_INNER]
Memory allocate failed during the event table creation for that event registering. [POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]
The event table is full during event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed during event table creation. [POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, try to allocate the memory of the thread table creation, but failed. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed. [POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [POS_RET_ERROR_INNER]

The count of thread is reach to max in event management table during event waiting.[POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting. [POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
Getting event timeout. [POS_RET_ERROR_INNER]
Error happened during event getting. [POS_RET_ERROR_INNER]
The result of positioning service is SENSOR_RET_ERROR_PARAM. [POS_RET_ERROR_PARAM]
The result of positioning service is SENSOR_RET_ERROR_BUFFULL. [POS_RET_ERROR_BUFFULL]
The result of positioning service is SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_RESOURCE]
The result of positioning service is not SENSOR_RET_ERROR_PARAM and SENSOR_RET_ERROR_BUFFULL and SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Method

**SENSOR_RET_API POS_RegisterListenerGPStime (HANDLE *hApp*, PCSTR *notifyName*, uint8_t *ucCtrlFlg*, uint8_t *ucDeliveryTiming*)**

**Brief**

Register GPS time delivery

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *notifyName* | PCSTR - Destination thread name |
| in | *ucCtrlFlg* | uint8_t - Delivery control flag(register) |
| in | *ucDeliveryTiming* | uint8_t - Delivery timing(change/update) |

Delivery control flag(ucCtrlFlg)
SENSOR_DELIVERY_REGIST - register
Register specified data delivery
Please note that if the same data delivery has been registered for multiple times, the data will also be deliveried for registered multiple times.
The specified GPS time information will be deliveried at registered time first delivery.
Delivery timing(ucDeliveryTiming)
SENSOR_DELIVERY_TIMING_CHANGE - change Specified data will be deliveried only when it changed.
SENSOR_DELIVERY_TIMING_UPDATE - update Specified data will be deliveried as long as it updated by vehicle sensor.

**Return values:**

| | |
|---|---|
| *SENSOR_RET_NORMAL* | normal finish |
| *SENSOR_RET_ERROR_CREATE_EVENT* | event creation error |
| *SENSOR_RET_ERROR_PARAM* | parameter error |
| *SENSOR_RET_ERROR_INNER* | internal error |
| *SENSOR_RET_ERROR_NOSUPPORT* | unsupported |
| *SENSOR_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [SENSOR_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* or *CWORD95__CWORD101___CWORD80* [SENSOR_RET_ERROR_NOSUPPORT]
The parameter notifyName is NULL [SENSOR_RET_ERROR_PARAM]
The parameter ucCtrlFlg is not register(SENSOR_DELIVERY_REGIST) [SENSOR_RET_ERROR_PARAM]
The parameter ucDeliveryTiming is neither update(SENSOR_DELIVERY_TIMING_UPDATE) nor change(SENSOR_DELIVERY_TIMING_CHANGE) [SENSOR_RET_ERROR_PARAM]
The count of message in message queue is reach to max [SENSOR_RET_ERROR_RESOURCE]
The count of mutex is reach to max [SENSOR_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [SENSOR_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max [SENSOR_RET_ERROR_CREATE_EVENT]

The event is created in system, but the count of reference is reach to max [SENSOR_RET_ERROR_CREATE_EVENT]
Memory allocate falied during the event table creation for event registering. [SENSOR_RET_ERROR_CREATE_EVENT]
The thread can not register in the event table. [SENSOR_RET_ERROR_CREATE_EVENT]
Memory allocate failed in event table during the thread table creation. [SENSOR_RET_ERROR_CREATE_EVENT]
After register the thread table in event table, the event flag has already been registered. [SENSOR_RET_ERROR_CREATE_EVENT]
After register the thread table in event table, the event flag register failed. [SENSOR_RET_ERROR_CREATE_EVENT]
The event table is full during event creation. [SENSOR_RET_ERROR_CREATE_EVENT]
The memory for event HANDLE allocate failed during event table creation. [SENSOR_RET_ERROR_CREATE_EVENT]
The thread can not be registered in event table. [SENSOR_RET_ERROR_CREATE_EVENT]
In event table, try to allocate the memory of the thread table creation, but failed. [SENSOR_RET_ERROR_CREATE_EVENT]
After register the thread in event table, the event flag has already been registered. [SENSOR_RET_ERROR_CREATE_EVENT]
After register the thread in event table, the event flag register failed. [SENSOR_RET_ERROR_CREATE_EVENT]
Specified event ID has not been registered in table. [SENSOR_RET_ERROR_CREATE_EVENT]
Initialize event object failed. [SENSOR_RET_ERROR_CREATE_EVENT]
ProcessNo has not been registered in message control table when message transfered between processes. [SENSOR_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes. [SENSOR_RET_ERROR_INNER]
Message transfer failed between processes. [SENSOR_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes. [SENSOR_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer between processes. [SENSOR_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes. [SENSOR_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes. [SENSOR_RET_ERROR_INNER]
Message transfer failed between processes. [SENSOR_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [SENSOR_RET_ERROR_INNER]
The count of thread arrived max in event management table during event waiting. [SENSOR_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event get. [SENSOR_RET_ERROR_INNER]
The ID of message event queue has not been created during event get. [SENSOR_RET_ERROR_INNER]
The flagID has not been registered during event get. [SENSOR_RET_ERROR_INNER]
The interruption happened during event get [SENSOR_RET_ERROR_INNER]
Whatever error happened during event get. [SENSOR_RET_ERROR_INNER]

Get event timeout. [SENSOR_RET_ERROR_INNER]
Error happened during event get. [SENSOR_RET_ERROR_INNER]

**Classification**

Public

**Type**

Method

**See also:**

None

**POS_RET_API POS_RegisterListenerGPSTimeSetReq (HANDLE *hApp*, PCSTR *notifyName*, uint8_t *ucCtrlFlg*)**

**Brief**

Register GPS time setting request delivery

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *notifyName* | PCSTR - Destination thread name |
| in | *ucCtrlFlg* | uint8_t - Delivery control flag(register) |

Delivery control flag(ucCtrlFlg)
SENSOR_DELIVERY_REGIST - register

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_BUFFULL* | the register count is full |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.

Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter notifyName is NULL [POS_RET_ERROR_PARAM]
The parameter ucCtrlFlg is not register(SENSOR_DELIVERY_REGIST)
[POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD95__CWORD101___CWORD80*
[POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max
[POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max
[POS_RET_ERROR_INNER]
The event is created in system, but the count of reference is reach to max
[POS_RET_ERROR_INNER]
Memory allocate falied during the event table creation for event registering.
[POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation.
[POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed.
[POS_RET_ERROR_INNER]
The event table is full during event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed during event table creation.
[POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, try to allocate the memory of the thread table creation, but failed.
[POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered.
[POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed.
[POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer failed in process. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer

between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [POS_RET_ERROR_INNER]
The count of thread arrived max in event management table during event waiting. [POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting. [POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
The result from positioning service is SENSOR_RET_ERROR_PARAM. [POS_RET_ERROR_PARAM]
The result from positioning service is SENSOR_RET_ERROR_BUFFULL. [POS_RET_ERROR_BUFFULL]
The result from positioning service is SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_RESOURCE]
The result from positioning service is not SENSOR_RET_ERROR_PARAM or SENSOR_RET_ERROR_BUFFULL or SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Method

**See also:**

None

**POS_RET_API POS_RegisterListenerHeading (HANDLE *hApp*, PCSTR *notifyName*, uint8_t *ucCtrlFlg*, uint8_t *ucDeliveryTiming*, uint8_t *ucGetMethod*)**

**Brief**

Register heading delivery

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | |

| | | |
|---|---|---|
| | | HANDLE - App Handle |
| in | *notifyName* | PCSTR - Destination thread name |
| in | *ucCtrlFlg* | uint8_t - Delivery control flag(register) |
| in | *ucDeliveryTiming* | uint8_t - Delivery timing(change/update) |
| in | *ucGetMethod* | uint8_t - Get method(GPS/Navi/Not specified) |

Delivery control flag(ucCtrlFlg)
SENSOR_DELIVERY_REGIST - register
Register specified heading delivery
Please note that if the same data delivery has been registered for mutiple times, the data will
also be deliveried for registered mutiple times.
The specified heading will be deliveried at registered time first delivery.
Delivery timing(ucDeliveryTiming)
SENSOR_DELIVERY_TIMING_CHANGE - change. Specified heading will be deliveried only when it is changed.
SENSOR_DELIVERY_TIMING_UPDATE - update.Specified heading will be deliveried when it is updated by
vehicle sensor.
Get method(ucGetMethod)
SENSOR_GET_METHOD_GPS - GPS. The heading from GPS will be deliveried.
SENSOR_GET_METHOD_NAVI - Navi. The heading from Navi will be deliveried.
SENSOR_GET_METHOD_AUTO - Not specified. The heading will be deliveried according to the current environment
Avaliable method is descriped as following in each environment.
In the following environment, if the SENSOR_GET_METHOD_AUTO is set,treated as default
In the following environment, if the unsupported method has been specified, POS_RET_ERROR_PARAM will be returned.
*CWORD96*___*CWORD84*_ *SENSOR_GET_METHOD_GPS(default)*
_*CWORD95* /_CWORD101___CWORD84_ SENSOR_GET_METHOD_NAVI(default)
*CWORD95* /_CWORD101___CWORD80_ SENSOR_GET_METHOD_GPS / SENSOR_GET_METHOD_NAVI(default)
*CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ) SENSOR_GET_METHOD_NAVI(default)
Please note that although it is possible to register delivery, but the data will not be deliveried

in *CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ).

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_BUFFULL* | the register count is full |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher
for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter ucDeliveryTiming is neither update(SENSOR_DELIVERY_TIMING_UPDATE) nor change
(SENSOR_DELIVERY_TIMING_CHANGE) [POS_RET_ERROR_PARAM]
The parameter ucCtrlFlg is not register(SENSOR_DELIVERY_REGIST) [POS_RET_ERROR_PARAM]
The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter notifyName is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* , and the parameter ucGetMethod is neither GPS(SENSOR_GET_METHOD_GPS)
nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD84* , and the parameter ucGetMethod is neither Navi(SENSOR_GET_METHOD_NAVI)
nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD80* , and the parameter ucGetMethod is not GPS(SENSOR_GET_METHOD_GPS)
or Navi(SENSOR_GET_METHOD_NAVI) or Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* or *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach max [POS_RET_ERROR_INNER]

The event is created in system, but the count of reference is reach to max [POS_RET_ERROR_INNER]
Memory allocate failed during the event table creation for event registering. [POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]
The event table is full during event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed during event table creation. [POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, try to allocate the memory of the thread table creation, but failed. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed. [POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [POS_RET_ERROR_INNER]
The count of thread is reach to max in event management table during event waiting. [POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting. [POS_RET_ERROR_INNER]

The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
Getting event timeout. [POS_RET_ERROR_INNER]
Error happened during event getting. [POS_RET_ERROR_INNER]
The result of positioning service is SENSOR_RET_ERROR_PARAM.
[POS_RET_ERROR_PARAM]
The result of positioning service is SENSOR_RET_ERROR_BUFFULL.
[POS_RET_ERROR_BUFFULL]
The result of positioning service is SENSOR_RET_ERROR_RESOURCE.
[POS_RET_ERROR_RESOURCE]
The result of positioning service is not SENSOR_RET_ERROR_PARAM and
SENSOR_RET_ERROR_BUFFULL
and SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Method

**POS_RET_API POS_RegisterListenerLonLat (HANDLE *hApp*, PCSTR *notifyName*, uint8_t *ucCtrlFlg*, uint8_t *ucDeliveryTiming*, uint8_t *ucGetMethod*)**

**Brief**

Register for longitude and lattitude delivery

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *notifyName* | PCSTR - Destination thread name |
| in | *ucCtrlFlg* | uint8_t - Delivery control flag(register) |
| in | *ucDeliveryTiming* | uint8_t - Delivery timing(change/update) |
| in | *ucGetMethod* | uint8_t - Get method(GPS/Navigation/Not specified) |

Delivery control flag(ucCtrlFlg)
SENSOR_DELIVERY_REGIST - register
Register for specified logtitude and latitude delivery
Please note if call this function for multiple times, the data will be deliveried for multiple times.
The specified longitude and latitude will be deliveried during the registrationfirst delivery
Delivery timing(ucDeliveryTiming)
SENSOR_DELIVERY_TIMING_CHANGE - change. Specified longitude and latitude will be deliveried
only when it has been changed.
SENSOR_DELIVERY_TIMING_UPDATE - update.Specified longitude and latitude will be deliveried
every time that has been updated by the vehicle sensors.
Get method(ucGetMethod)
SENSOR_GET_METHOD_GPS - GPS. The longitude and latitude from GPS will be deliveried.
SENSOR_GET_METHOD_NAVI - Navi. The longitude and latitude from Navigation will be deliveried.
SENSOR_GET_METHOD_AUTO - Not specified. The specified longitude and latitude will be deliveried
according to the current environment.
Avaliable method is descriped as following:
In the following environment, if the SENSOR_GET_METHOD_AUTO is set,treated as default
In the following environment, if the unsupported method has been specified,
POS_RET_ERROR_PARAM will be returned.
*CWORD96___CWORD84_ SENSOR_GET_METHOD_GPS(default)*
*CWORD95 /_CWORD101___CWORD84_ SENSOR_GET_METHOD_NAVI(default)*
*CWORD95 /_CWORD101___CWORD80_ SENSOR_GET_METHOD_GPS / SENSOR_GET_METHOD_NAVI(default)*
*CWORD95 /_CWORD101___CWORD84* error(no *CWORD80* ): SENSOR_GET_METHOD_NAVI(default)
In *CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ), please note that although it is possible to register sucessfully,

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_BUFFULL* | the register count is full |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the dispatcher for App are completed.

Availability of positioning service is TRUE.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter ucDeliveryTiming is neither update(SENSOR_DELIVERY_TIMING_UPDATE) nor change (SENSOR_DELIVERY_TIMING_CHANGE) [POS_RET_ERROR_PARAM]
The parameter ucCtrlFlg is not register(SENSOR_DELIVERY_REGIST) [POS_RET_ERROR_PARAM]
The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter notifyName is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* , and the parameter ucGetMethod is neither GPS(SENSOR_GET_METHOD_GPS) nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD84* , and the parameter ucGetMethod is neither Navi(SENSOR_GET_METHOD_NAVI) nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD80* , and the parameter ucGetMethod is not GPS(SENSOR_GET_METHOD_GPS) and Navi(SENSOR_GET_METHOD_NAVI) and Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* and *CWORD95__CWORD101___CWORD84* and *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to the max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to the max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to the max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference to the event is reach to the max [POS_RET_ERROR_INNER]
The event is created in system, but the count of reference to the event is reach to the max [POS_RET_ERROR_INNER]
Memory allocate failed during the event table creation for the event registering. [POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]
The event table is full. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed. [POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]

In event table, try to allocate the memory of the thread table creation, but failed. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed. [POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes.
[POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer between processes.
[POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between process. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes.
[POS_RET_ERROR_INNER]
Message transfer failed between process. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during waiting event.
[POS_RET_ERROR_INNER]
The count of thread beyond max in event management table during event waiting. [POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during getting event. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting. [POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
Getting event timeout. [POS_RET_ERROR_INNER]
Error happened during event getting. [POS_RET_ERROR_INNER]
The result of positioning service returned is SENSOR_RET_ERROR_PARAM. [POS_RET_ERROR_PARAM]
The result of positioning service returned is SENSOR_RET_ERROR_BUFFULL. [POS_RET_ERROR_BUFFULL]
The result of positioning service returned is SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_RESOURCE]
The result of positioning service returned is not SENSOR_RET_ERROR_PARAM and SENSOR_RET_ERROR_BUFFULL
and SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Method

**SENSOR_RET_API POS_RegisterListenerPkgSensData (HANDLE *hApp*, PCSTR *notifyName*, uint8_t *ucPkgNum*, DID * *pulDid*, uint8_t *ucCtrlFlg*, uint8_t *ucDeliveryTiming*)**

**Brief**

Send the extra package when first delivery.

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *notifyName* | PCSTR - Destination thread name |
| in | *ucPkgNum* | uint8_t - data number in package(116) |
| in | *pulDid* | DID * - buffer pointer of the data ID array in package |
| in | *ucCtrlFlg* | uint8_t - Delivery control flag(register) |
| in | *ucDeliveryTiming* | uint8_t - Delivery timing(change/update) |

data number in package(ucPkgNum)
The following 8 data ID can be registered. And the register data number range is 1~16.
buffer pointer of the data ID array in package(pulDid)
The data ID set to the first in pulDid is the delivery key.
If the data ID not one of the following be set, return SENSOR_RET_ERROR_PARAM.
POS_DID_SNS_COUNTER-sensor counter(get from sys)
POS_DID_GYRO-gyro output
POS_DID_SPEED_PULSE-speed pulse(count of pulse)
POS_DID_REV-REV signal(0forward1backward)
POS_DID_GSNS_X-Gsensor output X axis(154bit A/D value)
POS_DID_GSNS_Y-Gsensor output Y axis(154bit A/D value)

POS_DID_GYRO_TEMP-gyro temperature
POS_DID_PULSE_TIME-pulse time(32bit pulse time store count(0-32)
32bitx max32 time between pulse[s](1-4294967295s))
Time between is only valid in the environment after
*CWORD80* 2S(include *CWORD80* 2S). In other environment, the time will be deliveried as 0.

- The Gsensor output is 0 in the environment without Gsensor hardware.

Delivery control flag(ucCtrlFlg)
SENSOR_DELIVERY_REGIST - register
Register specified LonLat delivery
Please note that if the same data delivery has been registered for multiple times, the data will also be deliveried for registered multiple times.
The specified LonLat will be deliveried at register time no matter what delivery timing has been registered first delivery.
Delivery timing(ucDeliveryTiming)
SENSOR_DELIVERY_TIMING_CHANGE - change Specified LonLat be deliveried only when it is changed.
SENSOR_DELIVERY_TIMING_UPDATE - update Specified Lonlat be deliveried as long as it is updated by vehicle sensor.

**Return values:**

| | |
|---|---|
| *SENSOR_RET_NORMAL* | normal end |
| *SENSOR_RET_ERROR_CREATE_EVENT* | event create failed |
| *SENSOR_RET_ERROR_PARAM* | parameter error |
| *SENSOR_RET_ERROR_INNER* | internal error |
| *SENSOR_RET_ERROR_NOSUPPORT* | unsupported |
| *SENSOR_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization (_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of service positioning is TRUE.

**change of internal status**

There is no change of internal status

**Failure condition**

The parameter hApp is NULL [SENSOR_RET_ERROR_PARAM]
Hardware environment is not *CWORD95__CWORD101___CWORD80* [SENSOR_RET_ERROR_NOSUPPORT]

The parameter ucCtrlFlg is not register(SENSOR_DELIVERY_REGIST)
[SENSOR_RET_ERROR_PARAM]
The parameter ucDeliveryTiming is neither update(SENSOR_DELIVERY_TIMING_UPDATE)
nor change(SENSOR_DELIVERY_TIMING_CHANGE) [SENSOR_RET_ERROR_PARAM]
The parameter notifyName is NULL [SENSOR_RET_ERROR_PARAM]
The parameter ucPkgNum is 0 or it is larger than 16 [SENSOR_RET_ERROR_PARAM]
The parameter pulDid is NULL [SENSOR_RET_ERROR_PARAM]
The data ID in paramter buffer pulDid is not avaliable value
[SENSOR_RET_ERROR_PARAM]
The count of message in message queue is reach to max
[SENSOR_RET_ERROR_RESOURCE]
The count of mutex is reach to max [SENSOR_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max
[SENSOR_RET_ERROR_RESOURCE]
The event has been registered in event table and created in same process,
but the count of reference is reach to max [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table and created in system, but the
count of reference is reach to max [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, but memory for the thread event table
creation allocate falied. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, but the thread can not be registered in
event table. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, but memory for thread table creation
allocate falied. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, and the event flag has already been
registered. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, and the event flag register failed.
[SENSOR_RET_ERROR_CREATE_EVENT]
No empty field for registering the event HANDLE into event table.
[SENSOR_RET_ERROR_CREATE_EVENT]
Memory for event HANDLE get failed. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has not been registered in event table and not finished registering
[SENSOR_RET_ERROR_CREATE_EVENT]
The event has not been registered in event table, and memory for thread table
creation allocate falied. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has not been registered in event table, and the event flag has already been
registered. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has not been registered in event table, and the event flag register failed.
[SENSOR_RET_ERROR_CREATE_EVENT]
ProcessNo has not been registered in message control table when message transfered
between processes. [SENSOR_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered in process.
[SENSOR_RET_ERROR_INNER]
Message transfer failed in process. [SENSOR_RET_ERROR_INNER]
The destination process name is NULL. [SENSOR_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer
between processes. [SENSOR_RET_ERROR_INNER]
The message queue name has not been registered in control table when message

transfer between processes. [SENSOR_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes. [SENSOR_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes. [SENSOR_RET_ERROR_INNER]
Message transfer failed between processes. [SENSOR_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [SENSOR_RET_ERROR_CREATE_EVENT]
The count of thread is reach to max in event management table during event waiting. [SENSOR_RET_ERROR_CREATE_EVENT]
The EV_FLAG_BIT is not set in flagID during event getting. [SENSOR_RET_ERROR_CREATE_EVENT]
The ID of message event queue has not been created during event getting. [SENSOR_RET_ERROR_CREATE_EVENT]
The flagID has not been registered during event getting. [SENSOR_RET_ERROR_CREATE_EVENT]
The interruption happened during event getting [SENSOR_RET_ERROR_CREATE_EVENT]
Whatever error happened during event getting. [SENSOR_RET_ERROR_CREATE_EVENT]
Got event is SENSOR_RET_ERROR_INNER. [SENSOR_RET_ERROR_INNER]

**Detial**

Call this API to register vehicle sensor data delivery.
This API return the result of registering.
The data from sensor data received to registering will be deliveried.
The first delivery data is the sensor data(max 64 number of sensor counter, reverse signal,gyro temperature, max 640 number of gyro output,speed pulse,Gsensor output(X axis), Gsensor output(Y axis), max 2048 number of pulse time) in 6.4 second.
If the data number is more than max number, delivery the data in newest 6.4 second.
If the sensor data accumulated more than max number, set VEHICLE_SNS_BREAK to data missing information.
If sensor data number is more than the data number send in one time(10 number of sensor counter,reverse signal,gyro temperature, 100 number of gyro output,speed pulse, Gsensor output(X axis),Gsensor output(Y axis), 320 number of pulse time), the new data is divided into partitions(every partition with 10 number of sensor counter, reverse signal,gyro temperature, 100 number of gyro output,speed pulse,Gsensor output(X axis), Gsensor output(Y axis), 320 number of pulse time) to delivery.
The last message for first delivery is the message that the partition count equal to partition No.
After first delivery, the message data(1 number of sensor counter,reverse signal, gyro temperature, 10 number of gyro output,speed pulse,Gsensor output(X axis), Gsensor output(Y axis), 32 number of pulse time) deliveried.
And because the data missing information, divided partition count, diveided partition No is not used, they will be set to 0 in message.
(sample)The sensor data in 6.4 second divided to delivery
1st message(sensor counter,reverse signal,gyro temperature=10 number, gyro output,speed pulse,

Gsensor output(X axis),Gsensor output(Y axis)=100 numberpulse time=320 number), divided
partition count=7,divided partition No=1)
2nd message(sensor counter,reverse signal,gyro temperature=10 number, gyro output,speed pulse,
Gsensor output(X axis),Gsensor output(Y axis)=100 numberpulse time=320 number), divided
partition count=7,divided partition No=2)
3rd message(sensor counter,reverse signal,gyro temperature=10 number, gyro output,speed pulse,
Gsensor output(X axis),Gsensor output(Y axis)=100 numberpulse time=320 number), divided
partition count=7,divided partition No=3)
4th message(sensor counter,reverse signal,gyro temperature=10 number, gyro output,speed pulse,
Gsensor output(X axis),Gsensor output(Y axis)=100 numberpulse time=320 number), divided
partition count=7,divided partition No=4)
5th message(sensor counter,reverse signal,gyro temperature=10 number, gyro output,speed pulse,
Gsensor output(X axis),Gsensor output(Y axis)=100 numberpulse time=320 number), divided
partition count=7,divided partition No=5)
6th message(sensor counter,reverse signal,gyro temperature=10 number, gyro output,speed pulse,
Gsensor output(X axis),Gsensor output(Y axis)=100 numberpulse time=320 number), divided
partition count=7,divided partition No=6)
7th message(sensor counter,reverse signal,gyro temperature=10 number, gyro output,speed pulse,
Gsensor output(X axis),Gsensor output(Y axis)=100 numberpulse time=320 number), divided
partition count=7,divided partition No=7)

Please note the following points when use this API.
Duplication registering
The same destination thread name has already been registered
The registered delivery data updated and normal return.first delivery
To one delivery destination, the same data will not be duplication deliveried at same timing.
After delivery the sensor data accumulated in 6.4 second(first delivery), the sensor data will not be accumulated any more. So the same data will be deliveried as first delivery when registered again.
This API is only called by Navi proxy.
After call this API, if the delivery destination thread name is changed, please call this API again.

message structure

After success to register, vehicle sensor will send message as system API message with following format. Command ID : CID_POSIF_REGISTER_LISTENER_PKG_SENSOR_DATA

Definition of structure

```
#define  SENSOR_MSGBUF_DSIZE       2264  /* max size of message body */
#define  SENSOR_VSHEAD_DSIZE       36  /* vehicle sensor header size(1+3+16*2) */
#define  SENSOR_PKG_DELIVERY_MAX       16  /* max number of Data ID in package */
#define  SENSOR_VSINFO_DSIZE    (SENSOR_MSGBUF_DSIZE - SENSOR_VSHEAD_DSIZE)
typedef struct {
  uint8_t    ucDNum;          /* number of data */
  uint8_t    ucDataBreak;       /* data missing information */
  uint8_t    ucDivideCnt;       /* divided partition count */
  uint8_t    ucDivideSendCnt;      /* divided partition No. */
  uint16_t  usOffset[SENSOR_PKG_DELIVERY_MAX];  /* offset */
  uint8_t    ucData[SENSOR_VSINFO_DSIZE];    /* vehicle sensor data */
} SENSOR_PKG_MSG_VSINFO;
```

number of data

Data number in package

data missing information

VEHICLE_SNS_BREAK:not continuous data

VEHICLE_SNS_NORMAL:continuous data

divided partition count

All divided partition count

If it is more than 1, data divided to delivery

divided partition No.

The No. of the divided partition. If it equal to the divided partition count, that meanings this message is the last divided partition of the package.

offset

The array of the offset from the head of vehicle sensor data

vehicle sensor data

Data ID(4Byte)

Data size(2Byte)

reception flag(1Byte)

If the data get from CAN,magnetize line,GPS, set to 0x01

If data has not been received, set to 0x00

reserve(1Byte)

Data body is cycle fit with the packaged data.

**Classification**

Public

**Type**

Method

**SENSOR_RET_API POS_RegisterListenerSensData (HANDLE *hApp*, PCSTR *notifyName*, DID *ulDid*, uint8_t *ucCtrlFlg*, uint8_t *ucDeliveryTiming*)**

**Brief**

Register sensor data delivery.

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *notifyName* | PCSTR - Destination thread name |
| in | *ulDid* | DID - Data ID of vehicle info |
| in | *ucCtrlFlg* | uint8_t - Delivery control flag(register) |
| in | *ucDeliveryTiming* | uint8_t - Delivery timing(change/update) |

Data ID of vehicle info(ulDid)

- The Gsensor output is 0 in the environment without Gsensor hardware.

Delivery control flag(ucCtrlFlg)
SENSOR_DELIVERY_REGIST - register
Register specified LonLat delivery
Please note that if the same data delivery has been registered for multiple times,
the data will also be deliveried for registered multiple times.
The specified LonLat will be deliveried at register time no matter what delivery
timing has been registered first delivery.
Delivery timing(ucDeliveryTiming)
SENSOR_DELIVERY_TIMING_CHANGE - change Specified LonLat be deliveried only when
it is changed.
SENSOR_DELIVERY_TIMING_UPDATE - update Specified Lonlat be deliveried
as long as it is updated by vehicle sensor.

**Return values:**

| | |
|---|---|
| *SENSOR_RET_NORMAL* | normal end |

| | |
|---|---|
| *SENSOR_RET_ERROR_CREATE_EVENT* | event create failed |
| *SENSOR_RET_ERROR_PARAM* | parameter error |
| *SENSOR_RET_ERROR_INNER* | internal error |
| *SENSOR_RET_ERROR_NOSUPPORT* | unsupported |
| *SENSOR_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of service positioning is TRUE.

**change of internal status**

There is no change of internal status

**Failure condition**

The parameter ucDeliveryTiming is neither update (SENSOR_DELIVERY_TIMING_UPDATE) nor change(SENSOR_DELIVERY_TIMING_CHANGE) [SENSOR_RET_ERROR_PARAM]
The parameter ucCtrlFlg is not register(SENSOR_DELIVERY_REGIST) [SENSOR_RET_ERROR_PARAM]
The parameter hApp is NULL [SENSOR_RET_ERROR_PARAM]
The parameter notifyName is NULL [SENSOR_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* or *CWORD95__CWORD101___CWORD80* [SENSOR_RET_ERROR_NOSUPPORT]
The parameter ulDID is not avaliable value [SENSOR_RET_ERROR_PARAM]
The parameter ulDID is a value can not specified [SENSOR_RET_ERROR_PARAM]
The count of message in message queue is reach to max [SENSOR_RET_ERROR_RESOURCE]
The count of mutex is reach to max [SENSOR_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [SENSOR_RET_ERROR_RESOURCE]
The event has been registered in event table and created in same process, but the count of reference is reach to max [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table and created in system, but the count of reference is reach to max [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, but memory for the thread event table creation allocate falied. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, but the thread can not be registered in event table. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, but memory for thread table creation allocate falied. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, and the event flag has already been registered. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has been registered in event table, and the event flag register failed. [SENSOR_RET_ERROR_CREATE_EVENT]

No empty field for registering the event HANDLE into event table.
[SENSOR_RET_ERROR_CREATE_EVENT]
Memory for event HANDLE get failed. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has not been registered in event table and not finished registering
[SENSOR_RET_ERROR_CREATE_EVENT]
The event has not been registered in event table, and memory for thread table
creation allocate falied. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has not been registered in event table, and the event flag has already
been registered. [SENSOR_RET_ERROR_CREATE_EVENT]
The event has not been registered in event table, and the event flag register
failed. [SENSOR_RET_ERROR_CREATE_EVENT]
ProcessNo has not been registered in message control table when message transfered
in process. [SENSOR_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered in process.
[SENSOR_RET_ERROR_INNER]
Message transfer failed in process. [SENSOR_RET_ERROR_INNER]
The destination process name is NULL. [SENSOR_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message
transfer between process. [SENSOR_RET_ERROR_INNER]
The message queue name has not been registered in control table when message
transfer between process. [SENSOR_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between process.
[SENSOR_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered
between process. [SENSOR_RET_ERROR_INNER]
Message transfer failed between process. [SENSOR_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during waiting
event. [SENSOR_RET_ERROR_CREATE_EVENT]
The count of thread is reach to max in event management table during waiting
event. [SENSOR_RET_ERROR_CREATE_EVENT]
The EV_FLAG_BIT is not set in flagID during getting event.
[SENSOR_RET_ERROR_CREATE_EVENT]
The ID of message event queue has not been created during getting event.
[SENSOR_RET_ERROR_CREATE_EVENT]
The flagID has not been registered during getting event.
[SENSOR_RET_ERROR_CREATE_EVENT]
The interruption happened during getting event [SENSOR_RET_ERROR_CREATE_EVENT]
Whatever error happened during getting event. [SENSOR_RET_ERROR_CREATE_EVENT]

**Detial**

Call this API to register vehicle sensor data delivery.
This API return the result of registering.

Please note the following points when use this API.
Duplication registering
The same destination thread name has already been registered
The registered delivery data updated and normal return.first delivery

To one delivery destination, the same data will not be duplication deliveried at same timing.

After call this API, if the delivery destination thread name is changed, please call this API again.

message structure

After success to register, vehicle sensor will send message as system API message with following format.

If the register successed, certainly delivery first data. And then delivery data according to the delivery timing.

Command ID : CID_POSIF_REGISTER_LISTENER_SENSOR_DATA

```
1 #define  SENSOR_MSG_VSINFO_DSIZE  1272      /* max size of message body */
2 typedef struct
3 {
4   DID       did;         /* data ID */
5   uint16_t  size;        /* data size of vehicle sensor data */
6   uint8_t   rcvFlag;     /* reception flag */
7   uint8_t   reserve;     /* reserve */
8   uint8_t   data[SENSOR_MSG_VSINFO_DSIZE];  /* vehicle sensor data */
9 } SENSOR_MSG_VSINFO;
```

reception flag(1Byte)

If the data get from CAN or magnetize line, set to 0x01

If data has not been received, set to 0x00

**Classification**

Public

**Type**

Method

**POS_RET_API POS_RegisterListenerSpeed (HANDLE *hApp*, PCSTR *notifyName*, uint8_t *ucCtrlFlg*, uint8_t *ucDeliveryTiming*, uint8_t *ucGetMethod*)**

**Brief**

Register speed delivery

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *notifyName* | PCSTR - Destination thread name |
| in | *ucCtrlFlg* | |

| | | |
|---|---|---|
| | | uint8_t - Delivery control flag(register) |
| in | *ucDeliveryTiming* | uint8_t - Delivery timing(change/update) |
| in | *ucGetMethod* | uint8_t - Get method(POS/Navi/Not specified) |

Delivery control flag(ucCtrlFlg)
SENSOR_DELIVERY_REGIST - register
Register specified speed delivery
Please note that if the same data delivery has been registered for multiple times, the data will
also be deliveried for registered mutiple times.
The specified speed will be deliveried at registered time first delivery.
Delivery timing(ucDeliveryTiming)
SENSOR_DELIVERY_TIMING_CHANGE - change. Specified speed will be deliveried only when it is changed.
SENSOR_DELIVERY_TIMING_UPDATE - update.Specified speed will be deliveried when it is updated by
vehicle sensor.
Get method(ucGetMethod)
SENSOR_GET_METHOD_POS - The speed calculated in positioning based on speed pulse will be deliveried.
SENSOR_GET_METHOD_NAVI - Navi The speed from Navi will be deliveried.
SENSOR_GET_METHOD_AUTO - Not specified The speed will be deliveried according to the current environment deliveried.
Avaliable method is descriped as following in each environment.
In the following environment, if the SENSOR_GET_METHOD_AUTO is set,treated as default
In the following environment, if the unsupported method has been specified,
POS_RET_ERROR_PARAM will be returned.
*CWORD96___CWORD84_* *SENSOR_GET_METHOD_POS(default)*
*_CWORD95* /_CWORD101___CWORD84_ SENSOR_GET_METHOD_POS(default)
*CWORD95* /_CWORD101___CWORD80_ SENSOR_GET_METHOD_POS / SENSOR_GET_METHOD_NAVI(default)
*CWORD95* /_CWORD101___CWORD84_ error(no *CWORD80* ) SENSOR_GET_METHOD_POS(default)

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal end |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_BUFFULL* | the register count is full |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App
are completed.
Availability of service positioning is TRUE.

**change of internal status**

There is no change of internal status

**Failure condition**

The parameter ucDeliveryTiming is neither update(SENSOR_DELIVERY_TIMING_UPDATE) nor change(SENSOR_DELIVERY_TIMING_CHANGE) [POS_RET_ERROR_PARAM]
The parameter ucCtrlFlg is not register(SENSOR_DELIVERY_REGIST) [POS_RET_ERROR_PARAM]
The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter notifyName is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* , and the parameter ucGetMethod is neither POS(SENSOR_GET_METHOD_POS) nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD84* , and the parameter ucGetMethod is neither POS(SENSOR_GET_METHOD_POS) nor Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD95__CWORD101___CWORD80* , and the parameter ucGetMethod is not POS(SENSOR_GET_METHOD_POS) or Navi(SENSOR_GET_METHOD_NAVI) or Auto(SENSOR_GET_METHOD_AUTO) [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD84* or *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max [POS_RET_ERROR_INNER]
The event is created in system, but the count of reference is reach to max [POS_RET_ERROR_INNER]
Memory allocate failed during the event table creation for event registering. [POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table during the thread table creation. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]

The event table is full during event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed during event table creation. [POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, try to allocate the memory of the thread table creation, but failed. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed. [POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
ProcessNo has not been registered in message control table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
The destination process name size is larger than 20 characters when message transfer between processes. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when message transfer between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when message transfered between processes. [POS_RET_ERROR_INNER]
Message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [POS_RET_ERROR_INNER]
The count of thread is reach to max in event management table during event waiting. [POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event getting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event getting. [POS_RET_ERROR_INNER]
The flagID has not been registered during event getting. [POS_RET_ERROR_INNER]
The interruption happened during event getting [POS_RET_ERROR_INNER]
Whatever error happened during event getting. [POS_RET_ERROR_INNER]
Getting event timeout. [POS_RET_ERROR_INNER]
Error happened during event getting. [POS_RET_ERROR_INNER]
The result of positioning service is SENSOR_RET_ERROR_PARAM. [POS_RET_ERROR_PARAM]
The result of positioning service is SENSOR_RET_ERROR_BUFFULL. [POS_RET_ERROR_BUFFULL]

The result of positioning service is SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_RESOURCE]
The result of positioning service is not SENSOR_RET_ERROR_PARAM and SENSOR_RET_ERROR_BUFFULL and SENSOR_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Method

**POS_RET_API POS_ReqGPSReset (HANDLE *hApp*, PCSTR *ResName*, uint8_t *mode*)**

**Brief**

GPS reset request

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *ResName* | PCSTR - response thread name |
| in | *mode* | uint8_t - reset mode |

reset mode(mode)
If the mode is not one of the following, return POS_RET_ERROR_PARAM
GPS_RST_COLDSTART - GPS reset request(cold start)
Automatic search, track satellite and position fix after RAM initializedall config items be initialized to default value.
Initialize data objectsalmanac/ ephemeris data, current position, GPS receiver mode and etc.

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
The parameter ResName is NULL [POS_RET_ERROR_PARAM]
The parameter mode is not GPS_RST_COLDSTART [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The event is created in same process, but the count of reference is reach to max [POS_RET_ERROR_INNER]
The event is created in system, but the count of reference is reach to max [POS_RET_ERROR_INNER]
Memory allocate falied when the event table creation that for event registering. [POS_RET_ERROR_INNER]
The thread can not register in the event table. [POS_RET_ERROR_INNER]
Memory allocate failed in event table when the thread table creation. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread table in event table, the event flag register failed. [POS_RET_ERROR_INNER]
The event table is full when event creation. [POS_RET_ERROR_INNER]
The memory for event HANDLE allocate failed when event table creation. [POS_RET_ERROR_INNER]
The thread can not be registered in event table. [POS_RET_ERROR_INNER]
In event table, to allocate the memory of the thread table creation, but failed. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag has already been registered. [POS_RET_ERROR_INNER]
After register the thread in event table, the event flag register failed. [POS_RET_ERROR_INNER]
Specified event ID has not been registered in table. [POS_RET_ERROR_INNER]
Initialize event object failed. [POS_RET_ERROR_INNER]
The message queue name has not been registered in control table when GPS reset message transfer between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed when GPS reset message transfered

between processes. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table when GPS reset message transfered between processes. [POS_RET_ERROR_INNER]
GPS reset message transfer failed between processes. [POS_RET_ERROR_INNER]
Specified event HANDLE has not been registered in event HANDLE table during event waiting. [POS_RET_ERROR_INNER]
The count of thread is reach to max in event management table during event waiting. [POS_RET_ERROR_INNER]
The EV_FLAG_BIT is not set in flagID during event waiting. [POS_RET_ERROR_INNER]
The ID of message event queue has not been created during event waiting. [POS_RET_ERROR_INNER]
The flagID has not been registered during event waiting. [POS_RET_ERROR_INNER]
The interruption happened during event waiting [POS_RET_ERROR_INNER]
Whatever error happened during event waiting. [POS_RET_ERROR_INNER]
The result of sensor data get processing is POS_RET_ERROR_RESOURCE. [POS_RET_ERROR_RESOURCE]
The result of sensor data get processing is not POS_RET_ERROR_RESOURCE. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Fire and Forget

**See also:**

None

**int32_t POS_ReqGPSSetting (HANDLE *hApp*, SENSOR_MSG_SEND_DAT * *p_data*)**

**Brief**

GPS setting request

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *p_data* | SENSOR_MSG_SEND_DAT * - pointer of GPS setting data |

SENSOR_MSG_SEND_DAT structure

Data ID(did)
POS_DID_GPS__CWORD44__SETINITIAL - setting initial location and time data to GPS
(reference to initial location, time setting(*CWORD44* ))
POS_DID_GPS__CWORD44__SETRMODEEX - setting GPS receiver mode(pull extension sentence)
(reference to GPS receiver mode setting(*CWORD44* ))
POS_DID_GPS__CWORD44__SELSENT - setting command to GPS that output any sentence
(reference to output any sentence setting_CWORD44_)

**Return values:**

| | |
|---|---|
| *SENSOR_RET_NORMAL* | normal finish |
| *SENSOR_RET_ERROR_CREATE_EVENT* | event create error |
| *SENSOR_RET_ERROR_PARAM* | parameter error |
| *SENSOR_RET_ERROR_DID* | data ID not registered |
| *SENSOR_RET_ERROR_NOSUPPORT* | unsupported |
| *SENSOR_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status by this API

**Failure condition**

The parameter hApp is NULL [SENSOR_RET_ERROR_PARAM]
The parameter p_data is NULL [SENSOR_RET_ERROR_PARAM]
Data ID(p_data->did) is not supported [SENSOR_RET_ERROR_DID]
Hardware environment is not *CWORD95__CWORD101___CWORD80* [SENSOR_RET_ERROR_NOSUPPORT]
GPS setting data size(p_data->usSize) is not same with the data that related to data ID [SENSOR_RET_ERROR_PARAM]
The count of message in message queue is reach to max [SENSOR_RET_ERROR_RESOURCE]
The count of mutex is reach to max [SENSOR_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [SENSOR_RET_ERROR_RESOURCE]
The message queue name has not been registered in control table when GPS setting message transfer between processes. [SENSOR_RET_ERROR_CREATE_EVENT]
Message transfer HANDLE create failed. [SENSOR_RET_ERROR_CREATE_EVENT]
Message transfer HANDLE get failed from internal table. [SENSOR_RET_ERROR_CREATE_EVENT]
GPS setting message transfer failed. [SENSOR_RET_ERROR_CREATE_EVENT]

**Classification**

Public

**Type**

Fire and Forget

**See also:**

None

**NAVIINFO_RET_API POS_SetGPSInfo (HANDLE *hApp*, NAVIINFO_ALL * *navilocinfo*)**

**Brief**

Set GPS information

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *navilocinfo* | NAVIINFO_ALL * - pointer of GPS information |

NAVIINFO_ALL structure

```
1 typedef struct
2 {
3   uint8_t         ucSensorCnt;  /* sensor count */
4   uint8_t         reserve[3];   /* reserve */
5   NAVIINFO_DIAG_GPS stDiagGps;    /* position fix related information */
6   NAVIINFO_NAVI_GPS stNaviGps;    /* other GPS related information */
7 } NAVIINFO_ALL;
```

sensor count

sensor count value when GPS data received

NAVIINFO_DIAG_GPS structure

```
1 typedef struct
2 {
3   NAVIINFO_DIAG_GPS_FIX  stFix;  /* position fix information */
4   NAVIINFO_DIAG_GPS_SAT  stSat;  /* all satellite information */
5 } NAVIINFO_DIAG_GPS;
```

NAVIINFO_DIAG_GPS_FIX structure

```
1 typedef struct
2 {
3   uint8_t                ucFixSts;     /* position fix status */
4   uint8_t                ucReserve[3]; /* reserve */
5   NAVIINFO_DIAG_GPS_FIX_CNT stCnt;       /* position fix count data */
6   NAVIINFO_DIAG_GPS_FIX_XYZ stWgs84;      /* lonlat data(WGS84 geodetic)  */
7 } NAVIINFO_DIAG_GPS_FIX;
```

position fix status(ucFixSts)
NAVIINFO_DIAG_GPS_FIX_STS_NONnot fixed
NAVIINFO_DIAG_GPS_FIX_STS_2D 2D fix
NAVIINFO_DIAG_GPS_FIX_STS_3D 3D fix

NAVIINFO_DIAG_GPS_FIX_CNT structure

```
1 typedef struct
2 {
3   uint32_t  ulCnt3d;    /* position fix ratio:3D(unitsec) */
4   uint32_t  ulCnt2d;    /* position fix ratio:2D(unitsec) */
5   uint32_t  ulCntElse;  /* position fix ratio:not fix(unitsec) */
6 } NAVIINFO_DIAG_GPS_FIX_CNT;
```

NAVIINFO_DIAG_GPS_FIX_XYZ structure

```
1 typedef struct
2 {
3   int32_t   lLat;    /* GPS latitude(unit1/256sec) (+: north latitude-: south latitude) */
4   int32_t   lLon;    /* GPS longitude(unit1/256sec) (+: east longitude, -: west longitude) */
5 } NAVIINFO_DIAG_GPS_FIX_XYZ;
```

NAVIINFO_DIAG_GPS_SAT structure

```
1 typedef struct
2 {
3   NAVIINFO_DIAG_GPS_PRN   stPrn[12];  /* all satellite information */
4 } NAVIINFO_DIAG_GPS_SAT;
```

NAVIINFO_DIAG_GPS_PRN structure

```
1 typedef struct
2 {
3   uint8_t   ucRcvSts;     /* reception status */
4   uint8_t   ucPrn;        /* satellite No. */
5   uint8_t   ucelv;        /* satellite angle(unit1.0deg) */
6   uint8_t   ucLv;         /* satellite level */
7   uint16_t  usAzm;        /* satellite azimuth(unit1.0deg) (clockwise from north) */
```

```
8   uint8_t   ucReserve[2]; /* reserve */
9 } NAVIINFO_DIAG_GPS_PRN;
```

reception status(ucRcvSts)
NAVIINFO_DIAG_GPS_RCV_STS_NOTUSE not used
NAVIINFO_DIAG_GPS_RCV_STS_SEARCHING searching
NAVIINFO_DIAG_GPS_RCV_STS_TRACHING tracking
NAVIINFO_DIAG_GPS_RCV_STS_NOTUSEFIX not used for position fix
NAVIINFO_DIAG_GPS_RCV_STS_USEFIX used for position fix

NAVIINFO_NAVI_GPS structure

```
1 typedef struct
2 {
3   int32_t          altitude;   /* altitude(unit1m) */
4   uint16_t         speed;      /* speed(unit1.00km/h) */
5   uint16_t         heading;    /* heading(unit0.1deg) (clockwise from north) */
6   NAVIINFO_UTCTIME  utc;       /* UTC time */
7   uint8_t          tdsts;      /* date and time status */
8   uint8_t          reserve[3]; /* reserve */
9 } NAVIINFO_NAVI_GPS;
```

altitude(altitude)
As the altitude is used as unit [0.01m] in internal,
the altitude data range is (-21,474,839~21,474,839).
UTC(utc)
The time set after rollover.
date and time status(tdsts)
0= time has not been adjusted after GPS receiver reset(time input or master reset or CSF start)
1= time output from RTC Backup(have time adjustment result)
2= time adjustment completed

NAVIINFO_UTCTIME structure

```
1 typedef struct {
2   uint16_t    year;     /* A.D. (1~) */
3   uint8_t     month;    /* month(1~12) */
4   uint8_t     date;     /* date(1~31) */
5   uint8_t     hour;     /* hour(0~23) */
6   uint8_t     minute;   /* minute(0~59) */
7   uint8_t     second;   /* second(0~59) */
8   uint8_t     reserved; /* not used */
9 } NAVIINFO_UTCTIME;
```

**Return values:**

| | |
|---|---|
| *NAVIINFO_RET_NORMAL* | normal finish |
| *NAVIINFO_RET_ERROR_PARAM* | parameter error |
| *NAVIINFO_RET_ERROR_INNER* | internal error |
| *NAVIINFO_RET_ERROR_NOSUPPORT* | unsupported |

| | |
|---|---|
| *NAVIINFO_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter navilocinfo is NULL [NAVIINFO_RET_ERROR_PARAM]
The parameter hApp is NULL [NAVIINFO_RET_ERROR_PARAM]
Hardware environment is not *CWORD95__CWORD101___CWORD80* [NAVIINFO_RET_ERROR_NOSUPPORT]
GPS setting data size(p_data->usSize) is not same as the data that related to data ID [NAVIINFO_RET_ERROR_RESOURCE]
The count of message in message queue is reach to max [NAVIINFO_RET_ERROR_RESOURCE]
The count of mutex is reach to max [NAVIINFO_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [SENSOR_RET_ERROR_RESOURCE]
The message queue name has not been registered in control table. [NAVIINFO_RET_ERROR_INNER]
Message transfer HANDLE create failed. [NAVIINFO_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table. [NAVIINFO_RET_ERROR_INNER]
GPS setting message transfer failed between processes. [NAVIINFO_RET_ERROR_INNER]

**Classification**

Public

**Type**

Fire and ForgetMethodFire and ForgetFire and Forget

## POS_RET_API POS_SetGPStime (HANDLE *hApp*, POS_DATETIME * *pstDateTime*)

**Brief**

Temporarily set GPS time from Diag function

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *pstDateTime* | |

| | | |
|---|---|---|
| | | POS_DATETIME * - pointer of GPS time |

POS_DATETIME structure

```
1 typedef struct {
2   uint16_t  year;     /* A.D.(1~) */
3   uint8_t   month;    /* month(1~12) */
4   uint8_t   date;     /* date(1~31) */
5   uint8_t   hour;     /* hour(0~23) */
6   uint8_t   minute;   /* minute(0~59) */
7   uint8_t   second;   /* second(0~59) */
8   uint8_t   reserved; /* not used */
9 } POS_DATETIME;
```

**Return values:**

| | |
|---|---|
| *NAVIINFO_RET_NORMAL* | normal finish |
| *NAVIINFO_RET_ERROR_PARAM* | parameter error |
| *NAVIINFO_RET_ERROR_INNER* | internal error |
| *NAVIINFO_RET_ERROR_NOSUPPORT* | unsupported |
| *NAVIINFO_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the dispatcher for App are completed.
Availability of positioning service is TRUE.

**change of internal status**

There is no change of internal status

**Failure condition**

The parameter pstDateTime is NULL [POS_RET_ERROR_PARAM]
The parameter hApp is NULL [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD96___CWORD84* or *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
GPS time Data size is larger than 144 [POS_RET_ERROR_PARAM]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The message queue name has not been registered in control table. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table. [POS_RET_ERROR_INNER]
GPS time setting message transfer failed. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Fire and Forget

**POS_RET_API POS_SetLocationInfo (HANDLE *hApp*, POS_POSDATA * *pstPosData*)**

**Brief**

Set location information

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *pstPosData* | POS_POSDATA * - pointer to location information |

POS_POSDATA structure

```
1 typedef struct
2 {
3   int8_t    status;      /* data status */
4   uint8_t   posSts;      /* position status */
5   uint16_t  posAcc;      /* Position accuracy 1LSB=1m */
6   int32_t   Longitude;   /* Longitude(unit:1/128sec) */
7   int32_t   Latitude;    /* Latitude(unit:1/128sec) */
8   int32_t   altitude;    /* altitude (unit0.01m) */
9   int16_t   heading;     /* heading (unit1degree) */
10  uint8_t   reserved[2]; /* reserve */
11  } POS_POSDATA;
```

data status(status)
Do not set the invalid data(do not update data).
data valid/invalid status
Bit0 : latitude(1:valid 0:invalid, definition of maskbit:POS_LOC_INFO_LAT)
Bit1 : longitude(1:valid 0:invalid, definition of maskbit:POS_LOC_INFO_LON)
Bit2 : altitude (1:valid 0:invalid, definition of maskbit:POS_LOC_INFO_ALT)
Bit3 : heading (1:valid 0:invalid, definition of maskbit:POS_LOC_INFO_HEAD)
Bit4~7 : reserve
Position status(posSts)
It is valid only when "Get method is Navi" and "evironment is _CWORD80_".otherwise it will be set as 0
Bit0 : GPS data used result(1:used,0:not used, definition of maskbit:POS_LOC_INFO_USE_GSP)

Bit1 : DGPS data used result(1:used,0:not used, definition of maskbit:POS_LOC_INFO_USE_DGPS)
Bit2 : DR data used result(1:used,0:not used, definition of maskbit:POS_LOC_INFO_USE_DR)
Bit3 : MapMatching data use result(1:used,0:not used, definition of maskbit:POS_LOC_INFO_USE_MAPMATCHING)
Bit4~7 : reserve
Position accuracy(posAcc)
Detected accruray of current position1LSB=1m
It is valid only when "Get method is Navi" and "evironment is _CWORD80_".otherwise it will be set as 0
0000H: 0m
0001H: 1m
FFFDH: 65533m
FFFEH: 65534m and larger than 65534m
FFFFH: no data
longitude
data range: -180deg ~ +180deg
+: east longitude -: west longitude
latitude
data range-90deg ~ +90deg
+: north latitude -: south latitude
heading
data range-179deg ~ +180deg
Based on north, west(counter-clockwise) is +, east(clockwise) is -

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher
for App are completed.
Availability of positioning service is TRUE.

**change of internal status**

There is no change of internal status

**Failure condition**

The parameter pstPosData is NULL [POS_RET_ERROR_PARAM]
The parameter hApp is NULL [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]

The data status in parameter pstPosData is invalid value(0). [POS_RET_ERROR_PARAM]
The data status in parameter pstPosData is abnormal value(<15).
[POS_RET_ERROR_PARAM]
The data size is larger than specified value(POS_MSG_INFO_DSIZE)
[POS_RET_ERROR_RESOURCE]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max
[POS_RET_ERROR_RESOURCE]
The message queue name has not been registered in control table.
[POS_RET_ERROR_INNER]
Message transfer HANDLE create failed. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table. [POS_RET_ERROR_INNER]
Location data setting message transfer failed. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Fire and Forget

**See also:**

None

**POS_RET_API POS_SetLocationInfoNmea (HANDLE *hApp*, POS_LOCATIONINFO_NMEA * *locationInfo*)**

**Brief**

Set location NMEA information

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *locationInfo* | POS_POSDATA * - pointer to location information NMEA |

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *SENSOR_RET_ERROR_NOSUPPORT* | unsupported |

| | |
|---|---|
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
Hardware environment is *CWORD96___CWORD84* [POS_RET_ERROR_NOSUPPORT]
Hardware environment is *CWORD95__CWORD101___CWORD84* [POS_RET_ERROR_NOSUPPORT]
Hardware environment is not *CWORD95__CWORD101___CWORD80* , *CWORD95__CWORD101___CWORD84* and *CWORD96___CWORD84* [SENSOR_RET_ERROR_NOSUPPORT]
The length in parameter locationInfo is invalid value(0). [POS_RET_ERROR_PARAM]
The length in parameter locationInfo is abnormal value(> LOCATIONINFO_NMEA_MAX). [POS_RET_ERROR_PARAM]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The message queue name has not been registered in control table. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table. [POS_RET_ERROR_INNER]
Location NMEA data setting message transfer failed. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Fire and Forget

**See also:**

None

## POS_RET_API POS_SetSpeedInfo (HANDLE *hApp*, uint16_t *navispeed*)

**Brief**

Set speed information

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE - App Handle |
| in | *navispeed* | uint16_t - speed data[unit: 1.0km/h] |

**Return values:**

| | |
|---|---|
| *POS_RET_NORMAL* | normal finish |
| *POS_RET_ERROR_PARAM* | parameter error |
| *POS_RET_ERROR_INNER* | internal error |
| *POS_RET_ERROR_NOSUPPORT* | unsupported |
| *POS_RET_ERROR_RESOURCE* | lack of resource |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the Dispatcher for App are completed.
Availability of positioning service is TRUE.

**changes of internal status**

There is no changes of internal status

**Failure condition**

The parameter hApp is NULL [POS_RET_ERROR_PARAM]
Hardware environment is not *CWORD95__CWORD101___CWORD80* [POS_RET_ERROR_NOSUPPORT]
The data size is larger than specified value(POS_MSG_INFO_DSIZE) [POS_RET_ERROR_RESOURCE]
The count of message in message queue is reach to max [POS_RET_ERROR_RESOURCE]
The count of mutex is reach to max [POS_RET_ERROR_RESOURCE]
The count of item in ProcessName-ProcessNo convert table is reach to max [POS_RET_ERROR_RESOURCE]
The message queue name has not been registered in control table. [POS_RET_ERROR_INNER]
Message transfer HANDLE create failed. [POS_RET_ERROR_INNER]
Message transfer HANDLE get failed from internal table. [POS_RET_ERROR_INNER]
Speed data setting message transfer failed. [POS_RET_ERROR_INNER]

**Classification**

Public

**Type**

Fire and Forget

**See also:**

None

## Positioning_base_library

struct T_APIMSG_HEADER
struct T_APIMSG_MSGBUF_HEADER
struct _CWORD64_MSG_LOG_HDR
struct TimerDupData
struct RealTimeData
struct RealTimeDataBcd
struct GpsSetData
struct TimerToutMsg
struct TimerTimeReq
struct TimerAlrmReq
struct TimerStopReq
struct GpsSetTimeReq
struct SetDiffTimeReq
struct T_TIM_RCV_DATA
struct TimerSramData
struct SysTimeData
struct TimeCnvTbl_tag
struct DayCnvTbl_tag
struct TG_TIM_ROLOVR_GPSWEEKCORDATA
struct TIMECHGIND_MSG
struct T_TIM_CMDMSG
struct T_TIM_RETTIMEMSG
struct TimerModDat
struct _SYSTEMTIME
struct _SECURITY_ATTRIBUTES
struct YearCntTbl
struct DayCntTbl

**Macros**

#define **NULL** ((void *)0)
#define WTM_DUPTIME_MAX 5
*duplication timer max count*
#define TIMER_TYPE_SYN 1
*fixingcycletimer(unit: 100ms)*
#define TIMER_TYPE_USN 2
*asynchronous timer(unit: 100ms)*
#define TIMER_TYPE_ALM 5
*specific time timer*
#define SUN 0

*day*

#define MON 1

*month*

#define TUE 2

*tuesday*

#define WED 3

*wednesday*

#define THU 4

*thursday*

#define FRI 5

*friday*

#define SAT 6

*saturday*

#define DEFAULT_DIFFTIME 0x00

*time difference value of japan*

#define GPS_SAT 0x00

*correctedGPS time*

#define GPS_IST 0x01

*not corrected GPS time*

#define GPS_RTC 0x02

*RTC time.*

#define GPS_IRG 0x03

*initial time*

#define GPS_NON 0x7E

*GPS unused.*

#define GPS_IGN 0x7F

*time not received*

#define END_TIMER_RPT_OFF 0

*No continuation.*

#define END_TIMER_RPT_ON 1

*With continuation.*

#define CID_TIMER_SETGPS 0x4040

*command ID of GPS setting timer*

#define CID_TIMER_DIFSET 0x4045

*command ID of time difference setting timer*

#define CID_TIMER_CYCLE (CID)(CID_TIMER_BASE | CID_COM_SERIAL5)

*command ID of cycle data receive timer*

#define CNV_YEAR_MAX 136

*max value of conversion year(number of years + 1)*

#define SEC_PER_MIN 60

*60[sec]*

#define SEC_PER_HOUR 3600

*60[min]60[sec]*

#define SEC_PER_DAY 86400

*24[h]60[min]60[sec]*

#define TIM_ROLOVR_DISABLE 0

*GPS week rollover correct prohibition.*

#define TIM_ROLOVR_ENABLE 1

*GPS week rollover correct permission.*

#define RET_ERR_NONEINIT (-30)

*initialization untreated error*

#define RET_ERR_SEMLOCK (-31)

*get semaphore error*

#define RET_ERR_SEMUNLOCK (-32)

*open semaphore error*

#define RET_ERR_SRAMREAD (-33)

*read SRAM error*

#define TIM_NUM_SNDCMD 4

*command send number*

#define NON_TIMEUNIT_INVALID_TIME 0

*No time manager unit - system time not set.*

#define NON_TIMEUNIT_VALID_TIME 1

*No time manager unit - system time set complete.*

#define GPS_INVALID_TIME 2

*With GPS unit - system time not set.*

#define GPS_VALID_TIME 3

*With GPS unit - system time set complete.*

#define TMT_RETRY 3

*retry count*

#define TMT_RETRY_INTERVAL 500

*retry interval(500[msec])*

#define TMT_GPSDM_NAME "TIMER_GPS_DATA"

*GPS status storage data module name.*

#define TMT_GPSDM_SEM_NAME "TIM_SEM"

*GPS status storage data module semaphore name.*

#define INFINITE 0xFFFFFFFF

*infinite*

#define CID_CWORD43_BASE (CID)0xF010

*CWORD43 (Audio/_CWORD47_ Communication)*

#define CID_CWORD42_BASE (CID)0xF020

*CWORD42 (traditional CWORD36 )*

#define CID_GPS_BASE (CID)0xF030

*GPS.*

#define CID_TIMER_BASE (CID)0xF040

*timer management*

#define CID_VOUT_BASE (CID)0xF050

*voice output*

#define CID_CDCTRL_BASE (CID)0xF060

*CD control.*

#define CID_CDACS_BASE (CID)0xF070

*CD access.*

#define CID_DIAG_BASE (CID)0xF080

*diag*

#define CID_BRD_BASE (CID)0xF090

*search department*

#define CID_PCMCIA_BASE (CID)0xF0A0

*PCMCIA.*

#define CID_FM_BASE (CID)0xF0B0

*FM multiple.*

#define CID_VTHMNG_BASE (CID)0xF0D0

*sound output control*

#define CID_ETCCOM_BASE (CID)0xF0E0

*ETC Communication.*

#define CID_INITIAL_BASE (CID)0xF100

*initial*

#define CID_DEV_TIMER_BASE (CID)0xF200

*timer register driver*

#define CID_STORAGE_BASE (CID)0xF210

*storage*

#define CID_RESMGR_BASE (CID)0xF220

*ResourceManager.*

#define CID_COM_SERIAL0 (CID)0x0000

*serial command ID 0*

#define CID_COM_SERIAL1 (CID)0x0001

*serial command ID 1*

#define CID_COM_SERIAL2 (CID)0x0002

*serial command ID 2*

#define CID_COM_SERIAL3 (CID)0x0003
*serial command ID 3*

#define CID_COM_SERIAL4 (CID)0x0004
*serial command ID 4*

#define CID_COM_SERIAL5 (CID)0x0005
*serial command ID 5*

#define CID_COM_SERIAL6 (CID)0x0006
*serial command ID 6*

#define CID_COM_SERIAL7 (CID)0x0007
*serial command ID 7*

#define CID_COM_SERIAL8 (CID)0x0008
*serial command ID 8*

#define CID_COM_SERIAL9 (CID)0x0009
*serial command ID 9*

#define CID_COM_SERIAL10 (CID)0x000A
*serial command ID 10*

#define CID_COM_SERIAL11 (CID)0x000B
*serial command ID 11*

#define CID_COM_SERIAL12 (CID)0x000C
*serial command ID 12*

#define CID_COM_SERIAL13 (CID)0x000D
*serial command ID 13*

#define CID_COM_SERIAL14 (CID)0x000E
*serial command ID 14*

#define CID_COM_SERIAL15 (CID)0x000F
*serial command ID 15*

#define CID_COM_SERIAL16 (CID)0x0010
*serial command ID 16*

#define CID_COM_SERIAL17 (CID)0x0011
*serial command ID 17*

#define CID_COM_SERIAL18 (CID)0x0012
*serial command ID 18*

#define CID_COM_SERIAL19 (CID)0x0013
*serial command ID 19*

#define CID_CWORD43_CMDRCV (CID)(CID_CWORD43_BASE | CID_COM_SERIAL0)
*command reception notification*

#define CID_CWORD43_CMDSNDREQ (CID)(CID_CWORD43_BASE | CID_COM_SERIAL1)
*command send request notification*

#define CID_CWORD43_CMDSNDANS (CID)(CID_CWORD43_BASE | CID_COM_SERIAL2)

*command send result notification*

#define CID_CWORD43_ADRINF (CID)(CID_CWORD43_BASE | CID_COM_SERIAL3)

*equipment address info notification*

#define CID_CWORD42_SMALLCMDRCV (CID)(CID_CWORD42_BASE | CID_COM_SERIAL0)

*small scale command receive notify*

#define CID_CWORD42_LARGECMDRCV (CID)(CID_CWORD42_BASE | CID_COM_SERIAL1)

*large scale command receive notify*

#define CID_CWORD42_BUFFREQ (CID)(CID_CWORD42_BASE | CID_COM_SERIAL2)

*large scale buffer request*

#define CID_CWORD42_BUFFANS (CID)(CID_CWORD42_BASE | CID_COM_SERIAL3)

*large scale buffer response*

#define CID_CWORD42_CMDSNDREQ (CID)(CID_CWORD42_BASE | CID_COM_SERIAL4)

*command send request notify*

#define CID_CWORD42_CMDSNDANS (CID)(CID_CWORD42_BASE | CID_COM_SERIAL5)

*command send answers notify*

#define CID_CWORD42_SNDSTOPIND (CID)(CID_CWORD42_BASE | CID_COM_SERIAL6)

*send data cancel indication*

#define CID_CWORD42_LANSTATE (CID)(CID_CWORD42_BASE | CID_COM_SERIAL7)

*LAN state notify.*

#define CID_CWORD42_DEVENTRY (CID)(CID_CWORD42_BASE | CID_COM_SERIAL8)

*info change notify*

#define CID_GPS_CMDSNDREQ (CID)(CID_GPS_BASE | CID_COM_SERIAL1)

*command send request notify*

#define CID_GPS_CMDRCVANS (CID)(CID_GPS_BASE | CID_COM_SERIAL2)

*command receive answers notify*

#define CID_GPS_RETRESET (CID)(CID_GPS_BASE | CID_COM_SERIAL3)

*GPS reset response.*

#define CID_TIMER_TIMREQ (CID)(CID_TIMER_BASE | CID_COM_SERIAL0)

*timer start indication*

#define CID_TIMER_ALMREQ (CID)(CID_TIMER_BASE | CID_COM_SERIAL1)

*timer start indication with specified time*

#define CID_TIMER_STPREQ (CID)(CID_TIMER_BASE | CID_COM_SERIAL2)

*timer stop indication*

#define CID_TIMER_DUPREQ (CID)(CID_TIMER_BASE | CID_COM_SERIAL3)

*multi timer start indication*

#define CID_TIMER_TOUT (CID)(CID_TIMER_BASE | CID_COM_SERIAL4)

*timeout notify*

#define CID_TIMER_TIMCHG (CID)(CID_TIMER_BASE | CID_COM_SERIAL5)

*time change notify*

#define CID_TIMER_ALMANAC (CID)(CID_TIMER_BASE | CID_COM_SERIAL6)

*almanac abnormality notify*

#define CID_TIMER_TIME_VALID (CID)(CID_TIMER_BASE | CID_COM_SERIAL7)

*time enable notify*

#define CID_TIMER_SETTIME (CID)(CID_TIMER_BASE | CID_COM_SERIAL8)

*GPS time setting response notify.*

#define CID_CLKMNG_SETTIME (CID)(CID_TIMER_BASE | CID_COM_SERIAL9)

*time setting Indication*

#define CID_CLKMNG_NOTIACCURACY (CID)(CID_TIMER_BASE | CID_COM_SERIAL10)

*notify*

#define CID_CLKMNG_NOTIADJUST (CID)(CID_TIMER_BASE | CID_COM_SERIAL11)

*time setting change notify(unused)*

#define CID_CLKMNG_NOTIOBSERVERS (CID)(CID_TIMER_BASE | CID_COM_SERIAL12)

*fixingcyclenotify*

#define CID_CLKMNG_NOTIACCURACY_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL13)

*notify entry*

#define CID_CLKMNG_NOTIADJUST_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL14)

*time setting change notify entry(unused)*

#define CID_CLKMNG_NOTIOBSERVERS_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL15)

*fixingcyclenotify entry*

#define CID_CLKMNG_AUTOSETREALTIME_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL16)

*fixingcyclenotify entry*

#define CID_CLKMNG_SETTIMEZONE (CID)(CID_TIMER_BASE | CID_COM_SERIAL17)

*fixingcyclenotify entry*

#define CID_CLKMNG_AUTOSETTIMEZONE_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL18)

*fixingcyclenotify entry*

#define CID_CLKMNG_SETREALTIME (CID)(CID_TIMER_BASE | CID_COM_SERIAL19)

*fixingcyclenotify entry*

#define CID_CDCTRL_SPECIFYDRIVE (CID)(CID_CDCTRL_BASE | CID_COM_SERIAL0)

*< unused*

#define CID_CDCTRL_CD_STATUS (CID)(CID_CDCTRL_BASE | CID_COM_SERIAL1)

*CD status notify.*

#define CID_CDACS_RET_MAP_MAN_PTR (CID)(CID_CDACS_BASE | CID_COM_SERIAL0)

*info address return*

#define CID_CDACS_RET_FILE_PTR (CID)(CID_CDACS_BASE | CID_COM_SERIAL1)

*file read address return*

#define CID_CDACS_RET_MALLOC (CID)(CID_CDACS_BASE | CID_COM_SERIAL2)

*malloc return*

#define CID_CDACS_RET_CD_READ (CID)(CID_CDACS_BASE | CID_COM_SERIAL3)

*CD read return.*

#define CID_DIAG_MODENOTC (CID)(CID_DIAG_BASE | CID_COM_SERIAL0)

*normal diag transition Indication*

#define CID_DIAG_SYSCHKREQ (CID)(CID_DIAG_BASE | CID_COM_SERIAL1)

*diag system check request*

#define CID_BRD_IDRET (CID)(CID_BRD_BASE | CID_COM_SERIAL0)

*ID obtain response.*

#define CID_PCMCIA_CARD (CID)(CID_PCMCIA_BASE | CID_COM_SERIAL0)

*PCMCIA command receive notify.*

#define CID_FMDRV_DATARCV (CID)(CID_FM_BASE | CID_COM_SERIAL0)

*data receive end notify*

#define CID_VTH_MNG_VOICECTREND (CID)(CID_VTHMNG_BASE | CID_COM_SERIAL0)

*control sound end notify*

#define CID_ETC_CMDRCV (CID)(CID_ETCCOM_BASE | CID_COM_SERIAL0)

*ETC receive notify.*

#define CID_ETC_CMDSNDANS (CID)(CID_ETCCOM_BASE | CID_COM_SERIAL1)

*ETC send result notify.*

#define CID_INI_ACCOFF (CID)(CID_INITIAL_BASE | CID_COM_SERIAL0)

*ACC-OFF Indication.*

#define CID_INI_SD_INOUT (CID)(CID_INITIAL_BASE | CID_COM_SERIAL1)

*SD insertion notify.*

#define CID_INI_MAPMEDIA_STATUS (CID)(CID_INITIAL_BASE | CID_COM_SERIAL2)

*map media state notify*

#define CID_INI_ACCOFF_COMP (CID)(CID_INITIAL_BASE | CID_COM_SERIAL3)

*ACC-OFF process complete notify.*

#define CID_INI_STEPFORK (CID)(CID_INITIAL_BASE | CID_COM_SERIAL4)

*step start Indication*

#define CID_INI_STEPFORK_COMP (CID)(CID_INITIAL_BASE | CID_COM_SERIAL5)

*step start complete Indication*

#define CID_INI_SYSTEMERR (CID)(CID_INITIAL_BASE | CID_COM_SERIAL6)

*system abnormality notify*

#define CID_INI_INTERNAL_ACCOFF_START (CID)(CID_INITIAL_BASE | CID_COM_SERIAL7)

*internal ACC-OFF start Indication*

#define CID_INI_INTERNAL_ACCOFF_COMP (CID)(CID_INITIAL_BASE | CID_COM_SERIAL8)

*internal ACC-OFF complete Indication*

#define CID_DEV_TIMER_UP (CID)(CID_DEV_TIMER_BASE | CID_COM_SERIAL0)

*timer timeout notify*

#define CID_DEV_TIMER_START (CID)(CID_DEV_TIMER_BASE | CID_COM_SERIAL1)

*timer start Indication*

#define CID_DEV_TIMER_STOP (CID)(CID_DEV_TIMER_BASE | CID_COM_SERIAL2)

*timer stop Indication*

#define CID_STRMGR_DEVINSERTNOT (CID)(CID_STORAGE_BASE | CID_COM_SERIAL0)

*device insertion notify*

#define CID_STRMGR_DEVFIXNOT (CID)(CID_STORAGE_BASE | CID_COM_SERIAL1)

*device fix notify*

#define CID_STRMGR_DEVERRNOT (CID)(CID_STORAGE_BASE | CID_COM_SERIAL2)

*device abnormality notify*

#define CID_USBCTL_CONNDEV_NTF (CID)(CID_STORAGE_BASE | CID_COM_SERIAL6)

*USB Device Connect Notify.*

#define CID_USBCTL_CTRLPOWER_IND (CID)(CID_STORAGE_BASE | CID_COM_SERIAL7)

*Control Device Power Indication.*

#define CID_USBCTL_COMPTERM (CID)(CID_STORAGE_BASE | CID_COM_SERIAL8)

*USB observation complete notify.*

#define CID_USBCTL_CONNECTSTS (CID)(CID_STORAGE_BASE | CID_COM_SERIAL9)

*USB connect status notify.*

#define CID_USBCTL_INSERTSTS (CID)(CID_STORAGE_BASE | CID_COM_SERIAL10)

*USB insertion notify.*

#define CID_USBCTL_ERRORSTS (CID)(CID_STORAGE_BASE | CID_COM_SERIAL11)

*USB abnormality notify.*

#define CID_SDSDRV_INSERTSTS (CID)(CID_STORAGE_BASE | CID_COM_SERIAL12)

*SD insertion notify.*

#define CID_SDSDRV_COMPTERM (CID)(CID_STORAGE_BASE | CID_COM_SERIAL13)

*SD observation complete notify.*

#define CID_SDSDRV_CONNDEV_NTF (CID)(CID_STORAGE_BASE | CID_COM_SERIAL14)

*SD Device Connect Notify::GPF_001#.*

#define CID_RESMGR_ERROR_INFO_CHG (CID)(CID_RESMGR_BASE | CID_COM_SERIAL0)

*error info change notify*

#define CID_DEV_REQGPS MAKECID(CID_DEVHIGH, 0x00)

*GPS request.*

#define CID_DEV_REQRST MAKECID(CID_DEVHIGH, 0x01)

*gps reset request*

#define CID_GPS_SERIAL0 (CID)0x0100

*GPS reset request func definition.*

#define CID_GPS_REQRESET (CID)(CID_GPS_BASE | CID_GPS_SERIAL0)

*GPS reset request CID.*

#define DEBUG_DUMP_MAX_SIZE 4096

*_CWORD33_OnDebugDump max size*

#define KIND_DEBUG (0x00000001)

*for debug log*

#define KIND_ERROR (0x00000002)

*for error output log*

#define KIND_RELEASE (0x00000004)

*for release output log*

#define FLAG_IMPORTANT KIND_RELEASE

*flag for important log*

#define POS_DEBUG_LOGOUT_LEVEL 4

*debug log output level*

#define **DEBUG**

#define _CWORD33_LOG(zone, opt, fmt, ...)

*CWORD33 log output*

#define POS_DEBUG_LOGLEV_1 1

*debug log output level 1*

#define POS_DEBUG_LOGLEV_2 2

*debug log output level 2*

#define POS_DEBUG_LOGLEV_3 3

*debug log output level 3*

#define POS_DEBUG_LOGLEV_4 4

*debug log output level 4*

#define POS_DEBUG_LOGLEV_5 5

*debug log output level 5*

#define POS_SENSLOG_TYPE_NONE 0

*sensor log type NONE*

#define POS_SENSLOG_TYPE_SYS 1

*sensor log type SYS*

#define POS_SENSLOG_TYPE_GPS 2

*sensor log type GPS*

#define POS_SENSLOG_TYPE_NAV 3

*sensor log type NAV*

#define POS_SENSLOG_TYPE_CMD 4

*sensor log type CMD*

#define SNR_FUNC_DEBUG_MSG_CWORD71_FILE 0

*Sensor file Log valid:1,invalid:0.*

#define SNR_FUNC_DEBUG_MSG_CWORD71_ 1

*Sensor Log valid:1,invalid:0.*

#define GPS_FUNC_DEBUG_MSG_CWORD71_ 1

*GPS Log valid:1,invalid:0.*

#define COM_FUNC_DEBUG_MSG_CWORD71_ 1

*COMMON Log valid:1,invalid:0.*

#define BASE_FUNC_DEBUG_MSG _CWORD71_ 1

*BASE Log valid:1,invalid:0.*

#define MSG_FUNC_DEBUG_MSG _CWORD71_ 1

*MESSAGE Log valid:1,invalid:0.*

#define DEV_FUNC_DEBUG_MSG _CWORD71_ 1

*GPS DEV Log valid:1,invalid:0.*

#define RELTEXT(FLAG, QUOTE) __FUNCTION__, (QUOTE)

*output text format*

#define WPF_RETAILMSG_LEV(level, zone, ...) (((level) <= (POS_DEBUG_LOGOUT_LEVEL))?((void)((_CWORD33_LOG(zone, __VA_ARGS__)))):((void)(0)))

*message log output*

#define POS_WRITE_SENSLOG(zone, data, len) (((zone) != (0))?((void)((_CWORD33_LOG_DATA(zone, data, len)))):((void)(0)))

*write sensor log output*

#define WPF_RETAILMSG(PRINTF_FMT) ((void)(0))

*message log output*

#define POS_SNR_GET_ZONE(level)

*get sensor zone code*

#define POS_GPS_GET_ZONE(level)

*get GPS zone code*

#define POS_BASE_GET_ZONE(level)

*get base zone code*

#define POS_COM_GET_ZONE(level)

*get common zone code*

#define POS_MSG_GET_ZONE(level)

*get message zone code*

#define POS_DEV_GET_ZONE(level)

*get DEV zone code*

#define POS_SENSLOG_GET_ZONE(typ)

*get sensor log zone code*

#define POS_SNR_EXLOG(level, ...) _CWORD33_LOG(POS_SNR_GET_ZONE(level), __VA_ARGS__);

*sensor extension log output*

#define POS_GPS_EXLOG(level, ...) _CWORD33_LOG(POS_GPS_GET_ZONE(level), __VA_ARGS__);

*GPS extension log output.*

#define POS_BASE_EXLOG(level, ...) _CWORD33_LOG(POS_BASE_GET_ZONE(level), __VA_ARGS__);

*Base extension log output.*

#define POS_COM_EXLOG(level, ...) _CWORD33_LOG(POS_COM_GET_ZONE(level), __VA_ARGS__);

*Common extension log output.*

#define POS_MSG_EXLOG(level, ...) _CWORD33_LOG(POS_MSG_GET_ZONE(level), __VA_ARGS__);

*message extension log output*

#define POS_DEV_EXLOG(level, ...) __CWORD33_LOG(POS_DEV_GET_ZONE(level), __VA_ARGS__);

*DEV extension log output.*

#define POS_SENSLOG(type, data, len) printf("POS_SENSLOG\n");

*sensor log output*

#define PNO_NONE 0x0000

*initial value*

#define PNO__CWORD88__CWORD23 0x0100

*CWORD23 primary thread*

#define PNO_RDSDC_THREAD 0x0130

*RDS decoder control thread.*

#define PNO_RDSRCV_THREAD 0x0132

*RDS data communication thread.*

#define PNO_TMC_DATAMNG 0x0133

*TMC data manager.*

#define PNO__CWORD24__MAIN 0x01F0

*CWORD24 main thread*

#define PNO_DEV_TOUCH_READ 0x0202

*touch input control*

#define PNO_CLOCK_TIME_STAMP 0x0208

*clock timestamp thread*

#define PNO_CLK_MNG 0x0209

*time manager thread*

#define PNO_DEV_MNG_MAIN 0x020A

*device manager thread*

#define PNO_DEV_SYSCOM_RCV 0x0210

*Navi-SYS communication/reception control.*

#define PNO_DEV_SYSCOM_SND 0x0211

*Navi-SYS communication/transmission control.*

#define PNO_DEV_SYSCOM_TIMER 0x0212

*Navi-SYS communication driver timer thread.*

#define PNO_DEV_SYSCOM_MAIN 0x0213

*Navi-SYS communication driver main thread.*

#define PNO_DEV_MCSUB_MAIN 0x0215

*SUB communication control driver main thread.*

#define PNO_DEV_MCSUB_RECV 0x0216

*SUB communication control driver receive thread.*

#define PNO_CDFS_MAIN 0x0235

*CDFS thread.*

#define PNO_DVDFS_MAIN 0x0236

*DVDFS thread.*

#define PNO_SIMPLE_LOADER 0x0237

*loader thread*

#define PNO_FLSMNG_THREAD 0x02DA

*backup data manager thread*

#define PNO_DEV_CWORD57_MODEL 0x02E0

*_CWORD57_Model thread*

#define PNO_DEV_CWORD57_COMRCV 0x02E1

*CWORD57 command reception thread*

#define PNO_DEV_CWORD57_MONITOR 0x02E2

*CWORD57 communication monitor thread*

#define PNO_DEV_CWORD57_LIST 0x02E5

*CWORD57 list manager thread*

#define PNO_DEV_CWORD57_TAG_WRITER 0x02E6

*Tagging data read thread.*

#define PNO_DEV_CWORD57_AUTH 0x02EE

*CWORD57 certification control primary thread*

#define PNO_VIDEODECORD 0x02F8

*video decoder PNO*

#define PNO_DEV_USBCTL_MONITOR 0x02F9

*USB monitor driver.*

#define PNO_CMRDRVTHREAD 0x02FA

*camera driver thread*

#define PNO_DEV_SDSDRV 0x02FE

*SD monitor driver.*

#define PNO_GINI_ERRLOG 0x0303

*error log HDD storage thread*

#define PNO_GINI_VERUP 0x0306

*version up thread*

#define PNO_DEV_SDAS_DIAG 0x0312

*SDAS driver diag register thread.*

#define PNO_DEV_SDAS_IST 0x0313

*SDAS driver IST.*

#define PNO_DEV_SDAS_DRV 0x0314

*SDAS driver mount thread.*

#define PNO_DEV_SDAS_TIMER 0x0315

*SDAS driver timer thread.*

#define PNO_VEHICLE_SENSOR 0x0363

*vehicle sensor thread*

#define PNO_LINE_SENS_DRV 0x0364
*line sensor driver thread*

#define PNO_CAN_COM_PROT 0x0365
*CAN thread.*

#define PNO_CANCMR_MIDDLE 0x0366
*CAN camera middle thread.*

#define PNO_VEHICLE_INFO 0x0367
*vehicle information#053#*

#define PNO_VEHICLE_SWMANAGER 0x0368
*vehicle information#053#*

#define PNO_VEHICLE_VEHICLE 0x0371
*vehicle information#053# #GPF_60_001*

#define PNO_VEHICLE_TOUCH_INT 0x0372

#define PNO_SCC_SOUND_STOP 0x0383
*sound output end detection*

#define PNO_SCCSOUND_PLAY 0x0384
*SCCSound playback thread.*

#define PNO_SCCSOUND_SYNTH 0x0385
*SCCSound composition thread.*

#define PNO_VUP_CTRL 0x0394
*VUP control thread.*

#define PNO_VUPCTRL_CHK 0x0304
*version judgement thread*

#define PNO_CWORD87_LAN 0x03AB
*CWORD87 -LAN thread#015#*

#define PNO_CWORD38 0x03AC
*CWORD39 thread#017#*

#define PNO_LAN_SERVER 0x03AD
*LAN server thread#018#.*

#define PNO_LAN_CWORD105 0x03AE
*LAN CWORD105*

#define PNO_LAN_CWORD105_DUMMY 0x03AF
*LAN _CWORD105_dummy.*

#define PNO_MLB 0x03C0
*MLB.*

#define PNO_CWORD65 0x03C1
*CWORD65*

#define PNO_MLB_INTR 0x03C2
*MediaLB Peripheralinterruptionthread.*

#define PNO_DIRECTSHOW_MONITOR_MAIN 0x03B0

*Dshow monitor main thread.*

#define PNO_DIRECTSHOW_MONITOR_SUB 0x03B1

*Dshow monitor sub thread.*

#define PNO_USBAUDIO_WORKER_DIRECTSHOW 0x03E1

#define PNO_USBAUDIO_WORKER_TIMER 0x03E2

*USB audio worker thread(Timer)#008#.*

#define PNO_USBAUDIO_FILE_LIST 0x03E3

*USB audio file list thread#009#.*

#define PNO_USBAUDIO_MAIN 0x03E4

*USB audio main thread#010#.*

#define PNO_DISPMNG_MAIN 0x03F8

*image manager process main thread*

#define PNO_STRMNG_MAIN 0x03F9

*storage manager thread*

#define PNO_STRMNG_JUDGE 0x03FA

*storage device judgement thread*

#define PNO_CAM_MIDDLE 0x03FB

*camera middle thread*

#define PNO_ERR_MONITOR 0x03FC

*painting/image error monitor thread#001#*

#define PNO_DISPMNG_SINK_IO 0x03FD

*display manager_SINK_I/O thread#006#*

#define PNO_STR_DEV_LOGGER 0x03FE

*storage device logger thread#042#*

#define PNO_CTPMONITOR 0x03FF

*capacitance panel monitor thread#054#*

#define PNO_NAVI_STM_MAIN 0x0430

*storage manager thread*

#define PNO_NAVI_GPS_MAIN 0x0440

*GPS communication manager thread.*

#define PNO_NAVI_GPS_RCV 0x0441

*GPS receive thread.*

#define PNO_NAVI_LOCA_SENS 0x0442

*sensor thread*

#define PNO_NAVI_GPS_INT 0x0443

*GPS interruption thread#031#.*

#define PNO_NAVI_PROXY 0x0444

*Navi proxy thread#050#.*

#define PNO_NAVI_CORE 0x0445

*Navi core thread#050#.*

#define PNO_NAVI_CORE_IF 0x0446

*Navi core IF thread#050#.*

#define PNO_NAVI_MVACN_MID 0x0454

*voice recognition middle thread#002#*

#define PNO_RESMGR_THREAD_MAIN 0x0460

*resource manager thread::GPF_12_001#*

#define PNO_NAVI_DIAG_MANAGER 0x04A0

*diag manager thread*

#define PNO_NAVI_DIAG_SUB 0x04A1

*diag sub thread*

#define PNO_NAVI_SYSLOG_HDD 0x04A2

*SYS log HDD storage thread.*

#define PNO_DIAG_WORKER 0x04A3

*diag worker thread#016#*

#define PNO_BT_CMD 0x0530

*BT command send/receive manager thread.*

#define PNO_BT_MNG 0x0531

*BT control thread.*

#define PNO_BTAUD_MNG 0x0532

*BT audio control thread.*

#define PNO_BT_AVP 0x0533

*BTAVP thread.*

#define PNO_PB_MNG 0x0534

*telephone book thread*

#define PNO_PB_VCARD 0x0535

*telephone book VCARD thread*

#define PNO_BTCOM_MNG 0x0540

*BT communication manager thread.*

#define PNO_BTCOM_PRT 0x0541

*BT communication protocol thread.*

#define PNO_BTCOM_RCV 0x0542

*BT communication reception thread.*

#define PNO_VGUIDE_SNDSRCMNG 0x06FF

*register manager thread#021#*

#define PNO_ANA_THREAD 0x0769

*analysis support thread#020#*

#define PNO_COVEROS_OPERATE_BASE_OUT 0x07B0

*playback/recording thread(play base No.)*

#define PNO_COVEROS_OPERATE_OUT_MAX 0x07B9

*playback/recording thread(maximum Num of playback)*

#define PNO_COVEROS_OPERATE_BASE_IN 0x07C0

*playback/recording thread(input base No.)*

#define PNO_COVEROS_OPERATE_IN_MAX 0x07C9

*playback/recording thread(maximum Num of inputs)*

#define PNO_SOUNDAGENT_BASE 0x07D0

*sound agent thread base No.*

#define PNO_SOUNDAGENT_MAX 0x07E9

*Maximum Num of sound agent thread.*

#define PNO_PHONEME2PCMABORT 0x07FA

*voice composition interruption thread*

#define PNO_SOUND_DEVCTRL 0x07FB

*device control thread*

#define PNO_SOUND_INPUTCTRL 0x07FC

*input manager thread*

#define PNO_VGUIDE_SNDCTRL_BASIC_F 0x07FD

*playback manager thread(before seat basis playback)*

#define PNO_VGUIDE_SNDCTRL_BASIC_R 0x07FE

*playback manager thread(behind seat basic playback)*

#define PNO_VGUIDE_SNDCTRL_INTRPT 0x07FF

*playback manager thread(interruption voice)*

#define PNO__CWORD83_ 0x0900

*CWORD83 (Intersystem Communication Message)thread*

#define PNO_DOMAIN_SOUND 0x0770

*Sound thread #057#.*

#define PNO_DOMAIN_BLUETOOTH 0x0771

*BlueTooth thread #057#.*

#define PNO_DOMAIN_CAMERA 0x0772

*Camera thread #057#.*

#define PNO_DOMAIN_DIAG 0x0773

*Diag thread #057#.*

#define PNO_DOMAIN__CWORD57_ 0x0774

*CWORD57 thread #057#*

#define PNO_DOMAIN_SPEECHREC 0x0775

*SpeechRec thread #057#.*

#define PNO_DOMAIN_USBAUDIO 0x0776

*UsbAudio thread #057#.*

#define PNO_DOMAIN_SENSOR 0x0777

*Sensor thread #057#.*

#define PNO_DOMAIN_CWORD24 0x0778

*CWORD24 thread #057#*

#define PNO_DOMAIN_COMMUNICATION 0x0779

*Communication thread #057#.*

#define PNO_DOMAIN_NAVICORE 0x077A

*navi core thread*

#define PNO_DOMAIN_NAVIPROXY 0x077B

*navi proxy thread*

#define PNO_DOMAIN_ANALYSIS 0x077C

*Analysis thread #057#.*

#define PNO_DOMAIN_STORAGE 0x077D

*Storage thread #057#.*

#define PNO_DOMAIN_VEHICLE 0x077E

*VEHICLE thread #057#.*

#define PNO_DOMAIN_GRAPHICS 0x077F

*Graphics thread #057#.*

#define PNO_DOMAIN_VUP_CTRL 0x0780

*VUP_Ctrl thread #057#.*

#define PNO_DOMAIN_VEHICLEINFO 0x0781

*VehicleInfo thread #057#.*

#define PNO_DOMAIN_RESOURCEMANAGER 0x0782

*ResourceManager thread #057#.*

#define PNO_DOMAIN_DEVICEMANAGER 0x0783

*DeviceManager thread #057#.*

#define PNO_DOMAIN_PERIPHERAL 0x0784

*Peripheral thread #057#.*

#define PNO_DOMAIN_WDT 0x0785

*WDT thread #057#.*

#define PNO_DOMAIN_VUP 0x0786

*VUP thread #057#.*

#define PNO_DOMAIN_TIMERENTRY 0x0787

*TimerEntryDrv #060#.*

#define PNO_DOMAIN_LOADER 0x0788

*Loader thread #GPF_11_008.*

#define PNO_DOMAIN_WLAN 0x078A

*WLAN thread.*

#define PNO_DOMAIN_WLANMANAGER 0x078B

*WLAN Manager thread.*

#define PNO_DOMAIN_WLANADAPTER 0x078C

*WLAN Adapter thread #061.*

#define PNO_DOMAIN_WLANRECEIVER 0x078D

*WLAN Receiver thread #061.*

#define PNO_DOMAIN_HDRADIO 0x0820

*RadioAppthread#066 #GPF_10_022.*

#define PNO_HDRADIO_MAIN 0x0821

*RadioApp Main thread#066 #GPF_10_022.*

#define PNO_HDRADIO_CWORD51_RCVR 0x0822

*RadioApp _CWORD51_Rcvr thread#066 #GPF_10_022.*

#define PNO_DOMAIN_RADIO 0x0830

*radio.exe thread#066 #GPF_10_022*

#define PNO_RADIO_MAIN 0x0831

*radio main thread#067*

#define PNO_RADIO_BACKUP 0x0832

*radio backup manager thread#067*

#define PNO_RADIO_CDDB 0x0833

*radio CDDB manager thread#067*

#define PNO_RADIO_SYS_COMM 0x0834

*radio SYS communication manager thread#067*

#define PNO_DOMAIN_CD 0x0840

*CD middle thread#068.*

#define PNO_CD_MAIN 0x0841

*CD middle main thread#068.*

#define PNO_DOMAIN_CONTENTMGR 0x0850

*contents manager thread#068*

#define PNO_CONTENTMGR_MAIN 0x0851

*contents manager main thread#068*

#define PNO_CONTENTMGR_CDDB 0x0852

*contents manager CDDB thread#068*

#define PNO_CONTENTMGR_CONTENTS_DB 0x0853

*contents manager contentsDB thread#068*

#define PNO_CONTENTMGR_CONTENTS_DB_WORKER 0x0854

*contents manager contentsDB worker thread#068*

#define PNO_DOMAIN_XM 0x0860

*XM middle thread#068.*

#define PNO_XM_MAIN 0x0861

*XM middle main thread#068.*

#define PNO_XM_FILE_ACCESS 0x0862

*XM middle file access thread#068.*

#define PNO_XM_BACKUP 0x0863

*XM middle backup thread#068.*

#define PNO_XM_PACKET_DECODE 0x0864

*XM middle packet decode thread#068.*

#define PNO_XM_CDDB 0x0865

*XM middle CDDB access thread#068.*

#define PNO_CWORD24_JPEG_DECODE 0x01F1

*CWORD24 JPEG decode process thread#068*

#define PNO_USBAUDIO_AVIAGE 0x03E5

*USB middle Aviage reception thread#068.*

#define PNO_USBAUDIO_CWORD69 0x03E6

*USB middle CWORD69 reception thread#068.*

#define PNO_USBAUDIO_LIST_ASYNCHRONOUS 0x03E7

*USB middle list asynchronous thread#068.*

#define PNO_USBAUDIO_ARTWORK 0x03E8

*USB middle art worker task thread#068.*

#define PNO_DOMAIN_BTSTACK 0x08B0

*BluetoothStack#071.*

#define PNO_BTSTACK_ADAPTER 0x08B1

*BTStackAdapter_thread#071.*

#define PNO_BTSTACK_OSSTACK 0x08B2

*OS_StackThread#071.*

#define PNO_BTSTACK_UARTWT 0x08B3

*UART_WriteThread#071.*

#define PNO_BTSTACK_UARTRD 0x08B4

*UART_ReadThread#071.*

#define PNO_BTSTACK_OSAT 0x08B5

*OSAPIEx_Timer_Thread#071.*

#define PNO_BTSTACK_OSWAIT 0x08B6

*OS_WaitThread#071.*

#define PNO_DOMAIN_BTSRV 0x08C0

*BTService thread#071.*

#define PNO_BTSRV_SYSMGR 0x08C1

*bt_srv_sysmgr_threadMain thread#071*

#define PNO_BTSRV_CONNECTION 0x08C2

*bt_srv_con_threadMain thread#071*

#define PNO_BTSRV_HTP 0x08C3

*bt_srv_hfp_threadMain thread#071*

#define PNO_BTSRV_AVP 0x08C4

*bt_srv_avp_threadMain thread#071*

#define PNO_BTSRV_STRM 0x08C5

*bt_srv_strm_threadMain thread#071*

#define PNO_BTSRV_PBAP 0x08C6

*bt_srv_pbap_threadEntry thread#071*

#define PNO_BTSRV_ANLZ 0x08C7

*bt_srv_anlz_threadEntry thread#071*

#define PNO_BTSRV_TRNS 0x08C8

*bt_srv_trns_threadEntry thread#071*

#define PNO_BTSRV_EDIT 0x08C9

*bt_srv_edit_threadEntry thread#071*

#define PNO_BTSRV_MAP 0x08CA

*bt_srv_map_threadEntry thread#071*

#define PNO_BTSRV_MAPDATAPARSE 0x08CB

*bt_srv_map_DataParse_threadEntry thread#071*

#define PNO_BTSRV_MULTIENTRY 0x08CC

*bt_srv_multi_entry thread#071*

#define PNO_BTPM_IFCAVP 0x08D1

*btpm_ifc_avp_thread#071*

#define PNO_BTPM_DIST 0x08D2

*btpm_dist_thread#071*

#define PNO_BTPM_IFCHFP 0x08D3

*btpm_ifc_hfp_thread#071*

#define PNO_BTPM_IFCMAP 0x08D4

*tpm_ifc_map_thread#071*

#define PNO_BTPM_MPC 0x08D5

*btpm_mpc_thread#071*

#define PNO_BTPM_IFCPB 0x08D6

*btpm_ifc_pb_thread#071*

#define PNO_BT_APP 0x0800

*BT app thread.*

#define PNO_BT_STACK 0x0801

*BT stack adapter thread.*

#define PNO_BT_WAVE_IN 0x0804

*BT_HFP_WaveIn_thread.*

#define PNO_BT_WAVE_OUT 0x0805

*BT_HFP_WaveOut_thread.*

#define PNO_GGX_DISP_MAIN PNO_DISPMNG_MAIN

*Graphics image main control thread #065#.*

#define PNO_GGX_DISP_SEQ 0x0811

*image sequence control thread #065#*

#define PNO_GGX_DISP_DEV PNO_DISPMNG_SINK_IO

*image reception thread #065#*

#define PNO_GGX_DISP_VCAP 0x0813

*Graphics VideoCapture thread#065#.*

#define PNO_TGWCOM 0x0229

*SPI communication thread.*

#define TSKID_TGWCOM PNO_TGWCOM

*TGW communication.*

#define TSKID_TIM PNO_DEV_SYSCOM_TIMER

*timer thread*

#define MBXID_TGWCOM PNO_TGWCOM

*SPI communication thread.*

#define PNO_MLBSYNC 0x03C4

*MLB_Sync.*

#define PNO_MLBSYNC_INTR 0x03C5

*MLB_Sync_Intr.*

#define SYS_PNO_MIN PNO__CWORD88__CWORD23_

*minimum PNO.(normal process)*

#define SYS_PNO_MAX PNO__CWORD83_

*maximum PNO.(normal process)*

#define PNO_TMN__CWORD69_ PNO_GINI_MAIN

*CWORD69 taskmgr.exe*

#define PNO_TMN__CWORD51__BT 0x0320

*_CWORD51__BT taskmgr.exe*

#define PNO_TMN__CWORD51__DR 0x0321

*_CWORD51__DR taskmgr.exe*

#define PNO_TMN__CWORD51__HD_Radio 0x0322

*_CWORD51__HD-Radio taskmgr.exe*

#define PNO_TMN__CWORD51__CWORD105_ 0x0323

*CWORD51___CWORD105 taskmgr.exe*

#define PNO_LAN__CWORD105__AUDIO 0x03C6

*CWORD105 Audio*

#define PNO_LAN__CWORD105__INTR 0x03D0

*LAN _CWORD105__INTR.*

#define PNO__CWORD65__INTR 0x03D1

*CWORD65 INTR*

#define PNO_CLK_GPS 0x0316

*clock GPS thread*

#define PNO_DEV_EXT_INT 0x0267

*external INT driver*

#define PNO_GSYS_CMSG 0x0311

*message change thread*

#define PNO_DEV_RUNTIMEADD 0x0204

*run time addition thread*

#define PNO_DEV_TIMER_ENTRY 0x02F0

*timer entry driver thread*

#define PNO_GINI_MAIN 0x0300

*G initial.*

#define **CEPC_EM_NOHDD**

#define **PSL_USED**

#define UNIT_TYPE_NONE 0x00000000UL

*type none*

#define UNIT_TYPE_UNKNOWN 0x00000001UL

*type unknown*

#define UNIT_TYPE_CWORD71_CWORD96_CWORD84 0x00000002UL

*type CWORD96 CWORD84*

#define UNIT_TYPE_CWORD71_CWORD95_CWORD101_CWORD84 0x00000004UL

*type CWORD95 /_CWORD101_ CWORD84*

#define UNIT_TYPE_CWORD71_CWORD95_CWORD101_CWORD80 0x00000008UL

*type CWORD95 /_CWORD101_ CWORD80*

#define UNIT_TYPE_CWORD71_CWORD86 0x00000010UL

*type CWORD86*

#define MAX_NAME_LEN 32

*name max length*

#define _CWORD31_NO_SIZE 16

*CWORD31 No size*

#define RESERVE1_SIZE 16

*reserve 1*

#define HELPNET_ID_SIZE 21

*helpnet ID size*

#define RESERVE2_SIZE 11

*reserve 2*

#define DATE_SIZE 32

*make time size*

#define SERIAL_NO_SIZE 3

*serial No size*

#define PRODUCT_NAME_SIZE 1

*product name size*

#define **_pb_memcpy** memcpy
#define **_pb_memset** memset
#define **_pb_Exit**() _pb_Exit_d(__func__, __LINE__)
#define RET_NORMAL 0

*normal return*

#define RET_ERROR (-1)

*error return*

#define RET_ERRPARAM (-2)

*parameter error*

#define SAPI_EVSET_ABSOLUTE 1

*absolute vale set*

#define SAPI_EVWAIT_VAL 1

*event wait value set*

#define **TRUE** (1)
#define **FALSE** (!TRUE)

**Typedefs**

typedef signed char **int8**
typedef unsigned char **u_int8**
typedef signed short int **int16**
typedef unsigned short int **u_int16**
typedef signed int **int32**
typedef unsigned int **u_int32**
typedef struct TimeCnvTbl_tag **YearCntTbl**
typedef struct DayCnvTbl_tag **DayCntTbl**
typedef char **CHAR**
typedef unsigned long **DWORD**
typedef unsigned short **WORD**
typedef unsigned char **UCHAR**
typedef wchar_t **WCHAR**
typedef unsigned char **BYTE**
typedef char **TCHAR**
typedef void * **LPVOID**
typedef void * **PVOID**
typedef void * **HANDLE**
typedef unsigned long **ULONG**
typedef void **VOID**
typedef const CHAR * **PCSTR**
typedef const CHAR * **LPCSTR**
typedef LPCSTR **LPCTSTR**
typedef DWORD * **PDWORD**
typedef DWORD * **LPDWORD**
typedef UCHAR * **PUCHAR**
typedef BYTE * **PBYTE**

typedef struct _SYSTEMTIME **SYSTEMTIME**
typedef SYSTEMTIME * **PSYSTEMTIME**
typedef SYSTEMTIME * **LPSYSTEMTIME**
typedef struct _SECURITY_ATTRIBUTES **SECURITY_ATTRIBUTES**
typedef SECURITY_ATTRIBUTES * PSECURITY_ATTRIBUTES
*security attribute*
typedef SECURITY_ATTRIBUTES * LPSECURITY_ATTRIBUTES
*security attribute*
typedef uint32_t UNIT_TYPE
*unit type data type define*
typedef unsigned char **u_char**
typedef unsigned short **u_short**
typedef unsigned int **u_int**
typedef unsigned long **u_long**
typedef u_int32 **signal_code**
typedef u_int32 **process_id**
typedef u_int32 **EventID**
typedef u_int32 **status_code**
typedef u_int32 **path_id**
typedef u_int16 PNO
*PNo.*
typedef int32 RET_API
*_CWORD64_API return code*
typedef u_int32 RET_OS
*OS Error code.*
typedef u_int32 **event_id**
typedef event_id SemID
*semaphore ID*
typedef u_int16 MID
*message ID*
typedef u_int16 CID
*command ID*
typedef u_int8 RID
*resource ID*
typedef u_int16 T_ENDID
*message end ID*
typedef u_int32 ECODE
*CWORD64 error code*

**Enumerations**

enum { _CWORD64_EVENT_MANUALRESET_OFF = ***, _CWORD64_EVENT_MANUALRESET_ON, _CWORD64_EVENT_MANUALRESET_MAX }
enum _RcvMsgMode { RM_WAIT, RM_CHECKRCV }

**Functions**

EventID _pb_CreateEvent (u_int8 ucManualReset, int32 lInitData, char *cpEventName)
RET_API _pb_WaitEvent (EventID eventID, int32 lWaitMode, int32 lMinVal, int32 lMaxVal, int32 *plEventVal, u_int32 ulMillSecTime)
RET_API _pb_SetEvent (EventID eventID, int32 lSetMode, int32 lval)
RET_API _pb_DeleteEvent (EventID eventID)
RET_API _pb_Setup_CWORD64_API (HANDLE hApp)
VOID _pb_ExitThread (DWORD dwExitCode)
RET_API _pb_CreateMsg (PNO pno)
RET_API _pb_SndMsg (PNO pno, u_int16 size, void *msgbuf, u_int16 mode)
RET_API _pb_SndMsg_Ext (PCSTR name, CID cid, u_int16 size, const void *msgbuf, u_int16 mode)
SemID _pb_CreateSemaphore (char *semName)
RET_API _pb_SemLock (SemID semID)
RET_API _pb_SemUnlock (SemID semID)
RET_API _pb_CreateShareData (char *area_name, u_int32 size, void **mem_ptr)
RET_API _pb_LinkShareData (char *area_name, void **mem_ptr, u_int32 *size)
RET_API _pb_ReqTimerStart (PNO SndPno, u_int16 TimerSeq, u_int8 TimeType, u_int32 TimeOut)
RET_API _pb_TimerStop (PNO SndPno, u_int16 TimerSeq, u_int8 TimeType)
HANDLE _pb_CreateMutex (LPSECURITY_ATTRIBUTES lpMutexAttributes, BOOL bInitialOwner, LPCTSTR lpName)
void _pb_GetDebugMsgMngTbl (void *pBuf, uint8_t *pLen)
void _pb_GetDebugMutexMngTbl (void *pBuf, uint8_t *pLen)
void _pb_GetDebugTimerMngTbl (void *pBuf)
void _pb_GetDebugEventMngTbl (void *pBuf, uint8_t *pLen)
void _pb_GetDebugMemoryMngTbl (void *pBuf)
void _pb_GetDebugOtherMngTbl (void *pBuf)
HANDLE _pb_GetAppHandle (void)
void _pb_SetAppHandle (HANDLE hApp)
void _pb_Teardown_CWORD64_API (void)
void _pb_Exit_d (const char *pFunc, int line)
RET_API _pb_RcvMsg (PNO pno, u_int16 size, void **msgbuf, u_int16 mode)
RET_API _pb_GetZcSndBuf (PNO pno, void **pSndBuf)
RET_API _pb_ZcSndMsg (PNO pno, u_int16 size, u_int16 mode)
PNO _pb_CnvName2Pno (PCSTR name)
PCSTR _pb_CnvPno2Name (PNO pno)
BOOL _pb_GetMsgResource (void)
BOOL _pb_GetMutexResource (void)
BOOL _pb_GetOtherResource (void)
void _pb_ReleaseMsgResource (void)
void _pb_ReleaseMutexResource (void)
void _pb_ReleaseOtherResource (void)

---

**Detailed Description**

---

**Class Documentation**

**struct T_APIMSG_HEADER**

_CWORD64_API message header structure

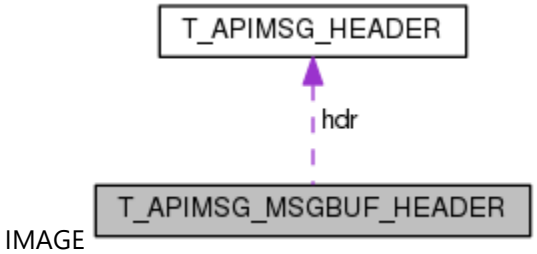
**Class Members:**

| | | |
|---|---|---|
| uint16_t | cid | message command ID |
| uint8_t | filler[2] | filter |
| uint16_t | msgbodysize | message body size |
| uint16_t | replyid | message send result notify dest ID |
| uint8_t | reserve | reserve |
| uint16_t | respno | message response process No |
| uint8_t | rid | message resource */ |
| uint16_t | sndpno | message send source process No |

**struct T_APIMSG_MSGBUF_HEADER**

_CWORD64_API message buffer header structure

Collaboration diagram for T_APIMSG_MSGBUF_HEADER:




IMAGE

**Class Members:**

| | | |
|---|---|---|
| T_APIMSG_HEADER | hdr | message header |
| uint32_t | signo | signal No, _pb_SndMsg to set 0 |

**struct _CWORD64_MSG_LOG_HDR**

_CWORD64_API message log header structure

**Class Members:**

| | | |
|---|---|---|
| uint32_t | dataBytes | data size |
| uint32_t | kickTime | kick time |
| uint32_t | pno | process No |

| | | |
|---|---|---|
| uint32_t | srCid | message command ID |

**struct TimerDupData**

define data type

**Class Members:**

| | | |
|---|---|---|
| u_int8 | ContFlg | continue flag |
| u_int8 | Dummy1 | dummy data |
| u_int16 | Dummy2 | dummy data |
| u_int32 | TimeOut[WTM_DUPTIME_MAX] | timeout value |

**struct RealTimeData**

current time data

**Class Members:**

| | | |
|---|---|---|
| u_int8 | CrtFlg | GPS time not correctedcorrected flag. |
| u_int8 | CrtHou | the last correction(hour) |
| u_int8 | CrtMin | min |
| u_int8 | CrtSec | sec |
| u_int8 | Dday | (day) |
| u_int8 | Hhour | (hour) |
| u_int8 | Mmin | (min) |
| u_int8 | Mmonth | (month) |
| u_int8 | Ssec | (sec) |
| u_int8 | Wweek | (week) |
| u_int16 | Yyear | current time(year:A.D.) |

**struct RealTimeDataBcd**

current time data(BCD format)

**Class Members:**

| | | |
|---|---|---|
| u_int8 | CrtFlg | GPS time not correctedcorrected flag(BIN 0x000x03) |
| u_int8 | CrtHou | the last correction(hour BCD 0x000x23) |

| | | |
|---|---|---|
| u_int8 | CrtMin | (min BCD ***) |
| u_int8 | CrtSec | (sec BCD ***) |
| u_int8 | Dday | (day BCD ***) |
| u_int8 | Hhour | (hour BCD ***) |
| u_int8 | Mmin | (min BCD ***) |
| u_int8 | Mmonth | (month BCD ***) |
| u_int8 | Ssec | (sec BCD ***) |
| u_int8 | Wweek | (week BIN ***) |
| u_int16 | Yyear | current time(year:A.D.BCD ex.0x1998) |

**struct GpsSetData**

GPS time setting data

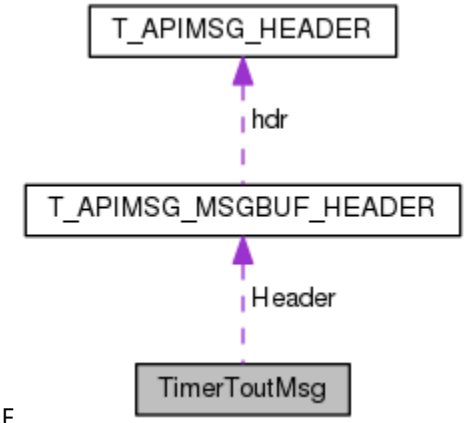
**Class Members:**

| | | |
|---|---|---|
| u_int8 | Dday | (day) |
| u_int8 | Dummy | dummy data |
| u_int8 | Hhour | (hour) |
| u_int8 | Mmin | (min) |
| u_int8 | Mmonth | (month) |
| u_int8 | Ssec | (sec) |
| u_int16 | Yyear | current time(year:A.D.) |

**struct TimerToutMsg**

timeout message

Collaboration diagram for TimerToutMsg:




IMAGE

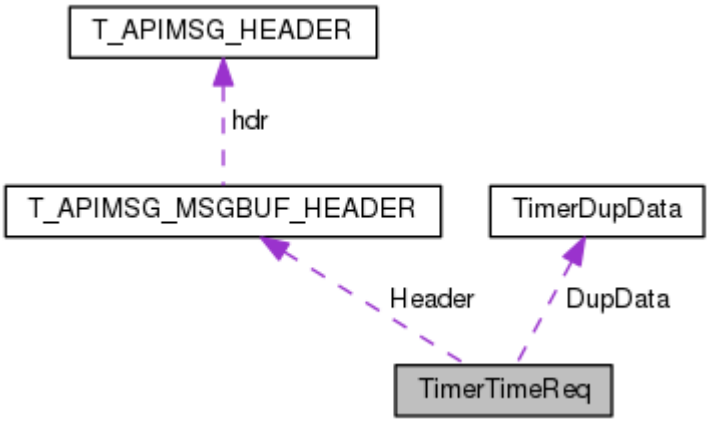
**Class Members:**

| | | |
|---|---|---|
| u_int16 | Dummy | dummy |
| T_APIMSG_MSGBUF_HEADER | Header | message header |
| u_int16 | TimerSeq | timer sequence No.(claimant specific) |

**struct TimerTimeReq**

timer start request

Collaboration diagram for TimerTimeReq:




IMAGE

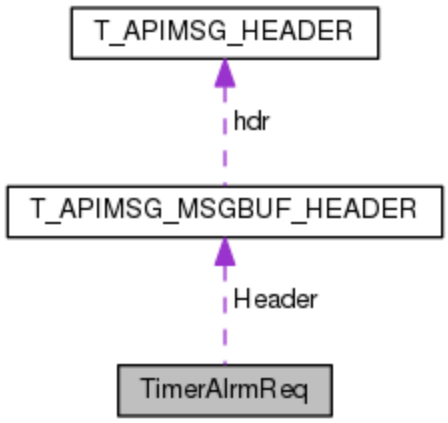
**Class Members:**

| | | |
|---|---|---|
| u_int8 | Dummy[3] | dummy data |
| TimerDupData | DupData | timeout value |
| T_APIMSG_MSGBUF_HEADER | Header | message header |
| PNO | SndPno | process No.(claimant specific) |
| u_int16 | TimerSeq | timer sequence No.(claimant specific) |
| u_int8 | TimeType | timer type |

**struct TimerAlrmReq**

time specified timer start request

Collaboration diagram for TimerAlrmReq:




IMAGE

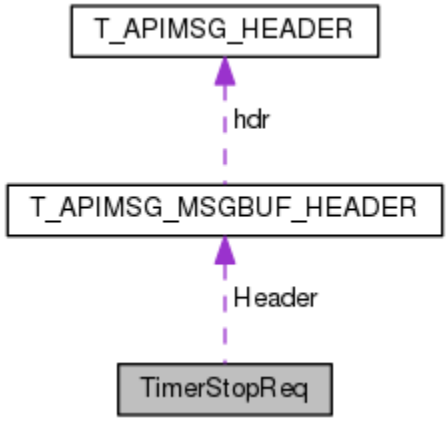
**Class Members:**

| | | |
|---|---|---|
| u_int8 | Dummy | dummy |
| T_APIMSG_MSGBUF_HEADER | Header | message header |
| PNO | SndPno | process No.(claimant specific) |
| u_int8 | TimeHou | specifictimeout hour |
| u_int8 | TimeMin | min |
| u_int16 | TimerSeq | timer sequence No.(claimant specific) |
| u_int8 | TimeSec | sec |

**struct TimerStopReq**

timer stop request

Collaboration diagram for TimerStopReq:




IMAGE

**Class Members:**

| | | |
|---|---|---|
| u_int8 | Dummy[3] | dummy data |

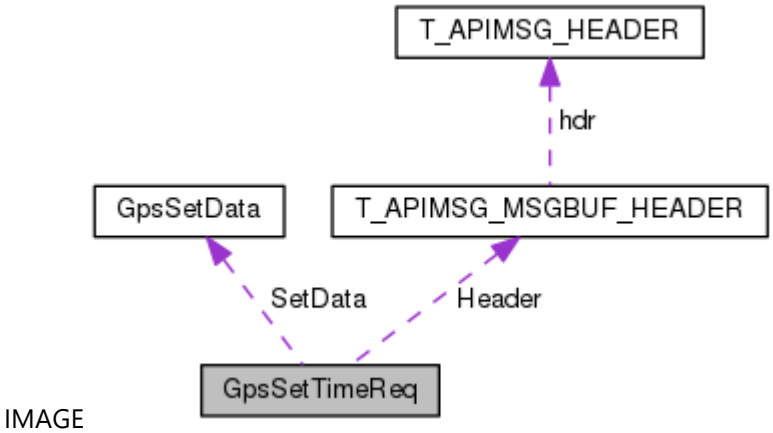
| | | |
|---|---|---|
| T_APIMSG_MSGBUF_HEADER | Header | message header |
| PNO | SndPno | process No.(claimant specific) |
| u_int16 | TimerSeq | timer sequence No.(claimant specific) |
| u_int8 | TimeType | timer type |

**struct GpsSetTimeReq**

GPS RTC time set request

time difference set request

Collaboration diagram for GpsSetTimeReq:



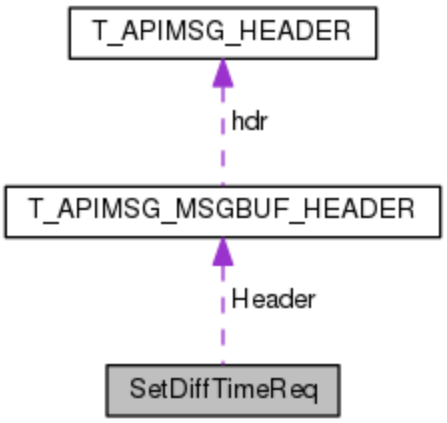

IMAGE

**Class Members:**

| | | |
|---|---|---|
| u_int8 | Dummy1 | dummy data |
| u_int8 | Dummy2 | dummy data |
| T_APIMSG_MSGBUF_HEADER | Header | message header |
| GpsSetData | SetData | GPS setting data. |
| PNO | SndPno | process No.(claimant specific) |

**struct SetDiffTimeReq**

Collaboration diagram for SetDiffTimeReq:



IMAGE

**Class Members:**

| | | |
|---|---|---|
| int8 | DiffTime | time difference value |
| int8 | Dummy1 | dummy data |
| int16 | Dummy2 | dummy data |
| T_APIMSG_MSGBUF_HEADER | Header | message header |

**struct T_TIM_RCV_DATA**

command notify information structure #005#

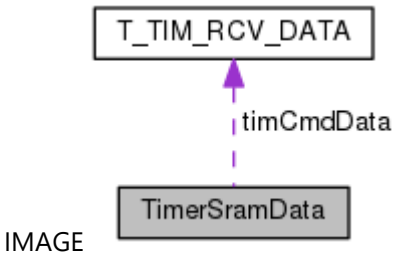
**Class Members:**

| | | |
|---|---|---|
| PNO | SndPno | command delivery PNO |
| u_int8 | ucAlmanc_Flg | almanac abnormal judgement notify complete flag |
| u_int8 | ucInvalid_Flg | time acquisition notify complete flag |

**struct TimerSramData**

timer SRAM data structure #005#

Collaboration diagram for TimerSramData:



IMAGE

**Class Members:**

| | | |
|---|---|---|
| int8 | _difftime | time difference value of current time |
| int8 | dummy[2] | dummy data |
| u_int32 | navitime | Navi operation addition time. |
| T_TIM_RCV_DATA | timCmdData[TIM_NUM_SNDCMD] | command receive information |
| int8 | timeUnit | specific time manager unit |

**struct SysTimeData**

system time data

**Class Members:**

| | | |
|---|---|---|
| u_int8 | Dday | (day BCD 0x010x31) |
| u_int8 | Hhour | (hour BCD 0x000x23) |
| u_int8 | Mmin | (min BCD 0x000x59) |
| u_int8 | Mmonth | (month BCD 0x010x12) |
| u_int8 | Ssec | (sec BCD 0x000x59) |
| u_int8 | Wweek | (week BIN 0x000x06) |
| u_int16 | Yyear | current time(year:A.D.BCD ex. 0x1998) |

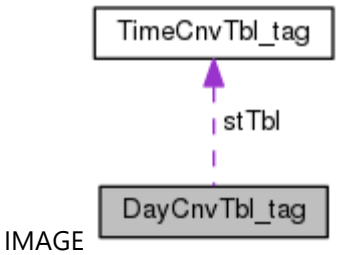
**struct TimeCnvTbl_tag**

**Class Members:**

| | | |
|---|---|---|
| uint32_t | ulCount | count |
| uint32_t | ulMonth[12] | month |
| uint32_t | ulYear | year |

**struct DayCnvTbl_tag**

Collaboration diagram for DayCnvTbl_tag:



IMAGE

**Class Members:**

| | | |
|---|---|---|
| YearCntTbl | stTbl[CNV_YEAR_MAX] | year convert table |

**struct TG_TIM_ROLOVR_GPSWEEKCORDATA**

GPS week correction data

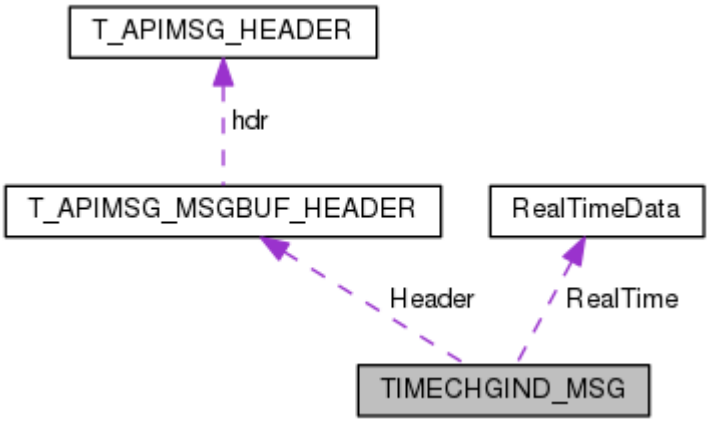
**Class Members:**

| | | |
|---|---|---|
| u_int8 | dummy[2] | dummy data |
| u_int8 | gpsweekcorcnt | GPS week rollover correct count. |
| u_int8 | permission | GPS week rollover correct permission. |

**struct TIMECHGIND_MSG**

time change information message structure

Collaboration diagram for TIMECHGIND_MSG:



IMAGE

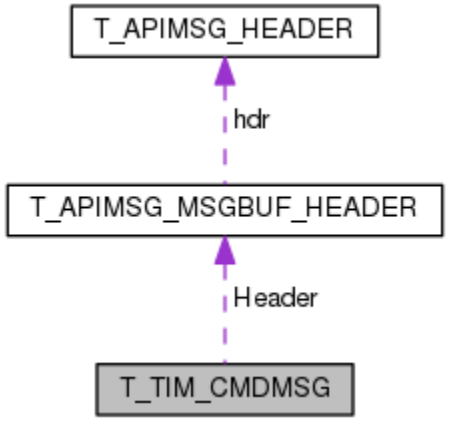
**Class Members:**

| | | |
|---|---|---|
| T_APIMSG_MSGBUF_HEADER | Header | message header |
| RealTimeData | RealTime | data body(current time data) |

**struct T_TIM_CMDMSG**

time available notification/almanac abnormality notification structure

Collaboration diagram for T_TIM_CMDMSG:




IMAGE

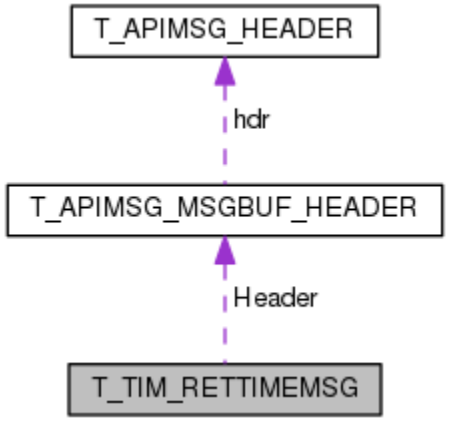
**Class Members:**

| | | |
|---|---|---|
| T_APIMSG_MSGBUF_HEADER | Header | message header |

**struct T_TIM_RETTIMEMSG**

time setting response message

Collaboration diagram for T_TIM_RETTIMEMSG:




IMAGE

**Class Members:**

| | | |
|---|---|---|
| int8 | cResult | process result |
| int8 | dummy[2] | dummy data |
| T_APIMSG_MSGBUF_HEADER | Header | message header |

**struct TimerModDat**

current time data module structure

**Class Members:**

| | | |
|---|---|---|
| u_int8 | CrtFlg | correctednot corrected flag |
| u_int8 | CrtHou | the last correction: hour |
| u_int8 | CrtMin | min |
| u_int8 | CrtSec | sec |
| u_int32 | InitFlg | initialization complete flag |

**struct _SYSTEMTIME**

system time

**Class Members:**

| | | |
|---|---|---|
| WORD | wDay | day |
| WORD | wDayOfWeek | day of week |
| WORD | wHour | hour |
| WORD | wMilliseconds | Milliseconds. |
| WORD | wMinute | minute |
| WORD | wMonth | month |
| WORD | wSecond | second |
| WORD | wYear | year |

**struct _SECURITY_ATTRIBUTES**

security attribute

**Class Members:**

| | | |
|---|---|---|
| BOOL | bInheritHandle | inherited handle |
| LPVOID | lpSecurityDescriptor | data pointer |
| DWORD | nLength | length |

**struct YearCntTbl**

year convert table

**struct DayCntTbl**

day convert table

**Macro Definition Documentation**

**#define _CWORD33_LOG( zone, opt, fmt, ...)**

**Value:**do { \
} while (0)

*CWORD33* log output

**#define CID_CDCTRL_SPECIFYDRIVE (CID)(CID_CDCTRL_BASE | CID_COM_SERIAL0)**

< unused

specific driver Indication

**#define PNO_USBAUDIO_WORKER_DIRECTSHOW 0x03E1**

USB audio worker thread(DirectShow control)

**#define PNO_VEHICLE_TOUCH_INT 0x0372**

touch interruption monitor thread#063# #GPF_60_012

**#define POS_BASE_GET_ZONE( level)**

**Value:**(((level) == (POS_DEBUG_LOGLEV_1))?(ZONE_ERR):\
((level) == (POS_DEBUG_LOGLEV_2))?(ZONE_INIT):\
((level) == (POS_DEBUG_LOGLEV_3))?(ZONE_INFO):\
((level) == (POS_DEBUG_LOGLEV_4))?(ZONE_WARN):\
(ZONE_20))

get base zone code

**#define POS_COM_GET_ZONE( level)**

**Value:**(((level) == (POS_DEBUG_LOGLEV_1))?(ZONE_ERR):\
((level) == (POS_DEBUG_LOGLEV_2))?(ZONE_INIT):\
((level) == (POS_DEBUG_LOGLEV_3))?(ZONE_INFO):\
((level) == (POS_DEBUG_LOGLEV_4))?(ZONE_WARN):\
(ZONE_28))

get common zone code

**#define POS_DEV_GET_ZONE( level)**

**Value:**((level) == (POS_DEBUG_LOGLEV_1)?(ZONE_ERR):\
(ZONE_14))

get DEV zone code

**#define POS_GPS_GET_ZONE( level)**

**Value:**(((level) == (POS_DEBUG_LOGLEV_1))?(ZONE_ERR):\
((level) == (POS_DEBUG_LOGLEV_2))?(ZONE_INIT):\

((level) == (POS_DEBUG_LOGLEV_3))?(ZONE_INFO):\
((level) == (POS_DEBUG_LOGLEV_4))?(ZONE_WARN):\
(ZONE_27))

get GPS zone code

**#define POS_MSG_GET_ZONE( level)**

**Value:**(((level) == (POS_DEBUG_LOGLEV_1))?(ZONE_ERR):\
(ZONE_17))

get message zone code

**#define POS_SENSLOG_GET_ZONE( typ)**

**Value:**(((typ) == (POS_SENSLOG_TYPE_SYS))?(ZONE_POS_SYS_IN):\
((typ) == (POS_SENSLOG_TYPE_GPS))?(ZONE_POS_GPS_IN):\
((typ) == (POS_SENSLOG_TYPE_NAV))?(ZONE_POS_NAV_IN):\
((typ) == (POS_SENSLOG_TYPE_CMD))?(ZONE_POS_CMD_IN):\
(0))

get sensor log zone code

**#define POS_SNR_GET_ZONE( level)**

**Value:**(((level) == (POS_DEBUG_LOGLEV_1))?(ZONE_ERR):\
((level) == (POS_DEBUG_LOGLEV_2))?(ZONE_INIT):\
((level) == (POS_DEBUG_LOGLEV_3))?(ZONE_INFO):\
((level) == (POS_DEBUG_LOGLEV_4))?(ZONE_WARN):\
(ZONE_26))

get sensor zone code

---

**Enumeration Type Documentation**

**anonymous enum**

**Enumerator**

***_CWORD64_EVENT_MANUALRESET_OFF*** manual reset off

***_CWORD64_EVENT_MANUALRESET_ON*** manual reset on

***_CWORD64_EVENT_MANUALRESET_MAX*** manual reset max value

**enum _RcvMsgMode**

mode of receive message

**Enumerator**

***RM_WAIT*** wait until received message

***RM_CHECKRCV*** do not wait message, return

**Function Documentation**

**PNO _pb_CnvName2Pno (PCSTR *name*)**

**Brief**

Convert process name to pno

**Parameters:**

| | | |
|---|---|---|
| in | *name* | PCSTR process name |

**Return values:**

| | |
|---|---|
| *0* | convert failed |
| *not* | 0 converted PNO |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter name is NULL
The size of parameter name is longer than THREAD_NAME_LEN_MAX

**Classification**

Public

**Type**

Sync

**PCSTR _pb_CnvPno2Name (PNO *pno*)**

**Brief**

Convert pno to process name

**Parameters:**

| | | |
|---|---|---|
| in | *pno* | |

| | | |
|---|---|---|
| | | PNO process No |

**Return values:**

| | |
|---|---|
| *NULL* | convert failed |
| *not* | NULL converted process name |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter pno has not been registered(not the return pno of _pb_CnvName2Pno).

**Classification**

Public

**Type**

Sync

**EventID _pb_CreateEvent (u_int8 *ucManualReset*, int32 *lInitData*, char * *cpEventName*)**

**Brief**

Create event by specified name

**Parameters:**

| | | |
|---|---|---|
| in | *ucManualReset* | u_int8 manual reset mode |
| in | *lInitData* | int32 event value |
| in | *cpEventName* | char* event name |

manual reset mode(ucManualReset)
_CWORD64_EVENT_MANUALRESET_OFF manual reset OFF
_CWORD64_EVENT_MANUALRESET_ON manual reset ON

**Return values:**

| | |
|---|---|
| *0* | event create failed |
| *not* | 0 created event ID |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter cpEventName is NULL
The parameter ucManualReset is more than or equal
_CWORD64_EVENT_MANUALRESET_MAX
The string of parameter cpEventName has no text
The string length of parameter cpEventName is longer than MAX_EVENT_NAME_LEN
The event is created in same process, but the count of thread reference to the event is reach to the max
The event is created in system, but the count of process reference to the event is reach to the max
Memory allocate failed for creating event table
During create the event table, the event flag register failed
The event table is full

**Classification**

Public

**Type**

Sync

## RET_API _pb_CreateMsg (PNO *pno*)

**Brief**

Create the message queue

**Parameters:**

| | | |
|---|---|---|
| in | *eventID* | EventID process No |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERRPARAM* | parameter error |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter pno is 0
The name length of thread called this API is longer than MAX_QUEUE_NAME_SIZE

**Classification**

Public

**Type**

Fire and Forget

**HANDLE _pb_CreateMutex (LPSECURITY_ATTRIBUTES *lpMutexAttributes*, BOOL *bInitialOwner*, LPCTSTR *lpName*)**

**Brief**

Create mutex

**Parameters:**

| | | |
|---|---|---|
| in | *lpMutexAttributes* | LPSECURITY_ATTRIBUTES not used |
| in | *bInitialOwner* | BOOL not used |
| in | *lpName* | LPCTSTR mutex name |

**Return values:**

| | |
|---|---|
| *NULL* | error return |
| *NULL* | created mutex Handle |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter lpName is NULL
The string length of pamareter lpName is longer than NAME_MAX
Mutex lock map failed

**Classification**

Public

**Type**

Sync

**SemID _pb_CreateSemaphore (char * *semName*)**

**Brief**

Create semaphore by specified name

**Parameters:**

| | | |
|---|---|---|
| in | *semName* | char* semaphore name |

**Return values:**

| | |
|---|---|
| *0* | semaphore create failed |
| *not* | 0 created semaphore ID |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter semName is NULL
The string of parameter semName has no text
The string length of parameter semName is longer than MAX_SEMAPHORE_NAME_LEN
Memory allocate failed for creating semaphore table
The semaphore table is full

**Classification**

Public

**Type**

Sync

**RET_API _pb_CreateShareData (char * *area_name*, u_int32 *size*, void ** *mem_ptr*)**

**Brief**

Create Share Data

**Parameters:**

| | | |
|---|---|---|
| in | *area_name* | char* share data name |
| in | *size* | u_int32 data size |
| out | *mem_ptr* | void** share memory pointer |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERROR* | error return |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter area_name is NULL
The parameter mem_ptr is NULL
The string of parameter area_name has no text
The string length of parameter area_name is longer than MAX_AREA_NAME_LEN
The parameter size is 0
The specified share data has already been created
Memory allocate failed for creating share data
The specified share data file create failed

**Classification**

Public

**Type**

Sync

**RET_API _pb_DeleteEvent (EventID *eventID*)**

**Brief**

Destroy the event specified by ID

**Parameters:**

| | | |
|---|---|---|
| in | *eventID* | EventID event ID |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERROR* | event flag destroy failed |
| *RET_EV_NONEspecified* | event not exist |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter eventID is more than MAX_PB_EVENTS
The specified event has not been registered in event table
Event flag destroy failed

**Classification**

Public

**Type**

Fire and Forget

**void _pb_Exit_d (const char * *pFunc*, int *line*)**

**Brief**

Terminate current process.

**Parameters:**

| | | |
|---|---|---|
| in | *pFunc* | char* source function |
| in | *line* | int source line |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

None

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**VOID _pb_ExitThread (DWORD *dwExitCode*)**

**Brief**

Terminate current thread.

**Parameters:**

| | | |
|---|---|---|
| in | *dwExitCode* | DWORD exit code(not used) |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

None

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**HANDLE _pb_GetAppHandle (void )**

**Brief**

Get current app Handle.

**Parameters:**

| | |
|---|---|
| *None* | |

**Return values:**

| | |
|---|---|
| *0* | get Handle failed |
| *not* | 0 Handle |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

Current app Handle has not been registered

**Classification**

Public

**Type**

Sync

**void _pb_GetDebugEventMngTbl (void * *pBuf*, uint8_t * *pLen*)**

**Brief**

Get event management table dump info.

**Parameters:**

| | | |
|---|---|---|
| out | *pBuf* | void* output buffer |

| | | |
|---|---|---|
| | *[in/out]* | pLen<br>uint8_t* output buffer size |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Sync

**void _pb_GetDebugMemoryMngTbl (void * *pBuf*)**

**Brief**

Get memory management table dump info.

**Parameters:**

| | | |
|---|---|---|
| out | *pBuf* | void* output buffer |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Sync

**void _pb_GetDebugMsgMngTbl (void * *pBuf*, uint8_t * *pLen*)**

**Brief**

Get message management table dump info.

**Parameters:**

| | | |
|---|---|---|
| out | *pBuf* | void* output buffer |
| | *[in/out]* | pLen<br>uint8_t* output buffer size |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Sync

**void _pb_GetDebugMutexMngTbl (void * *pBuf*, uint8_t * *pLen*)**

**Brief**

Get mutex management table dump info.

**Parameters:**

| | | |
|---|---|---|
| out | *pBuf* | void* output buffer |
| | *[in/out]* | pLen<br>uint8_t* output buffer size |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Sync

**void _pb_GetDebugOtherMngTbl (void *  *pBuf*)**

**Brief**

Get common info management table dump info.

**Parameters:**

| | | |
|---|---|---|
| out | *pBuf* | void* output buffer |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Sync

**void _pb_GetDebugTimerMngTbl (void * *pBuf*)**

**Brief**

Get timer management table dump info.

**Parameters:**

| | | |
|---|---|---|
| out | *pBuf* | void* output buffer |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Sync

**BOOL _pb_GetMsgResource (void )**

**Brief**

Get message resource

**Parameters:**

| | |
|---|---|
| *None* | |

**Return values:**

| | |
|---|---|
| *TRUE* | normal return |
| *FALSE* | error(lack of resource) |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**BOOL _pb_GetMutexResource (void )**

**Brief**

Get mutex resource

**Parameters:**

| | |
|---|---|
| *None* | |

**Return values:**

| | |
|---|---|
| *TRUE* | normal return |
| *FALSE* | error(lack of resource) |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**BOOL _pb_GetOtherResource (void )**

**Brief**

Get common resource

**Parameters:**

| | |
|---|---|
| *None* | |

**Return values:**

| | |
|---|---|
| *TRUE* | normal return |
| *FALSE* | error(lack of resource) |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_GetZcSndBuf (PNO *pno*, void ** *pSndBuf*)**

**Brief**

Get send buffer for message transferring in process

**Parameters:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERRPARAM* | parameter error |
| *RET_ERROR* | other error |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter pno is 0
The message queue specified by parameter pno has not been created
Get sender Handle failed
Get send buffer failed

**Classification**

Public

**Type**

Sync

**RET_API _pb_LinkShareData (char * *area_name*, void ** *mem_ptr*, u_int32 * *size*)**

**Brief**

Link Share Data

**Parameters:**

| | | |
|---|---|---|
| in | *area_name* | char* share data name |
| out | *mem_ptr* | void** share memory pointer |
| in | *size* | u_int32* data size |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERROR* | error return |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter area_name is NULL
The parameter mem_ptr is NULL
The string of parameter area_name has no text
The string length of parameter area_name is longer than MAX_AREA_NAME_LEN
The parameter size is NULL
Memory allocate failed for linking share data
The specified share data file open failed

**Classification**

Public

**Type**

Sync

## RET_API _pb_RcvMsg (PNO *pno*, u_int16 *size*, void ** *msgbuf*, u_int16 *mode*)

**Brief**

Receive message.

**Parameters:**

| | | |
|---|---|---|
| in | *pno* | PNO process No |
| in | *size* | u_int16 message size |
| out | *msgbuf* | void* message buffer |
| in | *mode* | u_int16 not used |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERRPARAM* | parameter error |
| *RET_ERROR* | other error |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter msgbuf is NULL
The parameter pno is 0
The message queue specified by parameter pno has not been created
Receive message failed
Receive message successrully, but the received message size is longer than parameter size
Get received message data failed

**Classification**

Public

**Type**

Sync

**void _pb_ReleaseMsgResource (void )**

**Brief**

Release message resource

**Parameters:**

| | |
|---|---|
| *None* | |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**void _pb_ReleaseMutexResource (void )**

**Brief**

Release mutex resource

**Parameters:**

| | |
|---|---|
| *None* | |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**void _pb_ReleaseOtherResource (void )**

**Brief**

Release common resource

**Parameters:**

| | |
|---|---|
| *None* | |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_ReqTimerStart (PNO *SndPno*, u_int16 *TimerSeq*, u_int8 *TimeType*, u_int32 *TimeOut*)**

**Brief**

Start timer

**Parameters:**

| | | |
|---|---|---|
| in | *SndPno* | PNO source process No |
| in | *TimerSeq* | u_int16 timer sequence No |

| | | |
|---|---|---|
| in | *TimerType* | u_int8 timer type |
| in | *TimeOut* | u_int32 timeout time |

timer type(TimerType)
TIMER_TYPE_SYN fixed term timer
TIMER_TYPE_USN asynchronous timer

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERRPARAM* | parameter error |
| *RET_ERROR* | other error |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

Initialization of positioning_base_library has not been done in current process
The parameter TimerType is neither TIMER_TYPE_SYN nor TIMER_TYPE_USN
The parameter TimeOut is 0
The same timer has already been started
Timer start failed

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_SemLock (SemID *semID*)**

**Brief**

Semaphore lock

**Parameters:**

| | | |
|---|---|---|
| in | *semID* | SemID semaphore ID |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_OSERROR* | error return |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter semName is 0
The parameter semName is more than MAX_PB_SEMAPHORES
The specified semaphore has not been registered in semaphore table
The specified semaphore's mutex has not been created
The specified semaphore's mutex lock failed

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_SemUnlock (SemID *semID*)**

**Brief**

Semaphore unlock

**Parameters:**

| | | |
|---|---|---|
| in | *semID* | SemID semaphore ID |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_OSERROR* | error return |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter semName is 0
The parameter semName is more than MAX_PB_SEMAPHORES
The specified semaphore has not been registered in semaphore table
The specified semaphore's mutex has not been created
The specified semaphore's mutex release failed

**Classification**

Public

**Type**

Fire and Forget

**void _pb_SetAppHandle (HANDLE *hApp*)**

**Brief**

Set current app Handle.

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE app Handle |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_SetEvent (EventID *eventID*, int32 *lSetMode*, int32 *lval*)**

**Brief**

Set the value of event specified by ID

**Parameters:**

| | | |
|---|---|---|
| in | *eventID* | EventID event ID |
| in | *lSetMode* | int32 event value set mode |
| in | *iVal* | int32 event value |

event value set mode(lSetMode)
SAPI_EVSET_ABSOLUTE absolute value set(set parameter value)
SAPI_EVSET_RELATE offset set(offset of current value set)

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERRPARAM* | parameter error |
| *RET_EV_NONEspecified* | event not exist |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter eventID is more than MAX_PB_EVENTS
The specified event has not been registered in event table
The parameter lSetMode is neither SAPI_EVSET_ABSOLUTE nor SAPI_EVSET_RELATE

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_Setup_CWORD64_API (HANDLE *hApp*)**

**Brief**

Initialization before call Base API(every process used Base API should call this API firstly).

**Parameters:**

| | | |
|---|---|---|
| in | *hApp* | HANDLE app Handle |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |

**Precondition**

The creation/initialization(_CWORD33_CreateDispatcherWithoutLoop and etc.) of the dispatcher for App are completed.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_SndMsg (PNO *pno*, u_int16 *size*, void * *msgbuf*, u_int16 *mode*)**

**Brief**

Send message in process

**Parameters:**

| | | |
|---|---|---|
| in | *pno* | PNO process No |
| in | *size* | u_int16 message size |

| | | |
|---|---|---|
| in | *msgbuf* | void* message buffer |
| in | *mode* | u_int16 not used |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERRPARAM* | parameter error |
| *RET_ERROR* | other error |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter msgbuf is NULL
The parameter pno is 0
The message queue specified by parameter pno has not been created
Get sender Handle failed
Send message failed

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_SndMsg_Ext (PCSTR *name*, CID *cid*, u_int16 *size*, const void * *msgbuf*, u_int16 *mode*)**

**Brief**

Send message between process

**Parameters:**

| | | |
|---|---|---|
| in | *name* | PCSTR destination process name |

| | | |
|---|---|---|
| in | *cid* | CID command ID |
| in | *size* | u_int16 message size |
| in | *msgbuf* | const void* message buffer |
| in | *mode* | u_int16 not used |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERRPARAM* | parameter error |
| *RET_ERROR* | other error |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter name is NULL
The parameter msgbuf is NULL
The length of parameter name is longer than MAX_QUEUE_NAME_SIZE
Get sender Handle failed
Send message failed

**Classification**

Public

**Type**

Fire and Forget

**void _pb_Teardown_CWORD64_API (void )**

**Brief**

Base API stop process(every process used Base API should call this API before stop).

**Parameters:**

| | |
|---|---|
| *None* | |

**Return values:**

| | |
|---|---|
| *None* | |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_TimerStop (PNO *SndPno*, u_int16 *TimerSeq*, u_int8 *TimeType*)**

**Brief**

Stop timer

**Parameters:**

| | | |
|---|---|---|
| in | *SndPno* | PNO source process No |
| in | *TimerSeq* | u_int16 timer sequence No |
| in | *TimerType* | u_int8 timer type |

timer type(TimerType)
TIMER_TYPE_SYN fixed term timer
TIMER_TYPE_USN asynchronous timer

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERROR* | error return |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

Initialization of positioning_base_library has not been done in current process
Timer stop failed

**Classification**

Public

**Type**

Fire and Forget

**RET_API _pb_WaitEvent (EventID *eventID*, int32 *lWaitMode*, int32 *lMinVal*, int32 *lMaxVal*, int32 * *plEventVal*, u_int32 *ulMillSecTime*)**

**Brief**

Wait for the event specified by ID

**Parameters:**

| | | |
|---|---|---|
| in | *eventID* | EventID event ID |
| in | *lWaitMode* | int32 event wait mode |
| in | *lMinVal* | int32 event wait min value |
| in | *lMaxVal* | int32 event wait max value |
| out | *plEventVal* | int32* event wait value output pointer |
| in | *ulMillSecTime* | u_int32 timeout time(ms) |

event wait mode(lWaitMode)
SAPI_EVWAIT_VAL

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_EV_NONEparameter* | error or specified event not exist |
| *RET_ERRTIMEOUTtimeout* | |
| *RET_ERROROther* | error |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter plEventVal is NULL
The parameter eventID is more than MAX_PB_EVENTS
The specified event has not been registered in event table
The count of threads which is waiting this event is reach max
The parameter ulMillSecTime is 0
The parameter ulMillSecTime is INFINITE, and wait event flag failed
The parameter ulMillSecTime is not INFINITE, and timeout
The parameter ulMillSecTime is not INFINITE, and get event flag failed

**Classification**

Public

**Type**

Sync

## RET_API _pb_ZcSndMsg (PNO *pno*, u_int16 *size*, u_int16 *mode*)

**Brief**

Send message in process

**Parameters:**

| | | |
|---|---|---|
| in | *pno* | PNO process No |

| | | |
|---|---|---|
| in | *size* | u_int16 message size |
| in | *mode* | u_int16 not used |

**Return values:**

| | |
|---|---|
| *RET_NORMAL* | normal return |
| *RET_ERRPARAM* | parameter error |
| *RET_ERROR* | other error |

**Precondition**

positioning_base_libraryinitialization are completed
(_pb_Setup_CWORD64_API has already been called) in current process.

**changes of the internal status**

There is no changes of the internal status by this API

**Failure condition**

The parameter pno is 0
The message queue specified by parameter pno has not been created
Get sender Handle failed
Get send buffer failed
Set send buffer failed
Send message failed

**Classification**

Public

**Type**

Fire and Forget

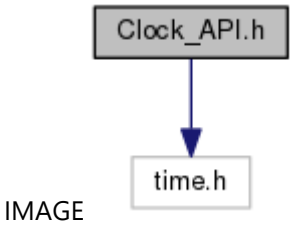
# File Documentation

## Clock_API.h File Reference

System community functions header file.

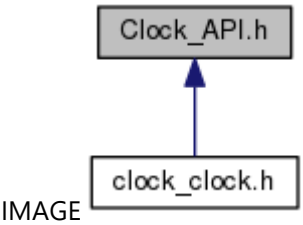
#include <time.h>

Include dependency graph for Clock_API.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

### Macros

#define CLOCK_CORRECTED_TIME (0U)
#define CLOCK_NOT_CORRECTED_TIME (1U)
#define CLOCK_RTC_TIME (2U)
#define CLOCK_INVALID_TIME (3U)
#define TIME_ZONE_MAX_SIZE 256

### Typedefs

typedef enum tagCLOCK_RETURN CLOCK_RETURN

### Enumerations

enum tagCLOCK_RETURN { CLOCK_OK, CLOCK_ERROR, CLOCK_ERROR_ARGUMENTS }

### Functions

CLOCK_RETURN Clock_getSystemTime (struct timespec *time, uint8_t *status)
CLOCK_RETURN Clock_getLocalTime (const time_t *base_time, struct tm *local_time)
CLOCK_RETURN Clock_CnvSecToDate (const time_t *timep, struct tm *result)
CLOCK_RETURN Clock_CnvDateToSec (const struct tm *tm, time_t *result)
CLOCK_RETURN Clock_setUserTime (const time_t *user_time)
CLOCK_RETURN Clock_getUserTime (time_t *user_time, uint8_t *status)

CLOCK_RETURN Clock_addUserTimeOffset (const struct tm *offset)
CLOCK_RETURN Clock_getSystemTimeY2K38 (uint32_t *time, uint8_t *status)
CLOCK_RETURN Clock_getLocalTimeY2K38 (const uint32_t *base_time, struct tm *local_time)
CLOCK_RETURN Clock_CnvSecToDateY2K38 (const uint32_t *time, struct tm *result)
CLOCK_RETURN Clock_CnvDateToSecY2K38 (const struct tm *tm, uint32_t *result)

---

**Detailed Description**

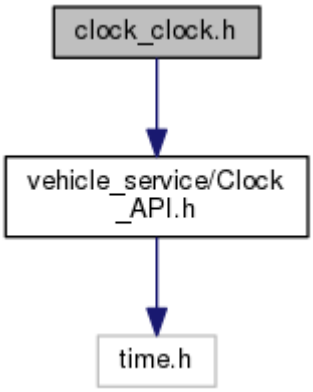
System community functions header file.

# clock_clock.h File Reference

clock module of clock unit common header file.

#include <vehicle_service/Clock_API.h>

Include dependency graph for clock_clock.h:




IMAGE

---

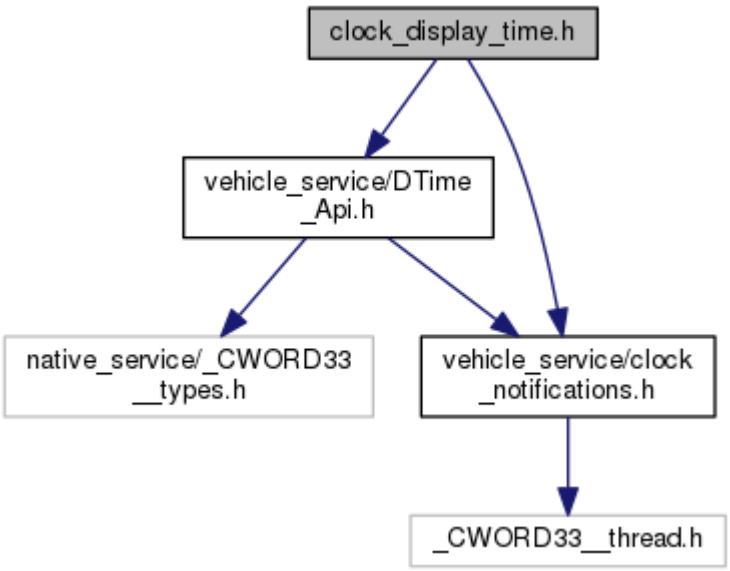
**Detailed Description**

clock module of clock unit common header file.

# clock_display_time.h File Reference

display time module of clock unit common header file.

#include <vehicle_service/DTime_Api.h>
#include <vehicle_service/clock_notifications.h>
Include dependency graph for clock_display_time.h:




IMAGE

---

**Detailed Description**

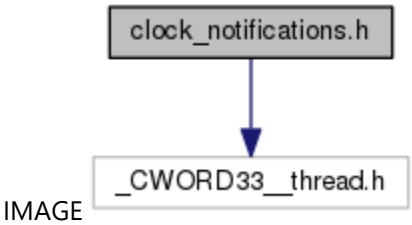
display time module of clock unit common header file.

# clock_notifications.h File Reference

Notification names defined in the file.

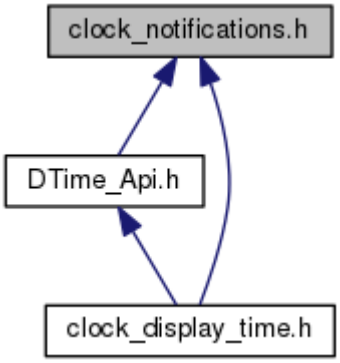
#include <_CWORD33__thread.h>

Include dependency graph for clock_notifications.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Macros**

#define NTFY_Clock_Availability MN_CLOCK"/Availability"

---

**Detailed Description**

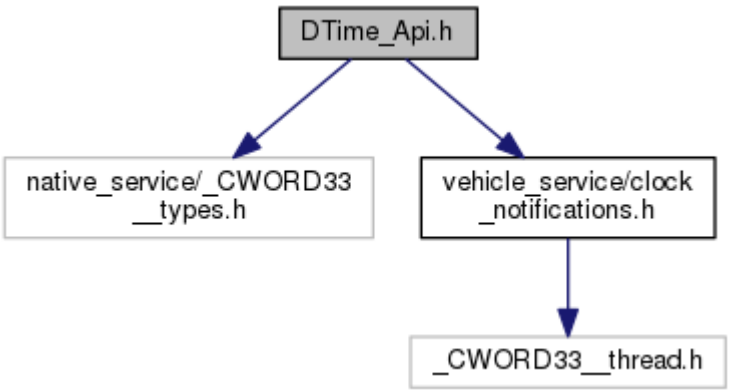
Notification names defined in the file.

# DTime_Api.h File Reference

display time middle API header file.

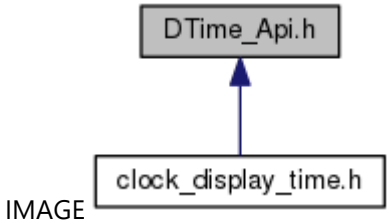
#include <native_service/_CWORD33__types.h>
#include <vehicle_service/clock_notifications.h>

Include dependency graph for DTime_Api.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Classes**

struct _tag_T_TimeData
struct _tag_T_TimeDiff
struct _tag_T_TimeOffset
struct _tag_T_InfoDate
struct _tag_T_InfoTime
struct _tag_T_DisplaySetting
struct _tag_T_DTimeData
struct _tag_T_DTIME_MSG_NOTIFY_DTIME
struct _tag_T_DTIME_GPSTIME
struct _tag_T_DTIME_MSG_GPSTIME

**Macros**

#define DTIME_TIMEDIFF_SIGN_PLUSE FALSE
#define DTIME_TIMEDIFF_SIGN_MINUS TRUE
#define DTIME_SETTING_GPS_MANUAL FALSE
#define DTIME_SETTING_GPS_AUTO TRUE

#define DTIME_SETTING_FORMAT_12H FALSE
#define DTIME_SETTING_FORMAT_24H TRUE
#define DTIME_SETTING_DST_MANUAL FALSE
#define DTIME_SETTING_DST_AUTO TRUE
#define DTIME_SETTING_DST_OFF FALSE
#define DTIME_SETTING_DST_ON TRUE
#define DTIME_SETTING_TIMEZONE_MANUAL FALSE
#define DTIME_SETTING_TIMEZONE_AUTO TRUE
#define DTIME_DISP_TIME_INVALID FALSE
#define DTIME_DISP_TIME_VALID TRUE
#define DTIME_DISP_MERIDIEM_INVALID 0
#define DTIME_DISP_MERIDIEM_AM 1
#define DTIME_DISP_MERIDIEM_PM 2
#define DTIME_DISP_MERIDIEM_12H_INVALID 3
#define DTIME_GPS_STATUS_INVALID 0x00
#define DTIME_GPS_STATUS_RTC 0x01
#define DTIME_GPS_STATUS_CORRECT 0x02
#define DTIME_SET_MERIDIEM_AM TRUE
#define DTIME_SET_MERIDIEM_PM FALSE

**Typedefs**

typedef enum _ClockServiceProtocol ClockServiceProtocol
typedef struct _tag_T_TimeData T_TimeData
typedef struct _tag_T_TimeDiff T_TimeDiff
typedef struct _tag_T_TimeOffset T_TimeOffset
typedef struct _tag_T_InfoDate T_InfoDate
typedef struct _tag_T_InfoTime T_InfoTime
typedef struct _tag_T_DisplaySetting T_DisplaySetting
typedef struct _tag_T_DTimeData T_DTimeData
typedef struct _tag_T_DTIME_MSG_NOTIFY_DTIME T_DTIME_MSG_NOTIFY_DTIME
typedef struct _tag_T_DTIME_GPSTIME T_DTIME_GPSTIME
typedef struct _tag_T_DTIME_MSG_GPSTIME T_DTIME_MSG_GPSTIME

**Enumerations**

enum _ClockServiceProtocol { CID_DTIME_NOTIFY = *** }

**Functions**

E_CWORD33_Status DTime_loadDisplaySetting (HANDLE h_app, T_DisplaySetting *setting)
E_CWORD33_Status DTime_getDisplaySetting (HANDLE h_app, T_DisplaySetting *setting)
E_CWORD33_Status DTime_setGpsTime (HANDLE h_app, T_DTIME_MSG_GPSTIME *gps_time)
E_CWORD33_Status DTime_registerListener_DTime (HANDLE h_app, PCSTR notify_name)
E_CWORD33_Status DTime_UnregisterListener_DTime (HANDLE h_app, PCSTR notify_name)
E_CWORD33_Status DTime_setAutoAdjust (HANDLE h_app, BOOL setting, T_TimeOffset *off_set)
E_CWORD33_Status DTime_increaseHour (HANDLE h_app)
E_CWORD33_Status DTime_decreaseHour (HANDLE h_app)
E_CWORD33_Status DTime_increaseMinute (HANDLE h_app)
E_CWORD33_Status DTime_decreaseMinute (HANDLE h_app)
E_CWORD33_Status DTime_clearMinute (HANDLE h_app)

E_CWORD33_Status DTime_setDate (const HANDLE h_app, const T_InfoDate *date)
E_CWORD33_Status DTime_setTime (const HANDLE h_app, const T_InfoTime *dtime)
E_CWORD33_Status DTime_setAMPM (HANDLE h_app, BOOL meridiem)
E_CWORD33_Status DTime_setFormat (HANDLE h_app, BOOL setting)
E_CWORD33_Status DTime_setTimeZone (HANDLE h_app, BOOL b_auto, T_TimeDiff *diff)
E_CWORD33_Status DTime_setDST (HANDLE h_app, BOOL b_auto, BOOL setting)
E_CWORD33_Status DTime_notifySwOff (HANDLE h_app)

---

**Detailed Description**

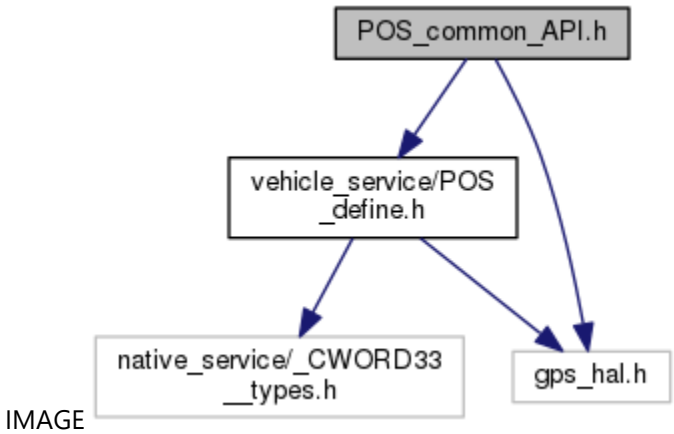
display time middle API header file.

# POS_common_API.h File Reference

API definition file for common function.

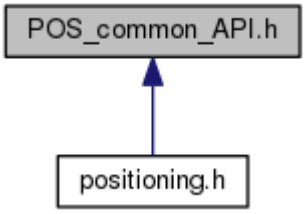
#include <vehicle_service/POS_define.h>
#include <gps_hal.h>

Include dependency graph for POS_common_API.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Classes**

struct POS_POSDATA
struct SENSORLOCATION_MSG_LONLATINFO
struct SENSORLOCATION_MSG_ALTITUDEINFO
struct SENSORMOTION_MSG_HEADINGINFO
struct SENSORMOTION_MSG_SPEEDINFO
struct POS_LOCATIONINFO_NMEA

**Macros**

#define SENSORLOCATION_STATUS_DISABLE (0)

*Not available.*

#define SENSORLOCATION_STATUS_ENABLE (1)

*Available.*

#define SENSORMOTION_STATUS_DISABLE (0)

*Not available.*

#define SENSORMOTION_STATUS_ENABLE (1)

*Available.*

#define SENSOR_GET_METHOD_AUTO (0)

*Not specified.*

#define SENSOR_GET_METHOD_GPS (1)

*GPS.*

#define SENSOR_GET_METHOD_NAVI (2)

*Navigation.*

#define SENSOR_GET_METHOD_DR (3)

*Dead Reckoning.*

#define SENSOR_GET_METHOD_POS (4)

*positioning (Internal)*

#define LOCATIONINFO_NMEA_MAX (1020)

*Max length of 'Location Information'.*

#define CID_POSIF_REGISTER_LISTENER_SPEED (0x0203)

*Delivery speed.*

#define CID_POSIF_REGISTER_LISTENER_LONLAT 0x0781

*Delivery longitude and latitude.*

#define CID_POSIF_REGISTER_LISTENER_ALTITUDE 0x0782

*Delivery altitude.*

#define CID_POSIF_REGISTER_LISTENER_HEADING 0x0783

*Delivery heading.*

**Functions**

POS_RET_API POS_RegisterListenerLonLat (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming, uint8_t ucGetMethod)

POS_RET_API POS_RegisterListenerAltitude (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming, uint8_t ucGetMethod)

POS_RET_API POS_RegisterListenerSpeed (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming, uint8_t ucGetMethod)

POS_RET_API POS_RegisterListenerHeading (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming, uint8_t ucGetMethod)

POS_RET_API POS_GetLonLat (HANDLE hApp, SENSORLOCATION_LONLATINFO_DAT *dat, uint8_t ucGetMethod)

POS_RET_API POS_GetAltitude (HANDLE hApp, SENSORLOCATION_ALTITUDEINFO_DAT *dat, uint8_t ucGetMethod)

POS_RET_API POS_GetSpeed (HANDLE hApp, SENSORMOTION_SPEEDINFO_DAT *dat, uint8_t ucGetMethod)

POS_RET_API POS_GetHeading (HANDLE hApp, SENSORMOTION_HEADINGINFO_DAT *dat, uint8_t ucGetMethod)

POS_RET_API POS_SetSpeedInfo (HANDLE hApp, uint16_t navispeed)

POS_RET_API POS_SetLocationInfo (HANDLE hApp, POS_POSDATA *pstPosData)

POS_RET_API POS_SetLocationInfoNmea (HANDLE hApp, POS_LOCATIONINFO_NMEA *locationInfo)

---

**Detailed Description**

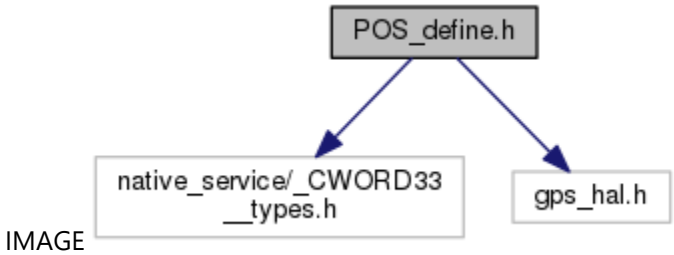
API definition file for common function.

# POS_define.h File Reference

Header file to define the constants and structure.

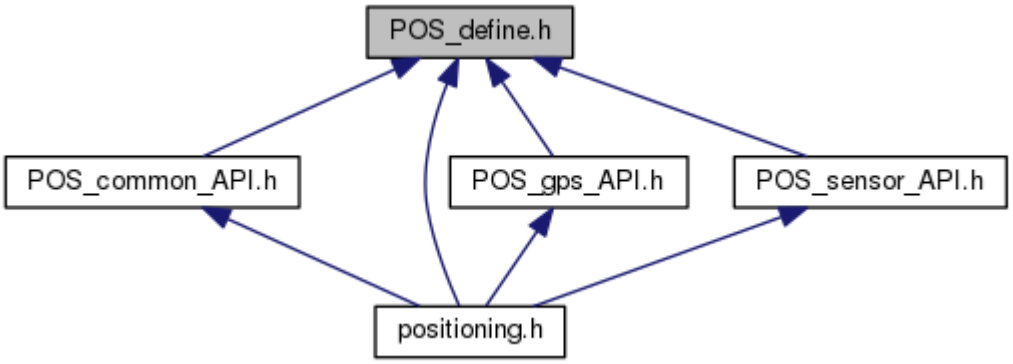
#include <native_service/_CWORD33__types.h>
#include <gps_hal.h>

Include dependency graph for POS_define.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Macros**

#define POS_AVAILABILITY "Positioning/Availability"
*Availability.*

#define POS_NTFY_SEND_THREAD "POS_Main"
*POS_Main thread.*

#define POS_NTFY_SEND_THREAD_GPS "POS_Gps"
*POS_Gps thread.*

#define SENSOR_RET_NORMAL 0
*normal finish*

#define SENSOR_RET_ERROR_PID (-1)
*thread ID error*

#define SENSOR_RET_ERROR_DID (-2)
*data ID error*

#define SENSOR_RET_ERROR_DID_DIS (-3)

*data ID not CAN ID*

#define SENSOR_RET_ERROR_PARAM (-4)

*parameter error*

#define SENSOR_RET_ERROR_BUFFULL (-5)

*buffer full error*

#define SENSOR_RET_ERROR_CREATE_EVENT (-6)

*create event error*

#define SENSOR_RET_ERROR_TIMER (-7)

*create timer error*

#define SENSOR_RET_ERROR_OUTOF_MEMORY (-8)

*share memory allocation size error*

#define SENSOR_RET_ERROR_SIZE (-9)

*memory size error*

#define SENSOR_RET_ERROR (-10)

*error occured*

#define SENSOR_RET_ERROR_NOSUPPORT (-11)

*no support*

#define SENSOR_RET_ERROR_INNER (-12)

*Internal error.*

#define SENSOR_RET_ERROR_RESOURCE (-13)

*lack of resources*

#define SENSOR_RET_ERROR_MIN POS_RET_ERROR_MIN

*min value of error*

#define POS_LOC_INFO_LAT 0x01

*current position latitude(bit0) 1:valid 0:invalid*

#define POS_LOC_INFO_LON 0x02

*current position longitude(bit1) 1:valid 0:invalid*

#define POS_LOC_INFO_ALT 0x04

*current position altitude(bit2) 1:valid 0:invalid*

#define POS_LOC_INFO_HEAD 0x08

*current position heading(bit3) 1:valid 0:invalid*

#define POS_LOC_INFO_USE_GSP 0x01

*GPS data used result(bit0) 1:valid 0:invalid.*

#define POS_LOC_INFO_USE_DGPS 0x02

*DGPS data used result(bit1) 1:valid 0:invalid.*

#define POS_LOC_INFO_USE_DR 0x04

*Dead Reckoning used result(bit2) 1:valid 0:invalid.*

#define POS_LOC_INFO_USE_MAPMATCHING 0x08

*MapMatching result(bit3) 1:valid 0:invalid.*

#define SENSOR_DELIVERY_REGIST 0x01

*register delivery*

#define SENSOR_DELIVERY_TIMING_UPDATE 0x01

*delivery update timing*

#define SENSOR_DELIVERY_TIMING_CHANGE 0x02

*delivery change timing*

**Typedefs**

typedef int32_t SENSOR_RET_API

*API return value.*

typedef int32_t POS_RET_API

*API return value.*

typedef uint32_t DID

*data ID*

typedef int32_t RET_API

*CWORD64 API return value*

---

**Detailed Description**

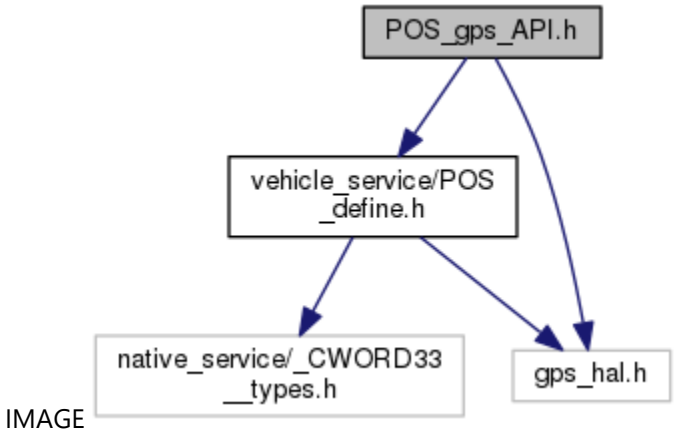
Header file to define the constants and structure.

# POS_gps_API.h File Reference

API definition file for GPS function.

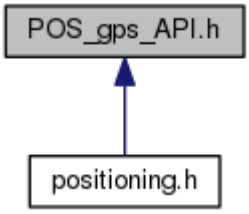
#include <vehicle_service/POS_define.h>
#include <gps_hal.h>

Include dependency graph for POS_gps_API.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

### Classes

struct SENSOR_MSG_SEND_DAT

### Macros

#define CID_POSIF_REGISTER_LISTENER_GPS_TIME_SET_REQ 0x0780

*GPS time setting result delivery command ID.*

### Typedefs

typedef int32_t NAVIINFO_RET_API

*define return value of Get/Set GPS information API*

### Functions

int32_t POS_ReqGPSSetting (HANDLE hApp, SENSOR_MSG_SEND_DAT *p_data)
NAVIINFO_RET_API POS_SetGPSInfo (HANDLE hApp, NAVIINFO_ALL *navilocinfo)
NAVIINFO_RET_API POS_GetGPSInfo (HANDLE hApp, NAVIINFO_DIAG_GPS *navidiaginfo)

POS_RET_API POS_ReqGPSReset (HANDLE hApp, PCSTR ResName, uint8_t mode)
POS_RET_API POS_GetGPSVersion (HANDLE hApp, uint8_t buf_size, int8_t *buf, uint8_t *size)
POS_RET_API POS_RegisterListenerGPSTimeSetReq (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg)
POS_RET_API POS_SetGPStime (HANDLE hApp, POS_DATETIME *pstDateTime)
SENSOR_RET_API POS_RegisterListenerGPStime (HANDLE hApp, PCSTR notifyName, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming)
POS_RET_API POS_GetGPStime (HANDLE hApp, SENSOR_GPSTIME *dat)

---

**Detailed Description**

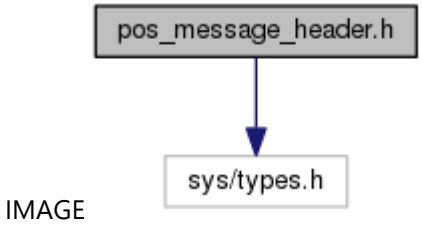
API definition file for GPS function.

# pos_message_header.h File Reference

Header file for message data definitions.

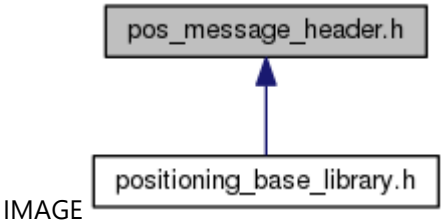
#include <sys/types.h>

Include dependency graph for pos_message_header.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Classes**

struct T_APIMSG_HEADER
struct T_APIMSG_MSGBUF_HEADER
struct _CWORD64_MSG_LOG_HDR

---

**Detailed Description**

Header file for message data definitions.

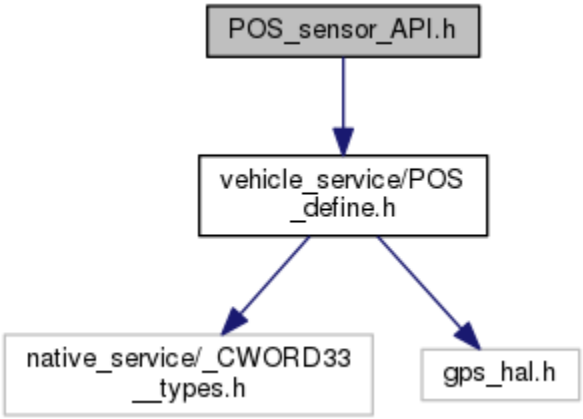
e

# POS_sensor_API.h File Reference

API definition file for Sensor function.

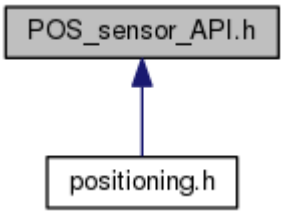
#include <vehicle_service/POS_define.h>

Include dependency graph for POS_sensor_API.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Classes**

struct SENSOR_PKG_MSG_VSINFO
struct SENSOR_MSG_VSINFO

**Macros**

#define POS_DID_SPEED_PULSE ***

*Data ID of speed pulse.*

#define POS_DID_SPEED_KMPH ***

*Data ID of KMPH speed.*

#define POS_DID_SNS_COUNTER ***

*Data ID of sensor counter.*

#define POS_DID_GYRO ***

*Data ID of gyro.*

#define POS_DID_GSNS_X ***

*Data ID of x axis gsensor.*

#define POS_DID_GSNS_Y ***

*Data ID of Y axis gsensor.*

#define POS_DID_REV ***

*Data ID of reverse signal.*

#define POS_DID_GPS_ANTENNA ***

*Data ID of GPS antenna status.*

#define POS_DID_SPEED_PULSE_FST ***

*Data ID of first time speed pulse.*

#define POS_DID_GYRO_FST ***

*Data ID of first time gyro.*

#define POS_DID_REV_FST ***

*Data ID of first time reverse signal.*

#define POS_DID_GYRO_TEMP ***

*Data ID of gyro temperature.*

#define POS_DID_GYRO_TEMP_FST ***

*Data ID of first time gyro temperature.*

#define POS_DID_GSNS_X_FST ***

*Data ID of first time x axis gsensor.*

#define POS_DID_GSNS_Y_FST ***

*Data ID of first time Y axis gsensor.*

#define POS_DID_PULSE_TIME ***

*Data ID of pulse time.*

#define POS_DID_GPS_CWORD44_NMEA ***

*Data ID of CWORD44 GPS NMEA sentence.*

#define POS_DID_GPS_CWORD44_P_CWORD44_DGP4 ***

*Data ID of CWORD44 GPS P_CWORD44_DGP4 data.*

#define POS_DID_GPS_CWORD44_FULLBINARY ***

*Data ID of CWORD44 GPS full binary data.*

#define POS_DID_GPS_NMEA ***

*Data ID of GPS NMEA sentence.*

#define POS_DID_GPS_CLOCK_DRIFT ***

*Data ID of GPS time drift data.*

#define POS_DID_GPS_CLOCK_FREQ ***

*Data ID of GPS time frequency data.*

#define CID_POSIF_REGISTER_LISTENER_PKG_SENSOR_DATA 0x0700

*sensor extra package register command ID*

#define CID_POSIF_REGISTER_LISTENER_SENSOR_DATA 0x0200

*sensor information register command ID*

#define SENSOR_MSGBUF_DSIZE 2264

*message body maximum size*

#define SENSOR_VSHEAD_DSIZE 36

*vehicle sensor header size(1+3+16*2)*

#define SENSOR_VSINFO_DSIZE (SENSOR_MSGBUF_DSIZE - SENSOR_VSHEAD_DSIZE)

*vehicle sensor data size*

#define SENSOR_MSG_VSINFO_DSIZE 1272

*vehicle sensor message body maximum size*

#define SENSOR_PKG_DELIVERY_MAX 16

*number of data ID per a package*

**Functions**

SENSOR_RET_API POS_RegisterListenerPkgSensData (HANDLE hApp, PCSTR notifyName, uint8_t ucPkgNum, DID *pulDid, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming)

SENSOR_RET_API POS_RegisterListenerSensData (HANDLE hApp, PCSTR notifyName, DID ulDid, uint8_t ucCtrlFlg, uint8_t ucDeliveryTiming)

POS_RET_API POS_GetSensData (HANDLE hApp, DID ulDid, void *pDestData, uint16_t usDestSize)

---

**Detailed Description**

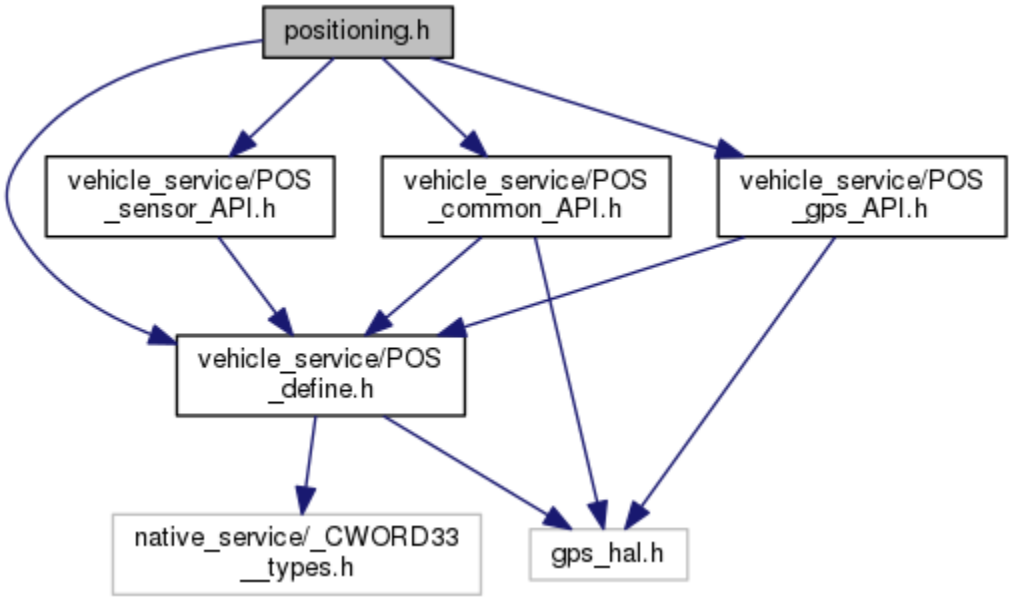
API definition file for Sensor function.

# positioning.h File Reference

Common header of positioning.

#include <vehicle_service/POS_define.h>
#include <vehicle_service/POS_sensor_API.h>
#include <vehicle_service/POS_gps_API.h>
#include <vehicle_service/POS_common_API.h>

Include dependency graph for positioning.h:




IMAGE

---

**Detailed Description**

Common header of positioning.

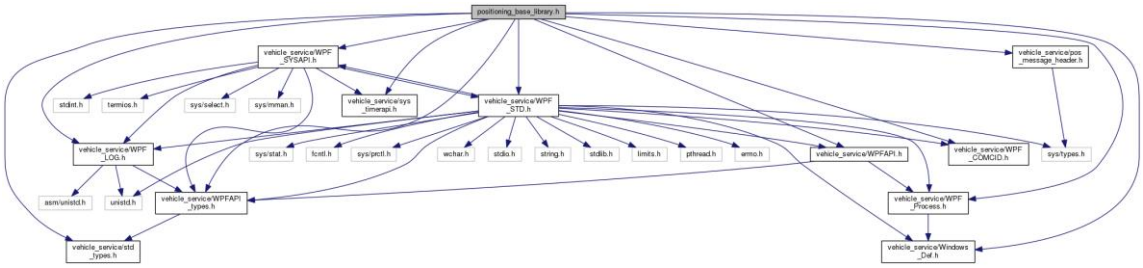
# positioning_base_library.h File Reference

Common header of positioning_base_library.

#include <vehicle_service/WPF_SYSAPI.h>
#include <vehicle_service/WPF_STD.h>
#include <vehicle_service/WPF_Process.h>
#include <vehicle_service/WPF_LOG.h>
#include <vehicle_service/WPF_COMCID.h>
#include <vehicle_service/WPFAPI_types.h>
#include <vehicle_service/WPFAPI.h>
#include <vehicle_service/Windows_Def.h>
#include <vehicle_service/sys_timerapi.h>
#include <vehicle_service/std_types.h>
#include <vehicle_service/pos_message_header.h>

Include dependency graph for positioning_base_library.h:

IMAGE




---

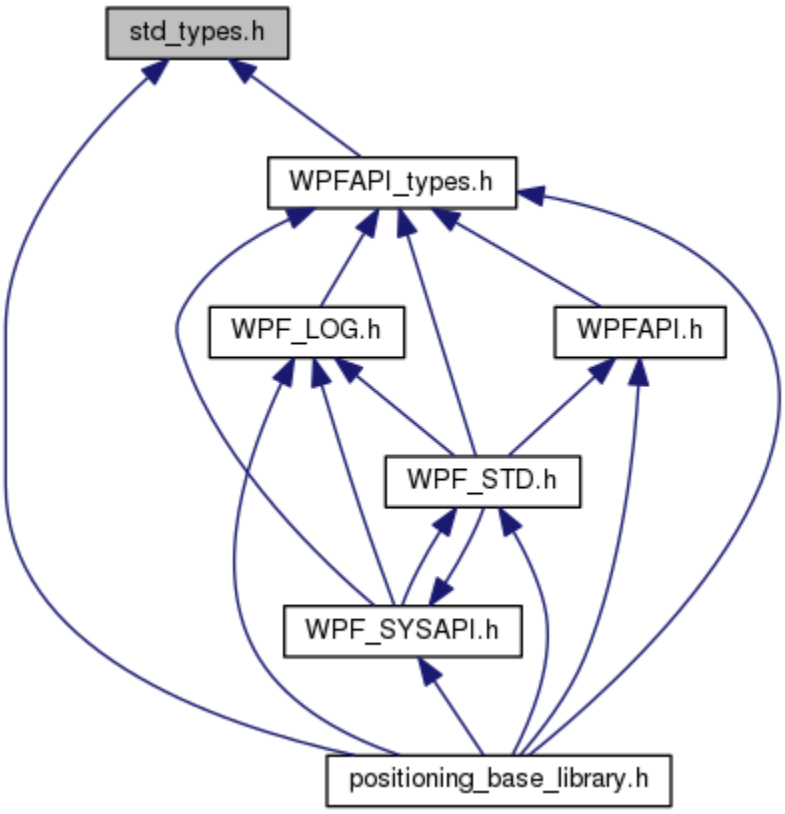
**Detailed Description**

Common header of positioning_base_library.

# std_types.h File Reference

Header file for base type definitions.

This graph shows which files directly or indirectly include this file:




IMAGE

**Macros**

#define **NULL** ((void *)0)

**Typedefs**

typedef signed char **int8**
typedef unsigned char **u_int8**
typedef signed short int **int16**
typedef unsigned short int **u_int16**
typedef signed int **int32**
typedef unsigned int **u_int32**

---

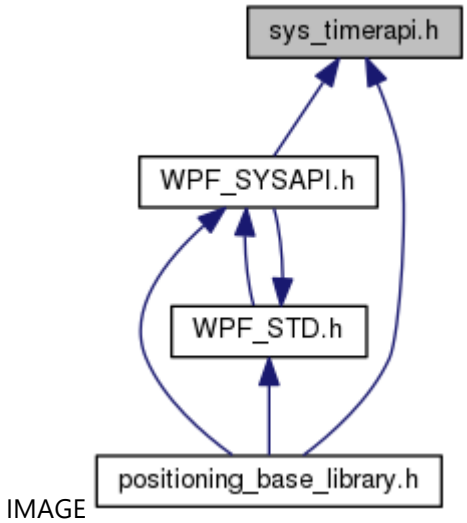
**Detailed Description**

Header file for base type definitions.

# sys_timerapi.h File Reference

Header file for timer management.

This graph shows which files directly or indirectly include this file:



IMAGE

**Classes**

struct TimerDupData
struct RealTimeData
struct RealTimeDataBcd
struct GpsSetData
struct TimerToutMsg
struct TimerTimeReq
struct TimerAlrmReq
struct TimerStopReq
struct GpsSetTimeReq
struct SetDiffTimeReq
struct T_TIM_RCV_DATA
struct TimerSramData
struct SysTimeData
struct TimeCnvTbl_tag
struct DayCnvTbl_tag
struct TG_TIM_ROLOVR_GPSWEEKCORDATA
struct TIMECHGIND_MSG
struct T_TIM_CMDMSG
struct T_TIM_RETTIMEMSG
struct TimerModDat

**Macros**

#define WTM_DUPTIME_MAX 5
*duplication timer max count*

#define TIMER_TYPE_SYN 1

*fixingcycletimer(unit: 100ms)*

#define TIMER_TYPE_USN 2

*asynchronous timer(unit: 100ms)*

#define TIMER_TYPE_ALM 5

*specific time timer*

#define SUN 0

*day*

#define MON 1

*month*

#define TUE 2

*tuesday*

#define WED 3

*wednesday*

#define THU 4

*thursday*

#define FRI 5

*friday*

#define SAT 6

*saturday*

#define DEFAULT_DIFFTIME 0x00

*time difference value of japan*

#define GPS_SAT 0x00

*correctedGPS time*

#define GPS_IST 0x01

*not corrected GPS time*

#define GPS_RTC 0x02

*RTC time.*

#define GPS_IRG 0x03

*initial time*

#define GPS_NON 0x7E

*GPS unused.*

#define GPS_IGN 0x7F

*time not received*

#define END_TIMER_RPT_OFF 0

*No continuation.*

#define END_TIMER_RPT_ON 1

*With continuation.*

#define CID_TIMER_SETGPS 0x4040

*command ID of GPS setting timer*

#define CID_TIMER_DIFSET 0x4045

*command ID of time difference setting timer*

#define CID_TIMER_CYCLE (CID)(CID_TIMER_BASE | CID_COM_SERIAL5)

*command ID of cycle data receive timer*

#define CNV_YEAR_MAX 136

*max value of conversion year(number of years + 1)*

#define SEC_PER_MIN 60

*60[sec]*

#define SEC_PER_HOUR 3600

*60[min]60[sec]*

#define SEC_PER_DAY 86400

*24[h]60[min]60[sec]*

#define TIM_ROLOVR_DISABLE 0

*GPS week rollover correct prohibition.*

#define TIM_ROLOVR_ENABLE 1

*GPS week rollover correct permission.*

#define RET_ERR_NONEINIT (-30)

*initialization untreated error*

#define RET_ERR_SEMLOCK (-31)

*get semaphore error*

#define RET_ERR_SEMUNLOCK (-32)

*open semaphore error*

#define RET_ERR_SRAMREAD (-33)

*read SRAM error*

#define TIM_NUM_SNDCMD 4

*command send number*

#define NON_TIMEUNIT_INVALID_TIME 0

*No time manager unit - system time not set.*

#define NON_TIMEUNIT_VALID_TIME 1

*No time manager unit - system time set complete.*

#define GPS_INVALID_TIME 2

*With GPS unit - system time not set.*

#define GPS_VALID_TIME 3

*With GPS unit - system time set complete.*

#define TMT_RETRY 3

*retry count*

#define TMT_RETRY_INTERVAL 500

*retry interval(500[msec])*

#define TMT_GPSDM_NAME "TIMER_GPS_DATA"

*GPS status storage data module name.*

#define TMT_GPSDM_SEM_NAME "TIM_SEM"

*GPS status storage data module semaphore name.*

**Typedefs**

typedef struct TimeCnvTbl_tag **YearCntTbl**
typedef struct DayCnvTbl_tag **DayCntTbl**

---

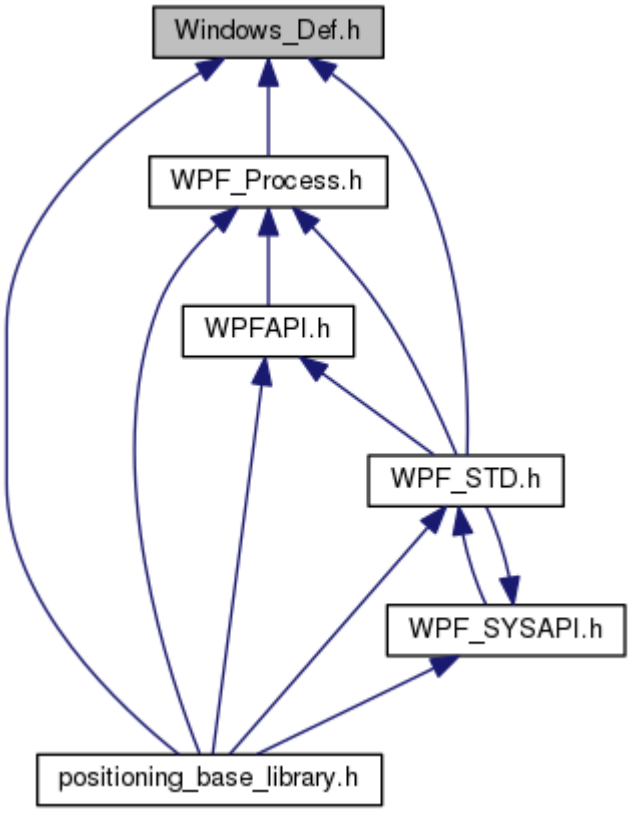
**Detailed Description**

Header file for timer management.

# Windows_Def.h File Reference

Header file for type definitions from windows.

This graph shows which files directly or indirectly include this file:




IMAGE

**Classes**

struct _SYSTEMTIME
struct _SECURITY_ATTRIBUTES

**Macros**

#define INFINITE 0xFFFFFFFF
*infinite*

**Typedefs**

typedef char **CHAR**
typedef unsigned long **DWORD**
typedef unsigned short **WORD**
typedef unsigned char **UCHAR**
typedef wchar_t **WCHAR**
typedef unsigned char **BYTE**
typedef char **TCHAR**

typedef void * **LPVOID**
typedef void * **PVOID**
typedef void * **HANDLE**
typedef unsigned long **ULONG**
typedef void **VOID**
typedef const CHAR * **PCSTR**
typedef const CHAR * **LPCSTR**
typedef LPCSTR **LPCTSTR**
typedef DWORD * **PDWORD**
typedef DWORD * **LPDWORD**
typedef UCHAR * **PUCHAR**
typedef BYTE * **PBYTE**
typedef struct _SYSTEMTIME **SYSTEMTIME**
typedef SYSTEMTIME * **PSYSTEMTIME**
typedef SYSTEMTIME * **LPSYSTEMTIME**
typedef struct _SECURITY_ATTRIBUTES **SECURITY_ATTRIBUTES**
typedef SECURITY_ATTRIBUTES * PSECURITY_ATTRIBUTES

*security attribute*

typedef SECURITY_ATTRIBUTES * LPSECURITY_ATTRIBUTES

*security attribute*

---

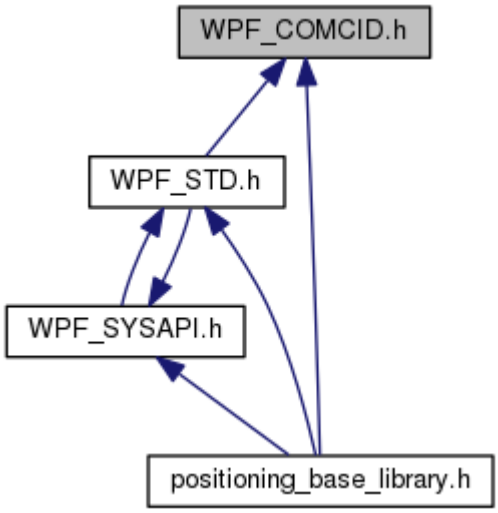
**Detailed Description**

Header file for type definitions from windows.

# WPF_COMCID.h File Reference

Header file for command ID definitions.

This graph shows which files directly or indirectly include this file:




IMAGE

**Macros**

#define CID_CWORD43_BASE (CID)0xF010

*CWORD43 (Audio/_CWORD47_ Communication)*

#define CID_CWORD42_BASE (CID)0xF020

*CWORD42 (traditional CWORD36 )*

#define CID_GPS_BASE (CID)0xF030

*GPS.*

#define CID_TIMER_BASE (CID)0xF040

*timer management*

#define CID_VOUT_BASE (CID)0xF050

*voice output*

#define CID_CDCTRL_BASE (CID)0xF060

*CD control.*

#define CID_CDACS_BASE (CID)0xF070

*CD access.*

#define CID_DIAG_BASE (CID)0xF080

*diag*

#define CID_BRD_BASE (CID)0xF090

*search department*

#define CID_PCMCIA_BASE (CID)0xF0A0

*PCMCIA.*

#define CID_FM_BASE (CID)0xF0B0

*FM multiple.*

#define CID_VTHMNG_BASE (CID)0xF0D0

*sound output control*

#define CID_ETCCOM_BASE (CID)0xF0E0

*ETC Communication.*

#define CID_INITIAL_BASE (CID)0xF100

*initial*

#define CID_DEV_TIMER_BASE (CID)0xF200

*timer register driver*

#define CID_STORAGE_BASE (CID)0xF210

*storage*

#define CID_RESMGR_BASE (CID)0xF220

*ResourceManager.*

#define CID_COM_SERIAL0 (CID)0x0000

*serial command ID 0*

#define CID_COM_SERIAL1 (CID)0x0001

*serial command ID 1*

#define CID_COM_SERIAL2 (CID)0x0002

*serial command ID 2*

#define CID_COM_SERIAL3 (CID)0x0003

*serial command ID 3*

#define CID_COM_SERIAL4 (CID)0x0004

*serial command ID 4*

#define CID_COM_SERIAL5 (CID)0x0005

*serial command ID 5*

#define CID_COM_SERIAL6 (CID)0x0006

*serial command ID 6*

#define CID_COM_SERIAL7 (CID)0x0007

*serial command ID 7*

#define CID_COM_SERIAL8 (CID)0x0008

*serial command ID 8*

#define CID_COM_SERIAL9 (CID)0x0009

*serial command ID 9*

#define CID_COM_SERIAL10 (CID)0x000A

*serial command ID 10*

#define CID_COM_SERIAL11 (CID)0x000B

*serial command ID 11*

#define CID_COM_SERIAL12 (CID)0x000C

*serial command ID 12*

#define CID_COM_SERIAL13 (CID)0x000D

*serial command ID 13*

#define CID_COM_SERIAL14 (CID)0x000E

*serial command ID 14*

#define CID_COM_SERIAL15 (CID)0x000F

*serial command ID 15*

#define CID_COM_SERIAL16 (CID)0x0010

*serial command ID 16*

#define CID_COM_SERIAL17 (CID)0x0011

*serial command ID 17*

#define CID_COM_SERIAL18 (CID)0x0012

*serial command ID 18*

#define CID_COM_SERIAL19 (CID)0x0013

*serial command ID 19*

#define CID_CWORD43_CMDRCV (CID)(CID_CWORD43_BASE | CID_COM_SERIAL0)

*command reception notification*

#define CID_CWORD43_CMDSNDREQ (CID)(CID_CWORD43_BASE | CID_COM_SERIAL1)

*command send request notification*

#define CID_CWORD43_CMDSNDANS (CID)(CID_CWORD43_BASE | CID_COM_SERIAL2)

*command send result notification*

#define CID_CWORD43_ADRINF (CID)(CID_CWORD43_BASE | CID_COM_SERIAL3)

*equipment address info notification*

#define CID_CWORD42_SMALLCMDRCV (CID)(CID_CWORD42_BASE | CID_COM_SERIAL0)

*small scale command receive notify*

#define CID_CWORD42_LARGECMDRCV (CID)(CID_CWORD42_BASE | CID_COM_SERIAL1)

*large scale command receive notify*

#define CID_CWORD42_BUFFREQ (CID)(CID_CWORD42_BASE | CID_COM_SERIAL2)

*large scale buffer request*

#define CID_CWORD42_BUFFANS (CID)(CID_CWORD42_BASE | CID_COM_SERIAL3)

*large scale buffer response*

#define CID_CWORD42_CMDSNDREQ (CID)(CID_CWORD42_BASE | CID_COM_SERIAL4)

*command send request notify*

#define CID_CWORD42_CMDSNDANS (CID)(CID_CWORD42_BASE | CID_COM_SERIAL5)

*command send answers notify*

#define CID_CWORD42_SNDSTOPIND (CID)(CID_CWORD42_BASE | CID_COM_SERIAL6)

*send data cancel indication*

#define CID_CWORD42_LANSTATE (CID)(CID_CWORD42_BASE | CID_COM_SERIAL7)

*LAN state notify.*

#define CID_CWORD42_DEVENTRY (CID)(CID_CWORD42_BASE | CID_COM_SERIAL8)

*info change notify*

#define CID_GPS_CMDSNDREQ (CID)(CID_GPS_BASE | CID_COM_SERIAL1)

*command send request notify*

#define CID_GPS_CMDRCVANS (CID)(CID_GPS_BASE | CID_COM_SERIAL2)

*command receive answers notify*

#define CID_GPS_RETRESET (CID)(CID_GPS_BASE | CID_COM_SERIAL3)

*GPS reset response.*

#define CID_TIMER_TIMREQ (CID)(CID_TIMER_BASE | CID_COM_SERIAL0)

*timer start indication*

#define CID_TIMER_ALMREQ (CID)(CID_TIMER_BASE | CID_COM_SERIAL1)

*timer start indication with specified time*

#define CID_TIMER_STPREQ (CID)(CID_TIMER_BASE | CID_COM_SERIAL2)

*timer stop indication*

#define CID_TIMER_DUPREQ (CID)(CID_TIMER_BASE | CID_COM_SERIAL3)

*multi timer start indication*

#define CID_TIMER_TOUT (CID)(CID_TIMER_BASE | CID_COM_SERIAL4)

*timeout notify*

#define CID_TIMER_TIMCHG (CID)(CID_TIMER_BASE | CID_COM_SERIAL5)

*time change notify*

#define CID_TIMER_ALMANAC (CID)(CID_TIMER_BASE | CID_COM_SERIAL6)

*almanac abnormality notify*

#define CID_TIMER_TIME_VALID (CID)(CID_TIMER_BASE | CID_COM_SERIAL7)

*time enable notify*

#define CID_TIMER_SETTIME (CID)(CID_TIMER_BASE | CID_COM_SERIAL8)

*GPS time setting response notify.*

#define CID_CLKMNG_SETTIME (CID)(CID_TIMER_BASE | CID_COM_SERIAL9)

*time setting Indication*

#define CID_CLKMNG_NOTIACCURACY (CID)(CID_TIMER_BASE | CID_COM_SERIAL10)

*notify*

#define CID_CLKMNG_NOTIADJUST (CID)(CID_TIMER_BASE | CID_COM_SERIAL11)

*time setting change notify(unused)*

#define CID_CLKMNG_NOTIOBSERVERS (CID)(CID_TIMER_BASE | CID_COM_SERIAL12)

*fixingcyclenotify*

#define CID_CLKMNG_NOTIACCURACY_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL13)

*notify entry*

#define CID_CLKMNG_NOTIADJUST_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL14)

*time setting change notify entry(unused)*

#define CID_CLKMNG_NOTIOBSERVERS_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL15)
*fixingcyclenotify entry*

#define CID_CLKMNG_AUTOSETREALTIME_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL16)
*fixingcyclenotify entry*

#define CID_CLKMNG_SETTIMEZONE (CID)(CID_TIMER_BASE | CID_COM_SERIAL17)
*fixingcyclenotify entry*

#define CID_CLKMNG_AUTOSETTIMEZONE_ENTRY (CID)(CID_TIMER_BASE | CID_COM_SERIAL18)
*fixingcyclenotify entry*

#define CID_CLKMNG_SETREALTIME (CID)(CID_TIMER_BASE | CID_COM_SERIAL19)
*fixingcyclenotify entry*

#define CID_CDCTRL_SPECIFYDRIVE (CID)(CID_CDCTRL_BASE | CID_COM_SERIAL0)
*< unused*

#define CID_CDCTRL_CD_STATUS (CID)(CID_CDCTRL_BASE | CID_COM_SERIAL1)
*CD status notify.*

#define CID_CDACS_RET_MAP_MAN_PTR (CID)(CID_CDACS_BASE | CID_COM_SERIAL0)
*info address return*

#define CID_CDACS_RET_FILE_PTR (CID)(CID_CDACS_BASE | CID_COM_SERIAL1)
*file read address return*

#define CID_CDACS_RET_MALLOC (CID)(CID_CDACS_BASE | CID_COM_SERIAL2)
*malloc return*

#define CID_CDACS_RET_CD_READ (CID)(CID_CDACS_BASE | CID_COM_SERIAL3)
*CD read return.*

#define CID_DIAG_MODENOTC (CID)(CID_DIAG_BASE | CID_COM_SERIAL0)
*normal diag transition Indication*

#define CID_DIAG_SYSCHKREQ (CID)(CID_DIAG_BASE | CID_COM_SERIAL1)
*diag system check request*

#define CID_BRD_IDRET (CID)(CID_BRD_BASE | CID_COM_SERIAL0)
*ID obtain response.*

#define CID_PCMCIA_CARD (CID)(CID_PCMCIA_BASE | CID_COM_SERIAL0)
*PCMCIA command receive notify.*

#define CID_FMDRV_DATARCV (CID)(CID_FM_BASE | CID_COM_SERIAL0)
*data receive end notify*

#define CID_VTH_MNG_VOICECTREND (CID)(CID_VTHMNG_BASE | CID_COM_SERIAL0)
*control sound end notify*

#define CID_ETC_CMDRCV (CID)(CID_ETCCOM_BASE | CID_COM_SERIAL0)
*ETC receive notify.*

#define CID_ETC_CMDSNDANS (CID)(CID_ETCCOM_BASE | CID_COM_SERIAL1)
*ETC send result notify.*

#define CID_INI_ACCOFF (CID)(CID_INITIAL_BASE | CID_COM_SERIAL0)

*ACC-OFF Indication.*

#define CID_INI_SD_INOUT (CID)(CID_INITIAL_BASE | CID_COM_SERIAL1)

*SD insertion notify.*

#define CID_INI_MAPMEDIA_STATUS (CID)(CID_INITIAL_BASE | CID_COM_SERIAL2)

*map media state notify*

#define CID_INI_ACCOFF_COMP (CID)(CID_INITIAL_BASE | CID_COM_SERIAL3)

*ACC-OFF process complete notify.*

#define CID_INI_STEPFORK (CID)(CID_INITIAL_BASE | CID_COM_SERIAL4)

*step start Indication*

#define CID_INI_STEPFORK_COMP (CID)(CID_INITIAL_BASE | CID_COM_SERIAL5)

*step start complete Indication*

#define CID_INI_SYSTEMERR (CID)(CID_INITIAL_BASE | CID_COM_SERIAL6)

*system abnormality notify*

#define CID_INI_INTERNAL_ACCOFF_START (CID)(CID_INITIAL_BASE | CID_COM_SERIAL7)

*internal ACC-OFF start Indication*

#define CID_INI_INTERNAL_ACCOFF_COMP (CID)(CID_INITIAL_BASE | CID_COM_SERIAL8)

*internal ACC-OFF complete Indication*

#define CID_DEV_TIMER_UP (CID)(CID_DEV_TIMER_BASE | CID_COM_SERIAL0)

*timer timeout notify*

#define CID_DEV_TIMER_START (CID)(CID_DEV_TIMER_BASE | CID_COM_SERIAL1)

*timer start Indication*

#define CID_DEV_TIMER_STOP (CID)(CID_DEV_TIMER_BASE | CID_COM_SERIAL2)

*timer stop Indication*

#define CID_STRMGR_DEVINSERTNOT (CID)(CID_STORAGE_BASE | CID_COM_SERIAL0)

*device insertion notify*

#define CID_STRMGR_DEVFIXNOT (CID)(CID_STORAGE_BASE | CID_COM_SERIAL1)

*device fix notify*

#define CID_STRMGR_DEVERRNOT (CID)(CID_STORAGE_BASE | CID_COM_SERIAL2)

*device abnormality notify*

#define CID_USBCTL_CONNDEV_NTF (CID)(CID_STORAGE_BASE | CID_COM_SERIAL6)

*USB Device Connect Notify.*

#define CID_USBCTL_CTRLPOWER_IND (CID)(CID_STORAGE_BASE | CID_COM_SERIAL7)

*Control Device Power Indication.*

#define CID_USBCTL_COMPTERM (CID)(CID_STORAGE_BASE | CID_COM_SERIAL8)

*USB observation complete notify.*

#define CID_USBCTL_CONNECTSTS (CID)(CID_STORAGE_BASE | CID_COM_SERIAL9)

*USB connect status notify.*

#define CID_USBCTL_INSERTSTS (CID)(CID_STORAGE_BASE | CID_COM_SERIAL10)

*USB insertion notify.*

#define CID_USBCTL_ERRORSTS (CID)(CID_STORAGE_BASE | CID_COM_SERIAL11)

*USB abnormality notify.*

#define CID_SDSDRV_INSERTSTS (CID)(CID_STORAGE_BASE | CID_COM_SERIAL12)

*SD insertion notify.*

#define CID_SDSDRV_COMPTERM (CID)(CID_STORAGE_BASE | CID_COM_SERIAL13)

*SD observation complete notify.*

#define CID_SDSDRV_CONNDEV_NTF (CID)(CID_STORAGE_BASE | CID_COM_SERIAL14)

*SD Device Connect Notify::GPF_001#.*

#define CID_RESMGR_ERROR_INFO_CHG (CID)(CID_RESMGR_BASE | CID_COM_SERIAL0)

*error info change notify*

#define CID_DEV_REQGPS MAKECID(CID_DEVHIGH, 0x00)

*GPS request.*

#define CID_DEV_REQRST MAKECID(CID_DEVHIGH, 0x01)

*gps reset request*

#define CID_GPS_SERIAL0 (CID)0x0100

*GPS reset request func definition.*

#define CID_GPS_REQRESET (CID)(CID_GPS_BASE | CID_GPS_SERIAL0)

*GPS reset request CID.*

---

**Detailed Description**

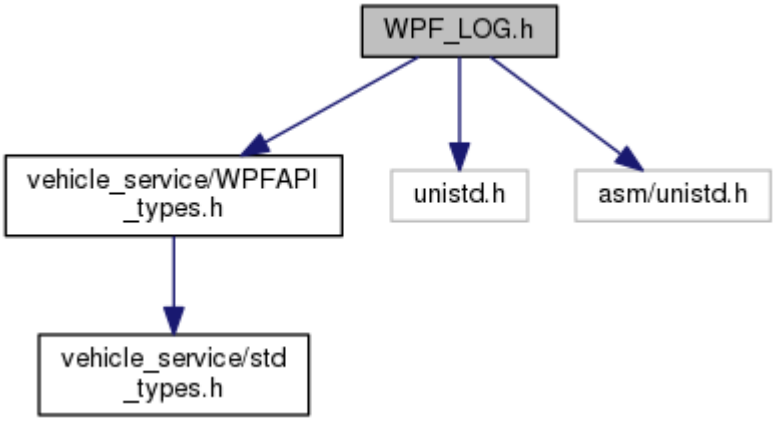
Header file for command ID definitions.

# WPF_LOG.h File Reference

Header file for log definitions.

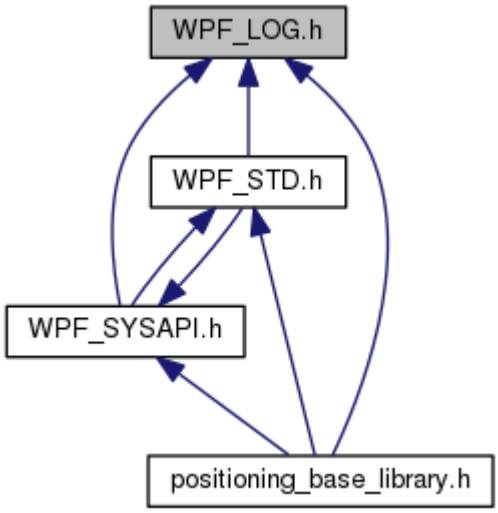
#include <vehicle_service/WPFAPI_types.h>
#include <unistd.h>
#include <asm/unistd.h>

Include dependency graph for WPF_LOG.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Macros**

#define DEBUG_DUMP_MAX_SIZE 4096

*_CWORD33_OnDebugDump max size*

#define KIND_DEBUG (0x00000001)

*for debug log*

#define KIND_ERROR (0x00000002)

*for error output log*

#define KIND_RELEASE (0x00000004)

*for release output log*

#define FLAG_IMPORTANT KIND_RELEASE

*flag for important log*

#define POS_DEBUG_LOGOUT_LEVEL 4

*debug log output level*

#define **DEBUG**

#define _CWORD33_LOG(zone, opt, fmt, ...)

*CWORD33 log output*

#define POS_DEBUG_LOGLEV_1 1

*debug log output level 1*

#define POS_DEBUG_LOGLEV_2 2

*debug log output level 2*

#define POS_DEBUG_LOGLEV_3 3

*debug log output level 3*

#define POS_DEBUG_LOGLEV_4 4

*debug log output level 4*

#define POS_DEBUG_LOGLEV_5 5

*debug log output level 5*

#define POS_SENSLOG_TYPE_NONE 0

*sensor log type NONE*

#define POS_SENSLOG_TYPE_SYS 1

*sensor log type SYS*

#define POS_SENSLOG_TYPE_GPS 2

*sensor log type GPS*

#define POS_SENSLOG_TYPE_NAV 3

*sensor log type NAV*

#define POS_SENSLOG_TYPE_CMD 4

*sensor log type CMD*

#define SNR_FUNC_DEBUG_MSG_CWORD71_FILE 0

*Sensor file Log valid:1,invalid:0.*

#define SNR_FUNC_DEBUG_MSG_CWORD71 1

*Sensor Log valid:1,invalid:0.*

#define GPS_FUNC_DEBUG_MSG_CWORD71 1

*GPS Log valid:1,invalid:0.*

#define COM_FUNC_DEBUG_MSG_CWORD71 1

*COMMON Log valid:1,invalid:0.*

#define BASE_FUNC_DEBUG_MSG_CWORD71 1

*BASE Log valid:1,invalid:0.*

#define MSG_FUNC_DEBUG_MSG__CWORD71_ 1

*MESSAGE Log valid:1,invalid:0.*

#define DEV_FUNC_DEBUG_MSG__CWORD71_ 1

*GPS DEV Log valid:1,invalid:0.*

#define RELTEXT(FLAG, QUOTE) __FUNCTION__, (QUOTE)

*output text format*

#define WPF_RETAILMSG_LEV(level, zone, ...) (((level) <= (POS_DEBUG_LOGOUT_LEVEL))?((void)(( CWORD33_LOG(zone, __VA_ARGS__)))):((void)(0)))

*message log output*

#define POS_WRITE_SENSLOG(zone, data, len) (((zone) != (0))?((void)((_CWORD33_LOG_DATA(zone, data, len)))):((void)(0)))

*write sensor log output*

#define WPF_RETAILMSG(PRINTF_FMT) ((void)(0))

*message log output*

#define POS_SNR_GET_ZONE(level)

*get sensor zone code*

#define POS_GPS_GET_ZONE(level)

*get GPS zone code*

#define POS_BASE_GET_ZONE(level)

*get base zone code*

#define POS_COM_GET_ZONE(level)

*get common zone code*

#define POS_MSG_GET_ZONE(level)

*get message zone code*

#define POS_DEV_GET_ZONE(level)

*get DEV zone code*

#define POS_SENSLOG_GET_ZONE(typ)

*get sensor log zone code*

#define POS_SNR_EXLOG(level, ...) _CWORD33_LOG(POS_SNR_GET_ZONE(level), __VA_ARGS__);

*sensor extension log output*

#define POS_GPS_EXLOG(level, ...) _CWORD33_LOG(POS_GPS_GET_ZONE(level), __VA_ARGS__);

*GPS extension log output.*

#define POS_BASE_EXLOG(level, ...) _CWORD33_LOG(POS_BASE_GET_ZONE(level), __VA_ARGS__);

*Base extension log output.*

#define POS_COM_EXLOG(level, ...) _CWORD33_LOG(POS_COM_GET_ZONE(level), __VA_ARGS__);

*Common extension log output.*

#define POS_MSG_EXLOG(level, ...) _CWORD33_LOG(POS_MSG_GET_ZONE(level), __VA_ARGS__);

*message extension log output*

#define POS_DEV_EXLOG(level, ...) _CWORD33_LOG(POS_DEV_GET_ZONE(level), __VA_ARGS__);

*DEV extension log output.*

```
#define POS_SENSLOG(type, data, len) printf("POS_SENSLOG\n");
```

*sensor log output*

---

**Detailed Description**

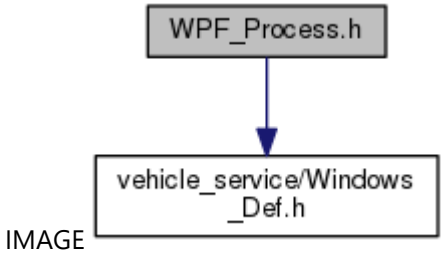
Header file for log definitions.

# WPF_Process.h File Reference

Header file for PNO definitions.

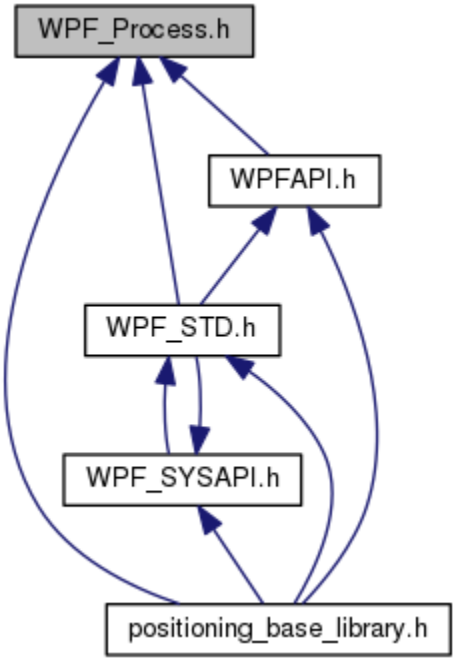
#include <vehicle_service/Windows_Def.h>

Include dependency graph for WPF_Process.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Macros**

#define PNO_NONE 0x0000

*initial value*

#define PNO_CWORD88_CWORD23 0x0100

*CWORD23 primary thread*

#define PNO_RDSDC_THREAD 0x0130

*RDS decoder control thread.*

#define PNO_RDSRCV_THREAD 0x0132

*RDS data communication thread.*

#define PNO_TMC_DATAMNG 0x0133

*TMC data manager.*

#define PNO_CWORD24_MAIN 0x01F0

*CWORD24 main thread*

#define PNO_DEV_TOUCH_READ 0x0202

*touch input control*

#define PNO_CLOCK_TIME_STAMP 0x0208

*clock timestamp thread*

#define PNO_CLK_MNG 0x0209

*time manager thread*

#define PNO_DEV_MNG_MAIN 0x020A

*device manager thread*

#define PNO_DEV_SYSCOM_RCV 0x0210

*Navi-SYS communication/reception control.*

#define PNO_DEV_SYSCOM_SND 0x0211

*Navi-SYS communication/transmission control.*

#define PNO_DEV_SYSCOM_TIMER 0x0212

*Navi-SYS communication driver timer thread.*

#define PNO_DEV_SYSCOM_MAIN 0x0213

*Navi-SYS communication driver main thread.*

#define PNO_DEV_MCSUB_MAIN 0x0215

*SUB communication control driver main thread.*

#define PNO_DEV_MCSUB_RECV 0x0216

*SUB communication control driver receive thread.*

#define PNO_CDFS_MAIN 0x0235

*CDFS thread.*

#define PNO_DVDFS_MAIN 0x0236

*DVDFS thread.*

#define PNO_SIMPLE_LOADER 0x0237

*loader thread*

#define PNO_FLSMNG_THREAD 0x02DA

*backup data manager thread*

#define PNO_DEV_CWORD57_MODEL 0x02E0

*_CWORD57_Model thread*

#define PNO_DEV_CWORD57_COMRCV 0x02E1

*CWORD57 command reception thread*

#define PNO_DEV_CWORD57_MONITOR 0x02E2

*CWORD57 communication monitor thread*

#define PNO_DEV_CWORD57_LIST 0x02E5

*CWORD57 list manager thread*

#define PNO_DEV_CWORD57_TAG_WRITER 0x02E6

*Tagging data read thread.*

#define PNO_DEV_CWORD57_AUTH 0x02EE

*CWORD57 certification control primary thread*

#define PNO_VIDEODECORD 0x02F8

*video decoder PNO*

#define PNO_DEV_USBCTL_MONITOR 0x02F9

*USB monitor driver.*

#define PNO_CMRDRVTHREAD 0x02FA

*camera driver thread*

#define PNO_DEV_SDSDRV 0x02FE

*SD monitor driver.*

#define PNO_GINI_ERRLOG 0x0303

*error log HDD storage thread*

#define PNO_GINI_VERUP 0x0306

*version up thread*

#define PNO_DEV_SDAS_DIAG 0x0312

*SDAS driver diag register thread.*

#define PNO_DEV_SDAS_IST 0x0313

*SDAS driver IST.*

#define PNO_DEV_SDAS_DRV 0x0314

*SDAS driver mount thread.*

#define PNO_DEV_SDAS_TIMER 0x0315

*SDAS driver timer thread.*

#define PNO_VEHICLE_SENSOR 0x0363

*vehicle sensor thread*

#define PNO_LINE_SENS_DRV 0x0364

*line sensor driver thread*

#define PNO_CAN_COM_PROT 0x0365

*CAN thread.*

#define PNO_CANCMR_MIDDLE 0x0366

*CAN camera middle thread.*

#define PNO_VEHICLE_INFO 0x0367

*vehicle information#053#*

#define PNO_VEHICLE_SWMANAGER 0x0368

*vehicle information#053#*

#define PNO_VEHICLE_VEHICLE 0x0371

*vehicle information#053# #GPF_60_001*

#define PNO_VEHICLE_TOUCH_INT 0x0372

#define PNO_SCC_SOUND_STOP 0x0383

*sound output end detection*

#define PNO_SCCSOUND_PLAY 0x0384

*SCCSound playback thread.*

#define PNO_SCCSOUND_SYNTH 0x0385

*SCCSound composition thread.*

#define PNO_VUP_CTRL 0x0394

*VUP control thread.*

#define PNO_VUPCTRL_CHK 0x0304

*version judgement thread*

#define PNO_CWORD87_LAN 0x03AB

*CWORD87 -LAN thread#015#*

#define PNO_CWORD38 0x03AC

*CWORD39 thread#017#*

#define PNO_LAN_SERVER 0x03AD

*LAN server thread#018#.*

#define PNO_LAN_CWORD105 0x03AE

*LAN CWORD105*

#define PNO_LAN_CWORD105_DUMMY 0x03AF

*LAN _CWORD105_dummy.*

#define PNO_MLB 0x03C0

*MLB.*

#define PNO_CWORD65 0x03C1

*CWORD65*

#define PNO_MLB_INTR 0x03C2

*MediaLB Peripheralinterruptionthread.*

#define PNO_DIRECTSHOW_MONITOR_MAIN 0x03B0

*Dshow monitor main thread.*

#define PNO_DIRECTSHOW_MONITOR_SUB 0x03B1

*Dshow monitor sub thread.*

#define PNO_USBAUDIO_WORKER_DIRECTSHOW 0x03E1

#define PNO_USBAUDIO_WORKER_TIMER 0x03E2

*USB audio worker thread(Timer)#008#.*

#define PNO_USBAUDIO_FILE_LIST 0x03E3

*USB audio file list thread#009#.*

#define PNO_USBAUDIO_MAIN 0x03E4

*USB audio main thread#010#.*

#define PNO_DISPMNG_MAIN 0x03F8

*image manager process main thread*

#define PNO_STRMNG_MAIN 0x03F9

*storage manager thread*

#define PNO_STRMNG_JUDGE 0x03FA

*storage device judgement thread*

#define PNO_CAM_MIDDLE 0x03FB

*camera middle thread*

#define PNO_ERR_MONITOR 0x03FC

*painting/image error monitor thread#001#*

#define PNO_DISPMNG_SINK_IO 0x03FD

*display manager_SINK_I/O thread#006#*

#define PNO_STR_DEV_LOGGER 0x03FE

*storage device logger thread#042#*

#define PNO_CTPMONITOR 0x03FF

*capacitance panel monitor thread#054#*

#define PNO_NAVI_STM_MAIN 0x0430

*storage manager thread*

#define PNO_NAVI_GPS_MAIN 0x0440

*GPS communication manager thread.*

#define PNO_NAVI_GPS_RCV 0x0441

*GPS receive thread.*

#define PNO_NAVI_LOCA_SENS 0x0442

*sensor thread*

#define PNO_NAVI_GPS_INT 0x0443

*GPS interruption thread#031#.*

#define PNO_NAVI_PROXY 0x0444

*Navi proxy thread#050#.*

#define PNO_NAVI_CORE 0x0445

*Navi core thread#050#.*

#define PNO_NAVI_CORE_IF 0x0446

*Navi core IF thread#050#.*

#define PNO_NAVI_MVACN_MID 0x0454

*voice recognition middle thread#002#*

#define PNO_RESMGR_THREAD_MAIN 0x0460

*resource manager thread::GPF_12_001#*

#define PNO_NAVI_DIAG_MANAGER 0x04A0

*diag manager thread*

#define PNO_NAVI_DIAG_SUB 0x04A1

*diag sub thread*

#define PNO_NAVI_SYSLOG_HDD 0x04A2

*SYS log HDD storage thread.*

#define PNO_DIAG_WORKER 0x04A3

*diag worker thread#016#*

#define PNO_BT_CMD 0x0530

*BT command send/receive manager thread.*

#define PNO_BT_MNG 0x0531

*BT control thread.*

#define PNO_BTAUD_MNG 0x0532

*BT audio control thread.*

#define PNO_BT_AVP 0x0533

*BTAVP thread.*

#define PNO_PB_MNG 0x0534

*telephone book thread*

#define PNO_PB_VCARD 0x0535

*telephone book VCARD thread*

#define PNO_BTCOM_MNG 0x0540

*BT communication manager thread.*

#define PNO_BTCOM_PRT 0x0541

*BT communication protocol thread.*

#define PNO_BTCOM_RCV 0x0542

*BT communication reception thread.*

#define PNO_VGUIDE_SNDSRCMNG 0x06FF

*register manager thread#021#*

#define PNO_ANA_THREAD 0x0769

*analysis support thread#020#*

#define PNO_COVEROS_OPERATE_BASE_OUT 0x07B0

*playback/recording thread(play base No.)*

#define PNO_COVEROS_OPERATE_OUT_MAX 0x07B9

*playback/recording thread(maximum Num of playback)*

#define PNO_COVEROS_OPERATE_BASE_IN 0x07C0

*playback/recording thread(input base No.)*

#define PNO_COVEROS_OPERATE_IN_MAX 0x07C9

*playback/recording thread(maximum Num of inputs)*

#define PNO_SOUNDAGENT_BASE 0x07D0

*sound agent thread base No.*

#define PNO_SOUNDAGENT_MAX 0x07E9

*Maximum Num of sound agent thread.*

#define PNO_PHONEME2PCMABORT 0x07FA

*voice composition interruption thread*

#define PNO_SOUND_DEVCTRL 0x07FB

*device control thread*

#define PNO_SOUND_INPUTCTRL 0x07FC

*input manager thread*

#define PNO_VGUIDE_SNDCTRL_BASIC_F 0x07FD

*playback manager thread(before seat basis playback)*

#define PNO_VGUIDE_SNDCTRL_BASIC_R 0x07FE

*playback manager thread(behind seat basic playback)*

#define PNO_VGUIDE_SNDCTRL_INTRPT 0x07FF

*playback manager thread(interruption voice)*

#define PNO_CWORD83 0x0900

*CWORD83 (Intersystem Communication Message)thread*

#define PNO_DOMAIN_SOUND 0x0770

*Sound thread #057#.*

#define PNO_DOMAIN_BLUETOOTH 0x0771

*BlueTooth thread #057#.*

#define PNO_DOMAIN_CAMERA 0x0772

*Camera thread #057#.*

#define PNO_DOMAIN_DIAG 0x0773

*Diag thread #057#.*

#define PNO_DOMAIN_CWORD57 0x0774

*CWORD57 thread #057#*

#define PNO_DOMAIN_SPEECHREC 0x0775

*SpeechRec thread #057#.*

#define PNO_DOMAIN_USBAUDIO 0x0776

*UsbAudio thread #057#.*

#define PNO_DOMAIN_SENSOR 0x0777

*Sensor thread #057#.*

#define PNO_DOMAIN_CWORD24 0x0778

*CWORD24 thread #057#*

#define PNO_DOMAIN_COMMUNICATION 0x0779

*Communication thread #057#.*

#define PNO_DOMAIN_NAVICORE 0x077A

*navi core thread*

#define PNO_DOMAIN_NAVIPROXY 0x077B

*navi proxy thread*

#define PNO_DOMAIN_ANALYSIS 0x077C

*Analysis thread #057#.*

#define PNO_DOMAIN_STORAGE 0x077D

*Storage thread #057#.*

#define PNO_DOMAIN_VEHICLE 0x077E

*VEHICLE thread #057#.*

#define PNO_DOMAIN_GRAPHICS 0x077F

*Graphics thread #057#.*

#define PNO_DOMAIN_VUP_CTRL 0x0780

*VUP_Ctrl thread #057#.*

#define PNO_DOMAIN_VEHICLEINFO 0x0781

*VehicleInfo thread #057#.*

#define PNO_DOMAIN_RESOURCEMANAGER 0x0782

*ResourceManager thread #057#.*

#define PNO_DOMAIN_DEVICEMANAGER 0x0783

*DeviceManager thread #057#.*

#define PNO_DOMAIN_PERIPHERAL 0x0784

*Peripheral thread #057#.*

#define PNO_DOMAIN_WDT 0x0785

*WDT thread #057#.*

#define PNO_DOMAIN_VUP 0x0786

*VUP thread #057#.*

#define PNO_DOMAIN_TIMERENTRY 0x0787

*TimerEntryDrv #060#.*

#define PNO_DOMAIN_LOADER 0x0788

*Loader thread #GPF_11_008.*

#define PNO_DOMAIN_WLAN 0x078A

*WLAN thread.*

#define PNO_DOMAIN_WLANMANAGER 0x078B

*WLAN Manager thread.*

#define PNO_DOMAIN_WLANADAPTER 0x078C

*WLAN Adapter thread #061.*

#define PNO_DOMAIN_WLANRECEIVER 0x078D

*WLAN Receiver thread #061.*

#define PNO_DOMAIN_HDRADIO 0x0820

*RadioAppthread#066 #GPF_10_022.*

#define PNO_HDRADIO_MAIN 0x0821

*RadioApp Main thread#066 #GPF_10_022.*

#define PNO_HDRADIO__CWORD51_RCVR 0x0822

*RadioApp _CWORD51_Rcvr thread#066 #GPF_10_022.*

#define PNO_DOMAIN_RADIO 0x0830

*radio.exe thread#066 #GPF_10_022*

#define PNO_RADIO_MAIN 0x0831

*radio main thread#067*

#define PNO_RADIO_BACKUP 0x0832

*radio backup manager thread#067*

#define PNO_RADIO_CDDB 0x0833

*radio CDDB manager thread#067*

#define PNO_RADIO_SYS_COMM 0x0834

*radio SYS communication manager thread#067*

#define PNO_DOMAIN_CD 0x0840

*CD middle thread#068.*

#define PNO_CD_MAIN 0x0841

*CD middle main thread#068.*

#define PNO_DOMAIN_CONTENTMGR 0x0850

*contents manager thread#068*

#define PNO_CONTENTMGR_MAIN 0x0851

*contents manager main thread#068*

#define PNO_CONTENTMGR_CDDB 0x0852

*contents manager CDDB thread#068*

#define PNO_CONTENTMGR_CONTENTS_DB 0x0853

*contents manager contentsDB thread#068*

#define PNO_CONTENTMGR_CONTENTS_DB_WORKER 0x0854

*contents manager contentsDB worker thread#068*

#define PNO_DOMAIN_XM 0x0860

*XM middle thread#068.*

#define PNO_XM_MAIN 0x0861

*XM middle main thread#068.*

#define PNO_XM_FILE_ACCESS 0x0862

*XM middle file access thread#068.*

#define PNO_XM_BACKUP 0x0863

*XM middle backup thread#068.*

#define PNO_XM_PACKET_DECODE 0x0864

*XM middle packet decode thread#068.*

#define PNO_XM_CDDB 0x0865

*XM middle CDDB access thread#068.*

#define PNO_CWORD24_JPEG_DECODE 0x01F1

*CWORD24 JPEG decode process thread#068*

#define PNO_USBAUDIO_AVIAGE 0x03E5

*USB middle Aviage reception thread#068.*

#define PNO_USBAUDIO__CWORD69_ 0x03E6

*USB middle CWORD69 reception thread#068.*

#define PNO_USBAUDIO_LIST_ASYNCHRONOUS 0x03E7

*USB middle list asynchronous thread#068.*

#define PNO_USBAUDIO_ARTWORK 0x03E8

*USB middle art worker task thread#068.*

#define PNO_DOMAIN_BTSTACK 0x08B0

*BluetoothStack#071.*

#define PNO_BTSTACK_ADAPTER 0x08B1

*BTStackAdapter_thread#071.*

#define PNO_BTSTACK_OSSTACK 0x08B2

*OS_StackThread#071.*

#define PNO_BTSTACK_UARTWT 0x08B3

*UART_WriteThread#071.*

#define PNO_BTSTACK_UARTRD 0x08B4

*UART_ReadThread#071.*

#define PNO_BTSTACK_OSAT 0x08B5

*OSAPIEx_Timer_Thread#071.*

#define PNO_BTSTACK_OSWAIT 0x08B6

*OS_WaitThread#071.*

#define PNO_DOMAIN_BTSRV 0x08C0

*BTService thread#071.*

#define PNO_BTSRV_SYSMGR 0x08C1

*bt_srv_sysmgr_threadMain thread#071*

#define PNO_BTSRV_CONNECTION 0x08C2

*bt_srv_con_threadMain thread#071*

#define PNO_BTSRV_HTP 0x08C3

*bt_srv_hfp_threadMain thread#071*

#define PNO_BTSRV_AVP 0x08C4

*bt_srv_avp_threadMain thread#071*

#define PNO_BTSRV_STRM 0x08C5

*bt_srv_strm_threadMain thread#071*

#define PNO_BTSRV_PBAP 0x08C6

*bt_srv_pbap_threadEntry thread#071*

#define PNO_BTSRV_ANLZ 0x08C7

*bt_srv_anlz_threadEntry thread#071*

#define PNO_BTSRV_TRNS 0x08C8

*bt_srv_trns_threadEntry thread#071*

#define PNO_BTSRV_EDIT 0x08C9

*bt_srv_edit_threadEntry thread#071*

#define PNO_BTSRV_MAP 0x08CA

*bt_srv_map_threadEntry thread#071*

#define PNO_BTSRV_MAPDATAPARSE 0x08CB

*bt_srv_map_DataParse_threadEntry thread#071*

#define PNO_BTSRV_MULTIENTRY 0x08CC

*bt_srv_multi_entry thread#071*

#define PNO_BTPM_IFCAVP 0x08D1

*btpm_ifc_avp_thread#071*

#define PNO_BTPM_DIST 0x08D2

*btpm_dist_thread#071*

#define PNO_BTPM_IFCHFP 0x08D3

*btpm_ifc_hfp_thread#071*

#define PNO_BTPM_IFCMAP 0x08D4

*tpm_ifc_map_thread#071*

#define PNO_BTPM_MPC 0x08D5

*btpm_mpc_thread#071*

#define PNO_BTPM_IFCPB 0x08D6

*btpm_ifc_pb_thread#071*

#define PNO_BT_APP 0x0800

*BT app thread.*

#define PNO_BT_STACK 0x0801

*BT stack adapter thread.*

#define PNO_BT_WAVE_IN 0x0804

*BT_HFP_WaveIn_thread.*

#define PNO_BT_WAVE_OUT 0x0805

*BT_HFP_WaveOut_thread.*

#define PNO_GGX_DISP_MAIN PNO_DISPMNG_MAIN

*Graphics image main control thread #065#.*

#define PNO_GGX_DISP_SEQ 0x0811

*image sequence control thread #065#*

#define PNO_GGX_DISP_DEV PNO_DISPMNG_SINK_IO

*image reception thread #065#*

#define PNO_GGX_DISP_VCAP 0x0813

*Graphics VideoCapture thread#065#.*

#define PNO_TGWCOM 0x0229

*SPI communication thread.*

#define TSKID_TGWCOM PNO_TGWCOM

*TGW communication.*

#define TSKID_TIM PNO_DEV_SYSCOM_TIMER

*timer thread*

#define MBXID_TGWCOM PNO_TGWCOM

*SPI communication thread.*

#define PNO_MLBSYNC 0x03C4

*MLB_Sync.*

#define PNO_MLBSYNC_INTR 0x03C5

*MLB_Sync_Intr.*

#define SYS_PNO_MIN PNO__CWORD88__CWORD23_

*minimum PNO.(normal process)*

#define SYS_PNO_MAX PNO__CWORD83_

*maximum PNO.(normal process)*

#define PNO_TMN__CWORD69_ PNO_GINI_MAIN

*CWORD69 taskmgr.exe*

#define PNO_TMN__CWORD51__BT 0x0320

*_CWORD51__BT taskmgr.exe*

#define PNO_TMN__CWORD51__DR 0x0321

*_CWORD51__DR taskmgr.exe*

#define PNO_TMN__CWORD51__HD_Radio 0x0322

*_CWORD51__HD-Radio taskmgr.exe*

#define PNO_TMN__CWORD51__CWORD105_ 0x0323

*CWORD51___CWORD105 taskmgr.exe*

#define PNO_LAN__CWORD105__AUDIO 0x03C6

*CWORD105 Audio*

#define PNO_LAN__CWORD105__INTR 0x03D0

*LAN _CWORD105__INTR.*

#define PNO__CWORD65__INTR 0x03D1

*CWORD65 INTR*

#define PNO_CLK_GPS 0x0316

*clock GPS thread*

#define PNO_DEV_EXT_INT 0x0267

*external INT driver*

#define PNO_GSYS_CMSG 0x0311

*message change thread*

#define PNO_DEV_RUNTIMEADD 0x0204

*run time addition thread*

#define PNO_DEV_TIMER_ENTRY 0x02F0

*timer entry driver thread*

#define PNO_GINI_MAIN 0x0300

*G initial.*

---

**Detailed Description**

Header file for PNO definitions.

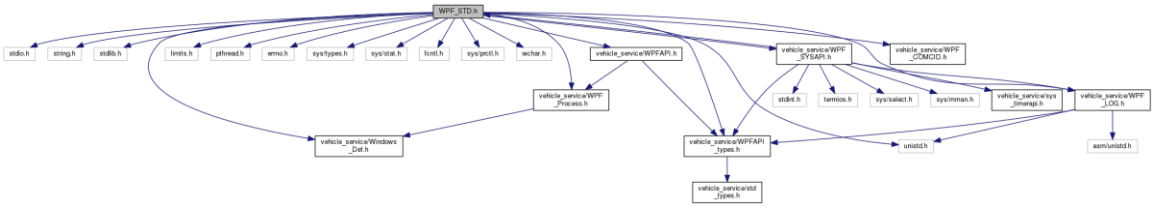
# WPF_STD.h File Reference

Header file for system standard definitions.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <limits.h>
#include <pthread.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/prctl.h>
#include <wchar.h>
#include <vehicle_service/WPFAPI_types.h>
#include <vehicle_service/Windows_Def.h>
#include <vehicle_service/WPF_Process.h>
#include <vehicle_service/WPFAPI.h>
#include <vehicle_service/WPF_SYSAPI.h>
#include <vehicle_service/WPF_COMCID.h>
#include <vehicle_service/WPF_LOG.h>
```

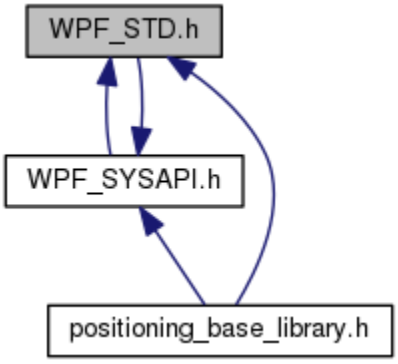
Include dependency graph for WPF_STD.h:

IMAGE




This graph shows which files directly or indirectly include this file:

IMAGE



**Detailed Description**

Header file for system standard definitions.

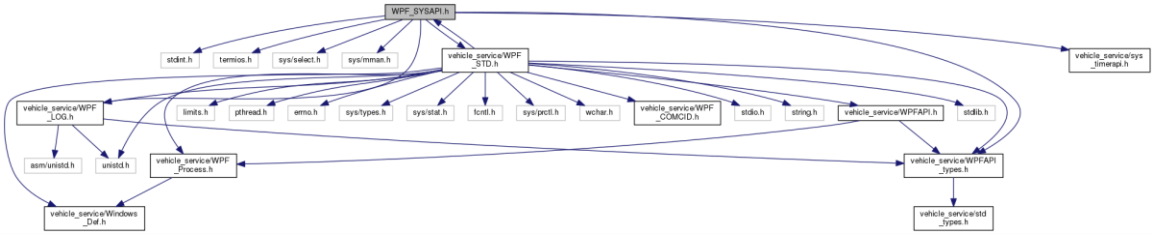
# WPF_SYSAPI.h File Reference

Header file for system API definitions.

#include <stdint.h>
#include <termios.h>
#include <sys/select.h>
#include <sys/mman.h>
#include <vehicle_service/WPF_STD.h>
#include <vehicle_service/WPFAPI_types.h>
#include <vehicle_service/WPF_LOG.h>
#include <vehicle_service/sys_timerapi.h>

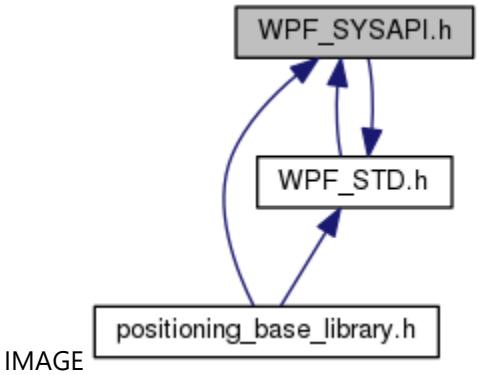
Include dependency graph for WPF_SYSAPI.h:

IMAGE




This graph shows which files directly or indirectly include this file:

IMAGE




**Macros**

#define **CEPC_EM_NOHDD**
#define **PSL_USED**
#define UNIT_TYPE_NONE 0x00000000UL

*type none*

#define UNIT_TYPE_UNKNOWN 0x00000001UL

*type unknown*

#define UNIT_TYPE_CWORD71_CWORD96_CWORD84 0x00000002UL

*type CWORD96 CWORD84*

#define UNIT_TYPE_CWORD71_CWORD95_CWORD101_CWORD84 0x00000004UL

*type CWORD95 /_CWORD101_ CWORD84*

#define UNIT_TYPE_CWORD71_CWORD95_CWORD101_CWORD80 0x00000008UL

*type CWORD95 /_CWORD101_ CWORD80*

#define UNIT_TYPE_CWORD71_CWORD86 0x00000010UL

*type CWORD86*

#define MAX_NAME_LEN 32

*name max length*

#define _CWORD31_NO_SIZE 16

*CWORD31 No size*

#define RESERVE1_SIZE 16

*reserve 1*

#define HELPNET_ID_SIZE 21

*helpnet ID size*

#define RESERVE2_SIZE 11

*reserve 2*

#define DATE_SIZE 32

*make time size*

#define SERIAL_NO_SIZE 3

*serial No size*

#define PRODUCT_NAME_SIZE 1

*product name size*

#define **_pb_memcpy** memcpy
#define **_pb_memset** memset
#define **_pb_Exit**() _pb_Exit_d(__func__, __LINE__)

**Typedefs**

typedef uint32_t UNIT_TYPE

*unit type data type define*

**Enumerations**

enum { _CWORD64_EVENT_MANUALRESET_OFF = ***, _CWORD64_EVENT_MANUALRESET_ON, _CWORD64_EVENT_MANUALRESET_MAX }

**Functions**

EventID _pb_CreateEvent (u_int8 ucManualReset, int32 lInitData, char *cpEventName)
RET_API _pb_WaitEvent (EventID eventID, int32 lWaitMode, int32 lMinVal, int32 lMaxVal, int32 *plEventVal, u_int32 ulMillSecTime)
RET_API _pb_SetEvent (EventID eventID, int32 lSetMode, int32 lval)
RET_API _pb_DeleteEvent (EventID eventID)
RET_API _pb_Setup_CWORD64_API (HANDLE hApp)
VOID _pb_ExitThread (DWORD dwExitCode)
RET_API _pb_CreateMsg (PNO pno)

RET_API _pb_SndMsg (PNO pno, u_int16 size, void *msgbuf, u_int16 mode)
RET_API _pb_SndMsg_Ext (PCSTR name, CID cid, u_int16 size, const void *msgbuf, u_int16 mode)
SemID _pb_CreateSemaphore (char *semName)
RET_API _pb_SemLock (SemID semID)
RET_API _pb_SemUnlock (SemID semID)
RET_API _pb_CreateShareData (char *area_name, u_int32 size, void **mem_ptr)
RET_API _pb_LinkShareData (char *area_name, void **mem_ptr, u_int32 *size)
RET_API _pb_ReqTimerStart (PNO SndPno, u_int16 TimerSeq, u_int8 TimeType, u_int32 TimeOut)
RET_API _pb_TimerStop (PNO SndPno, u_int16 TimerSeq, u_int8 TimeType)
HANDLE _pb_CreateMutex (LPSECURITY_ATTRIBUTES lpMutexAttributes, BOOL bInitialOwner, LPCTSTR lpName)
void _pb_GetDebugMsgMngTbl (void *pBuf, uint8_t *pLen)
void _pb_GetDebugMutexMngTbl (void *pBuf, uint8_t *pLen)
void _pb_GetDebugTimerMngTbl (void *pBuf)
void _pb_GetDebugEventMngTbl (void *pBuf, uint8_t *pLen)
void _pb_GetDebugMemoryMngTbl (void *pBuf)
void _pb_GetDebugOtherMngTbl (void *pBuf)
HANDLE _pb_GetAppHandle (void)
void _pb_SetAppHandle (HANDLE hApp)
void _pb_Teardown_CWORD64_API (void)
void _pb_Exit_d (const char *pFunc, int line)
RET_API _pb_RcvMsg (PNO pno, u_int16 size, void **msgbuf, u_int16 mode)
RET_API _pb_GetZcSndBuf (PNO pno, void **pSndBuf)
RET_API _pb_ZcSndMsg (PNO pno, u_int16 size, u_int16 mode)
PNO _pb_CnvName2Pno (PCSTR name)
PCSTR _pb_CnvPno2Name (PNO pno)
BOOL _pb_GetMsgResource (void)
BOOL _pb_GetMutexResource (void)
BOOL _pb_GetOtherResource (void)
void _pb_ReleaseMsgResource (void)
void _pb_ReleaseMutexResource (void)
void _pb_ReleaseOtherResource (void)

---

**Detailed Description**

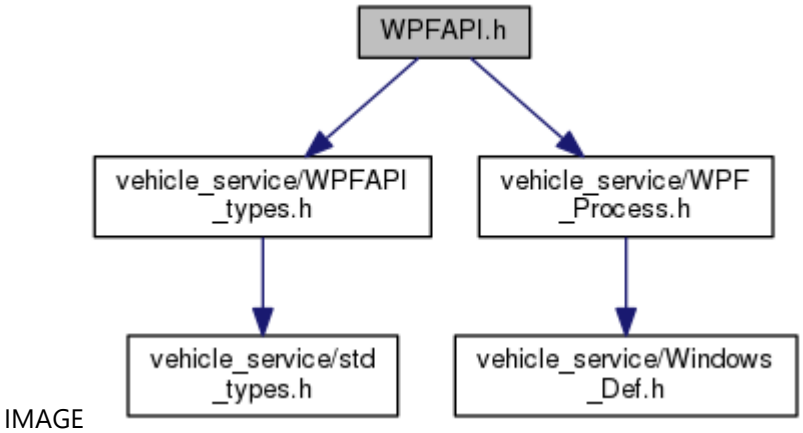
Header file for system API definitions.

# WPFAPI.h File Reference

Header file for message API definitions.

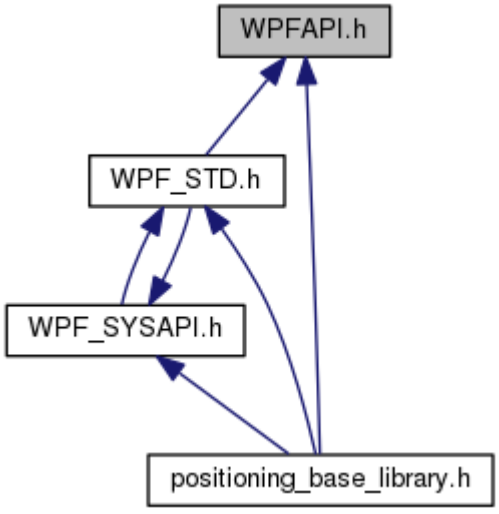
#include <vehicle_service/WPFAPI_types.h>
#include <vehicle_service/WPF_Process.h>

Include dependency graph for WPFAPI.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

**Macros**

#define RET_NORMAL 0

*normal return*

#define RET_ERROR (-1)

*error return*

#define RET_ERRPARAM (-2)

*parameter error*

#define SAPI_EVSET_ABSOLUTE 1

*absolute vale set*

#define SAPI_EVWAIT_VAL 1

*event wait value set*

**Enumerations**

enum _RcvMsgMode { RM_WAIT, RM_CHECKRCV }

---

**Detailed Description**

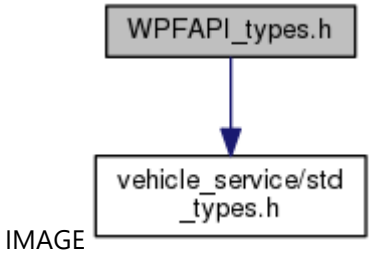
Header file for message API definitions.

## WPFAPI_types.h File Reference

Header file for common type definitions.

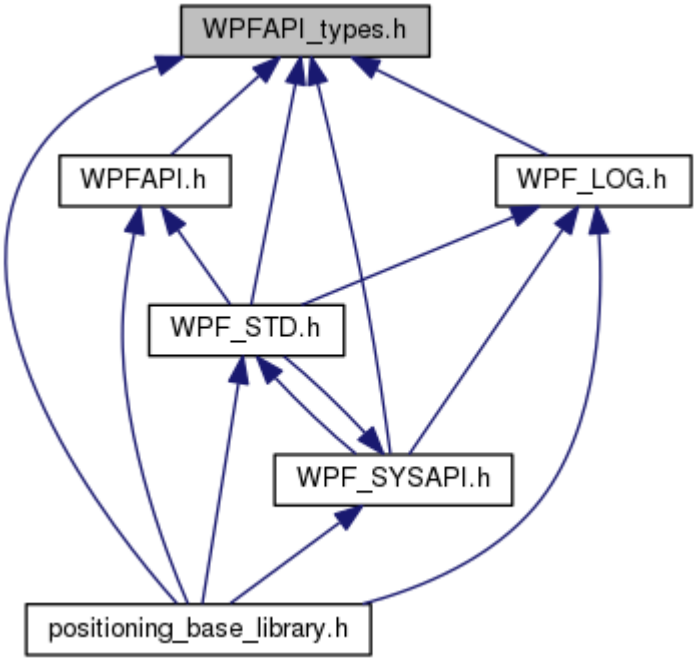
#include <vehicle_service/std_types.h>

Include dependency graph for WPFAPI_types.h:




IMAGE

This graph shows which files directly or indirectly include this file:




IMAGE

### Macros

#define **TRUE** (1)
#define **FALSE** (!TRUE)

### Typedefs

typedef unsigned char **u_char**
typedef unsigned short **u_short**
typedef unsigned int **u_int**
typedef unsigned long **u_long**

typedef u_int32 **signal_code**
typedef u_int32 **process_id**
typedef u_int32 **EventID**
typedef u_int32 **status_code**
typedef u_int32 **path_id**
typedef u_int16 PNO

*PNo.*

typedef int32 RET_API

*_CWORD64_API return code*

typedef u_int32 RET_OS

*OS Error code.*

typedef u_int32 **event_id**
typedef event_id SemID

*semaphore ID*

typedef u_int16 MID

*message ID*

typedef u_int16 CID

*command ID*

typedef u_int8 RID

*resource ID*

typedef u_int16 T_ENDID

*message end ID*

typedef u_int32 ECODE

*CWORD64 error code*

---

**Detailed Description**

Header file for common type definitions.