# TCNOpen TRDP Light

ReleaseV1.4

Generated by Doxygen 1.8.13

# **Contents**

| 1 | The  | TRDP L    | ight Libra   | ry API Sp   | ecifica  | ation |      |      |      |      |      |       |      | 1      |
|---|------|-----------|--------------|-------------|----------|-------|------|------|------|------|------|-------|------|--------|
|   | 1.1  | Genera    | al Informat  | ion         |          |       | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>1  |
|   |      | 1.1.1     | Purpose      |             |          |       | <br> | <br> | <br> | <br> | <br> | <br>  | <br> | <br>1  |
|   |      | 1.1.2     | Scope .      |             |          |       | <br> | <br> | <br> | <br> | <br> | <br>  | <br> | <br>1  |
|   |      | 1.1.3     | Related o    | documents   | <b>3</b> |       | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>1  |
|   |      | 1.1.4     | Abbrevia     | tions and   | Definiti | ons . | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>1  |
|   | 1.2  | Termin    | ology        |             |          |       | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>2  |
|   | 1.3  | Conve     | ntions of th | ne API .    |          |       | <br> | <br> | <br> | <br> | <br> | <br>  | <br> | <br>3  |
| 2 | Data | Structi   | ure Index    |             |          |       |      |      |      |      |      |       |      | 5      |
| _ | 2.1  |           | tructures    |             |          |       |      |      |      |      |      |       |      | 5      |
|   | 2.1  | Dala 3    | dructures    |             |          |       | <br> | <br> | <br> | <br> | <br> | <br>• | <br> | <br>5  |
| 3 | File | Index     |              |             |          |       |      |      |      |      |      |       |      | 7      |
|   | 3.1  | File Lis  | st           |             |          |       | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>7  |
| 4 | Data | a Structi | ure Docur    | nentation   |          |       |      |      |      |      |      |       |      | 9      |
|   | 4.1  | DNS I     | HEADER S     | Struct Refe | erence   |       | <br> | <br> | <br> | <br> | <br> | <br>  | <br> | <br>9  |
|   |      | 4.1.1     |              | Descriptio  |          |       |      |      |      |      |      |       |      | 9      |
|   | 4.2  | GNU_I     | PACKED S     |             |          |       |      |      |      |      |      |       |      | 9      |
|   |      | 4.2.1     | Detailed     | Descriptio  | n        |       | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>14 |
|   |      | 4.2.2     | Field Doo    | cumentation | on       |       | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>15 |
|   |      |           | 4.2.2.1      | confVeh     | Cnt      |       | <br> | <br> | <br> | <br> | <br> | <br>  | <br> | <br>15 |
|   |      |           | 4.2.2.2      | confVehl    | List     |       | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>15 |
|   |      |           | 4.2.2.3      | cstList     |          |       | <br> | <br> | <br> | <br> | <br> |       | <br> | <br>15 |
|   |      |           | 4.2.2.4      | cstUUID     |          |       |      |      |      |      |      |       |      | 16     |
|   |      |           |              |             |          |       |      |      |      |      |      |       |      |        |

ii CONTENTS

| 4.2.2.5  | datasetLength    | 16 |
|----------|------------------|----|
| 4.2.2.6  | deviceName       | 16 |
| 4.2.2.7  | etbld            | 16 |
| 4.2.2.8  | etbTopoCnt       | 16 |
| 4.2.2.9  | inhibit          | 17 |
| 4.2.2.10 | isLead           | 17 |
| 4.2.2.11 | leadDir          | 17 |
| 4.2.2.12 | leadVehOfCst     | 17 |
| 4.2.2.13 | lifesign         | 17 |
| 4.2.2.14 | msgType          | 17 |
| 4.2.2.15 | opCstList        | 18 |
| 4.2.2.16 | opTrnDirState    | 18 |
| 4.2.2.17 | opTrnTopoCnt     | 18 |
| 4.2.2.18 | opVehList        | 18 |
| 4.2.2.19 | ownOpCstNo       | 18 |
| 4.2.2.20 | protocolVersion  | 19 |
| 4.2.2.21 | reserved01 [1/2] | 19 |
| 4.2.2.22 | reserved01 [2/2] | 19 |
| 4.2.2.23 | reserved02 [1/2] | 19 |
| 4.2.2.24 | reserved02 [2/2] | 19 |
| 4.2.2.25 | reserved03       | 19 |
| 4.2.2.26 | reserved04       | 20 |
| 4.2.2.27 | reserved06       | 20 |
| 4.2.2.28 | safetyTrail      | 20 |
| 4.2.2.29 | trnCstNo         | 20 |
| 4.2.2.30 | trnDirState      | 20 |
| 4.2.2.31 | trnld            | 21 |
| 4.2.2.32 | trnNetDir        | 21 |
| 4.2.2.33 | trnOperator      | 21 |
| 4.2.2.34 | trnTopoCnt       | 21 |

CONTENTS

|      |        | 4.2.2.35 trnVehNo                  | . 21 |
|------|--------|------------------------------------|------|
|      |        | 4.2.2.36 vehld                     | . 21 |
|      |        | 4.2.2.37 vehOrient                 | . 22 |
|      |        | 4.2.2.38 version                   | . 22 |
| 4.3  | PD_EL  | E Struct Reference                 | . 22 |
|      | 4.3.1  | Detailed Description               | . 24 |
|      | 4.3.2  | Field Documentation                | . 24 |
|      |        | 4.3.2.1 pFrame                     | . 24 |
| 4.4  | TAU_N  | MARSHALL_INFO_T Struct Reference   | . 24 |
|      | 4.4.1  | Detailed Description               | . 25 |
| 4.5  | TCN_L  | IRI Struct Reference               | . 25 |
|      | 4.5.1  | Detailed Description               | . 25 |
| 4.6  | TRDP_  | CLTR_CST_INFO_T Struct Reference   | . 25 |
|      | 4.6.1  | Detailed Description               | . 26 |
| 4.7  | TRDP_  | COMID_DSID_MAP_T Struct Reference  | . 26 |
|      | 4.7.1  | Detailed Description               | . 26 |
| 4.8  | TRDP_  | CONSIST_INFO_T Struct Reference    | . 26 |
|      | 4.8.1  | Detailed Description               | . 27 |
|      | 4.8.2  | Field Documentation                | . 28 |
|      |        | 4.8.2.1 cstld                      | . 28 |
|      |        | 4.8.2.2 cstOwner                   | . 28 |
| 4.9  | TRDP_  | DATASET Struct Reference           | . 28 |
|      | 4.9.1  | Detailed Description               | . 29 |
| 4.10 | TRDP_  | DATASET_ELEMENT_T Struct Reference | . 29 |
|      | 4.10.1 | Detailed Description               | . 30 |
| 4.11 | TRDP_  | DBG_CONFIG_T Struct Reference      | . 30 |
|      | 4.11.1 | Detailed Description               | . 30 |
| 4.12 | TRDP_  | DNS_REPLY Struct Reference         | . 31 |
|      | 4.12.1 | Detailed Description               | . 31 |
|      | 4.12.2 | Field Documentation                | . 32 |

iv CONTENTS

| 4.12.2.1 tcnUriCnt                           | 32 |
|--|----|
| 4.13 TRDP_DNS_REQUEST Struct Reference       | 32 |
| 4.13.1 Detailed Description                  | 33 |
| 4.13.2 Field Documentation                   | 33 |
| 4.13.2.1 tcnUriCnt                           | 33 |
| 4.14 TRDP_ETB_INFO_T Struct Reference        | 33 |
| 4.14.1 Detailed Description                  | 33 |
| 4.14.2 Field Documentation                   | 33 |
| 4.14.2.1 cnCnt                               | 34 |
| 4.15 TRDP_FUNCTION_INFO_T Struct Reference   | 34 |
| 4.15.1 Detailed Description                  | 34 |
| 4.15.2 Field Documentation                   | 34 |
| 4.15.2.1 cnld                                | 35 |
| 4.15.2.2 cstVehNo                            | 35 |
| 4.15.2.3 etbld                               | 35 |
| 4.15.2.4 fctld                               | 35 |
| 4.16 TRDP_HANDLE Struct Reference            | 35 |
| 4.16.1 Detailed Description                  | 36 |
| 4.17 TRDP_LIST_STATISTICS_T Struct Reference | 36 |
| 4.17.1 Detailed Description                  | 37 |
| 4.18 TRDP_MARSHALL_CONFIG_T Struct Reference | 37 |
| 4.18.1 Detailed Description                  | 37 |
| 4.19 TRDP_MD_CONFIG_T Struct Reference       | 38 |
| 4.19.1 Detailed Description                  | 39 |
| 4.20 TRDP_MD_INFO_T Struct Reference         | 39 |
| 4.20.1 Detailed Description                  | 40 |
| 4.21 TRDP_MD_STATISTICS_T Struct Reference   | 40 |
| 4.21.1 Detailed Description                  | 41 |
| 4.22 TRDP_MEM_CONFIG_T Struct Reference      | 41 |
| 4.22.1 Detailed Description                  | 41 |

CONTENTS

| 4.23 TRDP_MEM_STATISTICS_T Struct Reference | . 42 |
|---|------|
| 4.23.1 Detailed Description                 | . 42 |
| 4.24 TRDP_PD_CONFIG_T Struct Reference      | . 42 |
| 4.24.1 Detailed Description                 | . 43 |
| 4.25 TRDP_PD_INFO_T Struct Reference        | . 43 |
| 4.25.1 Detailed Description                 | . 44 |
| 4.26 TRDP_PD_STATISTICS_T Struct Reference  | . 44 |
| 4.26.1 Detailed Description                 | . 45 |
| 4.27 TRDP_PROCESS_CONFIG_T Struct Reference | . 45 |
| 4.27.1 Detailed Description                 | . 45 |
| 4.28 TRDP_PROP_T Struct Reference           | . 46 |
| 4.28.1 Detailed Description                 | . 46 |
| 4.29 TRDP_PUB_STATISTICS_T Struct Reference | . 46 |
| 4.29.1 Detailed Description                 | . 47 |
| 4.29.2 Field Documentation                  | . 47 |
| 4.29.2.1 destAddr                           | . 47 |
| 4.30 TRDP_RED_STATISTICS_T Struct Reference | . 47 |
| 4.30.1 Detailed Description                 | . 47 |
| 4.31 TRDP_SDT_PAR_T Struct Reference        | . 47 |
| 4.31.1 Detailed Description                 | . 48 |
| 4.32 TRDP_SEND_PARAM_T Struct Reference     | . 48 |
| 4.32.1 Detailed Description                 | . 49 |
| 4.33 TRDP_SEQ_CNT_ENTRY_T Struct Reference  | . 49 |
| 4.33.1 Detailed Description                 | . 49 |
| 4.34 TRDP_SESSION Struct Reference          | . 49 |
| 4.34.1 Detailed Description                 | . 50 |
| 4.35 TRDP_SOCKET_TCP Struct Reference       | . 51 |
| 4.35.1 Detailed Description                 | . 51 |
| 4.36 TRDP_SOCKETS Struct Reference          | . 51 |
| 4.36.1 Detailed Description                 | . 52 |

vi

| 4.36.2 Field Documentation                      | 52 |
|---|----|
| 4.36.2.1 usage                                  | 52 |
| 4.37 TRDP_STATISTICS_REQUEST_T Struct Reference | 52 |
| 4.37.1 Detailed Description                     | 53 |
| 4.38 TRDP_STATISTICS_T Struct Reference         | 53 |
| 4.38.1 Detailed Description                     | 54 |
| 4.39 TRDP_SUBS_STATISTICS_T Struct Reference    | 54 |
| 4.39.1 Detailed Description                     | 55 |
| 4.39.2 Field Documentation                      | 55 |
| 4.39.2.1 filterAddr                             | 55 |
| 4.39.2.2 timeout                                | 55 |
| 4.39.2.3 toBehav                                | 56 |
| 4.40 TRDP_VEHICLE_INFO_T Struct Reference       | 56 |
| 4.40.1 Detailed Description                     | 57 |
| 4.40.2 Field Documentation                      | 57 |
| 4.40.2.1 vehld                                  | 57 |
| 4.41 TRDP_XML_DOC_HANDLE_T Struct Reference     | 57 |
| 4.41.1 Detailed Description                     | 57 |
| 4.42 VOS_SOCK_OPT_T Struct Reference            | 57 |
| 4.42.1 Detailed Description                     | 58 |
| 4.43 VOS_VERSION_T Struct Reference             | 58 |
| 4.43.1 Detailed Description                     | 58 |

CONTENTS vii

| 5 | File | Docum   | entation     |                                 | 59 |
|---|------|---------|--------------|---------------------------------|----|
|   | 5.1  | iec613  | 75-2-3.h Fi  | le Reference                    | 59 |
|   |      | 5.1.1   | Detailed I   | Description                     | 63 |
|   |      | 5.1.2   | Macro De     | finition Documentation          | 64 |
|   |      |         | 5.1.2.1      | ETB_CTRL_COMID                  | 64 |
|   |      |         | 5.1.2.2      | TRDP_ETBCTRL_DSID               | 64 |
|   |      |         | 5.1.2.3      | TRDP_MAX_FILE_NAME_LEN          | 64 |
|   |      |         | 5.1.2.4      | TRDP_MAX_LABEL_LEN              | 64 |
|   |      |         | 5.1.2.5      | TRDP_MAX_MD_DATA_SIZE           | 64 |
|   |      |         | 5.1.2.6      | TRDP_MAX_URI_HOST_LEN           | 65 |
|   |      |         | 5.1.2.7      | TRDP_MAX_URI_LEN                | 65 |
|   |      |         | 5.1.2.8      | TRDP_MAX_URI_USER_LEN           | 65 |
|   |      |         | 5.1.2.9      | TRDP_MD_DEFAULT_REPLY_TIMEOUT   | 65 |
|   |      |         | 5.1.2.10     | TRDP_MD_INFINITE_TIME           | 65 |
|   |      |         | 5.1.2.11     | TRDP_MIN_PD_HEADER_SIZE         | 65 |
|   |      |         | 5.1.2.12     | TRDP_MSG_PD                     | 66 |
|   |      |         | 5.1.2.13     | TRDP_PD_UDP_PORT                | 66 |
|   |      |         | 5.1.2.14     | TRDP_PROCESS_DEFAULT_CYCLE_TIME | 66 |
|   |      |         | 5.1.2.15     | TRDP_USR_URI_SIZE               | 66 |
|   |      |         | 5.1.2.16     | TTDB_NET_DIR_REQ_COMID          | 66 |
|   |      |         | 5.1.2.17     | TTDB_OP_DIR_INFO_COMID          | 66 |
|   |      |         | 5.1.2.18     | TTDB_STAT_CST_REQ_COMID         | 67 |
|   |      |         | 5.1.2.19     | TTDB_TRN_DIR_REQ_COMID          | 67 |
|   | 5.2  | tau_cs  | tinfo.c File | Reference                       | 67 |
|   |      | 5.2.1   | Detailed I   | Description                     | 68 |
|   |      | 5.2.2   | Function     | Documentation                   | 68 |
|   |      |         | 5.2.2.1      | cstInfoGetPropSize()            | 68 |
|   | 5.3  | tau_ctr | l.c File Ref | erence                          | 69 |
|   |      | 5.3.1   | Detailed I   | Description                     | 70 |
|   |      | 5.3.2   | Function     | Documentation                   | 71 |

viii CONTENTS

|     |         | 5.3.2.1      | tau_getEcspStat()        | 71 |
|-----|---------|--------------|--------------------------|----|
|     |         | 5.3.2.2      | tau_initEcspCtrl()       | 71 |
|     |         | 5.3.2.3      | tau_requestEcspConfirm() | 73 |
|     |         | 5.3.2.4      | tau_setEcspCtrl()        | 73 |
|     |         | 5.3.2.5      | tau_terminateEcspCtrl()  | 74 |
| 5.4 | tau_ctr | l.h File Re  | eference                 | 74 |
|     | 5.4.1   | Detailed     | Description              | 76 |
|     | 5.4.2   | Function     | Documentation            | 77 |
|     |         | 5.4.2.1      | tau_getEcspStat()        | 77 |
|     |         | 5.4.2.2      | tau_initEcspCtrl()       | 77 |
|     |         | 5.4.2.3      | tau_requestEcspConfirm() | 78 |
|     |         | 5.4.2.4      | tau_setEcspCtrl()        | 78 |
|     |         | 5.4.2.5      | tau_terminateEcspCtrl()  | 79 |
| 5.5 | tau_ctr | rl_types.h   | File Reference           | 79 |
|     | 5.5.1   | Detailed     | Description              | 81 |
| 5.6 | tau_dn  | ır.c File Re | eference                 | 82 |
|     | 5.6.1   | Detailed     | Description              | 83 |
|     | 5.6.2   | Function     | Documentation            | 84 |
|     |         | 5.6.2.1      | tau_addr2Uri()           | 84 |
|     |         | 5.6.2.2      | tau_deInitDnr()          | 85 |
|     |         | 5.6.2.3      | tau_DNRstatus()          | 85 |
|     |         | 5.6.2.4      | tau_getOwnAddr()         | 85 |
|     |         | 5.6.2.5      | tau_initDnr()            | 86 |
|     |         | 5.6.2.6      | tau_uri2Addr()           | 86 |
| 5.7 | tau_dn  | ır.h File Re | eference                 | 87 |
|     | 5.7.1   | Detailed     | Description              | 89 |
|     | 5.7.2   | Function     | Documentation            | 89 |
|     |         | 5.7.2.1      | tau_addr2Uri()           | 89 |
|     |         | 5.7.2.2      | tau_deInitDnr()          | 90 |
|     |         | 5.7.2.3      | tau_DNRstatus()          | 91 |
|     |         |              |                          |    |

CONTENTS

|      |          | 5.7.2.4      | tau_getOwnAddr()             | <br>91  |
|------|----------|--------------|------------------------------|---------|
|      |          | 5.7.2.5      | tau_initDnr()                | <br>92  |
|      |          | 5.7.2.6      | tau_uri2Addr()               | <br>93  |
| 5.8  | tau_dn   | r_types.h    | File Reference               | <br>93  |
|      | 5.8.1    | Detailed     | Description                  | <br>95  |
| 5.9  | tau_ma   | arshall.c Fi | ille Reference               | <br>96  |
|      | 5.9.1    | Detailed     | Description                  | <br>97  |
|      | 5.9.2    | Function     | Documentation                | <br>97  |
|      |          | 5.9.2.1      | tau_calcDatasetSize()        | <br>97  |
|      |          | 5.9.2.2      | tau_calcDatasetSizeByComId() | <br>98  |
|      |          | 5.9.2.3      | tau_initMarshall()           | <br>99  |
|      |          | 5.9.2.4      | tau_marshall()               | <br>99  |
|      |          | 5.9.2.5      | tau_marshallDs()             | <br>100 |
|      |          | 5.9.2.6      | tau_unmarshall()             | <br>101 |
|      |          | 5.9.2.7      | tau_unmarshallDs()           | <br>101 |
| 5.10 | tau_ma   | arshall.h F  | ile Reference                | <br>102 |
|      | 5.10.1   | Detailed     | Description                  | <br>104 |
|      | 5.10.2   | Function     | Documentation                | <br>104 |
|      |          | 5.10.2.1     | tau_calcDatasetSize()        | <br>105 |
|      |          | 5.10.2.2     | tau_calcDatasetSizeByComId() | <br>106 |
|      |          | 5.10.2.3     | tau_initMarshall()           | <br>107 |
|      |          | 5.10.2.4     | tau_marshall()               | <br>108 |
|      |          | 5.10.2.5     | tau_marshallDs()             | <br>109 |
|      |          | 5.10.2.6     | tau_unmarshall()             | <br>110 |
|      |          | 5.10.2.7     | tau_unmarshallDs()           | <br>111 |
| 5.11 | tau_tti. | c File Refe  | erence                       | <br>112 |
|      | 5.11.1   | Detailed     | Description                  | <br>113 |
|      | 5.11.2   | Macro De     | efinition Documentation      | <br>114 |
|      |          | 5.11.2.1     | TTI_CACHED_CONSISTS          | <br>114 |
|      | 5.11.3   | Function     | Documentation                | <br>114 |

CONTENTS

|               | 5.11.3.1    | tau_deInitTTI()                   | 114 |
|---------------|-------------|-----------------------------------|-----|
|               | 5.11.3.2    | tau_getCstFctCnt()                | 115 |
|               | 5.11.3.3    | tau_getCstFctInfo()               | 115 |
|               | 5.11.3.4    | tau_getCstInfo()                  | 116 |
|               | 5.11.3.5    | tau_getCstVehCnt()                | 116 |
|               | 5.11.3.6    | tau_getOpTrDirectory()            | 117 |
|               | 5.11.3.7    | tau_getOpTrnDirectoryStatusInfo() | 117 |
|               | 5.11.3.8    | tau_getOwnlds()                   | 118 |
|               | 5.11.3.9    | tau_getStaticCstInfo()            | 118 |
|               | 5.11.3.10   | tau_getTrDirectory()              | 119 |
|               | 5.11.3.11   | tau_getTrnCstCnt()                | 119 |
|               | 5.11.3.12   | 2 tau_getTrnVehCnt()              | 119 |
|               | 5.11.3.13   | B tau_getTTI()                    | 120 |
|               | 5.11.3.14   | tau_getVehInfo()                  | 120 |
|               | 5.11.3.15   | tau_getVehOrient()                | 121 |
|               | 5.11.3.16   | 6 tau_initTTlaccess()             | 121 |
| 5.12 tau_tti. | h File Refe | erence                            | 123 |
| 5.12.1        | Detailed I  | Description                       | 126 |
| 5.12.2        | Function    | Documentation                     | 126 |
|               | 5.12.2.1    | tau_deInitTTI()                   | 126 |
|               | 5.12.2.2    | tau_getCstFctCnt()                | 127 |
|               | 5.12.2.3    | tau_getCstFctInfo()               | 127 |
|               | 5.12.2.4    | tau_getCstInfo()                  | 128 |
|               | 5.12.2.5    | tau_getCstVehCnt()                | 128 |
|               | 5.12.2.6    | tau_getOpTrDirectory()            | 129 |
|               | 5.12.2.7    | tau_getOpTrnDirectoryStatusInfo() | 130 |
|               | 5.12.2.8    | tau_getOwnlds()                   | 130 |
|               | 5.12.2.9    | tau_getStaticCstInfo()            | 131 |
|               | 5.12.2.10   | tau_getTrDirectory()              | 131 |
|               | 5.12.2.11   | tau_getTrnCstCnt()                | 132 |
|               |             |                                   |     |

CONTENTS xi

|              | 5.12.2.12 tau_getTrnVehCnt()          |
|--------------|---------------------------------------|
|              | 5.12.2.13 tau_getTTI()                |
|              | 5.12.2.14 tau_getVehInfo()            |
|              | 5.12.2.15 tau_getVehOrient()          |
|              | 5.12.2.16 tau_initTTlaccess()         |
| 5.13 tau_tti | _types.h File Reference               |
| 5.13.1       | Detailed Description                  |
| 5.14 tau_xr  | nl.c File Reference                   |
| 5.14.1       | Detailed Description                  |
| 5.14.2       | Macro Definition Documentation        |
|              | 5.14.2.1 TRDP_SDT_DEFAULT_CMTHR       |
| 5.14.3       | Function Documentation                |
|              | 5.14.3.1 tau_freeTelegrams()          |
|              | 5.14.3.2 tau_freeXmlDatasetConfig()   |
|              | 5.14.3.3 tau_freeXmlDoc()             |
|              | 5.14.3.4 tau_prepareXmlDoc()          |
|              | 5.14.3.5 tau_readXmlDatasetConfig()   |
|              | 5.14.3.6 tau_readXmlDeviceConfig()    |
|              | 5.14.3.7 tau_readXmlInterfaceConfig() |
| 5.15 tau_xr  | nl.h File Reference                   |
| 5.15.1       | Detailed Description                  |
| 5.15.2       | Macro Definition Documentation        |
|              | 5.15.2.1 TRDP_DBG_DEFAULT             |
| 5.15.3       | Enumeration Type Documentation        |
|              | 5.15.3.1 TRDP_EXCHG_OPTION_T          |
| 5.15.4       | Function Documentation                |
|              | 5.15.4.1 tau_freeTelegrams()          |
|              | 5.15.4.2 tau_freeXmlDatasetConfig()   |
|              | 5.15.4.3 tau_freeXmlDoc()             |
|              | 5.15.4.4 tau_prepareXmlDoc()          |

xii CONTENTS

|              | 5.15.4.5 tau_readXmlDatasetConfig()   |
|--------------|---------------------------------------|
|              | 5.15.4.6 tau_readXmlDeviceConfig()    |
|              | 5.15.4.7 tau_readXmlInterfaceConfig() |
| 5.16 trdp_d  | Ilmain.c File Reference               |
| 5.16.1       | Detailed Description                  |
| 5.17 trdp_if | .c File Reference                     |
| 5.17.1       | Detailed Description                  |
| 5.17.2       | Function Documentation                |
|              | 5.17.2.1 tlc_closeSession()           |
|              | 5.17.2.2 tlc_configSession()          |
|              | 5.17.2.3 tlc_getETBTopoCount()        |
|              | 5.17.2.4 tlc_getInterval()            |
|              | 5.17.2.5 tlc_getOpTrainTopoCount()    |
|              | 5.17.2.6 tlc_getOwnlpAddress()        |
|              | 5.17.2.7 tlc_getVersion()             |
|              | 5.17.2.8 tlc_getVersionString()       |
|              | 5.17.2.9 tlc_init()                   |
|              | 5.17.2.10 tlc_openSession()           |
|              | 5.17.2.11 tlc_process()               |
|              | 5.17.2.12 tlc_reinitSession()         |
|              | 5.17.2.13 tlc_setETBTopoCount()       |
|              | 5.17.2.14 tlc_setOpTrainTopoCount()   |
|              | 5.17.2.15 tlc_terminate()             |
|              | 5.17.2.16 tlp_get()                   |
|              | 5.17.2.17 tlp_getRedundant()          |
|              | 5.17.2.18 tlp_publish()               |
|              | 5.17.2.19 tlp_put()                   |
|              | 5.17.2.20 tlp_republish()             |
|              | 5.17.2.21 tlp_request()               |
|              | 5.17.2.22 tlp_resubscribe()           |
|              |                                       |

CONTENTS xiii

|             | 5.17.2.23 tlp_setRedundant()         |
|-------------|--------------------------------------|
|             | 5.17.2.24 tlp_subscribe()            |
|             | 5.17.2.25 tlp_unpublish()            |
|             | 5.17.2.26 tlp_unsubscribe()          |
|             | 5.17.2.27 trdp_isValidSession()      |
|             | 5.17.2.28 trdp_sessionQueue()        |
| 5.18 trdp_i | f.h File Reference                   |
| 5.18.1      | Detailed Description                 |
| 5.18.2      | Prunction Documentation              |
|             | 5.18.2.1 trdp_isValidSession()       |
|             | 5.18.2.2 trdp_sessionQueue()         |
| 5.19 trdp_i | f_light.h File Reference             |
| 5.19.1      | Detailed Description                 |
| 5.19.2      | Prunction Documentation              |
|             | 5.19.2.1 tlc_closeSession()          |
|             | 5.19.2.2 tlc_configSession()         |
|             | 5.19.2.3 tlc_freeBuf()               |
|             | 5.19.2.4 tlc_getETBTopoCount()       |
|             | 5.19.2.5 tlc_getInterval()           |
|             | 5.19.2.6 tlc_getJoinStatistics()     |
|             | 5.19.2.7 tlc_getOpTrainTopoCount()   |
|             | 5.19.2.8 tlc_getOwnlpAddress()       |
|             | 5.19.2.9 tlc_getPubStatistics()      |
|             | 5.19.2.10 tlc_getRedStatistics()     |
|             | 5.19.2.11 tlc_getStatistics()        |
|             | 5.19.2.12 tlc_getSubsStatistics()    |
|             | 5.19.2.13 tlc_getTcpListStatistics() |
|             | 5.19.2.14 tlc_getUdpListStatistics() |
|             | 5.19.2.15 tlc_getVersion()           |
|             | 5.19.2.16 tlc_getVersionString()     |

xiv CONTENTS

|      |        | 5.19.2.17 tlc_init()                | 189 |
|------|--------|-------------------------------------|-----|
|      |        | 5.19.2.18 tlc_openSession()         | 190 |
|      |        | 5.19.2.19 tlc_process()             | 190 |
|      |        | 5.19.2.20 tlc_reinitSession()       | 191 |
|      |        | 5.19.2.21 tlc_resetStatistics()     | 191 |
|      |        | 5.19.2.22 tlc_setETBTopoCount()     | 192 |
|      |        | 5.19.2.23 tlc_setOpTrainTopoCount() | 192 |
|      |        | 5.19.2.24 tlc_terminate()           | 193 |
|      |        | 5.19.2.25 tlm_abortSession()        | 193 |
|      |        | 5.19.2.26 tlm_addListener()         | 194 |
|      |        | 5.19.2.27 tlm_confirm()             | 195 |
|      |        | 5.19.2.28 tlm_delListener()         | 195 |
|      |        | 5.19.2.29 tlm_notify()              | 196 |
|      |        | 5.19.2.30 tlm_readdListener()       | 197 |
|      |        | 5.19.2.31 tlm_reply()               | 197 |
|      |        | 5.19.2.32 tlm_replyQuery()          | 198 |
|      |        | 5.19.2.33 tlm_request()             | 199 |
|      |        | 5.19.2.34 tlp_get()                 | 200 |
|      |        | 5.19.2.35 tlp_getRedundant()        | 201 |
|      |        | 5.19.2.36 tlp_publish()             | 202 |
|      |        | 5.19.2.37 tlp_put()                 | 203 |
|      |        | 5.19.2.38 tlp_republish()           | 203 |
|      |        | 5.19.2.39 tlp_request()             | 204 |
|      |        | 5.19.2.40 tlp_resubscribe()         | 206 |
|      |        | 5.19.2.41 tlp_setRedundant()        | 207 |
|      |        | 5.19.2.42 tlp_subscribe()           | 207 |
|      |        | 5.19.2.43 tlp_unpublish()           | 209 |
|      |        | 5.19.2.44 tlp_unsubscribe()         | 210 |
| 5.20 | trdp_m | dcom.c File Reference               | 210 |
|      | 5.20.1 | Detailed Description                | 212 |

CONTENTS xv

|      | 5.20.2  | Function    | Documentation             | 213 |
|------|---------|-------------|---------------------------|-----|
|      |         | 5.20.2.1    | trdp_mdCall()             | 213 |
|      |         | 5.20.2.2    | trdp_mdCheckListenSocks() | 214 |
|      |         | 5.20.2.3    | trdp_mdCheckPending()     | 214 |
|      |         | 5.20.2.4    | trdp_mdCheckTimeouts()    | 214 |
|      |         | 5.20.2.5    | trdp_mdConfirm()          | 215 |
|      |         | 5.20.2.6    | trdp_mdFreeSession()      | 215 |
|      |         | 5.20.2.7    | trdp_mdGetTCPSocket()     | 216 |
|      |         | 5.20.2.8    | trdp_mdReply()            | 216 |
|      |         | 5.20.2.9    | trdp_mdSend()             | 217 |
| 5.21 | trdp_m  | dcom.h Fi   | le Reference              | 217 |
|      | 5.21.1  | Detailed    | Description               | 219 |
|      | 5.21.2  | Function    | Documentation             | 220 |
|      |         | 5.21.2.1    | trdp_mdCall()             | 220 |
|      |         | 5.21.2.2    | trdp_mdCheckListenSocks() | 221 |
|      |         | 5.21.2.3    | trdp_mdCheckPending()     | 221 |
|      |         | 5.21.2.4    | trdp_mdCheckTimeouts()    | 221 |
|      |         | 5.21.2.5    | trdp_mdConfirm()          | 222 |
|      |         | 5.21.2.6    | trdp_mdFreeSession()      | 222 |
|      |         | 5.21.2.7    | trdp_mdGetTCPSocket()     | 223 |
|      |         | 5.21.2.8    | trdp_mdReply()            | 223 |
|      |         | 5.21.2.9    | trdp_mdSend()             | 224 |
| 5.22 | trdp_pc | dcom.c File | e Reference               | 224 |
|      | 5.22.1  | Detailed    | Description               | 226 |
|      | 5.22.2  | Function    | Documentation             | 227 |
|      |         | 5.22.2.1    | trdp_pdCheck()            | 227 |
|      |         | 5.22.2.2    | trdp_pdCheckListenSocks() | 227 |
|      |         | 5.22.2.3    | trdp_pdCheckPending()     | 228 |
|      |         | 5.22.2.4    | trdp_pdDistribute()       | 228 |
|      |         | 5.22.2.5    | trdp_pdHandleTimeOuts()   | 229 |

xvi CONTENTS

| 5.22.2.6 trdp_pdInit()                | 229 |
|---------------------------------------|-----|
| 5.22.2.7 trdp_pdPut()                 | 230 |
| 5.22.2.8 trdp_pdReceive()             | 231 |
| 5.22.2.9 trdp_pdSend()                | 231 |
| 5.22.2.10 trdp_pdSendQueued()         | 232 |
| 5.22.2.11 trdp_pdUpdate()             | 232 |
| 5.23 trdp_pdcom.h File Reference      | 233 |
| 5.23.1 Detailed Description           | 235 |
| 5.23.2 Function Documentation         | 235 |
| 5.23.2.1 trdp_pdCheck()               | 235 |
| 5.23.2.2 trdp_pdCheckListenSocks()    | 236 |
| 5.23.2.3 trdp_pdCheckPending()        | 236 |
| 5.23.2.4 trdp_pdDistribute()          | 236 |
| 5.23.2.5 trdp_pdHandleTimeOuts()      | 237 |
| 5.23.2.6 trdp_pdInit()                | 238 |
| 5.23.2.7 trdp_pdPut()                 | 239 |
| 5.23.2.8 trdp_pdReceive()             | 239 |
| 5.23.2.9 trdp_pdSend()                | 240 |
| 5.23.2.10 trdp_pdSendQueued()         | 240 |
| 5.23.2.11 trdp_pdUpdate()             | 241 |
| 5.24 trdp_private.h File Reference    | 241 |
| 5.24.1 Detailed Description           | 244 |
| 5.24.2 Enumeration Type Documentation | 245 |
| 5.24.2.1 TRDP_MD_ELE_ST_T             | 245 |
| 5.24.2.2 TRDP_SOCK_TYPE_T             | 245 |
| 5.25 trdp_stats.c File Reference      | 245 |
| 5.25.1 Detailed Description           | 247 |
| 5.25.2 Function Documentation         | 247 |
| 5.25.2.1 tlc_getJoinStatistics()      | 247 |
| 5.25.2.2 tlc_getPubStatistics()       | 248 |

CONTENTS xvii

|      |         | 5.25.2.3     | tlc_getRedStatistic  | s()      | <br> | <br> | <br> | <br> | <br>248 |
|------|---------|--------------|----------------------|----------|------|------|------|------|---------|
|      |         | 5.25.2.4     | tlc_getStatistics()  |          | <br> | <br> | <br> | <br> | <br>249 |
|      |         | 5.25.2.5     | tlc_getSubsStatist   | ics()    | <br> | <br> | <br> | <br> | <br>249 |
|      |         | 5.25.2.6     | tlc_resetStatistics( | )        | <br> | <br> | <br> | <br> | <br>250 |
|      |         | 5.25.2.7     | trdp_initStats()     |          | <br> | <br> | <br> | <br> | <br>250 |
|      |         | 5.25.2.8     | trdp_pdPrepareSta    | ats()    | <br> | <br> | <br> | <br> | <br>251 |
|      |         | 5.25.2.9     | trdp_UpdateStats(    | )        | <br> | <br> | <br> | <br> | <br>251 |
| 5.26 | trdp_st | ats.h File l | Reference            |          | <br> | <br> | <br> | <br> | <br>252 |
|      | 5.26.1  | Detailed     | Description          |          | <br> | <br> | <br> | <br> | <br>253 |
|      | 5.26.2  | Function     | Documentation        |          | <br> | <br> | <br> | <br> | <br>253 |
|      |         | 5.26.2.1     | trdp_initStats()     |          | <br> | <br> | <br> | <br> | <br>253 |
|      |         | 5.26.2.2     | trdp_pdPrepareSta    | ats()    | <br> | <br> | <br> | <br> | <br>254 |
| 5.27 | trdp_ty | pes.h File   | Reference            |          | <br> | <br> | <br> | <br> | <br>254 |
|      | 5.27.1  | Detailed     | Description          |          | <br> | <br> | <br> | <br> | <br>259 |
|      | 5.27.2  | Macro De     | finition Documenta   | tion     | <br> | <br> | <br> | <br> | <br>260 |
|      |         | 5.27.2.1     | TRDP_FLAGS_DE        | EFAULT . | <br> | <br> | <br> | <br> | <br>260 |
|      | 5.27.3  | Typedef I    | Occumentation        |          | <br> | <br> | <br> | <br> | <br>260 |
|      |         | 5.27.3.1     | TRDP_IP_ADDR_        | T        | <br> | <br> | <br> | <br> | <br>260 |
|      |         | 5.27.3.2     | TRDP_MARSHAL         | L_T      | <br> | <br> | <br> | <br> | <br>260 |
|      |         | 5.27.3.3     | TRDP_MD_CALLI        | BACK_T . | <br> | <br> | <br> | <br> | <br>261 |
|      |         | 5.27.3.4     | TRDP_PD_CALLE        | BACK_T . | <br> | <br> | <br> | <br> | <br>261 |
|      |         | 5.27.3.5     | TRDP_PRINT_DB        | G_T      | <br> | <br> | <br> | <br> | <br>261 |
|      |         | 5.27.3.6     | TRDP_TIME_T.         |          | <br> | <br> | <br> | <br> | <br>261 |
|      |         | 5.27.3.7     | TRDP_UNMARSH         | IALL_T . | <br> | <br> | <br> | <br> | <br>262 |
|      | 5.27.4  | Enumera      | ion Type Documen     | tation   | <br> | <br> | <br> | <br> | <br>262 |
|      |         | 5.27.4.1     | TRDP_DATA_TYP        | PE_T     | <br> | <br> | <br> | <br> | <br>262 |
|      |         | 5.27.4.2     | TRDP_ERR_T           |          | <br> | <br> | <br> | <br> | <br>263 |
|      |         | 5.27.4.3     | TRDP_RED_STA         | ΓΕ_T     | <br> | <br> | <br> | <br> | <br>264 |
|      |         | 5.27.4.4     | TRDP_REPLY_ST        | TATUS_T  | <br> | <br> | <br> | <br> | <br>264 |
|      |         | 5.27.4.5     | TRDP_TO_BEHA         | VIOR_T . | <br> | <br> | <br> | <br> | <br>264 |

xviii CONTENTS

| 5.28 trdp_ut | tils.c File Reference                  |
|--------------|--|
| 5.28.1       | Detailed Description                   |
| 5.28.2       | Function Documentation                 |
|              | 5.28.2.1 trdp_checkSequenceCounter()   |
|              | 5.28.2.2 trdp_findMCjoins()            |
|              | 5.28.2.3 trdp_getCurrentMaxSocketCnt() |
|              | 5.28.2.4 trdp_getSeqCnt()              |
|              | 5.28.2.5 trdp_initSockets()            |
|              | 5.28.2.6 trdp_isAddressed()            |
|              | 5.28.2.7 trdp_isInIPrange()            |
|              | 5.28.2.8 trdp_packetSizeMD()           |
|              | 5.28.2.9 trdp_packetSizePD()           |
|              | 5.28.2.10 trdp_queueAppLast()          |
|              | 5.28.2.11 trdp_queueDelElement()       |
|              | 5.28.2.12 trdp_queueFindComId()        |
|              | 5.28.2.13 trdp_queueFindPubAddr()      |
|              | 5.28.2.14 trdp_queueFindSubAddr()      |
|              | 5.28.2.15 trdp_queueInsFirst()         |
|              | 5.28.2.16 trdp_releaseSocket()         |
|              | 5.28.2.17 trdp_requestSocket()         |
|              | 5.28.2.18 trdp_resetSequenceCounter()  |
|              | 5.28.2.19 trdp_validTopoCounters()     |
| 5.29 trdp_ut | tils.h File Reference                  |
| 5.29.1       | Detailed Description                   |
| 5.29.2       | Function Documentation                 |
|              | 5.29.2.1 trdp_checkSequenceCounter()   |
|              | 5.29.2.2 trdp_findMCjoins()            |
|              | 5.29.2.3 trdp_getCurrentMaxSocketCnt() |
|              | 5.29.2.4 trdp_getSeqCnt()              |
|              | 5.29.2.5 trdp_initSockets()            |

CONTENTS xix

|      |         | 5.29.2.6     | trdp_initUncompletedTCP()     | <br>282 |
|------|---------|--------------|-------------------------------|---------|
|      |         | 5.29.2.7     | trdp_isAddressed()            | <br>282 |
|      |         | 5.29.2.8     | trdp_isInIPrange()            | <br>282 |
|      |         | 5.29.2.9     | trdp_packetSizeMD()           | <br>284 |
|      |         | 5.29.2.10    | ) trdp_packetSizePD()         | <br>284 |
|      |         | 5.29.2.11    | I trdp_queueAppLast()         | <br>285 |
|      |         | 5.29.2.12    | 2 trdp_queueDelElement()      | <br>285 |
|      |         | 5.29.2.13    | 3 trdp_queueFindComId()       | <br>285 |
|      |         | 5.29.2.14    | 1 trdp_queueFindPubAddr()     | <br>286 |
|      |         | 5.29.2.15    | 5 trdp_queueFindSubAddr()     | <br>286 |
|      |         | 5.29.2.16    | 6 trdp_queueInsFirst()        | <br>287 |
|      |         | 5.29.2.17    | 7 trdp_releaseSocket()        | <br>287 |
|      |         | 5.29.2.18    | B trdp_requestSocket()        | <br>288 |
|      |         | 5.29.2.19    | O trdp_resetSequenceCounter() | <br>289 |
|      |         | 5.29.2.20    | trdp_validTopoCounters()      | <br>289 |
| 5.30 | trdp_xr | nl.c File Re | eference                      | <br>290 |
|      | 5.30.1  | Detailed I   | Description                   | <br>291 |
|      | 5.30.2  | Function     | Documentation                 | <br>291 |
|      |         | 5.30.2.1     | trdp_XMLClose()               | <br>291 |
|      |         | 5.30.2.2     | trdp_XMLCountStartTag()       | <br>292 |
|      |         | 5.30.2.3     | trdp_XMLEnter()               | <br>292 |
|      |         | 5.30.2.4     | trdp_XMLGetAttribute()        | <br>292 |
|      |         | 5.30.2.5     | trdp_XMLLeave()               | <br>293 |
|      |         | 5.30.2.6     | trdp_XMLOpen()                | <br>293 |
|      |         | 5.30.2.7     | trdp_XMLRewind()              | <br>294 |
|      |         | 5.30.2.8     | trdp_XMLSeekStartTag()        | <br>294 |
|      |         | 5.30.2.9     | trdp_XMLSeekStartTagAny()     | <br>294 |
| 5.31 | trdp_xr | nl.h File Ro | eference                      | <br>295 |
|      | 5.31.1  | Detailed I   | Description                   | <br>296 |
|      | 5.31.2  | Function     | Documentation                 | <br>297 |

CONTENTS

|            | 5.31.2.1     | trdp_XMLClos  | se()        |        | <br> | <br> | <br> | <br> | <br> | . 297 |
|------------|--------------|---------------|-------------|--------|------|------|------|------|------|-------|
|            | 5.31.2.2     | trdp_XMLCou   | ıntStartTa  | g()    | <br> | <br> | <br> | <br> | <br> | . 297 |
|            | 5.31.2.3     | trdp_XMLEnte  | er()        |        | <br> | <br> | <br> | <br> | <br> | . 298 |
|            | 5.31.2.4     | trdp_XMLGet   | Attribute() |        | <br> | <br> | <br> | <br> | <br> | . 298 |
|            | 5.31.2.5     | trdp_XMLLea   | ve()        |        | <br> | <br> | <br> | <br> | <br> | . 298 |
|            | 5.31.2.6     | trdp_XMLOpe   | en()        |        | <br> | <br> | <br> | <br> | <br> | . 299 |
|            | 5.31.2.7     | trdp_XMLRev   | vind()      |        | <br> | <br> | <br> | <br> | <br> | . 299 |
|            | 5.31.2.8     | trdp_XMLSee   | kStartTag   | J()    | <br> | <br> | <br> | <br> | <br> | . 299 |
|            | 5.31.2.9     | trdp_XMLSee   | kStartTag   | JAny() | <br> | <br> | <br> | <br> | <br> | . 300 |
| 5.32 vos_m | em.c File Re | eference      |             |        | <br> | <br> | <br> | <br> | <br> | . 300 |
| 5.32.1     | Detailed D   | escription .  |             |        | <br> | <br> | <br> | <br> | <br> | . 302 |
| 5.32.2     | Function D   | Oocumentation | ı           |        | <br> | <br> | <br> | <br> | <br> | . 302 |
|            | 5.32.2.1     | vos_bsearch(  | )           |        | <br> | <br> | <br> | <br> | <br> | . 302 |
|            | 5.32.2.2     | vos_memAllo   | c()         |        | <br> | <br> | <br> | <br> | <br> | . 303 |
|            | 5.32.2.3     | vos_memCou    | ınt()       |        | <br> | <br> | <br> | <br> | <br> | . 303 |
|            | 5.32.2.4     | vos_memDele   | ete()       |        | <br> | <br> | <br> | <br> | <br> | . 304 |
|            | 5.32.2.5     | vos_memFree   | ∍()         |        | <br> | <br> | <br> | <br> | <br> | . 304 |
|            | 5.32.2.6     | vos_memInit(  | )           |        | <br> | <br> | <br> | <br> | <br> | . 304 |
|            | 5.32.2.7     | vos_qsort() . |             |        | <br> | <br> | <br> | <br> | <br> | . 305 |
|            | 5.32.2.8     | vos_queueCr   | eate() .    |        | <br> | <br> | <br> | <br> | <br> | . 306 |
|            | 5.32.2.9     | vos_queueDe   | estroy() .  |        | <br> | <br> | <br> | <br> | <br> | . 306 |
|            | 5.32.2.10    | vos_queueRe   | ceive() .   |        | <br> | <br> | <br> | <br> | <br> | . 307 |
|            | 5.32.2.11    | vos_queueSe   | nd()        |        | <br> | <br> | <br> | <br> | <br> | . 307 |
|            | 5.32.2.12    | vos_strncat() |             |        | <br> | <br> | <br> | <br> | <br> | . 308 |
|            | 5.32.2.13    | vos_strncpy() |             |        | <br> | <br> | <br> | <br> | <br> | . 308 |
|            | 5.32.2.14    | vos_strnicmp  | ()          |        | <br> | <br> | <br> | <br> | <br> | . 308 |
| 5.33 vos_m | em.h File R  | eference      |             |        | <br> | <br> | <br> | <br> | <br> | . 309 |
| 5.33.1     | Detailed D   | escription .  |             |        | <br> | <br> | <br> | <br> | <br> | . 311 |
| 5.33.2     | Macro Def    | inition Docum | entation    |        | <br> | <br> | <br> | <br> | <br> | . 311 |
|            | 5.33.2.1     | VOS_MEM_E     | BLOCKSIZ    | ZES    | <br> | <br> | <br> | <br> | <br> | . 312 |

CONTENTS xxi

|            | 5.33.2.2 VOS_MEM_PREALLOCATE     |
|------------|----------------------------------|
| 5.33.3     | 3 Function Documentation         |
|            | 5.33.3.1 vos_bsearch()           |
|            | 5.33.3.2 vos_memAlloc()          |
|            | 5.33.3.3 vos_memCount()          |
|            | 5.33.3.4 vos_memDelete()         |
|            | 5.33.3.5 vos_memFree()           |
|            | 5.33.3.6 vos_memInit()           |
|            | 5.33.3.7 vos_qsort()             |
|            | 5.33.3.8 vos_queueCreate()       |
|            | 5.33.3.9 vos_queueDestroy()      |
|            | 5.33.3.10 vos_queueReceive()     |
|            | 5.33.3.11 vos_queueSend()        |
|            | 5.33.3.12 vos_strncat()          |
|            | 5.33.3.13 vos_strncpy()          |
|            | 5.33.3.14 vos_strnicmp()         |
| 5.34 vos_s | hared_mem.h File Reference       |
| 5.34.1     | Detailed Description             |
| 5.34.2     | 2 Function Documentation         |
|            | 5.34.2.1 vos_sharedClose()       |
|            | 5.34.2.2 vos_sharedOpen()        |
| 5.35 vos_s | ock.h File Reference             |
| 5.35.1     | Detailed Description             |
| 5.35.2     | 2 Macro Definition Documentation |
|            | 5.35.2.1 VOS_MAX_SOCKET_CNT      |
|            | 5.35.2.2 VOS_TTL_MULTICAST       |
| 5.35.3     | 3 Function Documentation         |
|            | 5.35.3.1 vos_determineBindAddr() |
|            | 5.35.3.2 vos_dottedIP()          |
|            | 5.35.3.3 vos_getInterfaces()     |

xxii CONTENTS

|      |         | 5.35.3.4    | VO   | os_h  | itonl() |                  |              |                  |      | <br> | <br>327 |
|------|---------|-------------|------|-------|---------|------------------|--------------|------------------|------|------|------|------|------|------|------|------|---------|
|      |         | 5.35.3.5    | VO   | os_h  | itonll( | )                |              |                  |      | <br> | <br>327 |
|      |         | 5.35.3.6    | VO   | os_h  | itons(  | ) .              |              |                  |      | <br> | <br>327 |
|      |         | 5.35.3.7    | VO   | os_ip | oDotte  | ed()             |              |                  |      | <br> | <br>328 |
|      |         | 5.35.3.8    | VO   | os_is | sMulti  | icas             | t() .        |                  |      | <br> | <br>328 |
|      |         | 5.35.3.9    | VO   | os_n  | etlfU   | p() .            |              |                  |      | <br> | <br>329 |
|      |         | 5.35.3.10   | ) vo | os_n  | ıtohl() |                  |              |                  |      | <br> | <br>329 |
|      |         | 5.35.3.11   | l vo | os_n  | itohll( | )                |              |                  |      | <br> | <br>329 |
|      |         | 5.35.3.12   | 2 vo | os_n  | itohs(  | ) .              |              |                  |      | <br> | <br>330 |
|      |         | 5.35.3.13   | 3 vo | os_s  | elect(  | () .             |              |                  |      | <br> | <br>330 |
|      |         | 5.35.3.14   | ł vo | os_s  | ockA    | ccep             | pt()         |                  |      | <br> | <br>330 |
|      |         | 5.35.3.15   | ō vo | os_s  | ockBi   | ind()            | )            |                  |      | <br> | <br>331 |
|      |         | 5.35.3.16   | S vo | os_s  | ockC    | lose             | ∍() .        |                  |      | <br> | <br>331 |
|      |         | 5.35.3.17   | 7 VO | os_s  | ockC    | onn              | ect(         | ) .              |      | <br> | <br>332 |
|      |         | 5.35.3.18   | 3 vo | os_s  | ockG    | etM              | IAC(         | . (              |      | <br> | <br>332 |
|      |         | 5.35.3.19   | ) vo | os_s  | ockIn   | iit()            |              |                  |      | <br> | <br>333 |
|      |         | 5.35.3.20   | ) vo | os_s  | ockJo   | oinN             | <b>/</b> С() |                  |      | <br> | <br>333 |
|      |         | 5.35.3.21   | l vo | os_s  | ockLe   | eave             | еМС          | () .             |      | <br> | <br>334 |
|      |         | 5.35.3.22   | 2 vo | os_s  | ockLi   | ster             | n() .        |                  |      | <br> | <br>334 |
|      |         | 5.35.3.23   | 3 vo | os_s  | ockO    | pen              | ıTCF         | <b>&gt;</b> () . |      | <br> | <br>335 |
|      |         | 5.35.3.24   | l vo | os_s  | ockO    | pen              | ıUDI         | P()              |      | <br> | <br>335 |
|      |         | 5.35.3.25   | 5 vo | os_s  | ockR    | ecei             | iveT         | CP(              | ) .  | <br> | <br>335 |
|      |         | 5.35.3.26   | S vo | os_s  | ockR    | ecei             | iveL         | JDP(             | ()   | <br> | <br>336 |
|      |         | 5.35.3.27   | 7 VC | os_s  | ockS    | enď              | TCF          | <b>?</b> () .    |      | <br> | <br>337 |
|      |         | 5.35.3.28   | 3 vo | os_s  | ockS    | end              | UDF          | ۰() .            |      | <br> | <br>337 |
|      |         | 5.35.3.29   | ) vo | os_s  | ockS    | etM              | ultic        | astlf            | () . | <br> | <br>338 |
|      |         | 5.35.3.30   | ) vo | os_s  | ockS    | etO <sub>l</sub> | ptior        | ns()             |      | <br> | <br>338 |
|      |         | 5.35.3.31   | l vo | os_s  | ockTe   | erm(             | () .         |                  |      | <br> | <br>339 |
| 5.36 | vos_thi | read.h File | Re   | efere | ence    |                  |              |                  |      | <br> | <br>339 |
|      | 5.36.1  | Detailed [  | Des  | scrip | otion   |                  |              |                  |      | <br> | <br> | <br> | <br> | <br> |      | <br> | <br>341 |

CONTENTS xxiii

|      | 5.36.2  | Function Documentation   |          | <br> | <br> | <br> | 342 |
|------|---------|--------------------------|----------|------|------|------|-----|
|      |         | 5.36.2.1 vos_addTime(    | )        | <br> | <br> | <br> | 342 |
|      |         | 5.36.2.2 vos_clearTime   | ()       | <br> | <br> | <br> | 342 |
|      |         | 5.36.2.3 vos_cmpTime(    | <u> </u> | <br> | <br> | <br> | 342 |
|      |         | 5.36.2.4 vos_cyclicThre  | ead()    | <br> | <br> | <br> | 344 |
|      |         | 5.36.2.5 vos_divTime()   |          | <br> | <br> | <br> | 344 |
|      |         | 5.36.2.6 vos_getTime()   |          | <br> | <br> | <br> | 345 |
|      |         | 5.36.2.7 vos_getTimeS    | tamp()   | <br> | <br> | <br> | 345 |
|      |         | 5.36.2.8 vos_getUuid()   |          | <br> | <br> | <br> | 345 |
|      |         | 5.36.2.9 vos_mulTime()   |          | <br> | <br> | <br> | 345 |
|      |         | 5.36.2.10 vos_mutexCre   | ate()    | <br> | <br> | <br> | 346 |
|      |         | 5.36.2.11 vos_mutexDel   | ete()    | <br> | <br> | <br> | 346 |
|      |         | 5.36.2.12 vos_mutexLoc   | k()      | <br> | <br> | <br> | 347 |
|      |         | 5.36.2.13 vos_mutexTryl  | _ock()   | <br> | <br> | <br> | 347 |
|      |         | 5.36.2.14 vos_mutexUnle  | ock()    | <br> | <br> | <br> | 347 |
|      |         | 5.36.2.15 vos_semaCrea   | ate()    | <br> | <br> | <br> | 348 |
|      |         | 5.36.2.16 vos_semaDele   | ete()    | <br> | <br> | <br> | 348 |
|      |         | 5.36.2.17 vos_semaGive   | e()      | <br> | <br> | <br> | 348 |
|      |         | 5.36.2.18 vos_semaTake   | e()      | <br> | <br> | <br> | 349 |
|      |         | 5.36.2.19 vos_subTime(   | )        | <br> | <br> | <br> | 349 |
|      |         | 5.36.2.20 vos_threadCre  | eate()   | <br> | <br> | <br> | 350 |
|      |         | 5.36.2.21 vos_threadDel  | ay()     | <br> | <br> | <br> | 350 |
|      |         | 5.36.2.22 vos_threadInit | ()       | <br> | <br> | <br> | 351 |
|      |         | 5.36.2.23 vos_threadIsA  | ctive()  | <br> | <br> | <br> | 351 |
|      |         | 5.36.2.24 vos_threadSel  | f()      | <br> | <br> | <br> | 351 |
|      |         | 5.36.2.25 vos_threadTer  | m()      | <br> | <br> | <br> | 352 |
|      |         | 5.36.2.26 vos_threadTer  | minate() | <br> | <br> | <br> | 352 |
| 5.37 | vos_typ | es.h File Reference      |          | <br> | <br> | <br> | 352 |
|      | 5.37.1  | Detailed Description .   |          | <br> | <br> | <br> | 354 |
|      | 5.37.2  | Typedef Documentation    |          | <br> | <br> | <br> | 355 |

xxiv CONTENTS

|             | 5.37.2.1      | VOS_PRINT_DBG_T         | . 355 |
|-------------|---------------|-------------------------|-------|
|             | 5.37.2.2      | VOS_TIMEVAL_T           | . 355 |
| 5.37.3      | Enumera       | tion Type Documentation | . 355 |
|             | 5.37.3.1      | VOS_ERR_T               | . 355 |
|             | 5.37.3.2      | VOS_LOG_T               | . 356 |
| 5.38 vos_ut | ils.c File Re | eference                | . 356 |
| 5.38.1      | Detailed I    | Description             | . 357 |
| 5.38.2      | Function      | Documentation           | . 358 |
|             | 5.38.2.1      | vos_crc32()             | . 358 |
|             | 5.38.2.2      | vos_getErrorString()    | . 359 |
|             | 5.38.2.3      | vos_getVersion()        | . 359 |
|             | 5.38.2.4      | vos_getVersionString()  | . 359 |
|             | 5.38.2.5      | vos_init()              | . 360 |
|             | 5.38.2.6      | vos_sc32()              | . 360 |
|             | 5.38.2.7      | vos_terminate()         | . 360 |
| 5.39 vos_ut | ils.h File R  | eference                | . 361 |
| 5.39.1      | Detailed I    | Description             | . 362 |
| 5.39.2      | Macro De      | efinition Documentation | . 363 |
|             | 5.39.2.1      | INITFCS                 | . 363 |
|             | 5.39.2.2      | VOS_MAX_ERR_STR_SIZE    | . 363 |
|             | 5.39.2.3      | VOS_MAX_FRMT_SIZE       | . 363 |
|             | 5.39.2.4      | VOS_MAX_PRNT_STR_SIZE   | . 363 |
| 5.39.3      | Function      | Documentation           | . 363 |
|             | 5.39.3.1      | vos_crc32()             | . 363 |
|             | 5.39.3.2      | vos_getErrorString()    | . 364 |
|             | 5.39.3.3      | vos_getVersion()        | . 364 |
|             | 5.39.3.4      | vos_getVersionString()  | . 365 |
|             | 5.39.3.5      | vos_init()              | . 365 |
|             | 5.39.3.6      | vos_sc32()              | . 366 |
|             | 5.39.3.7      | vos_terminate()         | . 367 |
| Index       |               |                         | 369   |

## **Chapter 1**

# The TRDP Light Library API Specification



#### 1.1 General Information

#### 1.1.1 Purpose

The TRDP protocol has been defined as the standard communication protocol in IP-enabled trains. It allows communication via process data (periodically transmitted data using UDP/IP) and message data (client - server messaging using UDP/IP or TCP/IP) This document describes the light API of the TRDP Library.

#### 1.1.2 Scope

The intended audience of this document is the developers and project members of the TRDP project. TRDP Client Applications are programs using the TRDP protocol library to access the services of TRDP. Programmers developing such applications are the main target audience for this documentation.

#### 1.1.3 Related documents

TCN-TRDP2-D-BOM-004-01 IEC61375-2-3\_CD\_ANNEXA Protocol definition of the TRDP standard

### 1.1.4 Abbreviations and Definitions

- -API Application Programming Interface -ECN Ethernet Consist Network -TRDP Train Real-time Data Protocol
- -TCMS Train Control Management System

### 1.2 Terminology

The API documented here is mainly concerned with three bodies of code:

- TRDP Client Applications (or 'client applications' for short): These are programs using the API to access the services of TRDP. Programmers developing such applications are the main target audience for this documentation.
- TRDP Light Implementations (or just 'TRDP implementation'): These are libraries realising the API as documented here. Programmers developing such implementations will find useful definitions about syntax and semantics of the API wihtin this documentation.
- VOS Subsystem (Virtual Operating System): An OS and hardware abstraction layer which offers memory, networking, threading, queues and debug functions. The VOS API is documented here.

The following diagram shows how these pieces of software are interrelated.

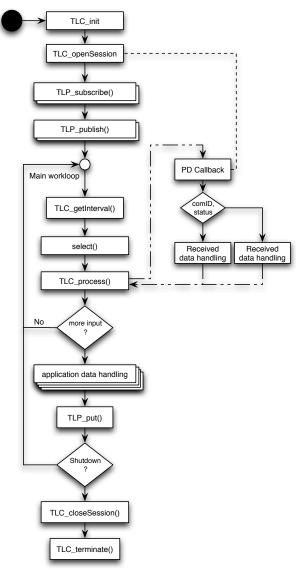


Figure 1.1 Sample client workflow

1.3 Conventions of the API

#### 1.3 Conventions of the API

The API comprises a set of C header files that can also be used from client applications written in C++. These header files are contained in a directory named trdp/api and a subdirectory called trdp/vos/api with declarations not topical to TRDP but needed by the stack. Client applications shall include these header files like:

```
#include "trdp_if_light.h"
```

and, if VOS functions are needed, also the corresponding headers:

```
#include "vos_thread.h"
```

for example.

The subdirectory trdp/doc contains files needed for the API documentation.

Generally client application source code including API headers will only compile if the parent directory of the trdp directory is part of the include path of the used compiler. No other subdirectories of the API should be added to the compiler's include path.

The client API doesn't support a "catch-all" header file that includes all declarations in one step; rather the client application has to include individual headers for each feature set it wants to use.

# **Chapter 2**

# **Data Structure Index**

### 2.1 Data Structures

Here are the data structures with brief descriptions:

| DNS_HEADER  |    |
|---|----|
| DNS header structure                                      | Ş  |
| GNU_PACKED  |    |
| Types for ETB control                                     | 9  |
| PD_ELE  |    |
| Queue element for PD packets to send or receive           | 22 |
| TAU_MARSHALL_INFO_T                                       |    |
| Marshalling info, used to and from wire                   | 24 |
| TCN_URI   |    |
| TCN-DNS simplified header structures                      | 25 |
| TRDP_CLTR_CST_INFO_T                                      |    |
| Closed train consists information                         | 25 |
| TRDP_COMID_DSID_MAP_T                                     |    |
| Comld - data set mapping element definition               | 26 |
| TRDP_CONSIST_INFO_T                                       |    |
| Consist information structure                             | 26 |
| TRDP_DATASET  |    |
| Dataset definition  | 28 |
| TRDP_DATASET_ELEMENT_T                                    |    |
| Dataset element definition                                | 29 |
| TRDP DBG CONFIG T   |    |
| Control for debug output device/file on application level | 30 |
| TRDP_DNS_REPLY  |    |
| TCN-DNS Reply telegram TCN DNS REP DS                     | 31 |
| TRDP DNS REQUEST  |    |
| TCN-DNS Request telegram TCN DNS REQ DS                   | 32 |
| TRDP ETB INFO T   |    |
| Types for train configuration information                 | 33 |
| TRDP FUNCTION INFO T                                      |    |
| Function/device information structure                     | 34 |
| TRDP HANDLE   |    |
| Hidden handle definition, used as unique addressing item  | 35 |
| TRDP LIST STATISTICS T                                    |    |
| Information about a particular MD listener                | 36 |
| TRDP_MARSHALL_CONFIG_T                                    |    |
| Marshaling/unmarshalling configuration                    | 37 |

6 Data Structure Index

| TRDP_MD_CONFIG_T   |     |
|--|-----|
| Default MD configuration   | 38  |
| TRDP_MD_INFO_T   | 00  |
| Message data info from received telegram; allows the application to generate responses | 39  |
| TRDP_MD_STATISTICS_T Structure containing all general MD statistics information        | 40  |
| TRDP MEM CONFIG T  | 40  |
| Enumeration type for memory pre-fragmentation, reuse of VOS definition                 | 41  |
| TRDP_MEM_STATISTICS_T  |     |
| Structure containing all general memory statistics information                         | 42  |
| TRDP PD CONFIG T   |     |
| Default PD configuration   | 42  |
| TRDP_PD_INFO_T   |     |
| Process data info from received telegram; allows the application to generate responses | 43  |
| TRDP_PD_STATISTICS_T   |     |
| Structure containing all general PD statistics information                             | 44  |
| TRDP_PROCESS_CONFIG_T  |     |
| Various flags/general TRDP options for library initialization                          | 45  |
| TRDP_PROP_T  | 4.0 |
| Application defined properties   | 46  |
| TRDP_PUB_STATISTICS_T  | 46  |
| Table containing particular PD publishing information                                  | 46  |
| A table containing PD redundant group information                                      | 47  |
| TRDP SDT PAR T   | 47  |
| Types to read out the XML configuration  | 47  |
| TRDP SEND PARAM T  |     |
| Quality/type of service and time to live   | 48  |
| TRDP SEQ CNT ENTRY T   |     |
| Tuples of last received sequence counter per comId                                     | 49  |
| TRDP_SESSION   |     |
| Session/application variables store  | 49  |
| TRDP_SOCKET_TCP  |     |
| TCP parameters   | 51  |
| TRDP_SOCKETS   |     |
| Socket item  | 51  |
| TRDP_STATISTICS_REQUEST_T  |     |
| TRDP statistics type definitions   | 52  |
| TRDP_STATISTICS_T  | EO  |
| Structure containing all general memory, PD and MD statistics information              | 53  |
| Table containing particular PD subscription information                                | 54  |
| TRDP VEHICLE INFO T  | J-  |
| Vehicle information structure  | 56  |
| TRDP XML DOC HANDLE T  |     |
| Parsed XML document handle   | 57  |
| VOS_SOCK_OPT_T   |     |
| Common socket options  | 57  |
| VOS_VERSION_T  |     |
| Version information  | 58  |

# **Chapter 3**

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

| iec61375-2-3.h                                  |
|---|
| All definitions from IEC 61375-2-3              |
| tau_cstinfo.c                                   |
| Functions for consist information access        |
| tau_ctrl.c                                      |
| Functions for train switch control              |
| tau_ctrl.h                                      |
| TRDP utility interface definitions              |
| tau_ctrl_types.h                                |
| TRDP utility interface definitions              |
| tau_dnr.c                                       |
| Functions for domain name resolution            |
| tau_dnr.h                                       |
| TRDP utility interface definitions              |
| tau_dnr_types.h                                 |
| TRDP utility interface definitions              |
| tau_marshall.c                                  |
| Marshalling functions for TRDP                  |
| tau_marshall.h                                  |
| TRDP utility interface definitions              |
| tau_tti.c                                       |
| Functions for train topology information access |
| tau_tti.h                                       |
| TRDP utility interface definitions              |
| tau_tti_types.h                                 |
| TRDP utility interface definitions              |
| tau_xml.c                                       |
| Functions for XML file parsing                  |
| tau_xml.h                                       |
| TRDP utility interface definitions              |
| trdp_dllmain.c                                  |
| Windows DLL main function                       |
| trdp_if.c                                       |
| Functions for ECN communication                 |
| trdp_if.h                                       |
| Typodofo for TDDD communication                 |

8 File Index

| trdp_if_light.h                                 |    |
|---|----|
| TRDP Light interface functions (API)            | 75 |
| trdp_mdcom.c                                    |    |
| Functions for MD communication                  | 10 |
| trdp_mdcom.h                                    |    |
| Functions for MD communication                  | 17 |
| trdp_pdcom.c                                    |    |
| Functions for PD communication                  | 24 |
| trdp_pdcom.h                                    |    |
| Functions for PD communication                  | 33 |
| trdp_private.h  Typedefs for TRDP communication | 44 |
| trdp stats.c                                    | +1 |
| Statistics functions for TRDP communication     | 15 |
| trdp_stats.h                                    | -5 |
|   | 52 |
| trdp types.h                                    | _  |
| Typedefs for TRDP communication                 | 54 |
| trdp_utils.c                                    |    |
| Helper functions for TRDP communication         | 35 |
| trdp_utils.h                                    |    |
| Common utilities for TRDP communication         | 76 |
| trdp_xml.c                                      |    |
| Simple XML parser                               | 90 |
| trdp_xml.h                                      |    |
| Simple XML parser                               | 95 |
| vos_mem.c                                       |    |
| ,   | 00 |
| vos_mem.h                                       | ~~ |
| Memory and queue functions for OS abstraction   | )9 |
|   | 19 |
| vos_sock.h                                      | 19 |
|   | 22 |
| vos_thread.h                                    |    |
|   | 39 |
| vos types.h                                     | -  |
| <del></del>                                     | 52 |
| vos_utils.c                                     |    |
| Common functions for VOS                        | 56 |
| vos_utils.h                                     |    |
| Typedefs for OS abstraction                     | 31 |

## **Chapter 4**

## **Data Structure Documentation**

### 4.1 DNS\_HEADER Struct Reference

DNS header structure.

#### 4.1.1 Detailed Description

DNS header structure.

The documentation for this struct was generated from the following file:

• tau\_dnr.c

### 4.2 GNU\_PACKED Struct Reference

Types for ETB control.

```
#include <trdp_private.h>
```

#### **Data Fields**

• UINT8 trnVehNo

vehicle sequence number within the train with vehicle 01 being the first vehicle in ETB reference direction 1 as defined in IEC61375-2-5 value range: 0..63 a value of 0 indicates that this vehicle has been inserted by correction

ANTIVALENT8 isLead

vehicle is leading

UINT8 leadDir

vehicle leading direction 0 = not relevant 1 = leading direction 1 2 = leading direction 2

UINT8 vehOrient

 $vehicle \ orientation \ 0 = not \ known \ (corrected \ vehicle) \ 1 = same \ as \ operational \ train \ direction \ 2 = inverse \ to \ operational \ train \ direction$ 

TRDP\_SHORT\_VERSION\_T version

telegram version information, main\_version = 1, sub\_version = 0

• UINT16 reserved01

reserved (=0)

UINT8 trnCstNo

own TCN consist number (= 1..32)

• UINT8 reserved02

reserved (=0)

UINT8 ownOpCstNo

own operational address (= 1..32) = 0 if unknown (e.g.

• UINT8 reserved03

reserved (=0)

UINT32 cstTopoCount

Consist topology counter.

UINT32 trnTopoCount

Train directory topology counter.

UINT32 opTrnTopoCount

Operational Train topology counter.

ANTIVALENT8 wasLead

consist was leading, '01'B = false, '10'B = true

ANTIVALENT8 regLead

leading request, '01'B = false, '10'B = true

UINT8 reqLeadDir

(request) leading direction, '01'B = consist direction 1, '10'B = consist direction 2

ANTIVALENT8 accLead

accept remote leading request, '01'B = false/not accepted, '10'B = true/accepted

ANTIVALENT8 clearConfComp

clear confirmed composition, '01'B = false, '10'B = true

ANTIVALENT8 corrRequest

request confirmation, '01'B = false, '10'B = true

ANTIVALENT8 corrInfoSet

correction info set, '01'B = false, '10'B = true

ANTIVALENT8 compStored

corrected composition stored, '01'B = false, '10'B = true

ANTIVALENT8 sleepRequest

request sleep mode, '01'B = false, '10'B = true

UINT8 leadVehOfCst

position of leading vehicle in consist, 0..31 (1: first vehicle in consist in Direction 1, 2: second vehicle, etc.)

UINT8 reserved04

reserved (=0)

• UINT16 reserved05

reserved (=0)

• UINT8 reserved06

reserved (=0)

UINT8 confVehCnt

number of confirmed vehicles in train (1..63)

TRDP\_CONF\_VEHICLE\_T confVehList [TRDP\_MAX\_VEH\_CNT]

dynamic ordered list of confirmed vehicles in train, starting with vehicle at train head, see sub-clause 5.3.3.2.6

TRDP\_ETB\_CTRL\_VDP\_T safetyTrail

ETBCTRL-VDP trailer, completely set to 0 == not used.

UINT8 reserved01

reserved (=0)

• TRDP\_NET\_LABEL\_T deviceName

function device of ECSC which sends the telegram

UINT8 inhibit

inauguration inhibit 0 = no inhibit request 1 = inhibit request

UINT8 leadingReq

leading request 0 = no leading request 1 = leading request

· UINT8 leadingDir

leading direction 0 = no leading request 1 = leading request direction 1 2 = leading request direction 2

UINT8 sleepReq

sleep request 0 = no sleep request 1 = sleep request

· UINT16 lifesign

wrap-around counter, incremented with each produced datagram.

UINT8 ecspState

ECSP state indication 0 = ECSP not operational(initial value) 1 = ECSP in operation.

UINT8 etbInhibit

inauguration inhibit indication 0 = n/a (default) 1 = inhibit not requested on ETB 2 = inhibit set on local ETBN 3 = inhibit set on remote ETBN 4 = inhibit set on local and remote ETBN

· UINT8 etbLength

indicates train lengthening in case train inauguration is inhibit 0 = no lengthening (default) 1 = lengthening detected

UINT8 etbShort

indicates train shortening in case train inauguration is inhibit 0 = no shortening (default) 1 = shortening detected

• UINT16 reserved02

reserved (=0)

UINT8 etbLeadState

indication of local consist leadership 5 = consist not leading (initial value) 6 = consist is leading requesting 9 = consist is leading 10 = leading conflict other values are not allowed

UINT8 etbLeadDir

direction of the leading end car in the local consist 0 = unknown (default) 1 = TCN direction 1 2 = TCN direction 2 other values are not allowed

UINT8 ttdbSrvState

TTDB server state indication 0 = n/a (initial value) 1 = Leader (default) 2 = Follower 3 = Error.

UINT8 dnsSrvState

DNS server state indication 0 = n/a (initial value) 1 = Leader (default) 2 = Follower 3 = Error.

UINT8 trnDirState

train directory state 1 = UNCONFIRMED 2 = CONFIRMED other values are not allowed

UINT8 opTrnDirState

train directory state 1 = INVALID 2 = VALID 4 = SHARED other values are not allowed

UINT8 sleepCtrlState

sleep control state (option) 0 = option not available 1 = RegularOperation 2 = WaitForSleepMode 3 = PrepareFor SleepMode

UINT8 sleepReqCnt

number of sleep requests (option) value range: 0..63, not used = 0

UINT32 opTrnTopoCnt

operational train topology counter

UINT8 command

confirmation order 1 = confirmation/correction request 2 = un-confirmation request

UINT16 confVehCnt

number of confirmed vehicles in the train (1..63).

TRDP\_OP\_VEHICLE\_T confVehList [TRDP\_MAX\_VEH\_CNT]

ordered list of confirmed vehicles in the train, starting with vehicle at train head, see chapter 5.3.3.2.10.

UINT8 status

status of storing correction info 0 = correctly stored 1 = not stored

UINT32 reqSafetyCode

SC-32 value of the request message.

UINT8 byPassCtrl

ETBN bypass control 0 = no action (keep old state) 1 = no bypass 2 = activate bypass.

UINT8 txCtrl

ETBN transmission control 0 = no action (keep old state) 1 = activate sending on ETB (default) 2 = stop sending on ETB.

UINT8 slCtrl

sleep mode control (option) 0 = no action (keep old state) 1 = deactivate sleep mode 2 = activate sleep mode (line activity sensing)

UINT8 etbnState

state indication of the (active) ETBN 0 = ETBN not operational(initial value) 1 = ETBN in operation

· UINT8 etbnlnaugState

ETBN inauguration state as defined in IEC61375-2-5 0 = init 1 = not inaugurated 2 = inaugurated 3 = ready for inauguration.

· UINT8 etbnPosition

position of the ETBN 0 = unknown (default) 1 = single node 2 = middle node 3 = end node TCN direction 1 4 = end node TCN direction 2

UINT8 etbnRole

ETBN node role as defined in IEC61375-2-5 0 = undefined 1 = master (redundancy leader) 2 = backup (redundancy follower) 3 = not redundant.

· BITSET8 etbLineState

indication of ETB line status (FALSE == not trusted, TRUE == trusted) bit0 = line A ETBN direction 1 bit1 = line B ETBN direction 1 bit2 = line C ETBN direction 1 bit3 = line D ETBN direction 1 bit4 = line A ETBN direction 2 bit5 = line B ETBN direction 2 bit6 = line C ETBN direction 2 bit7 = line D ETBN direction 2

UINT8 byPassState

state of bypass function 0 = bypass disabled 1 = bypass enabled

UINT8 slState

sleep mode state (option) 0 = no sleep mode 1 = sleep mode active (line activity sensing)

UINT32 etbTopoCnt

ETB topography counter.

TRDP TRAIN NET DIR T trnNetDir

dynamic train info

• UINT8 ver

Version - incremented for incompatible changes.

UINT8 rel

Release - incremented for compatible changes.

• UINT32 reserved01

reserved (=0)

TRDP\_SHORT\_VERSION\_T userDataVersion

version of the vital ETBCTRL telegram mainVersion = 1, subVersion = 0

UINT32 safeSegCount

safe sequence counter, as defined in B.9

UINT32 safetyCode

checksum, as defined in B.9

TRDP\_UUID\_T cstUUID

UUID of the consist, provided by ETBN (TrainNetworkDirectory) Reference to static consist attributes 0 if not available (e.g.

UINT32 cstTopoCnt

consist topology counter provided with the CSTINFO 0 if no CSTINFO available

UINT8 cstOrient

consist orientation '01'B = same as train direction '10'B = inverse to train direction

UINT8 cstCnt

number of consists in train; range: 1..63

• TRDP CONSIST T cstList [TRDP MAX CST CNT]

consist list.

UINT32 trnTopoCnt

trnTopoCnt value ctrlType == 0: actual value ctrlType == 1: set to 0

UINT8 etbld

identification of the ETB the TTDB is computed for bit0: ETB0 (operational network) bit1: ETB1 (multimedia network) bit2: ETB2 (other network) bit3: ETB3 (other network)

TRDP\_NET\_LABEL\_T vehId

Unique vehicle identifier, application defined (e.g.

UINT8 opVehNo

operational vehicle sequence number in train value range 1..63

UINT8 opCstNo

operational consist number in train (1..63)

UINT8 opCstOrient

consist orientation '00'B = not known (corrected vehicle) '01'B = same as operational train direction '10'B = inverse to operational train direction

• TRDP NET LABEL T trnld

train identifier, application defined (e.g.

TRDP NET LABEL T trnOperator

train operator, e.g.

UINT32 crc

sc-32 computed over record (seed value: 'FFFFFFFH'H)

UINT8 opTrnOrient

operational train orientation '00'B = unknown '01'B = same as train direction '10'B = inverse to train direction

UINT8 opCstCnt

number of consists in train (1..63)

TRDP\_OP\_CONSIST\_T opCstList [TRDP\_MAX\_CST\_CNT]

operational consist list starting with op.

• UINT8 reserved05

reserved for future use (= 0)

UINT8 opVehCnt

number of vehicles in train (1..63)

TRDP\_OP\_VEHICLE\_T opVehList [TRDP\_MAX\_VEH\_CNT]

operational vehicle list starting with op.

TRDP\_OP\_TRAIN\_DIR\_STATE\_T state

operational state of the train

UINT32 cstNetProp

consist network properties bit0..1: consist orientation bit2..7: 0 bit8..13: ETBN Id bit14..15: 0 bit16..21: subnet Id bit24..29: CN Id bit30..31: 0

UINT16 entryCnt

number of entries in train network directory

TRDP\_TRAIN\_NET\_DIR\_ENTRY\_T trnNetDir [TRDP\_MAX\_CST\_CNT]

train network directory

TRDP OP TRAIN DIR T opTrnDir

operational directory

TRDP\_TRAIN\_DIR\_T trnDir

train directory

• UINT32 sequenceCounter

Unique counter (autom incremented)

• UINT16 protocolVersion

fix value for compatibility (set by the API)

UINT16 msgType

of datagram: PD Request (0x5072) or PD\_MSG (0x5064)

UINT32 comld

set by user: unique id
• UINT32 datasetLength

length of the data to transmit 0...1432

UINT32 reserved

before used for ladder support

UINT32 replyComId

used in PD request

UINT32 replylpAddress

used for PD request

· UINT32 frameCheckSum

CRC32 of header.

INT32 replyStatus

0 = OK

• UINT8 sessionID [16u]

UUID as a byte stream.

UINT32 replyTimeout

in us

• UINT8 sourceURI [32u]

User part of URI.

• UINT8 destinationURI [32u]

User part of URI.

PD\_HEADER\_T frameHead

Packet header in network byte order.

UINT8 data [TRDP\_MAX\_PD\_DATA\_SIZE]

data ready to be sent or received (with CRCs)

# 4.2.1 Detailed Description

Types for ETB control.

TRDP PD packet.

TRDP message data header - network order and alignment.

TRDP process data header - network order and alignment.

Complete TTDB structure.

Train network directory structure.

Train network directory entry structure acc.

Operational Train directory status info structure.

Operational train structure.

Operational train directory state.

Operational consist structure. Operational vehicle structure. TCN train directory. CSTINFO Control telegram. TCN consist structure. Version information for communication buffers. to IEC61375-2-5 4.2.2 Field Documentation 4.2.2.1 confVehCnt UINT16 GNU\_PACKED::confVehCnt number of confirmed vehicles in the train (1..63). 4.2.2.2 confVehList TRDP\_OP\_VEHICLE\_T GNU\_PACKED::confVehList[TRDP\_MAX\_VEH\_CNT] ordered list of confirmed vehicles in the train, starting with vehicle at train head, see chapter 5.3.3.2.10. Parameters 'isLead' and 'leadDir' to be set to 0 4.2.2.3 cstList TRDP\_CONSIST\_T GNU\_PACKED::cstList

consist list ordered list starting with trnCstNo == 1 Note: This is a variable size array, only opCstCnt array elements are present on the network and for crc computation

If trnCstNo > 0 this shall be an ordered list starting with trnCstNo == 1 (exactly the same as in structure TRAIN $\leftarrow$  \_DIRECTORY). If trnCstNo == 0 it is not mandatory to list all consists (only consists which should send CSTINFO telegram). The parameters 'trnCstNo' and 'cstOrient' are optional and can be set to 0.

consist list.

### 4.2.2.4 cstUUID

```
TRDP_UUID_T GNU_PACKED::cstUUID
```

UUID of the consist, provided by ETBN (TrainNetworkDirectory) Reference to static consist attributes 0 if not available (e.g.

unique consist identifier

Reference to static consist attributes, 0 if not available (e.g.

correction)

### 4.2.2.5 datasetLength

```
UINT32 GNU_PACKED::datasetLength
```

length of the data to transmit 0...1432

defined by user: length of data to transmit

#### 4.2.2.6 deviceName

```
TRDP_NET_LABEL_T GNU_PACKED::deviceName
```

function device of ECSC which sends the telegram

function device of ED which sends the telegram

### 4.2.2.7 etbld

```
UINT8 GNU_PACKED::etbId
```

identification of the ETB the TTDB is computed for bit0: ETB0 (operational network) bit1: ETB1 (multimedia network) bit2: ETB2 (other network) bit3: ETB3 (other network)

identification of the ETB the TTDB is computed for 0: ETB0 (operational network) 1: ETB1 (multimedia network) 2: ETB2 (other network) 3: ETB3 (other network)

## 4.2.2.8 etbTopoCnt

UINT32 GNU\_PACKED::etbTopoCnt

ETB topography counter.

set by user: ETB to use, '0' for consist local traffic

train network directory CRC

#### 4.2.2.9 inhibit

UINT8 GNU\_PACKED::inhibit

inauguration inhibit 0 = no inhibit request 1 = inhibit request

ETBN inhibit 0 = no action (keep old state) 1 = no inhibit request 2 = inhibit request.

#### 4.2.2.10 isLead

ANTIVALENT8 GNU\_PACKED::isLead

vehicle is leading

consist contains leading vehicle, '01'B = false, '10'B = true

#### 4.2.2.11 leadDir

UINT8 GNU\_PACKED::leadDir

vehicle leading direction 0 = not relevant 1 = leading direction 1 2 = leading direction 2

'vehicle leading direction 0 = not relevant 1 = leading direction 1 2 = leading direction 2

# 4.2.2.12 leadVehOfCst

UINT8 GNU\_PACKED::leadVehOfCst

position of leading vehicle in consist, 0..31 (1: first vehicle in consist in Direction 1, 2: second vehicle, etc.)

position of leading vehicle in consist range 0...32 0 = not defined 1 = first vehicle in consist in direction 1 2 = second vehicle etc.

# 4.2.2.13 lifesign

UINT16 GNU\_PACKED::lifesign

wrap-around counter, incremented with each produced datagram.

### 4.2.2.14 msgType

UINT16 GNU\_PACKED::msgType

of datagram: PD Request (0x5072) or PD\_MSG (0x5064)

of datagram: Mn, Mr, Mp, Mq, Mc or Me

### 4.2.2.15 opCstList

```
TRDP_OP_CONSIST_T GNU_PACKED::opCstList[TRDP_MAX_CST_CNT]
```

operational consist list starting with op.

consist #1 Note: This is a variable size array, only opCstCnt array elements are present

### 4.2.2.16 opTrnDirState

```
UINT8 GNU_PACKED::opTrnDirState
```

train directory state 1 = INVALID 2 = VALID 4 = SHARED other values are not allowed

Operational train directory status: '01'B == invalid, '10'B == valid, '100'B == shared.

## 4.2.2.17 opTrnTopoCnt

```
UINT32 GNU_PACKED::opTrnTopoCnt
```

operational train topology counter

set by user: direction/side critical, '0' if ignored

operational train topology counter computed as defined in 5.3.3.2.16 (seed value : trnTopoCnt)

operational train topology counter set to 0 if opTrnDirState == invalid

operational train topocounter value of the operational train directory the correction is based on

## 4.2.2.18 opVehList

```
TRDP_OP_VEHICLE_T GNU_PACKED::opVehList[TRDP_MAX_VEH_CNT]
```

operational vehicle list starting with op.

vehicle #1 Note: This is a variable size array, only opCstCnt array elements are present

## 4.2.2.19 ownOpCstNo

```
UINT8 GNU_PACKED::ownOpCstNo
```

own operational address (= 1..32) = 0 if unknown (e.g.

operational consist number the vehicle belongs to

after Inauguration)

```
4.2.2.20 protocolVersion
UINT16 GNU_PACKED::protocolVersion
fix value for compatibility (set by the API)
fix value for compatibility
4.2.2.21 reserved01 [1/2]
UINT16 GNU_PACKED::reserved01
reserved (=0)
reserved for future use (= 0)
4.2.2.22 reserved01 [2/2]
UINT8 GNU_PACKED::reserved01
reserved (=0)
reserved for future use (= 0)
4.2.2.23 reserved02 [1/2]
UINT16 GNU_PACKED::reserved02
reserved (=0)
reserved (= 0)
reserved for future use (= 0)
4.2.2.24 reserved02 [2/2]
UINT16 GNU_PACKED::reserved02
reserved (=0)
reserved (= 0)
4.2.2.25 reserved03
UINT8 GNU_PACKED::reserved03
reserved (=0)
reserved for future use (= 0)
```

```
4.2.2.26 reserved04
UINT8 GNU_PACKED::reserved04
reserved (=0)
reserved for future use (= 0)
4.2.2.27 reserved06
UINT8 GNU_PACKED::reserved06
reserved (=0)
reserved for future use (= 0)
4.2.2.28 safetyTrail
TRDP_ETB_CTRL_VDP_T GNU_PACKED::safetyTrail
ETBCTRL-VDP trailer, completely set to 0 == not used.
ETBCTRL-VDP trailer, parameter 'safeSequCount' == 0 completely set to 0 == not used.
ETBCTRL-VDP trailer, parameter 'safeSequCount' == 0 completely set to 0 == SDTv2 not used.
ETBCTRL-VDP trailer, completely set to 0 == SDTv2 not used.
4.2.2.29 trnCstNo
UINT8 GNU_PACKED::trnCstNo
own TCN consist number (= 1..32)
sequence number of consist in train with vehicle 01 being the first vehicle in ETB reference direction 1 as defined in
IEC61375-2-5, value range: 1..63, 0 = inserted by correction
train consist number telegram control type 0 = with trnTopoCnt tracking 1 = without trnTopoCnt tracking
Sequence number of consist in train (1..63)
4.2.2.30 trnDirState
UINT8 GNU_PACKED::trnDirState
train directory state 1 = UNCONFIRMED 2 = CONFIRMED other values are not allowed
TTDB status: '01'B == unconfirmed, '10'B == confirmed.
```

# 4.2.2.31 trnld

```
TRDP_NET_LABEL_T GNU_PACKED::trnId
```

train identifier, application defined (e.g.

'ICE75', 'IC346'), informal

#### 4.2.2.32 trnNetDir

```
TRDP_TRAIN_NET_DIR_T GNU_PACKED::trnNetDir
```

dynamic train info

network directory

### 4.2.2.33 trnOperator

```
TRDP_NET_LABEL_T GNU_PACKED::trnOperator
```

train operator, e.g.

'trenitalia.it', informal

### 4.2.2.34 trnTopoCnt

```
UINT32 GNU_PACKED::trnTopoCnt
```

trnTopoCnt value ctrlType == 0: actual value ctrlType == 1: set to 0

computed as defined in 5.3.3.2.16 (seed value: etbTopoCnt)

# 4.2.2.35 trnVehNo

```
UINT8 GNU_PACKED::trnVehNo
```

vehicle sequence number within the train with vehicle 01 being the first vehicle in ETB reference direction 1 as defined in IEC61375-2-5 value range: 0..63 a value of 0 indicates that this vehicle has been inserted by correction

vehicle sequence number within the train with vehicle 01 being the first vehicle in ETB reference direction 1 as defined in IEC61375-2-5, value range: 1..63, a value of 0 indicates that this vehicle has been inserted by correction

### 4.2.2.36 vehld

```
TRDP_NET_LABEL_T GNU_PACKED::vehId
```

Unique vehicle identifier, application defined (e.g.

UIC Identifier)

#### 4.2.2.37 vehOrient

UINT8 GNU\_PACKED::vehOrient

vehicle orientation 0 = not known (corrected vehicle) 1 = same as operational train direction 2 = inverse to operational train direction

vehicle orientation, '00'B = not known (corrected vehicle) '01'B = same as operational train direction '10'B = inverse to operational train direction

### 4.2.2.38 version

TRDP\_SHORT\_VERSION\_T GNU\_PACKED::version

telegram version information, main version = 1, sub version = 0

Train info structure version.

TrainDirectoryState data structure version parameter 'mainVersion' shall be set to 1.

TrainDirectory data structure version parameter 'mainVersion' shall be set to 1.

Consist Info Control structure version parameter 'mainVersion' shall be set to 1.

The documentation for this struct was generated from the following files:

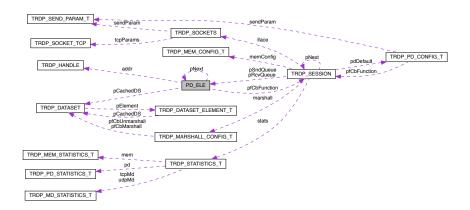
- tau\_ctrl\_types.h
- tau\_tti\_types.h
- · trdp private.h

# 4.3 PD\_ELE Struct Reference

Queue element for PD packets to send or receive.

#include <trdp\_private.h>

Collaboration diagram for PD\_ELE:



### **Data Fields**

struct PD\_ELE \* pNext

pointer to next element or NULL

UINT32 magic

prevent acces through dangeling pointer

TRDP\_ADDRESSES\_T addr

handle of publisher/subscriber

TRDP IP ADDR T lastSrcIP

last source IP a subscribed packet was received from

TRDP\_IP\_ADDR\_T pullipAddress

In case of pulling a PD this is the requested Ip.

UINT32 redId

Redundancy group ID or zero.

UINT32 curSeqCnt

the last sent or received sequence counter

UINT32 curSeqCnt4Pull

the last sent sequence counter for PULL

• TRDP\_SEQ\_CNT\_LIST\_T \* pSeqCntList

pointer to list of received sequence numbers per comld

UINT32 numRxTx

Counter for received packets (statistics)

UINT32 updPkts

Counter for updated packets (statistics)

UINT32 getPkts

Counter for read packets (statistics)

UINT32 numMissed

Counter for skipped sequence number (statistics)

TRDP\_ERR\_T lastErr

Last error (timeout)

· TRDP PRIV FLAGS T privFlags

private flags

TRDP\_FLAGS\_T pktFlags

flags

TRDP\_TIME\_T interval

time out value for received packets or interval for packets to send (set from ms)

• TRDP\_TIME\_T timeToGo

next time this packet must be sent/rcv

TRDP\_TO\_BEHAVIOR\_T toBehavior

timeout behavior for packets

UINT32 dataSize

net data size

UINT32 grossSize

complete packet size (header, data)

UINT32 sendSize

data size sent out

TRDP\_DATASET\_T \* pCachedDS

Pointer to dataset element if known.

INT32 socketldx

index into the socket list

const void \* pUserRef

from subscribe()

• TRDP\_PD\_CALLBACK\_T pfCbFunction

Pointer to PD callback function.

• PD\_PACKET\_T \* pFrame

header ...

# 4.3.1 Detailed Description

Queue element for PD packets to send or receive.

# 4.3.2 Field Documentation

## 4.3.2.1 pFrame

```
PD_PACKET_T* PD_ELE::pFrame
```

header ...

data + FCS...

The documentation for this struct was generated from the following file:

• trdp\_private.h

# 4.4 TAU\_MARSHALL\_INFO\_T Struct Reference

Marshalling info, used to and from wire.

# **Data Fields**

• INT32 level

track recursive level

UINT8 \* pSrc

source pointer

UINT8 \* pSrcEnd

last source

UINT8 \* pDst

destination pointer

UINT8 \* pDstEnd

last destination

## 4.4.1 Detailed Description

Marshalling info, used to and from wire.

The documentation for this struct was generated from the following file:

· tau marshall.c

# 4.5 TCN\_URI Struct Reference

TCN-DNS simplified header structures.

```
#include <tau_dnr_types.h>
```

### **Data Fields**

• CHAR8 tcnUriStr [80]

if != 0 use TCN DNS as resolver

INT16 resolvState

on request: reserved (= 0), on reply: -1 unknown, 0 OK

UINT32 tcnUrilpAddr

IP address of URI.

UINT32 tcnUrilpAddr2

if != 0, end IP address of range

# 4.5.1 Detailed Description

TCN-DNS simplified header structures.

The documentation for this struct was generated from the following file:

• tau\_dnr\_types.h

# 4.6 TRDP\_CLTR\_CST\_INFO\_T Struct Reference

Closed train consists information.

```
#include <tau_tti_types.h>
```

# **Data Fields**

• TRDP\_UUID\_T cltrCstUUID

closed train consist UUID

UINT8 cltrCstOrient

closed train consist orientation '01'B = same as closed train direction '10'B = inverse to closed train direction

UINT8 cltrCstNo

sequence number of the consist within the closed train, value range 1..32

UINT16 reserved01

reserved for future use (= 0)

# 4.6.1 Detailed Description

Closed train consists information.

The documentation for this struct was generated from the following file:

• tau\_tti\_types.h

# 4.7 TRDP\_COMID\_DSID\_MAP\_T Struct Reference

Comld - data set mapping element definition.

```
#include <trdp_types.h>
```

## **Data Fields**

UINT32 comld

comld

· UINT32 datasetId

corresponding dataset Id

# 4.7.1 Detailed Description

Comld - data set mapping element definition.

The documentation for this struct was generated from the following file:

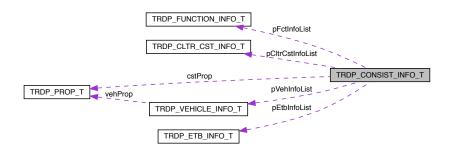
• trdp\_types.h

# 4.8 TRDP\_CONSIST\_INFO\_T Struct Reference

consist information structure

```
#include <tau_tti_types.h>
```

Collaboration diagram for TRDP\_CONSIST\_INFO\_T:



### **Data Fields**

TRDP\_SHORT\_VERSION\_T version

ConsistInfo data structure version, application defined mainVersion = 1, subVersion = 0.

UINT8 cstClass

consist info classification 1 = (single) consist 2 = closed train 3 = closed train consist

UINT8 reserved01

reserved for future use (= 0)

• TRDP\_NET\_LABEL\_T cstld

application defined consist identifier, e.g.

TRDP\_NET\_LABEL\_T cstType

consist type, application defined

TRDP\_NET\_LABEL\_T cstOwner

consist owner, e.g.

TRDP\_UUID\_T cstUUID

consist UUID

UINT32 reserved02

reserved for future use (= 0)

TRDP\_PROP\_T cstProp

static consist properties

• UINT16 reserved03

reserved for future use (= 0)

UINT16 etbCnt

number of ETB's, range: 1..4

TRDP\_ETB\_INFO\_T \* pEtbInfoList

ETB information list for the consist Ordered list starting with lowest etbld.

UINT16 reserved04

reserved for future use (= 0)

UINT16 vehCnt

number of vehicles in consist 1..32

• TRDP VEHICLE INFO T \* pVehInfoList

vehicle info list for the vehicles in the consist Ordered list starting with cstVehNo==1

UINT16 reserved05

reserved for future use (= 0)

UINT16 fctCnt

number of consist functions value range 0..1024

• TRDP\_FUNCTION\_INFO\_T \* pFctInfoList

function info list for the functions in consist lexicographical ordered by fctName

• UINT16 reserved06

reserved for future use (= 0)

UINT16 cltrCstCnt

number of original consists in closed train value range: 0..32, 0 = consist is no closed train

• TRDP\_CLTR\_CST\_INFO\_T \* pCltrCstInfoList

info on closed train composition Ordered list starting with cltrCstNo == 1

UINT32 cstTopoCnt

consist topology counter computed as defined in 5.3.3.2.16, seed value: 'FFFFFFF'H

### 4.8.1 Detailed Description

consist information structure

# 4.8.2 Field Documentation

### 4.8.2.1 cstld

```
TRDP_NET_LABEL_T TRDP_CONSIST_INFO_T::cstId
```

application defined consist identifier, e.g.

**UIC** identifier

### 4.8.2.2 cstOwner

```
TRDP_NET_LABEL_T TRDP_CONSIST_INFO_T::cstOwner
```

consist owner, e.g.

"trenitalia.it", "sncf.fr", "db.de"

The documentation for this struct was generated from the following file:

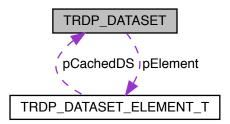
• tau\_tti\_types.h

# 4.9 TRDP\_DATASET Struct Reference

Dataset definition.

#include <trdp\_types.h>

Collaboration diagram for TRDP\_DATASET:



# **Data Fields**

• UINT32 id

dataset identifier > 1000

UINT16 reserved1

Reserved for future use, must be zero.

• UINT16 numElement

Number of elements.

TRDP\_DATASET\_ELEMENT\_T pElement []

Pointer to a dataset element, used as array.

# 4.9.1 Detailed Description

Dataset definition.

The documentation for this struct was generated from the following file:

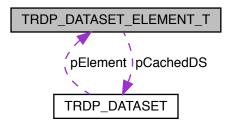
• trdp\_types.h

# 4.10 TRDP\_DATASET\_ELEMENT\_T Struct Reference

Dataset element definition.

#include <trdp\_types.h>

Collaboration diagram for TRDP\_DATASET\_ELEMENT\_T:



## **Data Fields**

· UINT32 type

Data type (TRDP\_DATA\_TYPE\_T 1...99) or dataset id > 1000.

UINT32 size

Number of items or TRDP\_VAR\_SIZE (0)

• CHAR8 \* name

Name param, on special request (Ticket #211)

· CHAR8 \* unit

Unit text for visualisation.

• REAL32 scale

Factor for visualisation.

INT32 offset

Offset for visualisation (val = scale \* x + offset)

struct TRDP\_DATASET \* pCachedDS

Used internally for marshalling speed-up.

# 4.10.1 Detailed Description

Dataset element definition.

The documentation for this struct was generated from the following file:

· trdp\_types.h

# 4.11 TRDP\_DBG\_CONFIG\_T Struct Reference

Control for debug output device/file on application level.

```
#include <tau_xml.h>
```

## **Data Fields**

TRDP\_DBG\_OPTION\_T option

Debug printout options for application use.

UINT32 maxFileSize

Maximal file size.

• TRDP\_FILE\_NAME\_T fileName

Debug file name and path.

# 4.11.1 Detailed Description

Control for debug output device/file on application level.

The documentation for this struct was generated from the following file:

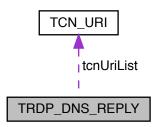
• tau\_xml.h

# 4.12 TRDP\_DNS\_REPLY Struct Reference

TCN-DNS Reply telegram TCN\_DNS\_REP\_DS.

#include <tau\_dnr\_types.h>

Collaboration diagram for TRDP\_DNS\_REPLY:



## **Data Fields**

• TRDP\_SHORT\_VERSION\_T version

1.0

• TRDP\_NET\_LABEL\_T deviceName

function device of ED which sends the telegram

UINT32 etbTopoCnt

ETB topography counter.

UINT32 opTrnTopoCnt

operational train topography counter needed for TCN-URIs related to the operational train view = 0 if not used

· UINT8 etbld

identification of the related ETB 0 = ETB0 (operational network) 1 = ETB1 (multimedia network) 2 = ETB2 (other network) 3 = ETB3 (other network) 255 = don't care (for access to local DNS server)

INT8 dnsStatus

0 = OK -1 = DNS Server not ready -2 = Inauguration in progress

UINT8 tcnUriCnt

number of TCN-URIs to be resolved value range: 0 .

• TCN\_URI\_T tcnUriList [255]

defined for max size

• TRDP\_ETB\_CTRL\_VDP\_T safetyTrail

SDT trailer.

# 4.12.1 Detailed Description

TCN-DNS Reply telegram TCN\_DNS\_REP\_DS.

### 4.12.2 Field Documentation

#### 4.12.2.1 tcnUriCnt

UINT8 TRDP\_DNS\_REPLY::tcnUriCnt

number of TCN-URIs to be resolved value range: 0 .

. 255

The documentation for this struct was generated from the following file:

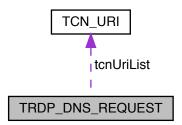
· tau\_dnr\_types.h

# 4.13 TRDP\_DNS\_REQUEST Struct Reference

TCN-DNS Request telegram TCN DNS REQ DS.

#include <tau\_dnr\_types.h>

Collaboration diagram for TRDP\_DNS\_REQUEST:



## **Data Fields**

TRDP\_SHORT\_VERSION\_T version

1.0

• TRDP\_NET\_LABEL\_T deviceName

function device of ED which sends the telegram

UINT32 etbTopoCnt

ETB topography counter.

UINT32 opTrnTopoCnt

operational train topography counter needed for TCN-URIs related to the operational train view = 0 if not used

UINT8 etblo

identification of the related ETB 0 = ETB0 (operational network) 1 = ETB1 (multimedia network) 2 = ETB2 (other network) 3 = ETB3 (other network) 255 = don't care (for access to local DNS server)

UINT8 tcnUriCnt

number of TCN-URIs to be resolved value range: 0 .

• TCN\_URI\_T tcnUriList [255]

defined for max size

TRDP\_ETB\_CTRL\_VDP\_T safetyTrail

SDT trailer.

# 4.13.1 Detailed Description

TCN-DNS Request telegram TCN\_DNS\_REQ\_DS.

### 4.13.2 Field Documentation

#### 4.13.2.1 tcnUriCnt

```
UINT8 TRDP_DNS_REQUEST::tcnUriCnt
```

number of TCN-URIs to be resolved value range: 0.

. 255

The documentation for this struct was generated from the following file:

· tau\_dnr\_types.h

# 4.14 TRDP\_ETB\_INFO\_T Struct Reference

Types for train configuration information.

```
#include <tau_tti_types.h>
```

## **Data Fields**

UINT8 etbld

identification of train backbone; value range: 0..3

UINT8 cnCnt

number of CNs within consist connected to this ETB value range 1..16 referring to cnld 0..15 acc.

• UINT16 reserved01

reserved for future use (= 0)

# 4.14.1 Detailed Description

Types for train configuration information.

ETB information

## 4.14.2 Field Documentation

### 4.14.2.1 cnCnt

```
UINT8 TRDP_ETB_INFO_T::cnCnt
```

number of CNs within consist connected to this ETB value range 1..16 referring to cnld 0..15 acc.

IEC61375-2-5

The documentation for this struct was generated from the following file:

· tau\_tti\_types.h

# 4.15 TRDP\_FUNCTION\_INFO\_T Struct Reference

function/device information structure

```
#include <tau_tti_types.h>
```

## **Data Fields**

· TRDP NET LABEL T fctName

function device or group label

UINT16 fctld

host identification of the function device or group as defined in IEC 61375-2-5, application defined.

BOOL8 grp

is a function group and will be resolved as IP multicast address

• UINT8 reserved01

reserved for future use (= 0)

UINT8 cstVehNo

Sequence number of the vehicle in the consist the function belongs to.

UINT8 etbld

number of connected train backbone.

• UINT8 cnld

identifier of connected consist network in the consist, related to the etbld.

• UINT8 reserved02

reserved for future use (= 0)

# 4.15.1 Detailed Description

function/device information structure

## 4.15.2 Field Documentation

### 4.15.2.1 cnld

```
UINT8 TRDP_FUNCTION_INFO_T::cnId
```

identifier of connected consist network in the consist, related to the etbld.

Value range: 0..31

#### 4.15.2.2 cstVehNo

```
UINT8 TRDP_FUNCTION_INFO_T::cstVehNo
```

Sequence number of the vehicle in the consist the function belongs to.

Value range: 1..16, 0 = not defined

#### 4.15.2.3 etbld

```
UINT8 TRDP_FUNCTION_INFO_T::etbId
```

number of connected train backbone.

Value range: 0..3

# 4.15.2.4 fctld

```
UINT16 TRDP_FUNCTION_INFO_T::fctId
```

host identification of the function device or group as defined in IEC 61375-2-5, application defined.

Value range: 1..16383 (device), 256..16383 (group)

The documentation for this struct was generated from the following file:

• tau\_tti\_types.h

# 4.16 TRDP\_HANDLE Struct Reference

Hidden handle definition, used as unique addressing item.

```
#include <trdp_private.h>
```

## **Data Fields**

UINT32 comId

comld for packets to send/receive

TRDP\_IP\_ADDR\_T srclpAddr

source IP for PD/MD

• TRDP\_IP\_ADDR\_T srclpAddr2

second source IP for PD/MD

TRDP\_IP\_ADDR\_T destlpAddr

destination IP for PD

• TRDP\_IP\_ADDR\_T mcGroup

multicast group to join for PD

UINT32 etbTopoCnt

etb topocount belongs to addressing item

UINT32 opTrnTopoCnt

opTrn topocount belongs to addressing item

# 4.16.1 Detailed Description

Hidden handle definition, used as unique addressing item.

The documentation for this struct was generated from the following file:

• trdp\_private.h

# 4.17 TRDP\_LIST\_STATISTICS\_T Struct Reference

Information about a particular MD listener.

```
#include <trdp_types.h>
```

### **Data Fields**

UINT32 comld

Comld to listen to.

TRDP\_URI\_USER\_T uri

URI user part to listen to.

• TRDP\_IP\_ADDR\_T joinedAddr

Joined IP address.

UINT32 callBack

Call back function if used.

UINT32 userRef

User reference if used.

• UINT32 numSessions

Number of sessions.

# 4.17.1 Detailed Description

Information about a particular MD listener.

The documentation for this struct was generated from the following file:

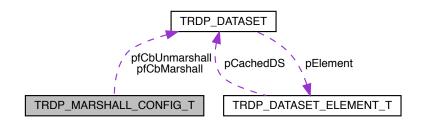
· trdp\_types.h

# 4.18 TRDP\_MARSHALL\_CONFIG\_T Struct Reference

Marshaling/unmarshalling configuration.

```
#include <trdp_types.h>
```

Collaboration diagram for TRDP\_MARSHALL\_CONFIG\_T:



# **Data Fields**

- TRDP\_MARSHALL\_T pfCbMarshall
  - Pointer to marshall callback function.
- TRDP\_UNMARSHALL\_T pfCbUnmarshall

Pointer to unmarshall callback function.

void \* pRefCon

Pointer to user context for call back.

# 4.18.1 Detailed Description

Marshaling/unmarshalling configuration.

The documentation for this struct was generated from the following file:

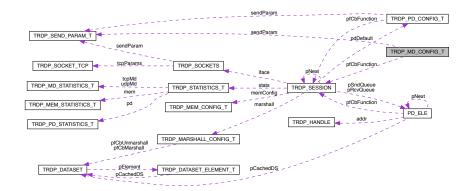
trdp\_types.h

# 4.19 TRDP\_MD\_CONFIG\_T Struct Reference

Default MD configuration.

#include <trdp\_types.h>

Collaboration diagram for TRDP\_MD\_CONFIG\_T:



## **Data Fields**

• TRDP\_MD\_CALLBACK\_T pfCbFunction

Pointer to MD callback function.

void \* pRefCon

Pointer to user context for call back.

• TRDP\_SEND\_PARAM\_T sendParam

Default send parameters.

TRDP\_FLAGS\_T flags

Default flags for MD packets.

• UINT32 replyTimeout

Default reply timeout in us.

UINT32 confirmTimeout

Default confirmation timeout in us.

UINT32 connectTimeout

Default connection timeout in us.

UINT32 sendingTimeout

Default sending timeout in us.

UINT16 udpPort

Port to be used for UDP MD communication.

UINT16 tcpPort

Port to be used for TCP MD communication.

• UINT32 maxNumSessions

Maximal number of replier sessions.

## 4.19.1 Detailed Description

Default MD configuration.

The documentation for this struct was generated from the following file:

· trdp\_types.h

# 4.20 TRDP\_MD\_INFO\_T Struct Reference

Message data info from received telegram; allows the application to generate responses.

```
#include <trdp_types.h>
```

### **Data Fields**

• TRDP\_IP\_ADDR\_T srclpAddr

source IP address for filtering

TRDP\_IP\_ADDR\_T destIpAddr

destination IP address for filtering

UINT32 seqCount

sequence counter

UINT16 protVersion

Protocol version.

TRDP\_MSG\_T msgType

Protocol ('PD', 'MD', ...)

UINT32 comld

ComID.

UINT32 etbTopoCnt

received topocount

• UINT32 opTrnTopoCnt

received topocount

BOOL8 aboutToDie

session is about to die

UINT32 numRepliesQuery

number of ReplyQuery received

UINT32 numConfirmSent

number of Confirm sent

UINT32 numConfirmTimeout

number of Confirm Timeouts (incremented by listeners

· UINT16 userStatus

error code, user stat

TRDP\_REPLY\_STATUS\_T replyStatus

reply status

TRDP\_UUID\_T sessionId

for response

UINT32 replyTimeout

reply timeout in us given with the request

• TRDP\_URI\_USER\_T srcUserURI

source URI user part from MD header

• TRDP URI HOST T srcHostURI

source URI host part (unused)

TRDP\_URI\_USER\_T destUserURI

destination URI user part from MD header

TRDP\_URI\_HOST\_T destHostURI

destination URI host part (unused)

UINT32 numExpReplies

number of expected replies, 0 if unknown

UINT32 numReplies

actual number of replies for the request

const void \* pUserRef

User reference given with the local call.

TRDP\_ERR\_T resultCode

error code

## 4.20.1 Detailed Description

Message data info from received telegram; allows the application to generate responses.

Note: Not all fields are relevant for each message type!

The documentation for this struct was generated from the following file:

· trdp\_types.h

# 4.21 TRDP\_MD\_STATISTICS\_T Struct Reference

Structure containing all general MD statistics information.

```
#include <trdp_types.h>
```

## **Data Fields**

UINT32 defQos

default QoS for MD

UINT32 defTtl

default TTL for MD

UINT32 defReplyTimeout

default reply timeout in us for MD

UINT32 defConfirmTimeout

default confirm timeout in us for MD

UINT32 numList

number of listeners

UINT32 numRcv

number of received MD packets

UINT32 numCrcErr

number of received MD packets with CRC err

UINT32 numProtErr

number of received MD packets with protocol err

UINT32 numTopoErr

number of received MD packets with wrong topo count

UINT32 numNoListener

number of received MD packets without listener

UINT32 numReplyTimeout

number of reply timeouts

UINT32 numConfirmTimeout

number of confirm timeouts

UINT32 numSend

number of sent MD packets

## 4.21.1 Detailed Description

Structure containing all general MD statistics information.

The documentation for this struct was generated from the following file:

• trdp\_types.h

# 4.22 TRDP\_MEM\_CONFIG\_T Struct Reference

Enumeration type for memory pre-fragmentation, reuse of VOS definition.

```
#include <trdp_types.h>
```

## **Data Fields**

UINT8 \* p

pointer to static or allocated memory

UINT32 size

size of static or allocated memory

UINT32 prealloc [VOS\_MEM\_NBLOCKSIZES]

memory block structure

## 4.22.1 Detailed Description

Enumeration type for memory pre-fragmentation, reuse of VOS definition.

Structure describing memory (and its pre-fragmentation)

The documentation for this struct was generated from the following file:

• trdp\_types.h

# 4.23 TRDP\_MEM\_STATISTICS\_T Struct Reference

Structure containing all general memory statistics information.

#include <trdp\_types.h>

## **Data Fields**

UINT32 total

total memory size

UINT32 free

free memory size

• UINT32 minFree

minimal free memory size in statistics interval

• UINT32 numAllocBlocks

allocated memory blocks

UINT32 numAllocErr

allocation errors

UINT32 numFreeErr

free errors

UINT32 blockSize [VOS\_MEM\_NBLOCKSIZES]

preallocated memory blocks

UINT32 usedBlockSize [VOS\_MEM\_NBLOCKSIZES]

used memory blocks

## 4.23.1 Detailed Description

Structure containing all general memory statistics information.

The documentation for this struct was generated from the following file:

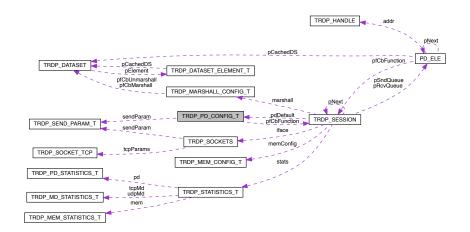
• trdp\_types.h

# 4.24 TRDP\_PD\_CONFIG\_T Struct Reference

Default PD configuration.

#include <trdp\_types.h>

Collaboration diagram for TRDP\_PD\_CONFIG\_T:



### **Data Fields**

TRDP\_PD\_CALLBACK\_T pfCbFunction

Pointer to PD callback function.

void \* pRefCon

Pointer to user context for call back.

TRDP\_SEND\_PARAM\_T sendParam

Default send parameters.

• TRDP\_FLAGS\_T flags

Default flags for PD packets.

UINT32 timeout

Default timeout in us.

• TRDP\_TO\_BEHAVIOR\_T toBehavior

Default timeout behavior.

UINT16 port

Port to be used for PD communication.

# 4.24.1 Detailed Description

Default PD configuration.

The documentation for this struct was generated from the following file:

• trdp\_types.h

# 4.25 TRDP\_PD\_INFO\_T Struct Reference

Process data info from received telegram; allows the application to generate responses.

```
#include <trdp_types.h>
```

### **Data Fields**

TRDP\_IP\_ADDR\_T srclpAddr

source IP address for filtering

TRDP\_IP\_ADDR\_T destlpAddr

destination IP address for filtering

UINT32 seqCount

sequence counter

UINT16 protVersion

Protocol version.

TRDP\_MSG\_T msgType

Protocol ('PD', 'MD', ...)

UINT32 comld

ComID.

UINT32 etbTopoCnt

received ETB topocount

UINT32 opTrnTopoCnt

received operational train directory topocount

UINT32 replyComId

ComID for reply (request only)

TRDP\_IP\_ADDR\_T replyIpAddr

IP address for reply (request only)

const void \* pUserRef

User reference given with the local subscribe.

• TRDP\_ERR\_T resultCode

error code

TRDP\_URI\_HOST\_T srcHostURI

source URI host part (unused)

TRDP\_URI\_HOST\_T destHostURI

destination URI host part (unused)

TRDP\_TO\_BEHAVIOR\_T toBehavior

callback can decide about handling of data on timeout

## 4.25.1 Detailed Description

Process data info from received telegram; allows the application to generate responses.

Note: Not all fields are relevant for each message type!

The documentation for this struct was generated from the following file:

• trdp\_types.h

# 4.26 TRDP\_PD\_STATISTICS\_T Struct Reference

Structure containing all general PD statistics information.

```
#include <trdp_types.h>
```

## **Data Fields**

UINT32 defQos

default QoS for PD

UINT32 defTtl

default TTL for PD

UINT32 defTimeout

default timeout in us for PD

UINT32 numSubs

number of subscribed Comld's

UINT32 numPub

number of published ComId's

UINT32 numRcv

number of received PD packets

UINT32 numCrcErr

number of received PD packets with CRC err

UINT32 numProtErr

number of received PD packets with protocol err

UINT32 numTopoErr

number of received PD packets with wrong topo count

UINT32 numNoSubs

number of received PD push packets without subscription

UINT32 numNoPub

number of received PD pull packets without publisher

UINT32 numTimeout

number of PD timeouts

UINT32 numSend

number of sent PD packets

UINT32 numMissed

number of packets skipped

## 4.26.1 Detailed Description

Structure containing all general PD statistics information.

The documentation for this struct was generated from the following file:

• trdp\_types.h

# 4.27 TRDP\_PROCESS\_CONFIG\_T Struct Reference

Various flags/general TRDP options for library initialization.

```
#include <trdp_types.h>
```

## **Data Fields**

TRDP\_LABEL\_T hostName

Host name.

• TRDP\_LABEL\_T leaderName

Leader name dependant on redundancy concept.

UINT32 cycleTime

TRDP main process cycle time in us.

· UINT32 priority

TRDP main process priority (0-255, 0=default, 255=highest)

TRDP\_OPTION\_T options

TRDP options.

## 4.27.1 Detailed Description

Various flags/general TRDP options for library initialization.

The documentation for this struct was generated from the following file:

trdp\_types.h

# 4.28 TRDP\_PROP\_T Struct Reference

Application defined properties.

```
#include <tau_tti_types.h>
```

## **Data Fields**

• TRDP\_SHORT\_VERSION\_T ver

properties version information, application defined

• UINT16 len

properties length in number of octets, application defined, must be a multiple of 4 octets for alignment reasons value range: 0..32768

• UINT8 prop [1]

properties, application defined

# 4.28.1 Detailed Description

Application defined properties.

The documentation for this struct was generated from the following file:

• tau\_tti\_types.h

# 4.29 TRDP\_PUB\_STATISTICS\_T Struct Reference

Table containing particular PD publishing information.

```
#include <trdp_types.h>
```

# **Data Fields**

UINT32 comId

Published Comld.

TRDP\_IP\_ADDR\_T destAddr

IP address of destination for this publishing.

UINT32 cycle

Publishing cycle in us.

UINT32 redId

Redundancy group id.

UINT32 redState

Redundant state.Leader or Follower.

UINT32 numPut

Number of packet updates.

UINT32 numSend

Number of packets sent out.

### 4.29.1 Detailed Description

Table containing particular PD publishing information.

#### 4.29.2 Field Documentation

### 4.29.2.1 destAddr

```
TRDP_IP_ADDR_T TRDP_PUB_STATISTICS_T::destAddr
```

IP address of destination for this publishing.

The documentation for this struct was generated from the following file:

• trdp\_types.h

# 4.30 TRDP\_RED\_STATISTICS\_T Struct Reference

A table containing PD redundant group information.

```
#include <trdp_types.h>
```

#### **Data Fields**

UINT32 id

Redundant Id.

UINT32 state

Redundant state.Leader or Follower.

### 4.30.1 Detailed Description

A table containing PD redundant group information.

The documentation for this struct was generated from the following file:

• trdp\_types.h

# 4.31 TRDP\_SDT\_PAR\_T Struct Reference

Types to read out the XML configuration.

```
#include <tau_xml.h>
```

UINT32 smi1

Safe message identifier - unique for this message at consist level.

UINT32 smi2

Safe message identifier - unique for this message at consist level.

UINT32 cmThr

Channel monitoring threshold.

UINT16 udv

User data version.

UINT16 rxPeriod

Sink cycle time.

UINT16 txPeriod

Source cycle time.

• UINT16 nGuard

Initial timeout cycles.

UINT8 nrxSafe

Timout cycles.

UINT8 reserved1

Reserved for future use.

• UINT16 reserved2

Reserved for future use.

### 4.31.1 Detailed Description

Types to read out the XML configuration.

The documentation for this struct was generated from the following file:

• tau\_xml.h

### 4.32 TRDP\_SEND\_PARAM\_T Struct Reference

Quality/type of service and time to live.

```
#include <trdp_types.h>
```

#### **Data Fields**

• UINT8 qos

Quality of service (default should be 5 for PD and 3 for MD)

• UINT8 ttl

Time to live (default should be 64)

• UINT8 retries

Retries from XML file.

### 4.32.1 Detailed Description

Quality/type of service and time to live.

The documentation for this struct was generated from the following file:

• trdp\_types.h

### 4.33 TRDP\_SEQ\_CNT\_ENTRY\_T Struct Reference

Tuples of last received sequence counter per comld.

```
#include <trdp_private.h>
```

#### **Data Fields**

UINT32 lastSeqCnt

Sequence counter value for comld.

TRDP\_IP\_ADDR\_T srclpAddr

Source IP address.

TRDP\_MSG\_T msgType

message type

### 4.33.1 Detailed Description

Tuples of last received sequence counter per comld.

The documentation for this struct was generated from the following file:

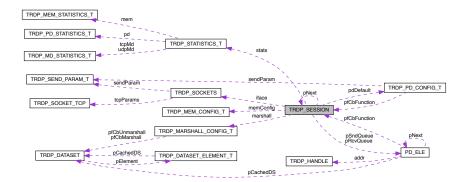
• trdp\_private.h

# 4.34 TRDP\_SESSION Struct Reference

Session/application variables store.

```
#include <trdp_private.h>
```

Collaboration diagram for TRDP\_SESSION:



struct TRDP\_SESSION \* pNext

Pointer to next session.

VOS\_MUTEX\_T mutex

protect this session

TRDP\_IP\_ADDR\_T realIP

Real IP address.

• TRDP IP ADDR T virtualIP

Virtual IP address.

UINT32 etbTopoCnt

current valid topocount or zero

UINT32 opTrnTopoCnt

current valid topocount or zero

TRDP\_TIME\_T nextJob

Store for next select interval.

TRDP\_PRINT\_DBG\_T pPrintDebugString

Pointer to function to print debug information.

• TRDP\_MARSHALL\_CONFIG\_T marshall

Marshalling(unMarshalling configuration.

TRDP\_PD\_CONFIG\_T pdDefault

Default configuration for process data.

TRDP\_MEM\_CONFIG\_T memConfig

Internal memory handling configuration.

• TRDP\_OPTION\_T option

Stack behavior options.

• TRDP\_SOCKETS\_T iface [VOS\_MAX\_SOCKET\_CNT]

Collection of sockets to use.

• PD ELE T \* pSndQueue

pointer to first element of send queue

• PD\_ELE\_T \* pRcvQueue

pointer to first element of rcv queue

PD\_PACKET\_T \* pNewFrame

pointer to received PD frame

• TRDP TIME T initTime

initialization time of session

TRDP\_STATISTICS\_T stats

statistics of this session

#### 4.34.1 Detailed Description

Session/application variables store.

The documentation for this struct was generated from the following file:

• trdp\_private.h

# 4.35 TRDP\_SOCKET\_TCP Struct Reference

#### TCP parameters.

#include <trdp\_private.h>

#### **Data Fields**

· TRDP IP ADDR T cornerlp

The other TCP corner Ip.

BOOL8 notSend

If the message has been sent uncompleted.

TRDP TIME T connectionTimeout

TCP socket connection Timeout.

BOOL8 sendNotOk

The sending timeout will be start.

• TRDP\_TIME\_T sendingTimeout

The timeout sending the message.

• BOOL8 addFileDesc

Ready to add the socket in the fd.

• BOOL8 morituri

about to die

### 4.35.1 Detailed Description

TCP parameters.

The documentation for this struct was generated from the following file:

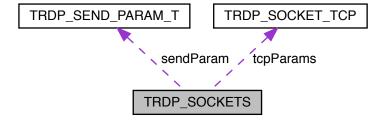
• trdp\_private.h

### 4.36 TRDP SOCKETS Struct Reference

### Socket item.

#include <trdp\_private.h>

Collaboration diagram for TRDP\_SOCKETS:



SOCKET sock

vos socket descriptor to use

TRDP\_IP\_ADDR\_T bindAddr

Defines the interface to use.

• TRDP\_SEND\_PARAM\_T sendParam

Send parameters.

• TRDP\_SOCK\_TYPE\_T type

Usage of this socket.

BOOL8 rcvMostly

Used for receiving.

• INT16 usage

No.

• TRDP\_SOCKET\_TCP\_T tcpParams

Params used for TCP.

• TRDP\_IP\_ADDR\_T mcGroups [VOS\_MAX\_MULTICAST\_CNT]

List of multicast addresses for this socket.

### 4.36.1 Detailed Description

Socket item.

### 4.36.2 Field Documentation

### 4.36.2.1 usage

INT16 TRDP\_SOCKETS::usage

No.

of current users of this socket

The documentation for this struct was generated from the following file:

• trdp\_private.h

# 4.37 TRDP\_STATISTICS\_REQUEST\_T Struct Reference

TRDP statistics type definitions.

#include <trdp\_types.h>

UINT32 comId

ComId to request: 35...41.

#### 4.37.1 Detailed Description

TRDP statistics type definitions.

Statistical data regarding the former info provided via SNMP the following information was left out/can be implemented additionally using MD:

- · PD subscr table: Comld, sourcelpAddr, destlpAddr, cbFct?, timout, toBehavior, counter
- PD publish table: Comld, destlpAddr, redld, redState cycle, ttl, qos, counter
- PD join table: joined MC address table
- · MD listener table: ComId destIpAddr, destUri, cbFct?, counter
- Memory usageStructure containing comld for MD statistics request (Comld 32).

The documentation for this struct was generated from the following file:

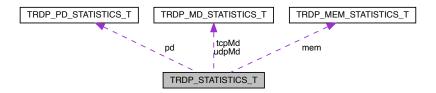
· trdp\_types.h

# 4.38 TRDP\_STATISTICS\_T Struct Reference

Structure containing all general memory, PD and MD statistics information.

```
#include <trdp_types.h>
```

Collaboration diagram for TRDP\_STATISTICS\_T:



UINT32 version

TRDP version.

TIMEDATE64 timeStamp

actual time stamp

• TIMEDATE32 upTime

time in sec since last initialisation

TIMEDATE32 statisticTime

time in sec since last reset of statistics

TRDP\_NET\_LABEL\_T hostName

host name

· TRDP NET LABEL T leaderName

leader host name

· TRDP IP ADDR TownlpAddr

own IP address

TRDP\_IP\_ADDR\_T leaderlpAddr

leader IP address

UINT32 processPrio

priority of TRDP process

• UINT32 processCycle

cycle time of TRDP process in microseconds

UINT32 numJoin

number of joins

UINT32 numRed

number of redundancy groups

TRDP\_MEM\_STATISTICS\_T mem

memory statistics

TRDP\_PD\_STATISTICS\_T pd

pd statistics

• TRDP MD STATISTICS TudpMd

UDP md statistics.

• TRDP\_MD\_STATISTICS\_T tcpMd

TCP md statistics.

## 4.38.1 Detailed Description

Structure containing all general memory, PD and MD statistics information.

The documentation for this struct was generated from the following file:

· trdp types.h

### 4.39 TRDP\_SUBS\_STATISTICS\_T Struct Reference

Table containing particular PD subscription information.

#include <trdp\_types.h>

UINT32 comId

Subscribed Comld.

TRDP\_IP\_ADDR\_T joinedAddr

Joined IP address.

• TRDP IP ADDR T filterAddr

Filter IP address, i.e IP address of the sender for this subscription, 0.0.0.0 in case all senders.

UINT32 callBack

call back function if used

UINT32 userRef

User reference if used.

UINT32 timeout

Time-out value in us.

· TRDP ERR T status

Receive status information TRDP\_NO\_ERR, TRDP\_TIMEOUT\_ERR.

UINT32 toBehav

Behavior at time-out.

UINT32 numRecv

Number of packets received for this subscription.

UINT32 numMissed

number of packets skipped for this subscription

### 4.39.1 Detailed Description

Table containing particular PD subscription information.

#### 4.39.2 Field Documentation

### 4.39.2.1 filterAddr

```
TRDP_IP_ADDR_T TRDP_SUBS_STATISTICS_T::filterAddr
```

Filter IP address, i.e IP address of the sender for this subscription, 0.0.0.0 in case all senders.

#### 4.39.2.2 timeout

UINT32 TRDP\_SUBS\_STATISTICS\_T::timeout

Time-out value in us.

### 0 = No time-out supervision

#### 4.39.2.3 toBehav

UINT32 TRDP\_SUBS\_STATISTICS\_T::toBehav

Behavior at time-out.

Set data to zero / keep last value

The documentation for this struct was generated from the following file:

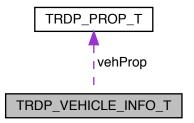
• trdp\_types.h

### 4.40 TRDP\_VEHICLE\_INFO\_T Struct Reference

vehicle information structure

#include <tau\_tti\_types.h>

Collaboration diagram for TRDP\_VEHICLE\_INFO\_T:



#### **Data Fields**

• TRDP\_NET\_LABEL\_T vehId

vehicle identifier label, application defined (e.g.

TRDP\_NET\_LABEL\_T vehType

vehicle type,application defined

UINT8 vehOrient

vehicle orientation '01'B = same as consist direction '10'B = inverse to consist direction

UINT8 cstVehNo

Sequence number of vehicle in consist(1..16)

ANTIVALENT8 tractVeh

vehicle is a traction vehicle '01'B = vehicle is not a traction vehicle '10'B = vehicle is a traction vehicle

UINT8 reserved01

for future use (= 0)

• TRDP\_PROP\_T vehProp

static vehicle properties

### 4.40.1 Detailed Description

vehicle information structure

#### 4.40.2 Field Documentation

#### 4.40.2.1 vehld

```
TRDP_NET_LABEL_T TRDP_VEHICLE_INFO_T::vehid
```

vehicle identifier label, application defined (e.g.

UIC vehicle identification number) vehId of vehicle with vehNo==1 is used also as cstId

The documentation for this struct was generated from the following file:

• tau\_tti\_types.h

# 4.41 TRDP\_XML\_DOC\_HANDLE\_T Struct Reference

Parsed XML document handle.

```
#include <tau_xml.h>
```

#### **Data Fields**

struct XML\_HANDLE \* pXmlDocument
 XML document context.

### 4.41.1 Detailed Description

Parsed XML document handle.

The documentation for this struct was generated from the following file:

• tau\_xml.h

# 4.42 VOS\_SOCK\_OPT\_T Struct Reference

Common socket options.

```
#include <vos_sock.h>
```

• UINT8 qos

quality/type of service 0...7

UINT8 ttl

time to live for unicast (default 64)

UINT8 ttl\_multicast

time to live for multicast

· BOOL8 reuseAddrPort

allow reuse of address and port

BOOL8 nonBlocking

use non blocking calls

• BOOL8 no\_mc\_loop

no multicast loop back

BOOL8 no\_udp\_crc

supress udp crc computation

#### 4.42.1 Detailed Description

Common socket options.

The documentation for this struct was generated from the following file:

vos\_sock.h

# 4.43 VOS\_VERSION\_T Struct Reference

Version information.

```
#include <vos_types.h>
```

#### **Data Fields**

UINT8 ver

Version - incremented for incompatible changes.

• UINT8 rel

Release - incremented for compatible changes.

• UINT8 upd

Update - incremented for bug fixes.

UINT8 evo

Evolution - incremented for build.

### 4.43.1 Detailed Description

Version information.

The documentation for this struct was generated from the following file:

vos\_types.h

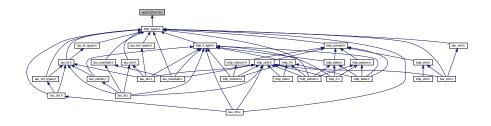
# **Chapter 5**

# **File Documentation**

### 5.1 iec61375-2-3.h File Reference

All definitions from IEC 61375-2-3.

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



### **Macros**

- #define ETB\_WAIT\_TIMER\_VALUE 5u /\* Compute train dir. IEC61375-2-3 Ch. 5.3.2.3 \*/
   Time out values (in seconds)
- #define TRDP\_PD\_UDP\_PORT 17224u

TRDP defines (from former trpd\_proto.h)

• #define TRDP\_MD\_UDP\_PORT 17225u

IANA assigned message data UDP port.

• #define TRDP\_MD\_TCP\_PORT 17225u

IANA assigned message data TCP port.

• #define TRDP\_PROTO\_VER 0x0100u

Protocol version.

• #define TRDP\_PROTOCOL\_VERSION\_CHECK\_MASK 0xFF00u

Version check, two digits are relevant.

• #define TRDP\_SESS\_ID\_SIZE 16u

Session ID (UUID) size in MD header.

• #define TRDP\_USR\_URI\_SIZE 32u

max

• #define TRDP\_MD\_INFINITE\_TIME (0)

Definitions for time out behaviour accd.

#define TRDP\_MD\_DEFAULT\_REPLY\_TIMEOUT 5000000u

Default MD communication parameters.

#define TRDP MD DEFAULT CONFIRM TIMEOUT 1000000u

[us] default confirm time out 1s

• #define TRDP MD DEFAULT CONNECTION TIMEOUT 60000000u

[us] Socket connection time out 1min

#define TRDP MD DEFAULT SENDING TIMEOUT 5000000u

[us] Socket sending time out 5s

#define TRDP\_PD\_DEFAULT\_QOS 5u

Default PD communication parameters.

• #define TRDP\_PD\_DEFAULT\_TIMEOUT 100000u

[us] 100ms default PD timeout

• #define TRDP\_PROCESS\_DEFAULT\_CYCLE\_TIME 10000u

Default TRDP process options.

#define TRDP PROCESS DEFAULT PRIORITY 64u

Default priority of TRDP process.

#define TRDP PROCESS DEFAULT OPTIONS TRDP OPTION TRAFFIC SHAPING

Default options for TRDP process.

#define TRDP MIN PD HEADER SIZE sizeof(PD HEADER T)

PD packet properties.

• #define TRDP\_MAX\_PD\_DATA\_SIZE 1432u

PD data

#define TRDP\_MAX\_MD\_DATA\_SIZE 65388u

MD packet properties.

• #define TRDP\_MAX\_MD\_RETRIES 2u

Maximum values.

#define TRDP\_MAX\_LABEL\_LEN 16u

label length incl.

• #define TRDP MAX URI USER LEN (2u \* TRDP MAX LABEL LEN)

URI user part excl.

#define TRDP\_MAX\_URI\_HOST\_LEN (4u \* TRDP\_MAX\_LABEL\_LEN)

URI host part length excl.

• #define TRDP\_MAX\_URI\_LEN ((6u \* TRDP\_MAX\_LABEL\_LEN) + 8u)

URI length excl.

• #define TRDP\_MAX\_FILE\_NAME\_LEN 128u

path and file name length incl.

• #define TRDP\_VAR\_SIZE 0u

Variable size dataset.

#define TRDP\_MSG\_PD 0x5064u

Message Types.

• #define TRDP\_MSG PP 0x5070u

'Pp' PD Data (Pull Reply)

#define TRDP\_MSG\_PR 0x5072u

'Pr' PD Reauest

#define TRDP\_MSG\_PE 0x5065u

'Pe' PD Error

• #define TRDP\_MSG\_MN 0x4D6Eu

'Mn' MD Notification (Request w/o reply)

#define TRDP\_MSG\_MR 0x4D72u

'Mr' MD Request with reply

```
    #define TRDP_MSG_MP 0x4D70u

     'Mp' MD Reply without confirmation
• #define TRDP MSG MQ 0x4D71u
     'Mq' MD Reply with confirmation
• #define TRDP_MSG_MC 0x4D63u
     'Mc' MD Confirm
• #define TRDP MSG ME 0x4D65u
     'Me' MD Error

    #define ETB0 ALL END DEVICES IP "239.193.0.0"

     from Table 22
• #define ETB CTRL COMID 1u
    Reserved COMIDs in the range 1 ...

    #define ETB_CTRL_CYC 500u

    [ms] 0.5s

    #define ETB CTRL TO 3000u

    [ms]3s

    #define TRDP_ETBCTRL_COMID ETB_CTRL_COMID

     alternative name
• #define CSTINFO COMID 2u
     Consist Info telegram (Message data notification 'Mn')

    #define TRDP_CSTINFO_COMID CSTINFO_COMID

    alternative name

    #define CSTINFOCTRL COMID 3u

     Consist Info control/request telegram (Message data notification 'Mn')

    #define TRDP_CSTINFOCTRL_COMID CSTINFOCTRL_COMID

    alternative name
• #define TRDP COMID ECHO 10u
     Reserved in Annex D & E.
• #define TTDB_STATUS_COMID 100u
     TTDB manager telegram PD.

    #define TTDB_STATUS_CYC 1000u

    [ms] Push
• #define TTDB_STATUS_TO 5000u
    [ms] 5s

    #define TTDB OP DIR INFO COMID 101u

     TTDB manager telegram MD: Push the OP_TRAIN_DIRECTORY.

    #define TTDB OP DIR INFO DS "TTDB OP TRAIN DIRECTORY INFO"

     OP TRAIN DIRECTORY.

    #define TTDB_TRN_DIR_REQ_COMID 102u

     TTDB manager telegram MD: Get the TRAIN_DIRECTORY.

    #define TTDB TRN DIR REQ TO 3000u

     3s timeout

    #define TTDB_TRN_DIR_REP_COMID 103u

     MD reply.
• #define TTDB_TRN_DIR_REP_DS "TTDB_TRAIN_DIRECTORY_INFO_REPLY"
     TRAIN_DIRECTORY.

    #define TTDB_STAT_CST_REQ_COMID 104u

     TTDB manager telegram MD: Get the static consist information.

    #define TTDB STAT CST REQ TO 3000u

    [ms] 3s timeout

    #define TTDB_STAT_CST_REP_DS "TTDB_STATIC_CONSIST_INFO_REPLY"
```

```
CONSIST INFO.

    #define TTDB_NET_DIR_REQ_COMID 106u

     TTDB manager telegram MD: Get the NETWORK_TRAIN_DIRECTORY.

    #define TTDB NET DIR REQ TO 3000u

    [ms] 3s timeout

    #define TTDB NET DIR REP COMID 107u

    MD reply.

    #define TTDB NET DIR REP DS "TTDB TRAIN NETWORK DIRECTORY INFO REPLY"

     TRAIN NETWORK DIRECTORY.

    #define TTDB_OP_DIR_INFO_REQ_COMID 108u

     TTDB manager telegram MD: Get the OP TRAIN DIRECTORY.
• #define TTDB_OP_DIR_INFO_REQ_TO 3000u
    [ms] 3s timeout
• #define TTDB_OP_DIR_INFO_REP_DS "TTDB_OP_TRAIN_DIR_INFO"
    OP_TRAIN_DIRECTORY.
• #define TTDB READ CMPLT REQ COMID 110u
     TTDB manager telegram MD: Get the TTDB.

    #define TTDB_READ_CMPLT_REQ_DS "TTDB_READ_COMPLETE_REQUEST"

    FTRx

    #define TTDB READ CMPLT REQ TO 3000u

    [ms] 3s timeout

    #define TTDB_READ_CMPLT_REP_COMID 111u

    MD reply.

    #define TTDB_READ_CMPLT_REP_DS "TTDB_READ_COMPLETE_REPLY"

     TRDP_READ_COMPLETE_REPLY_T.
• #define ECSP_CTRL_COMID 120u
    ECSP Control telegram.

    #define ECSP_CTRL_CYC 1000u

    [ms] 1s
• #define ECSP CTRL TO 5000u
    [ms] 5s

    #define ECSP_CTRL_DEST_URI "devECSP.anyVeh.ICst.ICITrn.ITrn"

    #define TRDP ECSP CTRL COMID ECSP CTRL COMID

    Etb control message.

    #define ECSP_STATUS_COMID 121u

    ECSP status telegram.
• #define ECSP_STATUS_CYC 1000u
    [ms] 1s

    #define ECSP_STATUS_TO 5000u

    #define ECSP STATUS DEST URI "devECSC.anyVeh.ICst.ICITrn.ITrn"

     10.0.0.100

    #define ECSP CONF REQ COMID 122u

    ECSP Confirmation Request telegram MD:
• #define ECSP_CONF_REQ_TO 3000u

    #define ECSP_CONF_REQ_URI "devECSP.anyVeh.ICst.ICITrn.ITrn"

     10.0.0.1
• #define ECSP_CONF_REP_TO 3000u
```

[ms]

```
    #define ETBN_CTRL_REQ_COMID 130u

     ETBN Control & Status Telegram MD.

    #define ETBN CTRL REQ DS "ETBN CTRL"

    #define ETBN CTRL REQ TO 3000u

    [ms] 3s timeout

    #define ETBN_CTRL_REP_DS "ETBN_STATUS"
```

ETBN status reply.

#define ETBN\_TRN\_NET\_DIR\_REQ\_COMID 132u

ETBN Control Telegram MD.

#define ETBN\_TRN\_NET\_DIR\_REQ\_TO 3000u

[ms] 3s timeout

• #define TCN\_DNS\_REQ\_COMID 140u

TCN-DNS Request Telegram MD.

#define TCN\_DNS\_REQ\_TO 3000u

[ms] 3s timeout

#define TCN\_DNS\_REQ\_TO\_US 3000000u

[us] 3s timeout

#define TRDP\_ETBCTRL\_DSID 1u

TRDP reserved data set ids in the range 1 ...

#### 5.1.1 Detailed Description

All definitions from IEC 61375-2-3.

Note

Project: TCNOpen TRDP

**Author** 

Bernd Loehr, NewTec GmbH, 2015-09-11

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

ld

### iec61375-2-3.h 1755 2018-08-07 12:10:03Z bloehr

```
BL 2018-01-29: Ticket #188 Typo in the TRDP_VAR_SIZE definition
AHW 2017-11-05: Ticket #179 Max. number of retries of a MD request needs to be checked
AHW 2017-05-22: Ticket #159 Infinit timeout at TRDB level is 0 acc. standard
BL 2017-04-28: Ticket #155: Kill trdp_proto.h - move definitions to iec61375-2-3.h
 BL 2017-02-08: Ticket #142: Compiler warnings / MISRA-C 2012 issues
BL 2016-05-04: Ticket #118: Fix defines to match IEC IS 2015
```

#### from trdp proto.h

```
BL 2017-03-13: Ticket #154 ComIds and DSIds literals (#define TRDP_...) in trdp_proto.h too long
BL 2017-03-01: Ticket #149 SourceUri and DestinationUri don't with 32 characters
BL 2017-02-08: Ticket #142: Compiler warnings / MISRA-C 2012 issues
BL 2016-11-09: Default PD/MD parameter defines moved from trdp_private.h
BL 2016-06-08: Ticket #120: ComIds for statistics changed to proposed 61375 errata
BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed
```

### 5.1.2 Macro Definition Documentation

```
5.1.2.1 ETB_CTRL_COMID
#define ETB_CTRL_COMID 1u
Reserved COMIDs in the range 1 ...
1000 ETB Control telegram
5.1.2.2 TRDP_ETBCTRL_DSID
#define TRDP_ETBCTRL_DSID 1u
TRDP reserved data set ids in the range 1 ...
1000
5.1.2.3 TRDP_MAX_FILE_NAME_LEN
#define TRDP_MAX_FILE_NAME_LEN 128u
path and file name length incl.
terminating '0'
5.1.2.4 TRDP_MAX_LABEL_LEN
#define TRDP_MAX_LABEL_LEN 16u
label length incl.
terminating '0'
5.1.2.5 TRDP_MAX_MD_DATA_SIZE
```

MD payload size

MD packet properties.

#define TRDP\_MAX\_MD\_DATA\_SIZE 65388u

#### 5.1.2.6 TRDP\_MAX\_URI\_HOST\_LEN

#define TRDP\_MAX\_URI\_HOST\_LEN (4u \* TRDP\_MAX\_LABEL\_LEN)

URI host part length excl.

terminating '0'

#### 5.1.2.7 TRDP\_MAX\_URI\_LEN

```
#define TRDP_MAX_URI_LEN ((6u * TRDP_MAX_LABEL_LEN) + 8u)
```

URI length excl.

terminating '0' and 1 padding byte

#### 5.1.2.8 TRDP\_MAX\_URI\_USER\_LEN

```
#define TRDP_MAX_URI_USER_LEN (2u * TRDP_MAX_LABEL_LEN)
```

URI user part excl.

terminating '0'

### 5.1.2.9 TRDP\_MD\_DEFAULT\_REPLY\_TIMEOUT

```
#define TRDP_MD_DEFAULT_REPLY_TIMEOUT 5000000u
```

Default MD communication parameters.

[us] default reply timeout 5s

### 5.1.2.10 TRDP\_MD\_INFINITE\_TIME

```
#define TRDP_MD_INFINITE_TIME (0)
```

Definitions for time out behaviour accd.

table A.18

### 5.1.2.11 TRDP\_MIN\_PD\_HEADER\_SIZE

```
#define TRDP_MIN_PD_HEADER_SIZE sizeof(PD_HEADER_T)
```

PD packet properties.

PD header size with FCS

```
#define TRDP_MSG_PD 0x5064u
```

5.1.2.12 TRDP\_MSG\_PD

Message Types.

'Pd' PD Data

5.1.2.13 TRDP\_PD\_UDP\_PORT

#define TRDP\_PD\_UDP\_PORT 17224u

TRDP defines (from former trpd\_proto.h)

IANA assigned process data UDP port

5.1.2.14 TRDP\_PROCESS\_DEFAULT\_CYCLE\_TIME

#define TRDP\_PROCESS\_DEFAULT\_CYCLE\_TIME 10000u

Default TRDP process options.

[us] 10ms cycle time for TRDP process

5.1.2.15 TRDP\_USR\_URI\_SIZE

#define TRDP\_USR\_URI\_SIZE 32u

max.

User URI size in MD header

5.1.2.16 TTDB\_NET\_DIR\_REQ\_COMID

#define TTDB\_NET\_DIR\_REQ\_COMID 106u

TTDB manager telegram MD: Get the NETWORK\_TRAIN\_DIRECTORY.

MD request

5.1.2.17 TTDB\_OP\_DIR\_INFO\_COMID

#define TTDB\_OP\_DIR\_INFO\_COMID 101u

TTDB manager telegram MD: Push the OP\_TRAIN\_DIRECTORY.

MD notification

#### 5.1.2.18 TTDB\_STAT\_CST\_REQ\_COMID

```
#define TTDB_STAT_CST_REQ_COMID 104u
```

TTDB manager telegram MD: Get the static consist information.

MD request

#### 5.1.2.19 TTDB TRN DIR REQ COMID

```
#define TTDB_TRN_DIR_REQ_COMID 102u
```

TTDB manager telegram MD: Get the TRAIN\_DIRECTORY.

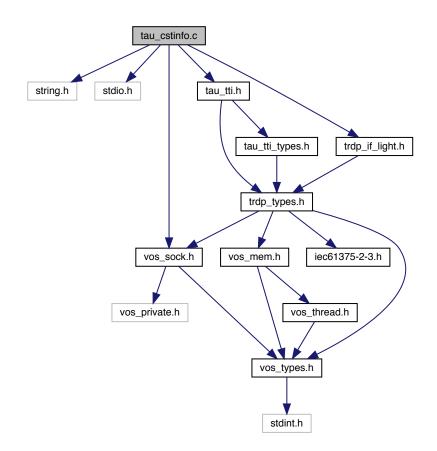
MD request

# 5.2 tau\_cstinfo.c File Reference

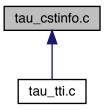
Functions for consist information access.

```
#include <string.h>
#include <stdio.h>
#include "trdp_if_light.h"
#include "tau_tti.h"
#include "vos sock.h"
```

Include dependency graph for tau\_cstinfo.c:



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



#### **Functions**

• UINT16 cstInfoGetPropSize (TRDP\_CONSIST\_INFO\_T \*pCstInfo)

Getter function to retrieve a value from the consist info telegram value.

### 5.2.1 Detailed Description

Functions for consist information access.

Note

Project: TCNOpen TRDP prototype stack

Author

B. Loehr (initial version)

### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2015. All rights reserved.

ld

tau\_cstinfo.c 1646 2017-07-05 14:34:41Z bloehr

```
BL 2017-05-08: Compiler warnings, doxygen comment errors
BL 2017-04-28: Ticket #155: Kill trdp_proto.h - move definitions to iec61375-2-3.h
BL 2016-02-24: C89 compatibility (thanks to Robert)
```

### 5.2.2 Function Documentation

### 5.2.2.1 cstInfoGetPropSize()

Getter function to retrieve a value from the consist info telegram value.

### **Parameters**

#### **Return values**



Here is the call graph for this function:

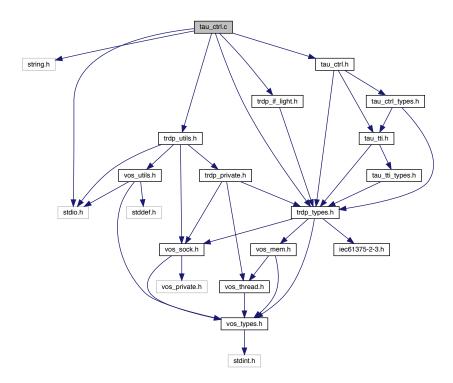


# 5.3 tau\_ctrl.c File Reference

Functions for train switch control.

```
#include <string.h>
#include <stdio.h>
#include "trdp_types.h"
#include "trdp_utils.h"
#include "trdp_if_light.h"
#include "tau_ctrl.h"
```

Include dependency graph for tau\_ctrl.c:



#### **Functions**

• EXT\_DECL\_TRDP\_ERR\_T\_tau\_initEcspCtrl (TRDP\_APP\_SESSION\_T\_appHandle, TRDP\_IP\_ADDR\_← T\_ecsplpAddr)

Function to init ECSP control interface.

• EXT\_DECL TRDP\_ERR\_T tau\_terminateEcspCtrl (TRDP\_APP\_SESSION\_T appHandle)

Function to close ECSP control interface.

• EXT\_DECL TRDP\_ERR\_T tau\_setEcspCtrl (TRDP\_APP\_SESSION\_T appHandle, TRDP\_ECSP\_CTRL\_T \*pEcspCtrl)

Function to set ECSP control information.

• EXT\_DECL TRDP\_ERR\_T tau\_getEcspStat (TRDP\_APP\_SESSION\_T appHandle, TRDP\_ECSP\_STAT\_T \*pEcspStat, TRDP\_PD\_INFO\_T \*pPdInfo)

Function to get ECSP status information.

• EXT\_DECL TRDP\_ERR\_T tau\_requestEcspConfirm (TRDP\_APP\_SESSION\_T appHandle, const void \*pUserRef, TRDP\_MD\_CALLBACK\_T pfCbFunction, TRDP\_ECSP\_CONF\_REQUEST\_T \*pEcspConf ← Request)

Function for ECSP confirmation/correction request, reply will be received via call back.

#### 5.3.1 Detailed Description

Functions for train switch control.

Note

Project: TCNOpen TRDP prototype stack

#### **Author**

Armin-H. Weiss (initial version)

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

tau\_ctrl.c 1711 2018-03-06 16:11:32Z bloehr

```
BL 2018-03-06: Ticket #101 Optional callback function on PD send
AHW 2017-11-08: Ticket #179 Max. number of retries (part of sendParam) of a MD request needs to be checked
BL 2017-04-28: Ticket #155: Kill trdp_proto.h - move definitions to iec61375-2-3.h
```

#### 5.3.2 Function Documentation

#### 5.3.2.1 tau\_getEcspStat()

```
EXT_DECL TRDP_ERR_T tau_getEcspStat (

TRDP_APP_SESSION_T appHandle,

TRDP_ECSP_STAT_T * pEcspStat,

TRDP_PD_INFO_T * pPdInfo )
```

Function to get ECSP status information.

#### **Parameters**

| in     | appHandle | Application handle                   |
|--------|-----------|--------------------------------------|
| in,out | pEcspStat | Pointer to the ECSP status structure |
| in,out | pPdInfo   | Pointer to PD status information     |

### Return values

| TRDP_NO_ERR     | no error               |
|-----------------|------------------------|
| TRDP_NOINIT_ERR | module not initialised |
| TRDP_PARAM_ERR  | Parameter error        |

#### 5.3.2.2 tau\_initEcspCtrl()

Function to init ECSP control interface.

#### **Parameters**

| in | appHandle  | Application handle |
|----|------------|--------------------|
| in | ecsplpAddr | ECSP address       |

#### Return values

| TRDP_NO_ERR   | no error             |
|---------------|----------------------|
| TRDP_INIT_ERR | initialisation error |

#### 5.3.2.3 tau\_requestEcspConfirm()

Function for ECSP confirmation/correction request, reply will be received via call back.

#### **Parameters**

| in | appHandle        | Application Handle                             |
|----|------------------|--|
| in | pUserRef         | user reference returned with reply             |
| in | pfCbFunction     | Pointer to callback function, NULL for default |
| in | pEcspConfRequest | Pointer to confirmation data                   |

#### **Return values**

| TRDP_NO_ERR     | no error               |
|-----------------|------------------------|
| TRDP_NOINIT_ERR | module not initialised |
| TRDP_PARAM_ERR  | Parameter error        |

#### 5.3.2.4 tau\_setEcspCtrl()

Function to set ECSP control information.

#### **Parameters**

| in | appHandle | Application handle                    |
|----|-----------|---------------------------------------|
| in | pEcspCtrl | Pointer to the ECSP control structure |

### Return values

| TRDP_NO_ERR     | no error               |
|-----------------|------------------------|
| TRDP_NOINIT_ERR | module not initialised |
| TRDP_PARAM_ERR  | Parameter error        |

### 5.3.2.5 tau\_terminateEcspCtrl()

Function to close ECSP control interface.

#### **Parameters**

| in appHandle | Application handle |
|--------------|--------------------|
|--------------|--------------------|

### Return values

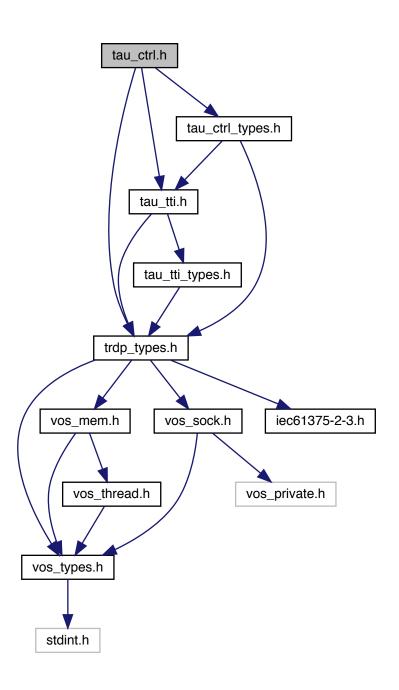
| TRDP_NO_ERR      | no error               |
|------------------|------------------------|
| TRDP_NOINIT_ERR  | module not initialised |
| TRDP_UNKNOWN_ERR | undefined error        |

# 5.4 tau\_ctrl.h File Reference

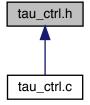
TRDP utility interface definitions.

```
#include "trdp_types.h"
#include "tau_tti.h"
#include "tau_ctrl_types.h"
```

Include dependency graph for tau\_ctrl.h:



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



#### **Functions**

 EXT\_DECL\_TRDP\_ERR\_T tau\_initEcspCtrl (TRDP\_APP\_SESSION\_T appHandle, TRDP\_IP\_ADDR\_← T ecsplpAddr)

Function to init ECSP control interface.

- EXT\_DECL TRDP\_ERR\_T tau\_terminateEcspCtrl (TRDP\_APP\_SESSION\_T appHandle) Function to close ECSP control interface.
- EXT\_DECL TRDP\_ERR\_T tau\_setEcspCtrl (TRDP\_APP\_SESSION\_T appHandle, TRDP\_ECSP\_CTRL\_T \*pEcspCtrl)

Function to set ECSP control information.

• EXT\_DECL TRDP\_ERR\_T tau\_getEcspStat (TRDP\_APP\_SESSION\_T appHandle, TRDP\_ECSP\_STAT\_T \*pEcspStat, TRDP\_PD\_INFO\_T \*pPdInfo)

Function to get ECSP status information.

• EXT\_DECL TRDP\_ERR\_T tau\_requestEcspConfirm (TRDP\_APP\_SESSION\_T appHandle, const void \*pUserRef, TRDP\_MD\_CALLBACK\_T pfCbFunction, TRDP\_ECSP\_CONF\_REQUEST\_T \*pEcspConf← Request)

Function for ECSP confirmation/correction request, reply will be received via call back.

#### 5.4.1 Detailed Description

TRDP utility interface definitions.

This module provides the interface to the following utilities

· ETB control

Note

Project: TCNOpen TRDP prototype stack

Author

Armin-H. Weiss (initial version)

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

tau\_ctrl.h 1483 2015-12-16 14:43:30Z bloehr

### 5.4.2 Function Documentation

### 5.4.2.1 tau\_getEcspStat()

Function to get ECSP status information.

#### **Parameters**

| in     | appHandle | Application Handle                   |
|--------|-----------|--------------------------------------|
| in,out | pEcspStat | Pointer to the ECSP status structure |
| in,out | pPdInfo   | Pointer to PD status information     |

#### Return values

| TRDP_NO_ERR     | no error               |
|-----------------|------------------------|
| TRDP_NOINIT_ERR | module not initialised |
| TRDP_PARAM_ERR  | Parameter error        |

#### **Parameters**

| in     | appHandle | Application handle                   |
|--------|-----------|--------------------------------------|
| in,out | pEcspStat | Pointer to the ECSP status structure |
| in,out | pPdInfo   | Pointer to PD status information     |

### Return values

| TRDP_NO_ERR     | no error               |
|-----------------|------------------------|
| TRDP_NOINIT_ERR | module not initialised |
| TRDP_PARAM_ERR  | Parameter error        |

### 5.4.2.2 tau\_initEcspCtrl()

Function to init ECSP control interface.

### **Parameters**

| in | appHandle  | Application handle |
|----|------------|--------------------|
| in | ecsplpAddr | ECSP address       |

#### Return values

| TRDP_NO_ERR   | no error             |
|---------------|----------------------|
| TRDP_INIT_ERR | initialisation error |

### 5.4.2.3 tau\_requestEcspConfirm()

Function for ECSP confirmation/correction request, reply will be received via call back.

#### **Parameters**

| in | appHandle        | Application Handle                             |
|----|------------------|--|
| in | pUserRef         | user reference returned with reply             |
| in | pfCbFunction     | Pointer to callback function, NULL for default |
| in | pEcspConfRequest | Pointer to confirmation data                   |

#### Return values

| TRDP_NO_ERR     | no error               |
|-----------------|------------------------|
| TRDP_NOINIT_ERR | module not initialised |
| TRDP_PARAM_ERR  | Parameter error        |

### 5.4.2.4 tau\_setEcspCtrl()

```
EXT_DECL TRDP_ERR_T tau_setEcspCtrl (

TRDP_APP_SESSION_T appHandle,

TRDP_ECSP_CTRL_T * pEcspCtrl )
```

Function to set ECSP control information.

#### **Parameters**

| in | appHandle | Application handle                    |
|----|-----------|---------------------------------------|
| in | pEcspCtrl | Pointer to the ECSP control structure |

#### Return values

| TRDP_NO_ERR     | no error               |
|-----------------|------------------------|
| TRDP_NOINIT_ERR | module not initialised |
| TRDP_PARAM_ERR  | Parameter error        |

#### 5.4.2.5 tau\_terminateEcspCtrl()

Function to close ECSP control interface.

#### **Parameters**

| in | appHandle | Application handle |
|----|-----------|--------------------|
|----|-----------|--------------------|

#### Return values

| TRDP_NO_ERR      | no error        |
|------------------|-----------------|
| TRDP_UNKNOWN_ERR | undefined error |

### **Parameters**

| in | appHandle | Application handle |
|----|-----------|--------------------|
|----|-----------|--------------------|

#### **Return values**

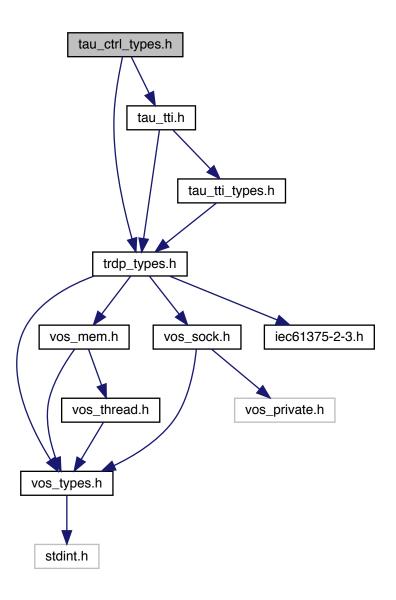
| TRDP_NO_ERR      | no error               |
|------------------|------------------------|
| TRDP_NOINIT_ERR  | module not initialised |
| TRDP_UNKNOWN_ERR | undefined error        |

# 5.5 tau\_ctrl\_types.h File Reference

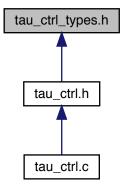
TRDP utility interface definitions.

```
#include "trdp_types.h"
#include "tau_tti.h"
```

Include dependency graph for tau\_ctrl\_types.h:



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



#### **Data Structures**

struct GNU\_PACKED

Types for ETB control.

• struct GNU\_PACKED

Types for ETB control.
• struct GNU\_PACKED

Types for ETB control.

• struct GNU\_PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

### 5.5.1 Detailed Description

TRDP utility interface definitions.

This module provides the interface to the following

• ETB control type definitions acc. to IEC61375-2-3

Note

Project: TCNOpen TRDP prototype stack

#### **Author**

Armin-H. Weiss (initial version)

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

tau\_ctrl\_types.h 1763 2018-09-21 16:03:13Z ahweiss

```
BL 2017-11-13: Ticket #176 TRDP_LABEL_T breaks field alignment -> TRDP_NET_LABEL_T
BL 2017-03-09: Ticket #131 errata: Last change undone (leadVehOfCst defined in corrigendum)
BL 2017-02-09: Ticket #131 TRDP_ECSP_CTRL_T in the tau_ctrl_types.h should not include the leadVehOfCst field
```

# 5.6 tau\_dnr.c File Reference

Functions for domain name resolution.

```
#include <string.h>
#include <stdio.h>
#include <ctype.h>
#include "tau_tti.h"
#include "tau_dnr.h"
#include "tau_dnr_types.h"
#include "trdp_utils.h"
#include "trdp_if_light.h"
#include "vos_mem.h"
#include "vos_sock.h"
Include dependency graph for tau_dnr.c:
```

string h trdp\_utilis.h trdp\_private.h tau\_dnr.h tau\_dnr.h tau\_ttl.light.h tau\_

#### **Data Structures**

struct DNS HEADER

DNS header structure.

#### **Macros**

• #define TAU\_MAX\_NO\_IF 4u

Default interface should be in the first 4.

• #define TAU\_DNS\_TIME\_OUT\_LONG 10u

Timeout in seconds for DNS server reply, if no hosts file provided.

• #define TAU\_DNS\_TIME\_OUT\_SHORT 1u

Timeout in seconds for DNS server reply, if hosts file was provided.

• #define TAU\_MAX\_HOST\_URI\_LEN 80u

Including EOS!

### **Typedefs**

 typedef struct DNS\_HEADER TAU\_DNS\_HEADER\_T DNS header structure.

### **Functions**

• EXT\_DECL TRDP\_ERR\_T tau\_initDnr (TRDP\_APP\_SESSION\_T appHandle, TRDP\_IP\_ADDR\_T dnsIp← Addr, UINT16 dnsPort, const CHAR8 \*pHostsFileName, TRDP\_DNR\_OPTS\_T dnsOptions)

Function to init the DNR subsystem Initialize the DNR resolver.

• EXT\_DECL void tau\_deInitDnr (TRDP\_APP\_SESSION\_T appHandle)

Function to deinit DNR.

- EXT\_DECL TRDP\_DNR\_STATE\_T tau\_DNRstatus (TRDP\_APP\_SESSION\_T appHandle)

  Function to get the status of DNR.
- EXT\_DECL TRDP\_IP\_ADDR\_T tau\_getOwnAddr (TRDP\_APP\_SESSION\_T appHandle)

Function to get the own IP address.

EXT\_DECL TRDP\_ERR\_T tau\_uri2Addr (TRDP\_APP\_SESSION\_T appHandle, TRDP\_IP\_ADDR\_T \*p
 — Addr, const TRDP\_URI\_T pUri)

Function to convert a URI to an IP address.

 EXT\_DECL TRDP\_ERR\_T tau\_addr2Uri (TRDP\_APP\_SESSION\_T appHandle, TRDP\_URI\_HOST\_T pUri, TRDP\_IP\_ADDR\_T addr)

Function to convert an IP address to a URI.

### 5.6.1 Detailed Description

Functions for domain name resolution.

Note

Project: TCNOpen TRDP prototype stack

#### **Author**

B. Loehr (initial version)

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

tau\_dnr.c 1763 2018-09-21 16:03:13Z ahweiss

```
BL 2018-08-07: Ticket #183 tau_getOwnIds declared but not defined
BL 2018-08-06: Ticket #210 IF condition for DNS Options incorrect in tau_uri2Addr()
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2018-05-03: Ticket #193 Unused parameter warnings
AHW 2017-11-08: Ticket #179 Max. number of retries (part of sendParam) of a MD request needs to be checked
BL 2017-07-25: Ticket #125: tau_dnr: TCN DNS support missing
BL 2017-05-08: Compiler warnings
BL 2017-03-01: Ticket #149 SourceUri and DestinationUri don't with 32 characters
BL 2017-02-08: Ticket #124 tau_dnr: Cache keeps etbTopoCount only
BL 2015-12-14: Ticket #8: DNR client
```

#### 5.6.2 Function Documentation

#### 5.6.2.1 tau\_addr2Uri()

Function to convert an IP address to a URI.

Receives an IP-Address and translates it into the host part of the corresponding URI. Both unicast and multicast addresses are accepted.

#### **Parameters**

|   | in  | appHandle | Handle returned by tlc_openSession()            |
|---|-----|-----------|---|
| ſ | out | pUri      | Pointer to a string to return the URI host part |
|   | in  | addr      | IP address, 0==own address                      |

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

### 5.6.2.2 tau\_delnitDnr()

Function to deinit DNR.

Release any resources allocated by DNR.

#### **Parameters**

| in | appHandle | Handle returned by tlc_openSession() |
|----|-----------|--------------------------------------|
|----|-----------|--------------------------------------|

#### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

# 5.6.2.3 tau\_DNRstatus()

Function to get the status of DNR.

#### **Parameters**

|  | in | appHandle | Handle returned by tlc_openSession() |
|--|----|-----------|--------------------------------------|
|--|----|-----------|--------------------------------------|

### Return values

| TRDP_DNR_NOT_AVAILABLE | no error                              |
|------------------------|---------------------------------------|
| TRDP_DNR_UNKNOWN       | enabled, but cache is empty           |
| TRDP_DNR_ACTIVE        | enabled, cache has values             |
| TRDP_DNR_HOSTSFILE     | enabled, hostsfile used (static mode) |

# 5.6.2.4 tau\_getOwnAddr()

Function to get the own IP address.

Returns the IP address set by openSession. If it was 0 (INADDR\_ANY), the address of the default adapter will be returned.

#### **Parameters**

| in | appHandle | Handle returned by tlc_openSession() |
|----|-----------|--------------------------------------|
|----|-----------|--------------------------------------|

#### Return values

```
own IP address
```

# 5.6.2.5 tau\_initDnr()

```
EXT_DECL TRDP_ERR_T tau_initDnr (

TRDP_APP_SESSION_T appHandle,

TRDP_IP_ADDR_T dnsIpAddr,

UINT16 dnsPort,

const CHAR8 * pHostsFileName,

TRDP_DNR_OPTS_T dnsOptions )
```

Function to init the DNR subsystem Initialize the DNR resolver.

Function to init DNR.

Depending on the supplied options, three operational modes are supported:

- 1. TRDP\_DNR\_COMMON\_THREAD (default) Expect tlc\_process running in a different, separate thread
- 2. TRDP\_DNR\_OWN\_THREAD For single threaded systems only! Internally call tlc\_process()
- 3. TRDP\_DNR\_STANDARD\_DNS Use standard DNS instead of TCN-DNS. Default dnsPort (= 0) for TCN-DNS is 17225, for standard DNS it is 53.

### **Parameters**

| in | appHandle      | Handle returned by tlc_openSession().   |
|----|----------------|---|
| in | dnslpAddr      | DNS/ECSP IP address.  |
| in | dnsPort        | DNS port number.  |
| in | pHostsFileName | Optional host file name as ECSP replacement/addition.                           |
| in | dnsOptions     | Use existing thread (recommended), use own tlc_process loop or use standard DNS |

### **Return values**

| TRDP_NO_ERR   | no error             |
|---------------|----------------------|
| TRDP_INIT_ERR | initialisation error |

### < default DNR/ECSP settings

### 5.6.2.6 tau\_uri2Addr()

```
TRDP_IP_ADDR_T * pAddr,
const TRDP_URI_T pUri )
```

Function to convert a URI to an IP address.

Receives an URI as input variable and translates this URI to an IP-Address. The URI may specify either a unicast or a multicast IP-Address.

### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession()                     |
|-----|-----------|--|
| out | pAddr     | Pointer to return the IP address                         |
| in  | pUri      | Pointer to an URI or an IP Address string, NULL==own URI |

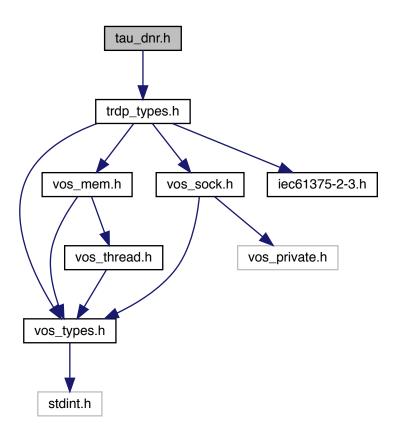
#### Return values

| TRDP_NO_ERR         | no error                  |
|---------------------|---------------------------|
| TRDP_PARAM_ERR      | Parameter error           |
| TRDP_UNRESOLVED_ERR | Could not resolve error   |
| TRDP_TOPO_ERR       | Cache/DB entry is invalid |

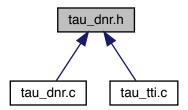
# 5.7 tau\_dnr.h File Reference

TRDP utility interface definitions.

#include "trdp\_types.h"
Include dependency graph for tau\_dnr.h:



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



# **Functions**

• EXT\_DECL TRDP\_ERR\_T tau\_initDnr (TRDP\_APP\_SESSION\_T appHandle, TRDP\_IP\_ADDR\_T dnsIp↔ Addr, UINT16 dnsPort, const CHAR8 \*pHostsFileName, TRDP\_DNR\_OPTS\_T dnsOptions)

Function to init DNR.

EXT\_DECL void tau\_deInitDnr (TRDP\_APP\_SESSION\_T appHandle)

Release any resources allocated by DNR.

• EXT\_DECL TRDP\_DNR\_STATE\_T tau\_DNRstatus (TRDP\_APP\_SESSION\_T appHandle)

Function to get the status of DNR.

EXT\_DECL TRDP\_IP\_ADDR\_T tau\_getOwnAddr (TRDP\_APP\_SESSION\_T appHandle)

Function to get the own IP address.

EXT\_DECL TRDP\_ERR\_T tau\_uri2Addr (TRDP\_APP\_SESSION\_T appHandle, TRDP\_IP\_ADDR\_T \*p
 — Addr, const TRDP\_URI\_T pUri)

Function to convert a URI to an IP address.

• EXT\_DECL TRDP\_ERR\_T tau\_addr2Uri (TRDP\_APP\_SESSION\_T appHandle, TRDP\_URI\_HOST\_T pUri, TRDP\_IP\_ADDR\_T addr)

Function to convert an IP address to a URI.

### 5.7.1 Detailed Description

TRDP utility interface definitions.

This module provides the interface to the following utilities

• IP - URI address translation

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Armin-H. Weiss (initial version)

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

tau\_dnr.h 1755 2018-08-07 12:10:03Z bloehr

```
BL 2018-08-07: Ticket #183 tau_getOwnIds moved here
BL 2017-07-25: Ticket #125: tau_dnr: TCN DNS support missing
BL 2015-12-14: Ticket #8: DNR client
```

### 5.7.2 Function Documentation

### 5.7.2.1 tau\_addr2Uri()

Function to convert an IP address to a URI.

Receives an IP-Address and translates it into the host part of the corresponding URI. Both unicast and multicast addresses are accepted.

### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().           |
|-----|-----------|---|
| out | pUri      | Pointer to a string to return the URI host part |
| in  | addr      | IP address, 0==own address                      |

### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

Receives an IP-Address and translates it into the host part of the corresponding URI. Both unicast and multicast addresses are accepted.

#### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession()            |
|-----|-----------|---|
| out | pUri      | Pointer to a string to return the URI host part |
| in  | addr      | IP address, 0==own address                      |

#### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

# 5.7.2.2 tau\_deInitDnr()

Release any resources allocated by DNR.

# **Parameters**

| in | appHandle | Handle returned by tlc_openSession(). |
|----|-----------|---------------------------------------|
|----|-----------|---------------------------------------|

# Return values

| none | Release any resources allocated by DNR. |
|------|---|
|------|---|

# **Parameters**

| ı |    |           | 11 11 1 0 1 0                        |
|---|----|-----------|--------------------------------------|
|   | ın | appHandle | Handle returned by tlc_openSession() |

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP PARAM ERR | Parameter error |

### 5.7.2.3 tau\_DNRstatus()

```
EXT_DECL TRDP_DNR_STATE_T tau_DNRstatus (

TRDP_APP_SESSION_T appHandle )
```

Function to get the status of DNR.

### **Parameters**

| in | appHandle | Handle returned by tlc_openSession() |
|----|-----------|--------------------------------------|
|----|-----------|--------------------------------------|

#### Return values

| TRDP_DNR_NOT_AVAILABLE | no error                              |
|------------------------|---------------------------------------|
| TRDP_DNR_UNKNOWN       | enabled, but cache is empty           |
| TRDP_DNR_ACTIVE        | enabled, cache has values             |
| TRDP_DNR_HOSTSFILE     | enabled, hostsfile used (static mode) |

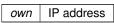
# 5.7.2.4 tau\_getOwnAddr()

Function to get the own IP address.

### **Parameters**

| ſ | in | appHandle | Handle returned by tlc_openSession(). |
|---|----|-----------|---------------------------------------|
|---|----|-----------|---------------------------------------|

### Return values



Returns the IP address set by openSession. If it was 0 (INADDR\_ANY), the address of the default adapter will be returned.

# **Parameters**

| in | appHandle | Handle returned by tlc_openSession() |
|----|-----------|--------------------------------------|
|----|-----------|--------------------------------------|

| own IP address |
|----------------|
|----------------|

#### 5.7.2.5 tau\_initDnr()

```
EXT_DECL TRDP_ERR_T tau_initDnr (

TRDP_APP_SESSION_T appHandle,

TRDP_IP_ADDR_T dnsIpAddr,

UINT16 dnsPort,

const CHAR8 * pHostsFileName,

TRDP_DNR_OPTS_T dnsOptions )
```

#### Function to init DNR.

#### **Parameters**

| in | appHandle      | Handle returned by tlc_openSession().   |
|----|----------------|---|
| in | dnslpAddr      | DNS/ECSP IP address.  |
| in | dnsPort        | DNS port number.  |
| in | pHostsFileName | Optional host file name as ECSP replacement/addition.                           |
| in | dnsOptions     | Use existing thread (recommended), use own tlc_process loop or use standard DNS |

#### Return values

| TRDP_NO_ERR   | no error             |
|---------------|----------------------|
| TRDP_INIT_ERR | initialisation error |

### Function to init DNR.

Depending on the supplied options, three operational modes are supported:

- 1. TRDP\_DNR\_COMMON\_THREAD (default) Expect tlc\_process running in a different, separate thread
- 2. TRDP\_DNR\_OWN\_THREAD For single threaded systems only! Internally call tlc\_process()
- 3. TRDP\_DNR\_STANDARD\_DNS Use standard DNS instead of TCN-DNS. Default dnsPort (= 0) for TCN-DNS is 17225, for standard DNS it is 53.

### Parameters

| in | appHandle      | Handle returned by tlc_openSession().   |
|----|----------------|---|
| in | dnslpAddr      | DNS/ECSP IP address.  |
| in | dnsPort        | DNS port number.  |
| in | pHostsFileName | Optional host file name as ECSP replacement/addition.                           |
| in | dnsOptions     | Use existing thread (recommended), use own tlc_process loop or use standard DNS |

### Return values

| TRDP_NO_ERR   | no error             |
|---------------|----------------------|
| TRDP_INIT_ERR | initialisation error |

### < default DNR/ECSP settings

#### 5.7.2.6 tau\_uri2Addr()

Function to convert a URI to an IP address.

Receives a URI as input variable and translates this URI to an IP-Address. The URI may specify either a unicast or a multicast IP-Address. The caller may specify a topographic counter, which will be checked.

#### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().                   |
|-----|-----------|---|
| out | pAddr     | Pointer to return the IP address                        |
| in  | pUri      | Pointer to a URI or an IP Address string, NULL==own URI |

#### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

Receives an URI as input variable and translates this URI to an IP-Address. The URI may specify either a unicast or a multicast IP-Address.

### Parameters

|   | in  | appHandle | Handle returned by tlc_openSession()                     |
|---|-----|-----------|--|
|   | out | pAddr     | Pointer to return the IP address                         |
| Ī | in  | pUri      | Pointer to an URI or an IP Address string, NULL==own URI |

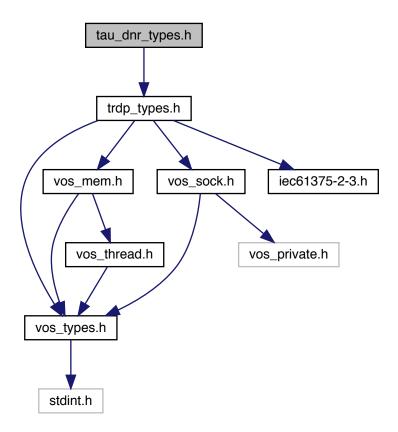
#### Return values

| TRDP_NO_ERR         | no error                  |
|---------------------|---------------------------|
| TRDP_PARAM_ERR      | Parameter error           |
| TRDP_UNRESOLVED_ERR | Could not resolve error   |
| TRDP_TOPO_ERR       | Cache/DB entry is invalid |

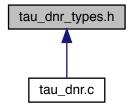
# 5.8 tau\_dnr\_types.h File Reference

TRDP utility interface definitions.

#include "trdp\_types.h"
Include dependency graph for tau\_dnr\_types.h:



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



# **Data Structures**

• struct TCN\_URI

TCN-DNS simplified header structures.

struct TRDP\_DNS\_REQUEST

TCN-DNS Request telegram TCN\_DNS\_REQ\_DS.

struct TRDP\_DNS\_REPLY

TCN-DNS Reply telegram TCN\_DNS\_REP\_DS.

# **Typedefs**

typedef struct TCN\_URI TCN\_URI\_T

TCN-DNS simplified header structures.

typedef struct TRDP\_DNS\_REQUEST\_T

TCN-DNS Request telegram TCN\_DNS\_REQ\_DS.

typedef struct TRDP\_DNS\_REPLY TRDP\_DNS\_REPLY\_T

TCN-DNS Reply telegram TCN\_DNS\_REP\_DS.

### 5.8.1 Detailed Description

TRDP utility interface definitions.

This module provides typedefs to the following utilities

• IP - URI address translation

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Löhr (initial version)

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright NewTec GmbH, 2017. All rights reserved.

ld

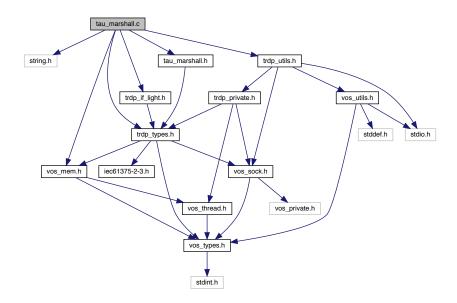
```
BL 2017-11-13: Ticket \#176 TRDP_LABEL_T breaks field alignment -> TRDP_NET_LABEL_T BL 2017-07-25: Ticket \#125: TCN-DNR client
```

# 5.9 tau marshall.c File Reference

### Marshalling functions for TRDP.

```
#include <string.h>
#include "trdp_types.h"
#include "trdp_if_light.h"
#include "trdp_utils.h"
#include "vos_mem.h"
#include "tau_marshall.h"
```

Include dependency graph for tau\_marshall.c:



#### **Data Structures**

struct TAU\_MARSHALL\_INFO\_T

Marshalling info, used to and from wire.

### **Functions**

Function to initialise the marshalling/unmarshalling.

- EXT\_DECL TRDP\_ERR\_T tau\_marshall (void \*pRefCon, UINT32 comId, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDest, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)
- EXT\_DECL TRDP\_ERR\_T tau\_unmarshall (void \*pRefCon, UINT32 comId, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDest, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)
   unmarshall function.
- EXT\_DECL TRDP\_ERR\_T tau\_marshallDs (void \*pRefCon, UINT32 dsId, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDest, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

marshall data set function.

EXT\_DECL TRDP\_ERR\_T tau\_unmarshallDs (void \*pRefCon, UINT32 dsld, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDest, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

unmarshall data set function.

EXT\_DECL TRDP\_ERR\_T tau\_calcDatasetSize (void \*pRefCon, UINT32 dsld, UINT8 \*pSrc, UINT32 src
 Size, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

Calculate data set size by given data set id.

EXT\_DECL TRDP\_ERR\_T tau\_calcDatasetSizeByComld (void \*pRefCon, UINT32 comld, UINT8 \*pSrc, U

INT32 srcSize, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

Calculate data set size by given Comld.

#### 5.9.1 Detailed Description

Marshalling functions for TRDP.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

# tau\_marshall.c 1748 2018-07-13 15:59:36Z bloehr

```
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32) SW 2018-06-12: Ticket #203 Incorrect unmarshalling of datasets containing TIMEDATE64 array BL 2018-05-17: Ticket #197 Incorrect Marshalling/Unmarshalling for nested datasets BL 2018-05-15: Wrong source size/range should not lead to marshalling error, check discarded BL 2018-05-03: Ticket #193 Unused parameter warnings BL 2018-05-02: Ticket #188 Typo in the TRDP_VAR_SIZE definition BL 2017-05-08: Compiler warnings, MISRA-C BL 2017-05-08: Ticket #156 Recursion counter never decremented (+ compiler warnings, MISRA) BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings), alignment casts fixed BL 2016-02-01: Ticket #108: missing initialisation of size-pointer BL 2016-02-04: Ticket #109: size_marshall -> size_unmarshall BL 2016-02-03: Ticket #108: Uninitialized info variable BL 2015-12-14: Ticket #33: source size check for marshalling
```

### 5.9.2 Function Documentation

#### 5.9.2.1 tau\_calcDatasetSize()

Calculate data set size by given data set id.

# **Parameters**

| in     | pRefCon     | Pointer to user context   |  |
|--------|-------------|---|--|
| in     | dsld        | Dataset id to identify the structure out of a configuration                     |  |
| in     | pSrc        | Pointer to received original message  |  |
| in     | srcSize     | size of the source buffer   |  |
| out    | pDestSize   | Pointer to the size of the data set   |  |
| in,out | ppDSPointer | pointer to pointer to cached dataset, set NULL if not used, set content NULL if |  |
|        |             | unknown   |  |

#### Return values

| TRDP_INIT_ERR        | marshalling not initialised  |
|----------------------|------------------------------|
| TRDP_NO_ERR          | no error                     |
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_COMID_ERR       | comid not existing           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

# 5.9.2.2 tau\_calcDatasetSizeByComld()

Calculate data set size by given Comld.

# **Parameters**

| in     | pRefCon     | Pointer to user context   |  |
|--------|-------------|---|--|
| in     | comld       | Comld id to identify the structure out of a configuration                       |  |
| in     | pSrc        | Pointer to received original message  |  |
| in     | srcSize     | size of the source buffer   |  |
| out    | pDestSize   | Pointer to the size of the data set   |  |
| in,out | ppDSPointer | pointer to pointer to cached dataset, set NULL if not used, set content NULL if |  |
|        |             | unknown   |  |

| TRDP_INIT_ERR  | marshalling not initialised |
|----------------|-----------------------------|
| TRDP_NO_ERR    | no error                    |
| TRDP_MEM_ERR   | provided buffer to small    |
| TRDP_PARAM_ERR | Parameter error             |
| TRDP_STATE_ERR | Too deep recursion          |

### Return values

| TRDP_COMID_ERR       | comid not existing           |
|----------------------|------------------------------|
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

### 5.9.2.3 tau\_initMarshall()

Function to initialise the marshalling/unmarshalling.

Types for marshalling / unmarshalling.

The supplied array must be sorted by Comlds. The array must exist during the use of the marshalling functions (until tlc\_terminate()).

#### **Parameters**

| in,out | ppRefCon      | Returns a pointer to be used for the reference context of marshalling/unmarshalling |
|--------|---------------|---|
| in     | numComId      | Number of datasets found in the configuration                                       |
| in     | pComldDsldMap | Pointer to an array of structures of type TRDP_DATASET_T                            |
| in     | numDataSet    | Number of datasets found in the configuration                                       |
| in     | pDataset      | Pointer to an array of pointers to structures of type TRDP_DATASET_T                |

### Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | Parameter error          |

# 5.9.2.4 tau\_marshall()

marshall function.

# **Parameters**

| in     | pRefCon  | pointer to user context  |
|--------|--|--|
| in     | comld Comld to identify the structure out of a configuration |  |
| in     | pSrc   | pointer to received original message   |
| in     | srcSize  | size of the source buffer  |
| in     | pDest  | pointer to a buffer for the treated message  |
| in,out | pDestSize  | size of the provide buffer / size of the treated message                               |
| in,out | ppDSPointer  | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |

### Return values

| TRDP_NO_ERR          | no error                     |
|----------------------|------------------------------|
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

# 5.9.2.5 tau\_marshallDs()

# marshall data set function.

### **Parameters**

| in     | pRefCon     | pointer to user context  |
|--------|-------------|--|
| in     | dsld        | Data set id to identify the structure out of a configuration                           |
| in     | pSrc        | pointer to received original message   |
| in     | srcSize     | size of the source buffer  |
| in     | pDest       | pointer to a buffer for the treated message  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |

| TRDP_INIT_ERR  | marshalling not initialised |
|----------------|-----------------------------|
| TRDP_NO_ERR    | no error                    |
| TRDP_MEM_ERR   | provided buffer to small    |
| TRDP_PARAM_ERR | Parameter error             |
| TRDP_STATE_ERR | Too deep recursion          |
| TRDP_COMID_ERR | comid not existing          |

# Return values

```
TRDP_MARSHALLING_ERR | dataset/source size mismatch
```

# 5.9.2.6 tau\_unmarshall()

#### unmarshall function.

#### **Parameters**

| in     | pRefCon     | pointer to user context  |  |
|--------|-------------|--|--|
| in     | comId       | comld Comld to identify the structure out of a configuration                           |  |
| in     | pSrc        | pointer to received original message   |  |
| in     | srcSize     | size of the source buffer  |  |
| in     | pDest       | pointer to a buffer for the treated message  |  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |  |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |  |

# Return values

| TRDP_INIT_ERR        | marshalling not initialised  |
|----------------------|------------------------------|
| TRDP_NO_ERR          | no error                     |
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_COMID_ERR       | comid not existing           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

# 5.9.2.7 tau\_unmarshallDs()

```
EXT_DECL TRDP_ERR_T tau_unmarshallDs (
    void * pRefCon,
    UINT32 dsId,
    UINT8 * pSrc,
    UINT32 srcSize,
    UINT8 * pDest,
```

```
UINT32 * pDestSize,
TRDP_DATASET_T ** ppDSPointer )
```

# unmarshall data set function.

#### **Parameters**

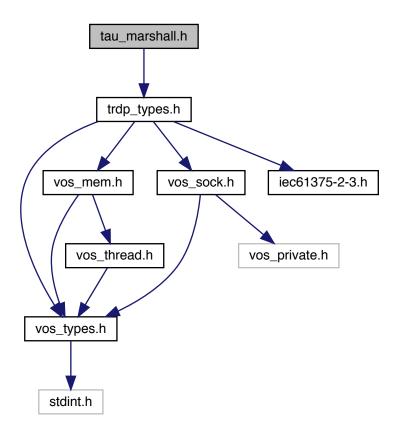
| in     | pRefCon     | pointer to user context  |  |
|--------|-------------|--|--|
| in     | dsld        | dsld Data set id to identify the structure out of a configuration                      |  |
| in     | pSrc        | pointer to received original message   |  |
| in     | srcSize     | size of the source buffer  |  |
| in     | pDest       | pointer to a buffer for the treated message  |  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |  |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |  |

# Return values

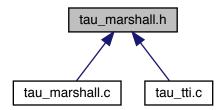
| TRDP_INIT_ERR        | marshalling not initialised  |
|----------------------|------------------------------|
| TRDP_NO_ERR          | no error                     |
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_COMID_ERR       | comid not existing           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

# 5.10 tau\_marshall.h File Reference

#include "trdp\_types.h"
Include dependency graph for tau\_marshall.h:



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



# **Functions**

Types for marshalling / unmarshalling.

• EXT\_DECL TRDP\_ERR\_T tau\_marshall (void \*pRefCon, UINT32 comId, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDest, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

marshall function.

 EXT\_DECL TRDP\_ERR\_T tau\_marshallDs (void \*pRefCon, UINT32 dsld, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDest, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

marshall data set function.

• EXT\_DECL TRDP\_ERR\_T tau\_unmarshall (void \*pRefCon, UINT32 comId, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDest, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

unmarshall function.

 EXT\_DECL TRDP\_ERR\_T tau\_unmarshallDs (void \*pRefCon, UINT32 dsld, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDest, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

unmarshall data set function.

• EXT\_DECL TRDP\_ERR\_T tau\_calcDatasetSize (void \*pRefCon, UINT32 dsld, UINT8 \*pSrc, UINT32 src
Size, UINT32 \*pDestSize, TRDP DATASET T \*\*ppDSPointer)

Calculate data set size by given data set id.

EXT\_DECL TRDP\_ERR\_T tau\_calcDatasetSizeByComld (void \*pRefCon, UINT32 comld, UINT8 \*pSrc, U
 INT32 srcSize, UINT32 \*pDestSize, TRDP\_DATASET\_T \*\*ppDSPointer)

Calculate data set size by given Comld.

#### 5.10.1 Detailed Description

TRDP utility interface definitions.

This module provides the interface to the following utilities

· marshalling/unmarshalling

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Armin-H. Weiss

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

tau\_marshall.h 1479 2015-12-14 14:53:45Z bloehr

```
BL 2015-12-14: Ticket #33: source size check for marshalling
```

### 5.10.2 Function Documentation

# 5.10.2.1 tau\_calcDatasetSize()

Calculate data set size by given data set id.

# **Parameters**

| in      | pRefCon     | Pointer to user context   |
|---------|-------------|---|
| in      | dsld        | Dataset id to identify the structure out of a configuration                     |
| in      | pSrc        | Pointer to received original message  |
| in      | srcSize     | size of the source buffer   |
| out     | pDestSize   | Pointer to the size of the data set   |
| in, out | ppDSPointer | pointer to pointer to cached dataset, set NULL if not used, set content NULL if |
|         |             | unknown   |

### Return values

| TRDP_NO_ERR    | no error                    |
|----------------|-----------------------------|
| TRDP_INIT_ERR  | marshalling not initialised |
| TRDP_PARAM_ERR | data set id not existing    |

### **Parameters**

| in      | pRefCon     | Pointer to user context   |
|---------|-------------|---|
| in      | dsld        | Dataset id to identify the structure out of a configuration                     |
| in      | pSrc        | Pointer to received original message  |
| in      | srcSize     | size of the source buffer   |
| out     | pDestSize   | Pointer to the size of the data set   |
| in, out | ppDSPointer | pointer to pointer to cached dataset, set NULL if not used, set content NULL if |
|         |             | unknown   |

| TRDP_INIT_ERR        | marshalling not initialised  |
|----------------------|------------------------------|
| TRDP_NO_ERR          | no error                     |
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_COMID_ERR       | comid not existing           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

# 5.10.2.2 tau\_calcDatasetSizeByComId()

# Calculate data set size by given Comld.

# **Parameters**

| in      | pRefCon     | Pointer to user context   |
|---------|-------------|---|
| in      | comId       | Comld id to identify the structure out of a configuration                       |
| in      | pSrc        | Pointer to received original message  |
| in      | srcSize     | size of the source buffer   |
| out     | pDestSize   | Pointer to the size of the data set   |
| in, out | ppDSPointer | pointer to pointer to cached dataset, set NULL if not used, set content NULL if |
|         |             | unknown   |

### Return values

| TRDP_NO_ERR    | no error                    |
|----------------|-----------------------------|
| TRDP_INIT_ERR  | marshalling not initialised |
| TRDP_PARAM_ERR | data set id not existing    |

### **Parameters**

| in      | pRefCon     | Pointer to user context   |
|---------|-------------|---|
| in      | comId       | Comld id to identify the structure out of a configuration                       |
| in      | pSrc        | Pointer to received original message  |
| in      | srcSize     | size of the source buffer   |
| out     | pDestSize   | Pointer to the size of the data set   |
| in, out | ppDSPointer | pointer to pointer to cached dataset, set NULL if not used, set content NULL if |
|         |             | unknown   |

| TRDP_INIT_ERR        | marshalling not initialised  |
|----------------------|------------------------------|
| TRDP_NO_ERR          | no error                     |
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_COMID_ERR       | comid not existing           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

### 5.10.2.3 tau\_initMarshall()

Types for marshalling / unmarshalling.

Function to initialise the marshalling/unmarshalling.

#### **Parameters**

| in,out | ppRefCon      | Returns a pointer to be used for the reference context of marshalling/unmarshalling |
|--------|---------------|---|
| in     | numComId      | Number of datasets found in the configuration                                       |
| in     | pComldDsldMap | Pointer to an array of structures of type TRDP_DATASET_T                            |
| in     | numDataSet    | Number of datasets found in the configuration                                       |
| in     | pDataset      | Pointer to an array of pointers to structures of type TRDP_DATASET_T                |

#### Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | Parameter error          |

Types for marshalling / unmarshalling.

The supplied array must be sorted by Comlds. The array must exist during the use of the marshalling functions (until tlc\_terminate()).

# **Parameters**

| in,out | ppRefCon      | Returns a pointer to be used for the reference context of marshalling/unmarshalling |  |
|--------|---------------|---|--|
| in     | numComId      | Number of datasets found in the configuration                                       |  |
| in     | pComIdDsIdMap | ap Pointer to an array of structures of type TRDP_DATASET_T                         |  |
| in     | numDataSet    | Number of datasets found in the configuration                                       |  |
| in     | pDataset      | Pointer to an array of pointers to structures of type TRDP_DATASET_T                |  |

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | Parameter error          |

# 5.10.2.4 tau\_marshall()

# marshall function.

#### **Parameters**

| in     | pRefCon     | pointer to user context  |
|--------|-------------|--|
| in     | comld       | Comld to identify the structure out of a configuration                                 |
| in     | pSrc        | pointer to received original message   |
| in     | srcSize     | size of the source buffer  |
| in     | pDest       | pointer to a buffer for the treated message  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |

# Return values

| TRDP_NO_ERR    | no error                    |
|----------------|-----------------------------|
| TRDP_MEM_ERR   | provided buffer to small    |
| TRDP_INIT_ERR  | marshalling not initialised |
| TRDP_COMID_ERR | comid not existing          |
| TRDP_PARAM_ERR | Parameter error             |

### **Parameters**

| in      | pRefCon     | pointer to user context  |
|---------|-------------|--|
| in      | comld       | Comld to identify the structure out of a configuration                                 |
| in      | pSrc        | pointer to received original message   |
| in      | srcSize     | size of the source buffer  |
| in      | pDest       | pointer to a buffer for the treated message  |
| in,out  | pDestSize   | size of the provide buffer / size of the treated message                               |
| in, out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |
|         |             |  |

| TRDP_NO_ERR          | no error                     |
|----------------------|------------------------------|
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP MARSHALLING ERR | dataset/source size mismatch |

# 5.10.2.5 tau\_marshallDs()

# marshall data set function.

#### **Parameters**

| in     | pRefCon     | pointer to user context  |
|--------|-------------|--|
| in     | dsld        | Data set id to identify the structure out of a configuration                           |
| in     | pSrc        | pointer to received original message   |
| in     | srcSize     | size of the source buffer  |
| in     | pDest       | pointer to a buffer for the treated message  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |

# Return values

| TRDP_NO_ERR    | no error                    |
|----------------|-----------------------------|
| TRDP_MEM_ERR   | provided buffer to small    |
| TRDP_INIT_ERR  | marshalling not initialised |
| TRDP_COMID_ERR | comid not existing          |
| TRDP_PARAM_ERR | Parameter error             |

### **Parameters**

| known |
|-------|
|       |

| TRDP_INIT_ERR        | marshalling not initialised  |
|----------------------|------------------------------|
| TRDP_NO_ERR          | no error                     |
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_COMID_ERR       | comid not existing           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

# 5.10.2.6 tau\_unmarshall()

### unmarshall function.

# **Parameters**

| in     | pRefCon     | pointer to user context  |
|--------|-------------|--|
| in     | comId       | Comld to identify the structure out of a configuration                                 |
| in     | pSrc        | pointer to received original message   |
| in     | srcSize     | size of the source buffer  |
| in     | pDest       | pointer to a buffer for the treated message  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |

# Return values

| TRDP_NO_ERR    | no error                    |
|----------------|-----------------------------|
| TRDP_MEM_ERR   | provided buffer to small    |
| TRDP_INIT_ERR  | marshalling not initialised |
| TRDP_COMID_ERR | comid not existing          |

# **Parameters**

| in     | pRefCon     | pointer to user context  |
|--------|-------------|--|
| in     | comId       | ComId to identify the structure out of a configuration                                 |
| in     | pSrc        | pointer to received original message   |
| in     | srcSize     | size of the source buffer  |
| in     | pDest       | pointer to a buffer for the treated message  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |

| TRDP_INIT_ERR        | marshalling not initialised  |
|----------------------|------------------------------|
| TRDP_NO_ERR          | no error                     |
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_COMID_ERR       | comid not existing           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

# 5.10.2.7 tau\_unmarshallDs()

### unmarshall data set function.

# **Parameters**

| in     | pRefCon     | pointer to user context  |
|--------|-------------|--|
| in     | dsld        | Data set id to identify the structure out of a configuration                           |
| in     | pSrc        | pointer to received original message   |
| in     | srcSize     | size of the source buffer  |
| in     | pDest       | pointer to a buffer for the treated message  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |

# Return values

| TRDP_NO_ERR    | no error                    |
|----------------|-----------------------------|
| TRDP_MEM_ERR   | provided buffer to small    |
| TRDP_INIT_ERR  | marshalling not initialised |
| TRDP_COMID_ERR | comid not existing          |

# **Parameters**

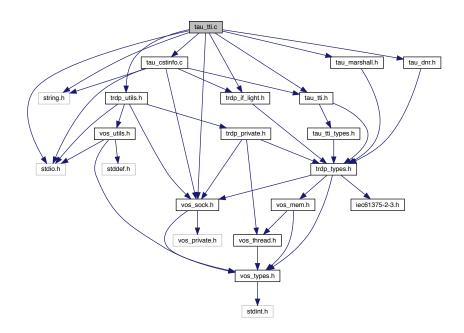
| in     | pRefCon     | pointer to user context  |  |
|--------|-------------|--|--|
| in     | dsld        | Data set id to identify the structure out of a configuration                           |  |
| in     | pSrc        | pointer to received original message   |  |
| in     | srcSize     | size of the source buffer  |  |
| in     | pDest       | pointer to a buffer for the treated message  |  |
| in,out | pDestSize   | size of the provide buffer / size of the treated message                               |  |
| in,out | ppDSPointer | pointer to pointer to cached dataset set NULL if not used, set content NULL if unknown |  |

| TRDP_INIT_ERR        | marshalling not initialised  |
|----------------------|------------------------------|
| TRDP_NO_ERR          | no error                     |
| TRDP_MEM_ERR         | provided buffer to small     |
| TRDP_PARAM_ERR       | Parameter error              |
| TRDP_STATE_ERR       | Too deep recursion           |
| TRDP_COMID_ERR       | comid not existing           |
| TRDP_MARSHALLING_ERR | dataset/source size mismatch |

# 5.11 tau tti.c File Reference

Functions for train topology information access.

```
#include <string.h>
#include <stdio.h>
#include "trdp_if_light.h"
#include "trdp_utils.h"
#include "tau_marshall.h"
#include "tau_tti.h"
#include "vos_sock.h"
#include "tau_dnr.h"
#include "tau_cstinfo.c"
Include dependency graph for tau_tti.c:
```



# Macros

• #define TTI\_CACHED\_CONSISTS 8u

We hold this number of consist infos (ca.

### **Functions**

• EXT\_DECL TRDP\_ERR\_T tau\_initTTlaccess (TRDP\_APP\_SESSION\_T appHandle, VOS\_SEMA\_T user ← Action, TRDP\_IP\_ADDR\_T ecsplpAddr, CHAR8 \*hostsFileName)

Function to init TTI access.

• EXT\_DECL void tau\_deInitTTI (TRDP\_APP\_SESSION\_T appHandle)

Release any resources allocated by TTI Must be called before closing the session.

• EXT\_DECL TRDP\_ERR\_T tau\_getOpTrDirectory (TRDP\_APP\_SESSION\_T appHandle, TRDP\_OP\_TRA

IN DIR STATE T \*pOpTrnDirState, TRDP OP TRAIN DIR T \*pOpTrnDir)

Function to retrieve the operational train directory state.

EXT\_DECL TRDP\_ERR\_T tau\_getOpTrnDirectoryStatusInfo (TRDP\_APP\_SESSION\_T appHandle, TRD
 — P\_OP\_TRAIN\_DIR\_STATUS\_INFO\_T \*pOpTrnDirStatusInfo)

Function to retrieve the operational train directory state info.

EXT\_DECL TRDP\_ERR\_T tau\_getTrDirectory (TRDP\_APP\_SESSION\_T appHandle, TRDP\_TRAIN\_DIR
 — T \*pTrnDir)

Function to retrieve the train directory.

• EXT\_DECL TRDP\_ERR\_T tau\_getStaticCstInfo (TRDP\_APP\_SESSION\_T appHandle, TRDP\_CONSIST → INFO T \*pCstInfo, TRDP UUID T const cstUUID)

Function to retrieve the consist info.

• EXT\_DECL TRDP\_ERR\_T tau\_getTTI (TRDP\_APP\_SESSION\_T appHandle, TRDP\_OP\_TRAIN\_DIR\_S 

TATE\_T \*pOpTrnDirState, TRDP\_OP\_TRAIN\_DIR\_T \*pOpTrnDir, TRDP\_TRAIN\_DIR\_T \*pTrnDir, TRDP 

\_TRAIN\_NET\_DIR\_T \*pTrnNetDir)

Function to retrieve the operational train directory.

- EXT\_DECL TRDP\_ERR\_T tau\_getTrnCstCnt (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pTrnCstCnt) Function to retrieve the total number of consists in the train.
- EXT\_DECL TRDP\_ERR\_T tau\_getTrnVehCnt (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pTrnVehCnt) Function to retrieve the total number of vehicles in the train.
- EXT\_DECL TRDP\_ERR\_T tau\_getCstVehCnt (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pCstVehCnt, const TRDP\_LABEL\_T pCstLabel)

Function to retrieve the total number of vehicles in a consist.

EXT\_DECL TRDP\_ERR\_T tau\_getCstFctCnt (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pCstFctCnt, const TRDP\_LABEL\_T pCstLabel)

Function to retrieve the total number of functions in a consist.

EXT\_DECL TRDP\_ERR\_T tau\_getCstFctInfo (TRDP\_APP\_SESSION\_T appHandle, TRDP\_FUNCTION\_
 —
 INFO\_T \*pFctInfo, const TRDP\_LABEL\_T pCstLabel, UINT16 maxFctCnt)

Function to retrieve the function information of the consist.

• EXT\_DECL TRDP\_ERR\_T tau\_getVehInfo (TRDP\_APP\_SESSION\_T appHandle, TRDP\_VEHICLE\_INF

O\_T \*pVehInfo, const TRDP\_LABEL\_T pVehLabel, const TRDP\_LABEL\_T pCstLabel)

Function to retrieve the vehicle information of a consist's vehicle.

• EXT\_DECL TRDP\_ERR\_T tau\_getCstInfo (TRDP\_APP\_SESSION\_T appHandle, TRDP\_CONSIST\_INF

O\_T \*pCstInfo, const TRDP\_LABEL\_T pCstLabel)

Function to retrieve the consist information of a train's consist.

• EXT\_DECL TRDP\_ERR\_T tau\_getVehOrient (TRDP\_APP\_SESSION\_T appHandle, UINT8 \*pVehOrient, UINT8 \*pCstOrient, TRDP\_LABEL\_T pVehLabel, TRDP\_LABEL\_T pCstLabel)

Function to retrieve the orientation of the given vehicle.

• EXT\_DECL TRDP\_ERR\_T tau\_getOwnIds (TRDP\_APP\_SESSION\_T appHandle, TRDP\_LABEL\_T \*p → DevId, TRDP\_LABEL\_T \*pVehId, TRDP\_LABEL\_T \*pCstId)

Who am I ?.

#### 5.11.1 Detailed Description

Functions for train topology information access.

The TTI subsystem maintains a pointer to the TAU\_TTDB struct in the TRDP session struct. That TAU\_TTDB struct keeps the subscription and listener handles, the current TTDB directories and a pointer list to consist infos (in network format). On init, most TTDB data is requested from the ECSP plus the own consist info. This data is automatically updated if an inauguration is detected. Additional consist infos are requested on demand, only. Because of the asynchronous behavior of the TTI subsystem, most functions in tau\_tti.c may return TRDP\_N ○ ODATA\_ERR on first invocation. They should be called again after 1...3 seconds (3s is the timeout for most MD replies).

Note

Project: TCNOpen TRDP prototype stack

Author

B. Loehr (initial version)

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2016. All rights reserved.

ld

#### tau tti.c 1755 2018-08-07 12:10:03Z bloehr

```
BL 2018-08-07: Ticket #183 tau_getOwnIds declared but not defined
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2017-11-28: Ticket #180 Filtering rules for DestinationURI does not follow the standard
BL 2017-11-13: Ticket #176 TRDP_LABEL_T breaks field alignment -> TRDP_NET_LABEL_T
AHW 2017-11-08: Ticket #179 Max. number of retries (part of sendParam) of a MD request needs to be checked
BL 2017-05-08: Compiler warnings, doxygen comment errors
BL 2017-04-28: Ticket #155: Kill trdp_proto.h - move definitions to iec61375-2-3.h
BL 2017-03-13: Ticket #154 ComIds and DSIds literals (#define TRDP_...) in trdp_proto.h too long
BL 2017-02-10: Ticket #129 Found a bug which yields wrong output params and potentially segfaults
BL 2017-02-8: Ticket #142 Compiler warnings / MISRA-C 2012 issues
BL 2016-02-18: Ticket #7: Add train topology information support
```

### 5.11.2 Macro Definition Documentation

```
5.11.2.1 TTI_CACHED_CONSISTS
```

```
#define TTI_CACHED_CONSISTS 8u
```

We hold this number of consist infos (ca.

105kB)

#### 5.11.3 Function Documentation

```
5.11.3.1 tau_delnitTTI()
```

Release any resources allocated by TTI Must be called before closing the session.

Function to terminate TTI access.

### **Parameters**

|  | in | appHandle | Handle returned by tlc_openSession(). |
|--|----|-----------|---------------------------------------|
|--|----|-----------|---------------------------------------|

#### **Return values**

```
none
```

# 5.11.3.2 tau\_getCstFctCnt()

Function to retrieve the total number of functions in a consist.

#### **Parameters**

| in  | appHandle  | Handle returned by tlc_openSession().               |
|-----|------------|---|
| out | pCstFctCnt | Pointer to the number of functions to be returned   |
| in  | pCstLabel  | Pointer to a consist label. NULL means own consist. |

### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

# 5.11.3.3 tau\_getCstFctInfo()

Function to retrieve the function information of the consist.

### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().   |  |
|-----|-----------|---|--|
| out | pFctInfo  | Pointer to function info list to be returned. Memory needs to be provided by application. |  |
|     |           | Set NULL if not used.   |  |
| in  | pCstLabel | Pointer to a consist label. NULL means own consist.                                       |  |
| in  | maxFctCnt | Maximal number of functions to be returned in provided buffer.                            |  |

### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

# 5.11.3.4 tau\_getCstInfo()

Function to retrieve the consist information of a train's consist.

#### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().               |  |
|-----|-----------|---|--|
| out | pCstInfo  | Pointer to the consist info to be returned.         |  |
| in  | pCstLabel | Pointer to a consist label. NULL means own consist. |  |

# Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

# 5.11.3.5 tau\_getCstVehCnt()

```
EXT_DECL TRDP_ERR_T tau_getCstVehCnt (

TRDP_APP_SESSION_T appHandle,

UINT16 * pCstVehCnt,

const TRDP_LABEL_T pCstLabel )
```

Function to retrieve the total number of vehicles in a consist.

#### **Parameters**

| in  | appHandle  | Handle returned by tlc_openSession().               |  |
|-----|------------|---|--|
| out | pCstVehCnt | Pointer to the number of vehicles to be returned    |  |
| in  | pCstLabel  | Pointer to a consist label. NULL means own consist. |  |

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | Parameter error |
| TRDP_NODATA_ERR | Try again       |

#### 5.11.3.6 tau\_getOpTrDirectory()

Function to retrieve the operational train directory state.

#### **Parameters**

| in  | appHandle      | Handle returned by tlc_openSession().                                     |
|-----|----------------|---|
| out | pOpTrnDirState | Pointer to an operational train directory state structure to be returned. |
| out | pOpTrnDir      | Pointer to an operational train directory structure to be returned.       |

#### Return values

| TRDP_NO_ERR     | no error                                      |
|-----------------|---|
| TRDP_PARAM_ERR  | Parameter error                               |
| TRDP_NODATA_ERR | Data currently not available, try again later |

### 5.11.3.7 tau\_getOpTrnDirectoryStatusInfo()

Function to retrieve the operational train directory state info.

Return a copy of the last received PD 100 telegram. Note: The values are in host endianess! When validating (SDTv2), network endianess must be ensured.

### **Parameters**

| in  | appHandle           | Handle returned by tlc_openSession().                                     |
|-----|---------------------|---|
| out | pOpTrnDirStatusInfo | Pointer to an operational train directory state structure to be returned. |

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

### 5.11.3.8 tau\_getOwnlds()

```
EXT_DECL TRDP_ERR_T tau_getOwnIds (

TRDP_APP_SESSION_T appHandle,

TRDP_LABEL_T * pDevId,

TRDP_LABEL_T * pVehId,

TRDP_LABEL_T * pCstId )
```

Who am I?.

Realizes a kind of 'Who am I' function. It is used to determine the own identifiers (i.e. the own labels), which may be used as host part of the own fully qualified domain name.

#### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession() |
|-----|-----------|--------------------------------------|
| out | pDevld    | Returns the device label (host name) |
| out | pVehld    | Returns the vehicle label            |
| out | pCstld    | Returns the consist label            |

#### Return values

| TRDP_NO_ERR     | no error                                 |
|-----------------|--|
| TRDP_PARAM_ERR  | Parameter error                          |
| TRDP_NODATA_ERR | Data currently not available, call again |

# 5.11.3.9 tau\_getStaticCstInfo()

```
EXT_DECL TRDP_ERR_T tau_getStaticCstInfo (

TRDP_APP_SESSION_T appHandle,

TRDP_CONSIST_INFO_T * pCstInfo,

TRDP_UUID_T const cstUUID )
```

Function to retrieve the consist info.

Function to retrieve the operational train directory.

#### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().                 |
|-----|-----------|---|
| out | pCstInfo  | Pointer to a consist info structure to be returned.   |
| in  | cstUUID   | UUID of the consist the consist info is rquested for. |

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.11.3.10 tau\_getTrDirectory()

Function to retrieve the train directory.

Function to retrieve the operational train directory.

### **Parameters**

| in <i>appl</i> |     | appHandle | Handle returned by tlc_openSession().                  |
|----------------|-----|-----------|--|
| Ī              | out | pTrnDir   | Pointer to a train directory structure to be returned. |

### **Return values**

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | Parameter error |
| TRDP_NODATA_ERR | Try later       |

### 5.11.3.11 tau\_getTrnCstCnt()

Function to retrieve the total number of consists in the train.

### **Parameters**

|  | in | appHandle  | Handle returned by tlc_openSession(). |  |
|--|----|--|---------------------------------------|--|
| out pTrnCstCnt Pointer to the number of consists |    | Pointer to the number of consists to be returned |                                       |  |

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | Parameter error |
| TRDP_NODATA_ERR | Try again       |

## 5.11.3.12 tau\_getTrnVehCnt()

Function to retrieve the total number of vehicles in the train.

## **Parameters**

| in                    | appHandle | Handle returned by tlc_openSession().            |  |
|-----------------------|-----------|--|--|
| out <i>pTrnVehCnt</i> |           | Pointer to the number of vehicles to be returned |  |

### Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | Parameter error |
| TRDP_NODATA_ERR | Try again       |

## 5.11.3.13 tau\_getTTI()

```
EXT_DECL TRDP_ERR_T tau_getTTI (

TRDP_APP_SESSION_T appHandle,

TRDP_OP_TRAIN_DIR_STATE_T * pOpTrnDirState,

TRDP_OP_TRAIN_DIR_T * pOpTrnDir,

TRDP_TRAIN_DIR_T * pTrnDir,

TRDP_TRAIN_NET_DIR_T * pTrnNetDir )
```

Function to retrieve the operational train directory.

### **Parameters**

| in                              | appHandle      | Handle returned by tlc_openSession().                                     |  |
|---------------------------------|----------------|---|--|
| out                             | pOpTrnDirState | Pointer to an operational train directory state structure to be returned. |  |
| out <i>pOpTrnDir</i> Pointer to |                | Pointer to an operational train directory structure to be returned.       |  |
| out                             | pTrnDir        | Pointer to a train directory structure to be returned.                    |  |
| out                             | pTrnNetDir     | Pointer to a train network directory structure to be returned.            |  |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.11.3.14 tau\_getVehInfo()

```
EXT_DECL TRDP_ERR_T tau_getVehInfo (

TRDP_APP_SESSION_T appHandle,

TRDP_VEHICLE_INFO_T * pVehInfo,

const TRDP_LABEL_T pVehLabel,

const TRDP_LABEL_T pCstLabel)
```

Function to retrieve the vehicle information of a consist's vehicle.

### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().   |  |
|-----|-----------|---|--|
| out | pVehInfo  | Pointer to the vehicle info to be returned.   |  |
| in  | pVehLabel | Pointer to a vehicle label. NULL means own vehicle if cstLabel refers to own consist. |  |
| in  | pCstLabel | Pointer to a consist label. NULL means own consist.                                   |  |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.11.3.15 tau\_getVehOrient()

```
EXT_DECL TRDP_ERR_T tau_getVehOrient (

TRDP_APP_SESSION_T appHandle,

UINT8 * pVehOrient,

UINT8 * pCstOrient,

TRDP_LABEL_T pVehLabel,

TRDP_LABEL_T pCstLabel )
```

Function to retrieve the orientation of the given vehicle.

## **Parameters**

| in  | appHandle  | Handle returned by tlc_openSession().  |  |
|-----|------------|--|--|
| out | pVehOrient | Pointer to the vehicle orientation to be returned '00'B = not known (corrected vehicle)    |  |
|     |            | '01'B = same as operational train direction '10'B = inverse to operational train direction |  |
| out | pCstOrient | Pointer to the consist orientation to be returned '00'B = not known (corrected vehicle)    |  |
|     |            | '01'B = same as operational train direction '10'B = inverse to operational train direction |  |
| in  | pVehLabel  | vehLabel = NULL means own vehicle if cstLabel == NULL, currently ignored.                  |  |
| in  | pCstLabel  | cstLabel = NULL means own consist  |  |

# Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.11.3.16 tau\_initTTlaccess()

Function to init TTI access.

Subscribe to necessary process data for correct ECSP handling, further calls need DNS!

## **Parameters**

| in | appHandle     | Handle returned by tlc_openSession().                                   |  |
|----|---------------|---|--|
| in | userAction    | Semaphore to fire if inauguration took place.                           |  |
| in | ecsplpAddr    | ECSP IP address. Currently not used.                                    |  |
| in | hostsFileName | Optional host file name as ECSP replacement. Currently not implemented. |  |

# Return values

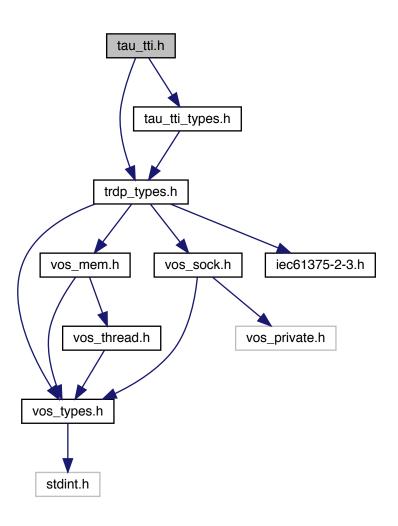
| TRDP_NO_ERR   | no error             |
|---------------|----------------------|
| TRDP_INIT_ERR | initialisation error |

# 5.12 tau\_tti.h File Reference

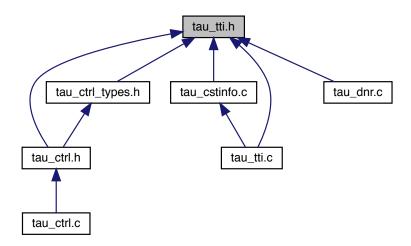
TRDP utility interface definitions.

```
#include "trdp_types.h"
#include "tau_tti_types.h"
```

Include dependency graph for tau\_tti.h:



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



### **Functions**

EXT\_DECL TRDP\_ERR\_T tau\_initTTlaccess (TRDP\_APP\_SESSION\_T appHandle, VOS\_SEMA\_T user

 Action, TRDP\_IP\_ADDR\_T ecsplpAddr, CHAR8 \*hostsFileName)

Function to init TTI access.

EXT\_DECL void tau\_delnitTTI (TRDP\_APP\_SESSION\_T appHandle)

Function to terminate TTI access.

EXT\_DECL TRDP\_ERR\_T tau\_getOpTrDirectory (TRDP\_APP\_SESSION\_T appHandle, TRDP\_OP\_TRA
 IN\_DIR\_STATE\_T \*pOpTrDirState, TRDP\_OP\_TRAIN\_DIR\_T \*pOpTrDir)

Function to retrieve the operational train directory state.

EXT\_DECL TRDP\_ERR\_T tau\_getOpTrnDirectoryStatusInfo (TRDP\_APP\_SESSION\_T appHandle, TRD
 — P\_OP\_TRAIN\_DIR\_STATUS\_INFO\_T \*pOpTrnDirStatusInfo)

Function to retrieve the operational train directory state info.

EXT\_DECL TRDP\_ERR\_T tau\_getTrDirectory (TRDP\_APP\_SESSION\_T appHandle, TRDP\_TRAIN\_DIR
 \_T \*pTrDir)

Function to retrieve the operational train directory.

EXT\_DECL TRDP\_ERR\_T tau\_getStaticCstInfo (TRDP\_APP\_SESSION\_T appHandle, TRDP\_CONSIST
 — INFO\_T \*pCstInfo, TRDP\_UUID\_T const cstUUID)

Function to retrieve the operational train directory.

• EXT\_DECL TRDP\_ERR\_T tau\_getTTI (TRDP\_APP\_SESSION\_T appHandle, TRDP\_OP\_TRAIN\_DIR\_S 
TATE\_T \*pOpTrnDirState, TRDP\_OP\_TRAIN\_DIR\_T \*pOpTrnDir, TRDP\_TRAIN\_DIR\_T \*pTrnDir, TRDP 
\_\_TRAIN\_NET\_DIR\_T \*pTrnNetDir)

Function to retrieve the operational train directory.

- EXT\_DECL TRDP\_ERR\_T tau\_getTrnCstCnt (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pTrnCstCnt) Function to retrieve the total number of consists in the train.
- EXT\_DECL TRDP\_ERR\_T tau\_getTrnVehCnt (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pTrnVehCnt) Function to retrieve the total number of vehicles in the train.
- EXT\_DECL TRDP\_ERR\_T tau\_getCstVehCnt (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pCstVehCnt, const TRDP\_LABEL\_T pCstLabel)

Function to retrieve the total number of vehicles in a consist.

• EXT\_DECL TRDP\_ERR\_T tau\_getCstFctCnt (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pCstFctCnt, const TRDP\_LABEL\_T pCstLabel)

Function to retrieve the total number of functions in a consist.

EXT\_DECL TRDP\_ERR\_T tau\_getCstFctInfo (TRDP\_APP\_SESSION\_T appHandle, TRDP\_FUNCTION\_
 —
 INFO\_T \*pFctInfo, const TRDP\_LABEL\_T pCstLabel, UINT16 maxFctCnt)

Function to retrieve the function information of the consist.

Function to retrieve the vehicle information of a consist's vehicle.

EXT\_DECL TRDP\_ERR\_T tau\_getCstInfo (TRDP\_APP\_SESSION\_T appHandle, TRDP\_CONSIST\_INF

 O\_T \*pCstInfo, const TRDP\_LABEL\_T pCstLabel)

Function to retrieve the consist information of a train's consist.

• EXT\_DECL\_TRDP\_ERR\_T\_tau\_getVehOrient (TRDP\_APP\_SESSION\_T appHandle, UINT8 \*pVehOrient, UINT8 \*pCstOrient, TRDP\_LABEL\_T pVehLabel, TRDP\_LABEL\_T pCstLabel)

Function to retrieve the orientation of the given vehicle.

• EXT\_DECL TRDP\_ERR\_T tau\_getOwnIds (TRDP\_APP\_SESSION\_T appHandle, TRDP\_LABEL\_T \*p↔ DevId, TRDP\_LABEL\_T \*pVehId, TRDP\_LABEL\_T \*pCstId)

Who am I?.

### 5.12.1 Detailed Description

TRDP utility interface definitions.

This module provides the interface to the following utilities

· train topology information access

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Armin-H. Weiss (initial version)

### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2014. All rights reserved.

ld

tau\_tti.h 1755 2018-08-07 12:10:03Z bloehr

```
BL 2018-08-07: Ticket #183 tau_getOwnIds moved here BL 2016-02-18: Ticket #7: Add train topology information support
```

### 5.12.2 Function Documentation

```
5.12.2.1 tau_deInitTTI()
```

Function to terminate TTI access.

### **Parameters**

| in | appHandle | Handle returned by tlc_openSession(). |
|----|-----------|---------------------------------------|
|----|-----------|---------------------------------------|

### Return values

### **Parameters**

|  | in | appHandle | Handle returned by tlc_openSession(). |
|--|----|-----------|---------------------------------------|
|--|----|-----------|---------------------------------------|

### **Return values**

```
none
```

## 5.12.2.2 tau\_getCstFctCnt()

Function to retrieve the total number of functions in a consist.

### **Parameters**

|   | in  | appHandle  | Handle returned by tlc_openSession().               |
|---|-----|------------|---|
| - | out | pCstFctCnt | Pointer to the number of functions to be returned   |
|   | in  | pCstLabel  | Pointer to a consist label. NULL means own consist. |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.12.2.3 tau\_getCstFctInfo()

Function to retrieve the function information of the consist.

## **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().   |
|-----|-----------|---|
| out | pFctInfo  | Pointer to function info list to be returned. Memory needs to be provided by application. |
|     |           | Set NULL if not used.   |
| in  | pCstLabel | Pointer to a consist label. NULL means own consist.                                       |
| in  | maxFctCnt | Maximal number of functions to be returned in provided buffer.                            |

### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

# 5.12.2.4 tau\_getCstInfo()

Function to retrieve the consist information of a train's consist.

### **Parameters**

| Ī | in  | appHandle | Handle returned by tlc_openSession().               |
|---|-----|-----------|---|
| ĺ | out | pCstInfo  | Pointer to the consist info to be returned.         |
| Ī | in  | pCstLabel | Pointer to a consist label. NULL means own consist. |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.12.2.5 tau\_getCstVehCnt()

Function to retrieve the total number of vehicles in a consist.

| i | .n | appHandle  | Handle returned by tlc_openSession().               |
|---|----|------------|---|
| 0 | ut | pCstVehCnt | Pointer to the number of vehicles to be returned    |
| i | .n | pCstLabel  | Pointer to a consist label. NULL means own consist. |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

# **Parameters**

| in  | appHandle  | Handle returned by tlc_openSession().               |
|-----|------------|---|
| out | pCstVehCnt | Pointer to the number of vehicles to be returned    |
| in  | pCstLabel  | Pointer to a consist label. NULL means own consist. |

### Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | Parameter error |
| TRDP_NODATA_ERR | Try again       |

# 5.12.2.6 tau\_getOpTrDirectory()

Function to retrieve the operational train directory state.

### **Parameters**

| in  | appHandle     | Handle returned by tlc_openSession().                                     |
|-----|---------------|---|
| out | pOpTrDirState | Pointer to an operational train directory state structure to be returned. |
| out | pOpTrDir      | Pointer to an operational train directory structure to be returned.       |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## **Parameters**

| in  | appHandle      | Handle returned by tlc_openSession().                                     |
|-----|----------------|---|
| out | pOpTrnDirState | Pointer to an operational train directory state structure to be returned. |
| out | pOpTrnDir      | Pointer to an operational train directory structure to be returned.       |

### Return values

| TRDP_NO_ERR     | no error                                      |
|-----------------|---|
| TRDP_PARAM_ERR  | Parameter error                               |
| TRDP_NODATA_ERR | Data currently not available, try again later |

### 5.12.2.7 tau\_getOpTrnDirectoryStatusInfo()

Function to retrieve the operational train directory state info.

Return a copy of the last received PD 100 telegram. Note: The values are in host endianess! When validating (SDTv2), network endianess must be ensured.

### **Parameters**

| in  | appHandle           | Handle returned by tlc_openSession().                                     |
|-----|---------------------|---|
| out | pOpTrnDirStatusInfo | Pointer to an operational train directory state structure to be returned. |

### **Return values**

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.12.2.8 tau\_getOwnlds()

```
EXT_DECL TRDP_ERR_T tau_getOwnIds (

TRDP_APP_SESSION_T appHandle,

TRDP_LABEL_T * pDevId,

TRDP_LABEL_T * pVehId,

TRDP_LABEL_T * pCstId )
```

## Who am I?.

Realizes a kind of 'Who am I' function. It is used to determine the own identifiers (i.e. the own labels), which may be used as host part of the own fully qualified domain name.

### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession() |
|-----|-----------|--------------------------------------|
| out | pDevld    | Returns the device label (host name) |
| out | pVehld    | Returns the vehicle label            |
| out | pCstld    | Returns the consist label            |

### **Return values**

| TRDP_NO_ERR     | no error                                 |
|-----------------|--|
| TRDP_PARAM_ERR  | Parameter error                          |
| TRDP_NODATA_ERR | Data currently not available, call again |

## 5.12.2.9 tau\_getStaticCstInfo()

Function to retrieve the operational train directory.

### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().                 |
|-----|-----------|---|
| out | pCstInfo  | Pointer to a consist info structure to be returned.   |
| in  | cstUUID   | UUID of the consist the consist info is rquested for. |

### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

Function to retrieve the operational train directory.

### **Parameters**

| in  | appHandle   | Handle returned by tlc_openSession(). |  |
|-----|---|---------------------------------------|--|
| out | pCstInfo Pointer to a consist info structure to be returned.    |                                       |  |
| in  | in cstUUID UUID of the consist the consist info is rquested for |                                       |  |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.12.2.10 tau\_getTrDirectory()

Function to retrieve the operational train directory.

| in  | appHandle | Handle returned by tlc_openSession().                  |
|-----|-----------|--|
| out | pTrDir    | Pointer to a train directory structure to be returned. |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

Function to retrieve the operational train directory.

### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().                  |
|-----|-----------|--|
| out | pTrnDir   | Pointer to a train directory structure to be returned. |

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | Parameter error |
| TRDP_NODATA_ERR | Try later       |

# 5.12.2.11 tau\_getTrnCstCnt()

Function to retrieve the total number of consists in the train.

## **Parameters**

| in  | appHandle  | Handle returned by tlc_openSession().            |
|-----|------------|--|
| out | pTrnCstCnt | Pointer to the number of consists to be returned |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## **Parameters**

| in  | appHandle  | Handle returned by tlc_openSession().            |
|-----|------------|--|
| out | pTrnCstCnt | Pointer to the number of consists to be returned |

### Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | Parameter error |
| TRDP_NODATA_ERR | Try again       |

## 5.12.2.12 tau\_getTrnVehCnt()

Function to retrieve the total number of vehicles in the train.

### **Parameters**

| in  | appHandle  | Handle returned by tlc_openSession().            |
|-----|------------|--|
| out | pTrnVehCnt | Pointer to the number of vehicles to be returned |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## **Parameters**

| j | in  | appHandle  | Handle returned by tlc_openSession().            |
|---|-----|------------|--|
|   | out | pTrnVehCnt | Pointer to the number of vehicles to be returned |

### Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | Parameter error |
| TRDP_NODATA_ERR | Try again       |

## 5.12.2.13 tau\_getTTI()

```
EXT_DECL TRDP_ERR_T tau_getTTI (

TRDP_APP_SESSION_T appHandle,

TRDP_OP_TRAIN_DIR_STATE_T * pOpTrnDirState,

TRDP_OP_TRAIN_DIR_T * pOpTrnDir,

TRDP_TRAIN_DIR_T * pTrnDir,

TRDP_TRAIN_NET_DIR_T * pTrnNetDir )
```

Function to retrieve the operational train directory.

| in  | appHandle  | Handle returned by tlc_openSession().                          |  |
|-----|--|--|--|
| out | pOpTrnDirState Pointer to an operational train directory state structure to be returned. |  |  |
| out | pOpTrnDir Pointer to an operational train directory structure to be returned.            |  |  |
| out | pTrnDir Pointer to a train directory structure to be returned.                           |  |  |
| out | pTrnNetDir   | Pointer to a train network directory structure to be returned. |  |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

# 5.12.2.14 tau\_getVehInfo()

```
EXT_DECL TRDP_ERR_T tau_getVehInfo (

TRDP_APP_SESSION_T appHandle,

TRDP_VEHICLE_INFO_T * pVehInfo,

const TRDP_LABEL_T pVehLabel,

const TRDP_LABEL_T pCstLabel)
```

Function to retrieve the vehicle information of a consist's vehicle.

### **Parameters**

| in  | appHandle | Handle returned by tlc_openSession().   |  |
|-----|-----------|---|--|
| out | pVehInfo  | Pointer to the vehicle info to be returned.   |  |
| in  | pVehLabel | Pointer to a vehicle label. NULL means own vehicle if cstLabel refers to own consist. |  |
| in  | pCstLabel | Pointer to a consist label. NULL means own consist.                                   |  |

### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.12.2.15 tau\_getVehOrient()

```
EXT_DECL TRDP_ERR_T tau_getVehOrient (
          TRDP_APP_SESSION_T appHandle,
          UINT8 * pVehOrient,
          UINT8 * pCstOrient,
          TRDP_LABEL_T pVehLabel,
          TRDP_LABEL_T pCstLabel )
```

Function to retrieve the orientation of the given vehicle.

| in  | appHandle  | Handle returned by tlc_openSession().  |
|-----|------------|--|
| out | pVehOrient | Pointer to the vehicle orientation to be returned '00'B = not known (corrected vehicle)    |
|     |            | '01'B = same as operational train direction '10'B = inverse to operational train direction |
| out | pCstOrient | Pointer to the consist orientation to be returned '00'B = not known (corrected vehicle)    |
|     |            | '01'B = same as operational train direction '10'B = inverse to operational train direction |
| in  | pVehLabel  | vehLabel = NULL means own vehicle if cstLabel == NULL                                      |
| in  | pCstLabel  | cstLabel = NULL means own consist  |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## **Parameters**

| in  | appHandle  | Handle returned by tlc_openSession().  |  |
|-----|------------|--|--|
| out | pVehOrient | Pointer to the vehicle orientation to be returned '00'B = not known (corrected vehicle)    |  |
|     |            | '01'B = same as operational train direction '10'B = inverse to operational train direction |  |
| out | pCstOrient | Pointer to the consist orientation to be returned '00'B = not known (corrected vehicle)    |  |
|     |            | '01'B = same as operational train direction '10'B = inverse to operational train direction |  |
| in  | pVehLabel  | vehLabel = NULL means own vehicle if cstLabel == NULL, currently ignored.                  |  |
| in  | pCstLabel  | cstLabel = NULL means own consist  |  |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | Parameter error |

## 5.12.2.16 tau\_initTTlaccess()

## Function to init TTI access.

## **Parameters**

| in | appHandle                      | Handle returned by tlc_openSession().         |  |
|----|--------------------------------|---|--|
| in | userAction                     | Semaphore to fire if inauguration took place. |  |
| in | in ecsplpAddr ECSP IP address. |   |  |
| in | hostsFileName                  | Optional host file name as ECSP replacement.  |  |

## Return values

| TRDP_NO_ERR   | no error             |
|---------------|----------------------|
| TRDP_INIT_ERR | initialisation error |

Subscribe to necessary process data for correct ECSP handling, further calls need DNS!

| in | appHandle  | Handle returned by tlc_openSession().         |
|----|------------|---|
| in | userAction | Semaphore to fire if inauguration took place. |
| in | ecsplpAddr | ECSP IP address. Currently not used.          |

## **Parameters**

| in | hostsFileName | Optional host file name as ECSP replacement. Currently not implemented. |  |
|----|---------------|---|--|
|----|---------------|---|--|

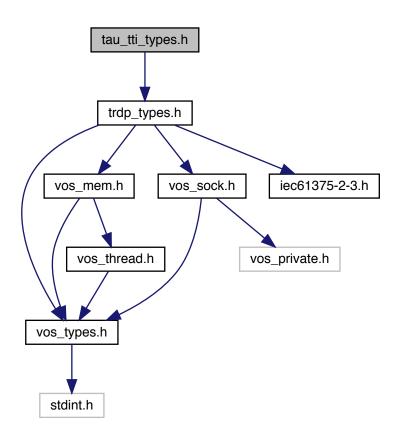
### Return values

| TRDP_NO_ERR   | no error             |
|---------------|----------------------|
| TRDP_INIT_ERR | initialisation error |

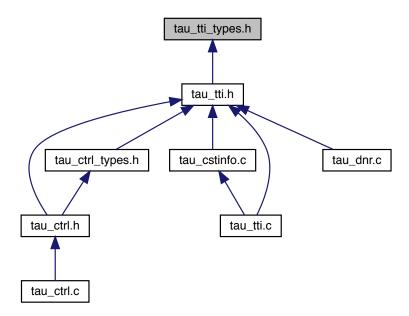
# 5.13 tau\_tti\_types.h File Reference

TRDP utility interface definitions.

#include "trdp\_types.h"
Include dependency graph for tau\_tti\_types.h:



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



# **Data Structures**

struct GNU\_PACKED

Types for ETB control.

struct TRDP\_ETB\_INFO\_T

Types for train configuration information.

struct TRDP\_CLTR\_CST\_INFO\_T

Closed train consists information.

struct TRDP\_PROP\_T

Application defined properties.

• struct TRDP\_FUNCTION\_INFO\_T

function/device information structure

struct TRDP\_VEHICLE\_INFO\_T

vehicle information structure

struct TRDP\_CONSIST\_INFO\_T

consist information structure

struct GNU\_PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

• struct GNU\_PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

struct GNU PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

### **Macros**

#define TRDP\_MAX\_CST\_CNT 63u
 max number of consists per train

#define TRDP\_MAX\_VEH\_CNT 63u

max number of vehicles per train

## 5.13.1 Detailed Description

TRDP utility interface definitions.

This module provides the interface to the following utilities

train topology information access type definitions acc. to IEC61375-2-3

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Armin-H. Weiss (initial version)

## Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2014. All rights reserved.

ld

tau\_tti\_types.h 1763 2018-09-21 16:03:13Z ahweiss

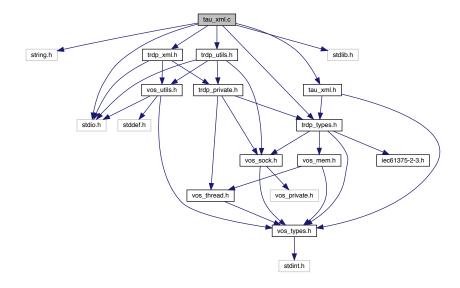
```
BL 2017-11-13: Ticket \#176 TRDP_LABEL_T breaks field alignment -> TRDP_NET_LABEL_T BL 2017-05-08: Compiler warnings, doxygen comment errors
```

# 5.14 tau\_xml.c File Reference

## Functions for XML file parsing.

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "trdp_types.h"
#include "trdp_utils.h"
#include "tau_xml.h"
#include "trdp_xml.h"
```

### Include dependency graph for tau\_xml.c:



## **Macros**

- #define TRDP\_SDT\_DEFAULT\_SMI2 0u
  - Default SDT safe message identifier.
- #define TRDP\_SDT\_DEFAULT\_NRXSAFE 3u

Default SDT timeout cycles.

• #define TRDP\_SDT\_DEFAULT\_NGUARD 100u

Default SDT initial timeout cycles.

• #define TRDP\_SDT\_DEFAULT\_CMTHR 10u

Default SDT chan.

### **Functions**

EXT\_DECL TRDP\_ERR\_T tau\_prepareXmlDoc (const CHAR8 \*pFileName, TRDP\_XML\_DOC\_HANDLE
 — T \*pDocHnd)

Open XML file, prepare XPath context.

• EXT DECL void tau freeXmlDoc (TRDP XML DOC HANDLE T \*pDocHnd)

Free all the memory allocated by tau\_prepareXmlDoc.

EXT\_DECL TRDP\_ERR\_T tau\_readXmlInterfaceConfig (const TRDP\_XML\_DOC\_HANDLE\_T \*pDocHnd, const CHAR8 \*plfName, TRDP\_PROCESS\_CONFIG\_T \*pProcessConfig, TRDP\_PD\_CONFIG\_T \*p← PdConfig, TRDP\_MD\_CONFIG\_T \*pMdConfig, UINT32 \*pNumExchgPar, TRDP\_EXCHG\_PAR\_T \*\*pp← ExchgPar)

Read the interface relevant telegram parameters (except data set configuration) out of the configuration file.

• EXT\_DECL void tau\_freeTelegrams (UINT32 numExchgPar, TRDP\_EXCHG\_PAR\_T \*pExchgPar)

Free array of telegram configurations allocated by tau\_readXmlInterfaceConfig.

• EXT\_DECL\_TRDP\_ERR\_T\_tau\_readXmlDeviceConfig (const\_TRDP\_XML\_DOC\_HANDLE\_T\_\*pDocHnd, TRDP\_MEM\_CONFIG\_T \*pMemConfig, TRDP\_DBG\_CONFIG\_T \*pDbgConfig, UINT32 \*pNumComPar, TRDP\_COM\_PAR\_T \*\*ppComPar, UINT32 \*pNumlfConfig, TRDP\_IF\_CONFIG\_T \*\*pplfConfig)

Function to read the TRDP device configuration parameters out of the XML configuration file.

 EXT\_DECL TRDP\_ERR\_T tau\_readXmlDatasetConfig (const TRDP\_XML\_DOC\_HANDLE\_T \*pDocHnd, UINT32 \*pNumComId, TRDP\_COMID\_DSID\_MAP\_T \*\*ppComIdDsIdMap, UINT32 \*pNumDataset, ap← TRDP\_DATASET\_T \*apDataset)

Function to read the DataSet configuration out of the XML configuration file.

EXT\_DECL void tau\_freeXmlDatasetConfig (UINT32 numComId, TRDP\_COMID\_DSID\_MAP\_T \*pComId←
 DsIdMap, UINT32 numDataset, TRDP\_DATASET\_T \*\*ppDataset)

Function to free the memory for the DataSet configuration.

## 5.14.1 Detailed Description

Functions for XML file parsing.

SOX parsing of XML configuration file

Note

Project: TCNOpen TRDP prototype stack

**Author** 

B. Loehr, NewTec GmbH, Tomas Svoboda, UniControls a.s.

### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright NewTec GmbH, 2016. All rights reserved.

ld

### tau\_xml.c 1770 2018-10-29 10:49:08Z s-bender

```
SB 2018-10-29: Ticket #214 Incorrect parsing of <source> and <destination> elements
BL 2018-10-01: Some default attribute values for com-parameter tag were missing
BL 2018-09-05: Ticket #211 XML handling: Dataset Name should be stored in TRDP_DATASET_ELEMENT_T
BL 2018-05-03: Ticket #194: Platform independent format specifiers in vos_printLog
BL 2018-01-30: Ticket #189 timeout-value not parsed in tau_xml
BL 2017-06-08: Compiler warning (unused dbgPrint)
BL 2017-05-08: Compiler warnings (static definitions)
BL 2017-03-01: Ticket #149 SourceUri and DestinationUri don't with 32 characters
BL 2017-02-27: Ticket #142 Compiler warnings / MISRA-C 2012 issues
BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings)
BL 2016-03-21: Ticket #116: Memory corruption using new XML library
BL 2016-03-04: Ticket #113: parsing of dataset element "type" always returns 0
BL 2016-02-11: Ticket #111: unit, scale, offset added
BL 2016-02-11: Ticket #102: Replacing libxm12
BL 2016-01-25: Ticket #106: Callback can be ON, OFF, ALWAYS
```

## 5.14.2 Macro Definition Documentation

## 5.14.2.1 TRDP\_SDT\_DEFAULT\_CMTHR

```
#define TRDP_SDT_DEFAULT_CMTHR 10u
```

Default SDT chan.

monitoring threshold

## 5.14.3 Function Documentation

### 5.14.3.1 tau\_freeTelegrams()

Free array of telegram configurations allocated by tau\_readXmlInterfaceConfig.

### **Parameters**

| in | numExchgPar | Number of telegram configurations in the array |
|----|-------------|--|
| in | pExchgPar   | Pointer to array of telegram configurations    |

## 5.14.3.2 tau\_freeXmlDatasetConfig()

Function to free the memory for the DataSet configuration.

Free the memory for the DataSet configuration which was allocated when parsing the XML configuration file.

| in | numComId   | The number of entries in the Comld DatasetId mapping list              |
|----|--|--|
| in | pComldDsldMap  | Pointer to an array of structures of type TRDP_COMID_DSID_MAP_T        |
| in | numDataset The number of datasets found in the configuration |  |
| in | ppDataset  | Pointer to an array of pointers to a structures of type TRDP_DATASET_T |

## Return values

none

### 5.14.3.3 tau\_freeXmlDoc()

```
EXT_DECL void tau_freeXmlDoc ( {\tt TRDP\_XML\_DOC\_HANDLE\_T\ *\ pDocHnd\ )}
```

Free all the memory allocated by tau\_prepareXmlDoc.

## **Parameters**

| in | pDocHnd | Handle of the parsed XML file |
|----|---------|-------------------------------|
|----|---------|-------------------------------|

## 5.14.3.4 tau\_prepareXmlDoc()

Open XML file, prepare XPath context.

Load XML file into DOM tree, prepare XPath context.

### **Parameters**

| in  | pFileName | Path and filename of the xml configuration file |
|-----|-----------|---|
| out | pDocHnd   | Handle of the parsed XML file                   |

### Return values

| TRDP_NO_ERR    | no error            |
|----------------|---------------------|
| TRDP_PARAM_ERR | File does not exist |

### 5.14.3.5 tau\_readXmlDatasetConfig()

Function to read the DataSet configuration out of the XML configuration file.

## **Parameters**

| in  | pDocHnd        | Handle of the XML document prepared by tau_prepareXmlDoc              |
|-----|----------------|---|
| out | pNumComId      | Pointer to the number of entries in the Comld DatasetId mapping list  |
| out | ppComldDsldMap | Pointer to an array of a structures of type TRDP_COMID_DSID_MAP_T     |
| out | pNumDataset    | Pointer to the number of datasets found in the configuration          |
| out | apDataset      | Pointer to an array of pointers to a structure of type TRDP_DATASET_T |

### Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | File not existing        |

## 5.14.3.6 tau\_readXmlDeviceConfig()

Function to read the TRDP device configuration parameters out of the XML configuration file.

The user must release the memory for ppComPar and pplfConfig (using vos\_memFree)

### **Parameters**

| in  | pDocHnd      | Handle of the XML document prepared by tau_prepareXmlDoc |
|-----|--------------|--|
| out | pMemConfig   | Memory configuration                                     |
| out | pDbgConfig   | Debug printout configuration for application use         |
| out | pNumComPar   | Number of configured com parameters                      |
| out | ppComPar     | Pointer to array of com parameters                       |
| out | pNumlfConfig | Number of configured interfaces                          |
| out | pplfConfig   | Pointer to an array of interface parameter sets          |

## Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | File not existing        |

### 5.14.3.7 tau\_readXmlInterfaceConfig()

Read the interface relevant telegram parameters (except data set configuration) out of the configuration file .

### **Parameters**

| in  | pDocHnd        | Handle of the XML document prepared by tau_prepareXmlDoc |
|-----|----------------|--|
| in  | plfName        | Interface name   |
| out | pProcessConfig | TRDP process (session) configuration for the interface   |
| out | pPdConfig      | PD default configuration for the interface               |
| out | pMdConfig      | MD default configuration for the interface               |
| out | pNumExchgPar   | Number of configured telegrams                           |
| out | ppExchgPar     | Pointer to array of telegram configurations              |

### Return values

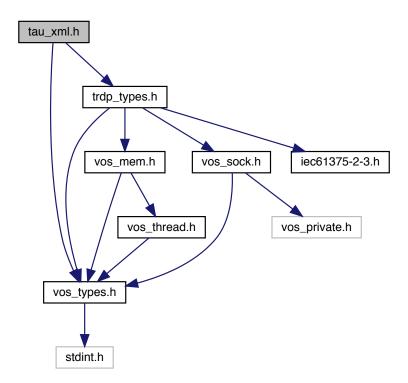
| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | File not existing        |

# 5.15 tau\_xml.h File Reference

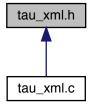
TRDP utility interface definitions.

```
#include "vos_types.h"
#include "trdp_types.h"
```

Include dependency graph for tau\_xml.h:



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



## **Data Structures**

• struct TRDP\_SDT\_PAR\_T

Types to read out the XML configuration.

• struct TRDP\_DBG\_CONFIG\_T

Control for debug output device/file on application level.

• struct TRDP\_XML\_DOC\_HANDLE\_T

Parsed XML document handle.

### **Macros**

• #define TRDP DBG DEFAULT 0.

Control for debug output format on application level.

• #define TRDP\_DBG\_OFF 0x01

Printout off.

#define TRDP\_DBG\_ERR 0x02

Printout error.

#define TRDP DBG WARN 0x04

Printout warning and error.

• #define TRDP DBG INFO 0x08

Printout info, warning and error.

#define TRDP DBG DBG 0x10

Printout debug, info, warning and error.

• #define TRDP\_DBG\_TIME 0x20

Printout timestamp.

#define TRDP DBG LOC 0x40

Printout file name and line.

#define TRDP DBG CAT 0x80

Printout category (DBG, INFO, WARN, ERR)

### **Enumerations**

```
    enum TRDP_EXCHG_OPTION_T {
        TRDP_EXCHG_UNSET = 0,
        TRDP_EXCHG_SOURCE = 1,
        TRDP_EXCHG_SINK = 2,
        TRDP_EXCHG_SOURCESINK = 3 }
```

Type attribute for telegrams.

### **Functions**

Load XML file into DOM tree, prepare XPath context.

• EXT DECL void tau freeXmlDoc (TRDP XML DOC HANDLE T\*pDocHnd)

Free all the memory allocated by tau\_prepareXmlDoc.

 EXT\_DECL\_TRDP\_ERR\_T tau\_readXmlDeviceConfig (const\_TRDP\_XML\_DOC\_HANDLE\_T \*pDocHnd, TRDP\_MEM\_CONFIG\_T \*pMemConfig, TRDP\_DBG\_CONFIG\_T \*pDbgConfig, UINT32 \*pNumComPar, TRDP\_COM\_PAR\_T \*\*ppComPar, UINT32 \*pNumIfConfig, TRDP\_IF\_CONFIG\_T \*\*pplfConfig)

Function to read the TRDP device configuration parameters out of the XML configuration file.

EXT\_DECL TRDP\_ERR\_T tau\_readXmlInterfaceConfig (const TRDP\_XML\_DOC\_HANDLE\_T \*pDocHnd, const CHAR8 \*plfName, TRDP\_PROCESS\_CONFIG\_T \*pProcessConfig, TRDP\_PD\_CONFIG\_T \*p← PdConfig, TRDP\_MD\_CONFIG\_T \*pMdConfig, UINT32 \*pNumExchgPar, TRDP\_EXCHG\_PAR\_T \*\*pp← ExchgPar)

 $\textit{Read the interface relevant telegram parameters (except data set configuration) out of the configuration file \ .}$ 

 EXT\_DECL TRDP\_ERR\_T tau\_readXmlDatasetConfig (const TRDP\_XML\_DOC\_HANDLE\_T \*pDocHnd, UINT32 \*pNumComId, TRDP\_COMID\_DSID\_MAP\_T \*\*ppComIdDsIdMap, UINT32 \*pNumDataset, pap← TRDP\_DATASET\_T papDataset)

Function to read the DataSet configuration out of the XML configuration file.

EXT\_DECL void tau\_freeXmlDatasetConfig (UINT32 numComId, TRDP\_COMID\_DSID\_MAP\_T \*pComId←DsIdMap, UINT32 numDataset, TRDP\_DATASET\_T \*\*pNumDataset)

Function to free the memory for the DataSet configuration.

• EXT\_DECL void tau\_freeTelegrams (UINT32 numExchgPar, TRDP\_EXCHG\_PAR\_T \*pExchgPar)

Free array of telegram configurations allocated by tau\_readXmlInterfaceConfig.

## 5.15.1 Detailed Description

TRDP utility interface definitions.

This module provides the interface to the following utilities

· read xml configuration interpreter

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Armin-H. Weiss (initial version)

### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

tau\_xml.h 1601 2017-05-08 15:27:38Z bloehr

```
BL 2017-05-08: Compiler warnings, flag enums -> defines
BL 2016-02-11: Ticket #102: Custom XML parser, libxml2 not needed anymore
```

### 5.15.2 Macro Definition Documentation

```
5.15.2.1 TRDP_DBG_DEFAULT
```

```
#define TRDP_DBG_DEFAULT 0,
```

Control for debug output format on application level.

Printout default

## 5.15.3 Enumeration Type Documentation

```
5.15.3.1 TRDP_EXCHG_OPTION_T
```

```
enum TRDP_EXCHG_OPTION_T
```

Type attribute for telegrams.

### Enumerator

| TRDP_EXCHG_UNSET      | default, direction is not defined          |
|-----------------------|--|
| TRDP_EXCHG_SOURCE     | telegram shall be published                |
| TRDP_EXCHG_SINK       | telegram shall be subscribed               |
| TRDP_EXCHG_SOURCESINK | telegram shall be published and subscribed |

## 5.15.4 Function Documentation

## 5.15.4.1 tau\_freeTelegrams()

Free array of telegram configurations allocated by tau\_readXmlInterfaceConfig.

### **Parameters**

| in | numExchgPar | Number of telegram configurations in the array |
|----|-------------|--|
| in | pExchgPar   | Pointer to array of telegram configurations    |

### 5.15.4.2 tau\_freeXmlDatasetConfig()

Function to free the memory for the DataSet configuration.

Free the memory for the DataSet configuration which was allocated when parsing the XML configuration file.

### **Parameters**

| in | numComId      | The number of entries in the Comld DatasetId mapping list              |
|----|---------------|--|
| in | pComldDsldMap | Pointer to an array of structures of type TRDP_COMID_DSID_MAP_T        |
| in | numDataset    | The number of datasets found in the configuration                      |
| in | pNumDataset   | Pointer to an array of pointers to a structures of type TRDP_DATASET_T |

### **Return values**

| none | Free the memory for the DataSet configuration which was allocated when parsing the XML |
|------|--|
|      | configuration file.  |

## **Parameters**

| in | numComId      | The number of entries in the Comld DatasetId mapping list              |
|----|---------------|--|
| in | pComIdDsIdMap | Pointer to an array of structures of type TRDP_COMID_DSID_MAP_T        |
| in | numDataset    | The number of datasets found in the configuration                      |
| in | ppDataset     | Pointer to an array of pointers to a structures of type TRDP_DATASET_T |

## Return values

```
none
```

## 5.15.4.3 tau\_freeXmlDoc()

```
EXT_DECL void tau_freeXmlDoc (  \label{eq:tau_dec} {\tt TRDP\_XML\_DOC\_HANDLE\_T} \ * \ pDocHnd \ )
```

Free all the memory allocated by tau\_prepareXmlDoc.

## **Parameters**

| in | pDocHnd | Handle of the parsed XML file |
|----|---------|-------------------------------|
|----|---------|-------------------------------|

## 5.15.4.4 tau\_prepareXmlDoc()

Load XML file into DOM tree, prepare XPath context.

## **Parameters**

| in  | pFileName | Path and filename of the xml configuration file |
|-----|-----------|---|
| out | pDocHnd   | Handle of the parsed XML file                   |

# Return values

| TRDP_NO_ERR    | no error            |
|----------------|---------------------|
| TRDP_PARAM_ERR | File does not exist |

Load XML file into DOM tree, prepare XPath context.

| in  | pFileName | Path and filename of the xml configuration file |
|-----|-----------|---|
| out | pDocHnd   | Handle of the parsed XML file                   |

# Return values

| TRDP_NO_ERR    | no error            |
|----------------|---------------------|
| TRDP_PARAM_ERR | File does not exist |

## 5.15.4.5 tau\_readXmlDatasetConfig()

Function to read the DataSet configuration out of the XML configuration file.

### **Parameters**

| in  | pDocHnd        | Handle of the XML document prepared by tau_prepareXmlDoc               |
|-----|----------------|--|
| out | pNumComId      | Pointer to the number of entries in the Comld DatasetId mapping list   |
| out | ppComIdDsIdMap | Pointer to an array of a structures of type TRDP_COMID_DSID_MAP_T      |
| out | pNumDataset    | Pointer to the number of datasets found in the configuration           |
| out | papDataset     | Pointer to an array of pointers to a structures of type TRDP_DATASET_T |

## Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | File not existing        |

### 5.15.4.6 tau\_readXmlDeviceConfig()

Function to read the TRDP device configuration parameters out of the XML configuration file.

| in  | pDocHnd    | Handle of the XML document prepared by tau_prepareXmlDoc |  |
|-----|------------|--|--|
| out | pMemConfig | Memory configuration                                     |  |

## **Parameters**

| out | pDbgConfig   | Debug printout configuration for application use |
|-----|--------------|--|
| out | pNumComPar   | Number of configured com parameters              |
| out | ppComPar     | Pointer to array of com parameters               |
| out | pNumlfConfig | Number of configured interfaces                  |
| out | pplfConfig   | Pointer to an array of interface parameter sets  |

### Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | File not existing        |

The user must release the memory for ppComPar and pplfConfig (using vos\_memFree)

### **Parameters**

| in  | pDocHnd      | Handle of the XML document prepared by tau_prepareXmlDoc |
|-----|--------------|--|
| out | pMemConfig   | Memory configuration                                     |
| out | pDbgConfig   | Debug printout configuration for application use         |
| out | pNumComPar   | Number of configured com parameters                      |
| out | ppComPar     | Pointer to array of com parameters                       |
| out | pNumlfConfig | Number of configured interfaces                          |
| out | pplfConfig   | Pointer to an array of interface parameter sets          |

### Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | File not existing        |

## 5.15.4.7 tau\_readXmlInterfaceConfig()

Read the interface relevant telegram parameters (except data set configuration) out of the configuration file .

| in | pDocHnd | Handle of the XML document prepared by tau_prepareXmlDoc |
|----|---------|--|

### **Parameters**

| in  | plfName        | Interface name   |
|-----|----------------|--|
| out | pProcessConfig | TRDP process (session) configuration for the interface |
| out | pPdConfig      | PD default configuration for the interface             |
| out | pMdConfig      | MD default configuration for the interface             |
| out | pNumExchgPar   | Number of configured telegrams                         |
| out | ppExchgPar     | Pointer to array of telegram configurations            |

### Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_PARAM_ERR | File not existing        |

# 5.16 trdp\_dllmain.c File Reference

Windows DLL main function.

## 5.16.1 Detailed Description

Windows DLL main function.

Note

Project: TCNOpen TRDP prototype stack

## Author

Armin-H. Weiss, Bombardier

### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

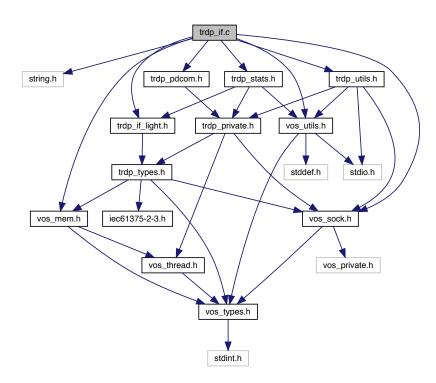
ld

trdp\_dllmain.c 1763 2018-09-21 16:03:13Z ahweiss

# 5.17 trdp\_if.c File Reference

Functions for ECN communication.

```
#include <string.h>
#include "trdp_if_light.h"
#include "trdp_utils.h"
#include "trdp_pdcom.h"
#include "trdp_stats.h"
#include "vos_sock.h"
#include "vos_mem.h"
#include "vos_utils.h"
Include dependency graph for trdp_if.c:
```



## **Functions**

• BOOL8 trdp\_isValidSession (TRDP\_APP\_SESSION\_T pSessionHandle)

Check if the session handle is valid.

• TRDP\_APP\_SESSION\_T \* trdp\_sessionQueue (void)

Get the session queue head pointer.

EXT\_DECL TRDP\_IP\_ADDR\_T tlc\_getOwnIpAddress (TRDP\_APP\_SESSION\_T appHandle)

Get the interface address.

• EXT\_DECL TRDP\_ERR\_T tlc\_init (const TRDP\_PRINT\_DBG\_T pPrintDebugString, void \*pRefCon, const TRDP\_MEM\_CONFIG\_T \*pMemConfig)

Initialize the TRDP stack.

• EXT\_DECL TRDP\_ERR\_T tlc\_openSession (TRDP\_APP\_SESSION\_T \*pAppHandle, TRDP\_IP\_ADDR ← \_ T ownIpAddr, TRDP\_IP\_ADDR\_T leaderIpAddr, const TRDP\_MARSHALL\_CONFIG\_T \*pMarshall, const TRDP\_PD\_CONFIG\_T \*pPdDefault, const TRDP\_MD\_CONFIG\_T \*pMdDefault, const TRDP\_PROCES ← S\_CONFIG\_T \*pProcessConfig)

Open a session with the TRDP stack.

 EXT\_DECL TRDP\_ERR\_T tlc\_configSession (TRDP\_APP\_SESSION\_T appHandle, const TRDP\_MAR← SHALL\_CONFIG\_T \*pMarshall, const TRDP\_PD\_CONFIG\_T \*pPdDefault, const TRDP\_MD\_CONFIG\_T \*pMdDefault, const TRDP\_PROCESS\_CONFIG\_T \*pProcessConfig)

(Re-)configure a session.

• EXT\_DECL TRDP\_ERR\_T tlc\_closeSession (TRDP\_APP\_SESSION\_T appHandle)

Close a session.

EXT\_DECL TRDP\_ERR\_T tlc\_terminate (void)

Un-Initialize.

• EXT DECL TRDP ERR T tlc reinitSession (TRDP APP SESSION T appHandle)

Re-Initialize

const char \* tlc\_getVersionString (void)

Return a human readable version representation.

EXT\_DECL const TRDP\_VERSION\_T \* tlc\_getVersion (void)

Return version.

- TRDP\_ERR\_T tlp\_setRedundant (TRDP\_APP\_SESSION\_T appHandle, UINT32 redId, BOOL8 leader)

  Do not send non-redundant PDs when we are follower.
- EXT\_DECL TRDP\_ERR\_T tlp\_getRedundant (TRDP\_APP\_SESSION\_T appHandle, UINT32 redId, BOOL8 \*pLeader)

Get status of redundant Comlds.

EXT\_DECL TRDP\_ERR\_T tlc\_setETBTopoCount (TRDP\_APP\_SESSION\_T appHandle, UINT32 etbTopo
 —
 Cnt)

Set new topocount for trainwide communication.

Set new operational train topocount for direction/orientation sensitive communication.

• EXT\_DECL UINT32 tlc\_getETBTopoCount (TRDP\_APP\_SESSION\_T appHandle)

Set new topocount for trainwide communication.

EXT\_DECL UINT32 tlc\_getOpTrainTopoCount (TRDP\_APP\_SESSION\_T appHandle)

Set new operational train topocount for direction/orientation sensitive communication.

EXT\_DECL TRDP\_ERR\_T tlp\_publish (TRDP\_APP\_SESSION\_T appHandle, TRDP\_PUB\_T \*pPubHandle, const void \*pUserRef, TRDP\_PD\_CALLBACK\_T pfCbFunction, UINT32 comId, UINT32 etbTopoCnt, UIN ← T32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr, TRDP\_IP\_ADDR\_T destlpAddr, UINT32 interval, UINT32 redId, TRDP\_FLAGS\_T pktFlags, const TRDP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, U ← INT32 dataSize)

Prepare for sending PD messages.

- EXT\_DECL TRDP\_ERR\_T tlp\_republish (TRDP\_APP\_SESSION\_T appHandle, TRDP\_PUB\_T pubHandle, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr, TRDP\_IP\_ADDR\_T destlpAddr)

  Prepare for sending PD messages.
- TRDP\_ERR\_T tlp\_unpublish (TRDP\_APP\_SESSION\_T appHandle, TRDP\_PUB\_T pubHandle)
   Stop sending PD messages.
- TRDP\_ERR\_T tlp\_put (TRDP\_APP\_SESSION\_T appHandle, TRDP\_PUB\_T pubHandle, const UINT8 \*p↔ Data, UINT32 dataSize)

Update the process data to send.

• EXT\_DECL TRDP\_ERR\_T tlc\_getInterval (TRDP\_APP\_SESSION\_T appHandle, TRDP\_TIME\_T \*pInterval, TRDP\_FDS\_T \*pFileDesc, INT32 \*pNoDesc)

Get the lowest time interval for PDs.

• EXT\_DECL TRDP\_ERR\_T tlc\_process (TRDP\_APP\_SESSION\_T appHandle, TRDP\_FDS\_T \*pRfds, INT32 \*pCount)

Work loop of the TRDP handler.

EXT\_DECL TRDP\_ERR\_T tlp\_request (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T subHandle, UINT32 comld, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr, TRDP\_IP\_AD → DR\_T destlpAddr, UINT32 redld, TRDP\_FLAGS\_T pktFlags, const TRDP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize, UINT32 replyComld, TRDP\_IP\_ADDR\_T replyIpAddr)

Initiate sending PD messages (PULL).

EXT\_DECL TRDP\_ERR\_T tlp\_subscribe (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T \*pSub← Handle, const void \*pUserRef, TRDP\_PD\_CALLBACK\_T pfCbFunction, UINT32 comId, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr1, TRDP\_IP\_ADDR\_T srclpAddr2, TRDP\_IP\_ADD← R\_T destlpAddr, TRDP\_FLAGS\_T pktFlags, UINT32 timeout, TRDP\_TO\_BEHAVIOR\_T toBehavior)

Prepare for receiving PD messages.

• EXT\_DECL TRDP\_ERR\_T tlp\_unsubscribe (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T sub↔ Handle)

Stop receiving PD messages.

• EXT\_DECL\_TRDP\_ERR\_T tlp\_resubscribe (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T sub ← Handle, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr1, TRDP\_IP\_ADDR\_T srclpAddr2, TRDP\_IP\_ADDR\_T destlpAddr)

Reprepare for receiving PD messages.

Get the last valid PD message.

## 5.17.1 Detailed Description

Functions for ECN communication.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

## Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

Ы

trdp if.c 1772 2018-10-30 12:18:12Z bloehr

```
BL 2018-10-09: Ticket #213 ComId 31 subscription removed (<-- undone!)
BL 2018-06-29: Default settings handling / compiler warnings
SW 2018-06-26: Ticket #205 tlm_addListener() does not acknowledge TRDP_FLAGS_DEFAULT flag
BL 2018-06-25: Ticket #201 tlp_setRedundant return value if redId is 0
BL 2018-06-12: Ticket #204 tlp_publish should take default callback function
BL 2018-05-03: Ticket #199 Setting redId on tlp_request() has no effect
BL 2018-04-20: Ticket #196 setRedundant with redId = 0 stops all publishers
BL 2018-04-18: MD notify: pass optional cb pointer to mdsend
BL 2018-03-06: Ticket #101 Optional callback function on PD send
BL 2018-02-03: Ticket #190 Source filtering (IP-range) for PD subscribe
BL 2017-11-28: Ticket #180 Filtering rules for DestinationURI does not follow the standard
```

```
BL 2017-11-17: superfluous session->redID replaced by sndQueue->redId
BL 2017-11-15: Ticket #1 Unjoin on unsubscribe/delListener (finally ;-)
 BL 2017-11-10: Ticket #172 Infinite loop of message sending after PD Pull Request when registered in multicas
 BL 2017-11-10: return error in resultCode of tlp_get()
BL 2017-11-09: Ticket \#171 Wrong socket binding for multicast request messages
AHW 2017-11-08: Ticket #179 Max. number of retries (part of sendParam) of a MD request needs to be checked
BL 2017-07-31: Ticket #168 Unnecessary multicast Join on tlp_publish()
BL 2017-07-12: Ticket #164 Fix for #151 (operator '&' instead of xor)
AHW 2017-05-30: Ticket #143 tlm_replyErr() only at TRDP level allowed
AHW 2017-05-22: Ticket #158 Infinit timeout at TRDB level is 0 acc. standard
BL 2017-05-08: Compiler warnings, local prototypes added
BL 2017-03-02: Ticket #151 tlp_request: timeout-flag is not cleared
BL 2017-03-01: Ticket #149 SourceUri and DestinationUri don't with 32 characters
 BL 2017-02-10: Ticket \#137 tlc_closeSession should close the tcp socket used for md communication
BL 2017-02-10: Ticket #128 PD: Support of ComId == 0
BL 2017-02-10: Ticket #130 PD Pull: Request is always sent to the same ip address
 BL 2017-02-09: Ticket #132 tlp_publish: Check of datasize wrong if using marshaller
 BL 2017-02-08: Ticket #142: Compiler warnings / MISRA-C 2012 issues
 BL 2017-02-08: Ticket #139: Swap parameter in tlm_reply
 BL 2016-07-06: Ticket #122: 64Bit compatibility (+ compiler warnings)
 BL 2016-06-08: Ticket #120: ComIds for statistics changed to proposed 61375 errata
 BL 2016-06-01: Ticket #119 tlc_getInterval() repeatedly returns 0 after timeout
 BL 2016-02-04: Late configuration update/merging
BL 2015-12-22: Mutex optimised in closeSession
 BL 2015-12-14: Setter for default configuration added
 BL 2015-11-24: Accessor for IP address of session
 BL 2015-11-24: Ticket #104: PD telegrams with no data is never sent
 BL 2015-09-04: Ticket #99: refCon for tlc_init()
 BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed
 BL 2014-06-03: Do not return error on data-less tlp_request
BL 2014-06-02: Ticket #41: Sequence counter handling fixed
               Removing receive queue on session close added
 BL 2014-02-27: Ticket #24: trdp_if.c won't compile without MD_SUPPORT
 BL 2013-06-24: ID 125: Time-out handling and ready descriptors fixed
 BL 2013-02-01: ID 53: Zero datset size fixed for PD
 BL 2013-01-25: ID 20: Redundancy handling fixed
 BL 2013-01-08: LADDER: Removed/Changed some ladder specific code in tlp_subscribe()
 BL 2012-12-03: ID 1: "using uninitialized PD_ELE_T.pullIpAddress variable"
                ID 2: "uninitialized PD_ELE_T newPD->pNext in tlp_subscribe()"
```

## 5.17.2 Function Documentation

## 5.17.2.1 tlc\_closeSession()

Close a session.

Clean up and release all resources of that session

#### **Parameters**

| in | appHandle | The handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |
| TRDP_PARAM_ERR  | handle NULL    |

#### 5.17.2.2 tlc\_configSession()

```
EXT_DECL TRDP_ERR_T tlc_configSession (

TRDP_APP_SESSION_T appHandle,

const TRDP_MARSHALL_CONFIG_T * pMarshall,

const TRDP_PD_CONFIG_T * pPdDefault,

const TRDP_MD_CONFIG_T * pMdDefault,

const TRDP_PROCESS_CONFIG_T * pProcessConfig )
```

(Re-)configure a session.

tlc\_configSession is called by openSession, but may also be called later on to change the defaults. Only the supplied settings (pointer != NULL) will be evaluated.

#### **Parameters**

| in   | appHandle   | ndle A handle for further calls to the trdp stack  |  |
|--|---|--|--|
| in   | in pMarshall Pointer to marshalling configuration |  |  |
| in   | in pPdDefault Pointer to default PD configuration |  |  |
| in   | in pMdDefault Pointer to default MD configuration |  |  |
| in pProcessConfig Pointer to process configuration only option parameter is used here to define sub- |   | Pointer to process configuration only option parameter is used here to define session behavior all other parameters are only used to feed statistics |  |

#### **Return values**

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_INIT_ERR  | not yet inited  |
| TRDP_PARAM_ERR | parameter error |

# 5.17.2.3 tlc\_getETBTopoCount()

Set new topocount for trainwide communication.

This value is used for validating outgoing and incoming packets only!

#### **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|

# Return values

etbTopoCnt

### 5.17.2.4 tlc\_getInterval()

Get the lowest time interval for PDs.

Return the maximum time interval suitable for 'select()' so that we can send due PD packets in time. If the PD send queue is empty, return zero time

#### **Parameters**

| in appHandle The handle returned by tlc_openSession              |  | The handle returned by tlc_openSession                       |
|--|--|--|
| out  | pInterval pointer to needed interval                   |  |
| in,out   | n, out <i>pFileDesc</i> pointer to file descriptor set |  |
| out pNoDesc pointer to put no of highest used descriptors (for s |  | pointer to put no of highest used descriptors (for select()) |

#### Return values

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

## 5.17.2.5 tlc\_getOpTrainTopoCount()

Set new operational train topocount for direction/orientation sensitive communication.

This value is used for validating outgoing and incoming packets only!

### **Parameters**

| in | appHandle | The handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

| opTrnTopoCnt | New operational topocount value |
|--------------|---------------------------------|

# 5.17.2.6 tlc\_getOwnlpAddress()

Get the interface address.

#### **Parameters**

|  | out | appHandle | A handle for further calls to the trdp stack |
|--|-----|-----------|--|
|--|-----|-----------|--|

## Return values



# 5.17.2.7 tlc\_getVersion()

Return version.

Return pointer to version structure

# Return values

```
TRDP_VERSION←
T
```

# 5.17.2.8 tlc\_getVersionString()

Return a human readable version representation.

Return string in the form 'v.r.u.b'

| const | string |
|-------|--------|

#### 5.17.2.9 tlc\_init()

Initialize the TRDP stack.

Support for message data can only be excluded during compile time!

tlc\_init initializes the memory subsystem and takes a function pointer to an output function for logging.

#### **Parameters**

|   | in | pPrintDebugString | Pointer to debug print function |
|---|----|-------------------|---------------------------------|
| ĺ | in | pRefCon           | user context                    |
|   | in | pMemConfig        | Pointer to memory configuration |

## Return values

| TRDP_NO_ERR    | no error                 |  |
|----------------|--------------------------|--|
| TRDP_MEM_ERR   | memory allocation failed |  |
| TRDP_PARAM_ERR | initialization error     |  |

# 5.17.2.10 tlc\_openSession()

```
EXT_DECL TRDP_ERR_T tlc_openSession (

TRDP_APP_SESSION_T * pAppHandle,

TRDP_IP_ADDR_T ownIpAddr,

TRDP_IP_ADDR_T leaderIpAddr,

const TRDP_MARSHALL_CONFIG_T * pMarshall,

const TRDP_PD_CONFIG_T * pPdDefault,

const TRDP_MD_CONFIG_T * pMdDefault,

const TRDP_PROCESS_CONFIG_T * pProcessConfig )
```

Open a session with the TRDP stack.

tlc\_openSession returns in pAppHandle a unique handle to be used in further calls to the stack.

#### **Parameters**

| · aramon  | didineters .  |  |  |  |
|---|---|--|--|--|
| outpAppHandleA handle for further calls to the trdp stackinownIpAddrOwn IP address, can be different for each process in multihom |   | A handle for further calls to the trdp stack                                       |  |  |
|   |   | Own IP address, can be different for each process in multihoming systems, if zero, |  |  |
|   |   | the default interface / IP will be used.   |  |  |
| in  | leaderlpAddr  | derlpAddr IP address of redundancy leader  |  |  |
| in  | pMarshall   | Pointer to marshalling configuration   |  |  |
| in  | pPdDefault  | Pointer to default PD configuration  |  |  |
| in  | pMdDefault  | Pointer to default MD configuration  |  |  |
| in  | pProcessConfig Pointer to process configuration only option parameter is used here to define se |  |  |  |
| behavior all other parameters are only used to feed statistics  |   | behavior all other parameters are only used to feed statistics                     |  |  |

#### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_INIT_ERR  | not yet inited  |
| TRDP_PARAM_ERR | parameter error |
| TRDP_SOCK_ERR  | socket error    |

# 5.17.2.11 tlc\_process()

Work loop of the TRDP handler.

Search the queue for pending PDs to be sent Search the receive queue for pending PDs (time out)

#### **Parameters**

| in     | appHandle         The handle returned by tlc_openSession           pRfds         pointer to set of ready descriptors |  |
|--------|--|--|
| in     |  |  |
| in,out | pCount   | pointer to number of ready descriptors |

# Return values

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

# 5.17.2.12 tlc\_reinitSession()

# Re-Initialize.

Should be called by the application when a link-down/link-up event has occured during normal operation. We need to re-join the multicast groups...

# **Parameters**

| in | appHandle | The handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

| TRDP NO ERR   no | error |
|------------------|-------|
|------------------|-------|

#### Return values

| TRDP_NOINIT_ERR | handle invalid |
|-----------------|----------------|
| TRDP_PARAM_ERR  | handle NULL    |

# 5.17.2.13 tlc\_setETBTopoCount()

Set new topocount for trainwide communication.

This value is used for validating outgoing and incoming packets only!

#### **Parameters**

| in | appHandle  | the handle returned by tlc_openSession |
|----|------------|--|
| in | etbTopoCnt | New etbTopoCnt value                   |

## Return values

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

# 5.17.2.14 tlc\_setOpTrainTopoCount()

Set new operational train topocount for direction/orientation sensitive communication.

This value is used for validating outgoing and incoming packets only!

## **Parameters**

| in | appHandle    | The handle returned by tlc_openSession |
|----|--------------|--|
| in | opTrnTopoCnt | New operational topocount value        |

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

## 5.17.2.15 tlc\_terminate()

Un-Initialize.

Clean up and close all sessions. Mainly used for debugging/test runs. No further calls to library allowed

## Return values

| TRDP_NO_ERR    | no error               |
|----------------|------------------------|
| TRDP_INIT_ERR  | no error               |
| TRDP_MEM_ERR   | TrafficStore nothing   |
| TRDP_MUTEX_ERR | TrafficStore mutex err |

# 5.17.2.16 tlp\_get()

```
EXT_DECL TRDP_ERR_T tlp_get (
          TRDP_APP_SESSION_T appHandle,
          TRDP_SUB_T subHandle,
          TRDP_PD_INFO_T * pPdInfo,
          UINT8 * pData,
          UINT32 * pDataSize )
```

Get the last valid PD message.

This allows polling of PDs instead of event driven handling by callbacks

# **Parameters**

| in     | appHandle | the handle returned by tlc_openSession |
|--------|-----------|--|
| in     | subHandle | the handle returned by subscription    |
| in,out | pPdInfo   | pointer to application's info buffer   |
| in,out | pData     | pointer to application's data buffer   |
| in,out | pDataSize | in: size of buffer, out: size of data  |

| TRDP_NO_ERR      | no error                         |
|------------------|----------------------------------|
| TRDP_PARAM_ERR   | parameter error                  |
| TRDP_SUB_ERR     | not subscribed                   |
| TRDP_TIMEOUT_ERR | packet timed out                 |
| TRDP_NOINIT_ERR  | handle invalid                   |
| TRDP_COMID_ERR   | ComID not found when marshalling |

## 5.17.2.17 tlp\_getRedundant()

Get status of redundant Comlds.

Only the status of the first found redundancy group entry will be returned!

#### **Parameters**

| in     | appHandle the handle returned by tlc_openSession                |   |
|--------|---|---|
| in     | redId   | will be returned for all ComID's with the given redId |
| in,out | in, out pLeader TRUE if we're sending this redundancy group (le |   |

#### **Return values**

| TRDP_NO_ERR     | no error                      |
|-----------------|-------------------------------|
| TRDP_PARAM_ERR  | redld invalid or not existing |
| TRDP_NOINIT_ERR | handle invalid                |

### 5.17.2.18 tlp\_publish()

```
EXT_DECL TRDP_ERR_T tlp_publish (
           TRDP_APP_SESSION_T appHandle,
            TRDP_PUB_T * pPubHandle,
            const void * pUserRef,
             TRDP_PD_CALLBACK_T pfCbFunction,
             UINT32 comId,
             UINT32 etbTopoCnt,
             UINT32 opTrnTopoCnt,
             TRDP_IP_ADDR_T srcIpAddr,
             TRDP_IP_ADDR_T destIpAddr,
             UINT32 interval,
             UINT32 redId,
             TRDP_FLAGS_T pktFlags,
             const TRDP_SEND_PARAM_T * pSendParam,
             const UINT8 * pData,
             UINT32 dataSize )
```

Prepare for sending PD messages.

Queue a PD message, it will be send when tlc\_publish has been called

#### **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

# **Parameters**

| out | pPubHandle   | returned handle for related re/unpublish                                      |  |
|-----|--------------|---|--|
| in  | pUserRef     | user supplied value returned within the info structure of callback function   |  |
| in  | pfCbFunction | Pointer to pre-send callback function, NULL if not used                       |  |
| in  | comld        | comld of packet to send   |  |
| in  | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                        |  |
| in  | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication |  |
| in  | srclpAddr    | own IP address, 0 - srcIP will be set by the stack                            |  |
| in  | destlpAddr   | where to send the packet to   |  |
| in  | interval     | frequency of PD packet (>= 10ms) in usec                                      |  |
| in  | redId        | 0 - Non-redundant, > 0 valid redundancy group                                 |  |
| in  | pktFlags     | OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE, TRDP_FLAGS_MARSHALL,             |  |
|     |              | TRDP_FLAGS_CALLBACK   |  |
| in  | pSendParam   | optional pointer to send parameter, NULL - default parameters are used        |  |
| in  | pData        | pointer to data packet / dataset, NULL if sending starts later with tlp_put() |  |
| in  | dataSize     | size of data packet >= 0 and <= TRDP_MAX_PD_DATA_SIZE                         |  |

## Return values

| TRDP_NO_ERR     | no error                         |
|-----------------|----------------------------------|
| TRDP_PARAM_ERR  | parameter error                  |
| TRDP_MEM_ERR    | could not insert (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                   |

# 5.17.2.19 tlp\_put()

Update the process data to send.

Update previously published data. The new telegram will be sent earliest when tlc\_process is called.

# **Parameters**

| in     | appHandle | the handle returned by tlc_openSession |
|--------|-----------|--|
| in     | pubHandle | the handle returned by publish         |
| in,out | pData     | pointer to application's data buffer   |
| in,out | dataSize  | size of data                           |

| TRDP_NO_ERR    | no error   |
|----------------|--|
| TRDP_PARAM_ERR | parameter error on uninitialized parameter or changed dataSize compared to |
|                | published one  |

## Return values

| TRDP_NOPUB_ERR  | not published                    |
|-----------------|----------------------------------|
| TRDP_NOINIT_ERR | handle invalid                   |
| TRDP_COMID_ERR  | ComID not found when marshalling |

#### 5.17.2.20 tlp\_republish()

```
EXT_DECL TRDP_ERR_T tlp_republish (

TRDP_APP_SESSION_T appHandle,

TRDP_PUB_T pubHandle,

UINT32 etbTopoCnt,

UINT32 opTrnTopoCnt,

TRDP_IP_ADDR_T srcIpAddr,

TRDP_IP_ADDR_T destIpAddr)
```

Prepare for sending PD messages.

Reinitialize and queue a PD message, it will be send when tlc\_publish has been called

#### **Parameters**

| in | appHandle    | the handle returned by tlc_openSession  |  |
|----|--------------|---|--|
| in | pubHandle    | handle for related unpublish  |  |
| in | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                        |  |
| in | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication |  |
| in | srclpAddr    | own IP address, 0 - srcIP will be set by the stack                            |  |
| in | destlpAddr   | where to send the packet to   |  |

#### **Return values**

| TRDP_NO_ERR     | no error                         |
|-----------------|----------------------------------|
| TRDP_PARAM_ERR  | parameter error                  |
| TRDP_MEM_ERR    | could not insert (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                   |

# 5.17.2.21 tlp\_request()

```
UINT32 redId,
TRDP_FLAGS_T pktFlags,
const TRDP_SEND_PARAM_T * pSendParam,
const UINT8 * pData,
UINT32 dataSize,
UINT32 replyComId,
TRDP_IP_ADDR_T replyIpAddr )
```

Initiate sending PD messages (PULL).

## Send a PD request message

#### **Parameters**

| in | appHandle    | the handle returned by tlc_openSession  |
|----|--------------|---|
| in | subHandle    | handle from related subscribe   |
| in | comld        | comld of packet to be sent  |
| in | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                                |
| in | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication         |
| in | srclpAddr    | own IP address, 0 - srcIP will be set by the stack                                    |
| in | destlpAddr   | where to send the packet to   |
| in | redId        | 0 - Non-redundant, > 0 valid redundancy group   |
| in | pktFlags     | OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE, TRDP_FLAGS_MARSHALL, TRDP_FLAGS_CALLBACK |
| in | pSendParam   | optional pointer to send parameter, NULL - default parameters are used                |
| in | pData        | pointer to packet data / dataset  |
| in | dataSize     | size of packet data   |
| in | replyComId   | comld of reply (default comID of subscription)  |
| in | replylpAddr  | IP for reply  |

#### Return values

| TRDP_NO_ERR     | no error                         |
|-----------------|----------------------------------|
| TRDP_PARAM_ERR  | parameter error                  |
| TRDP_MEM_ERR    | could not insert (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                   |
| TRDP_NOSUB_ERR  | no matching subscription found   |

# 5.17.2.22 tlp\_resubscribe()

```
EXT_DECL TRDP_ERR_T tlp_resubscribe (

TRDP_APP_SESSION_T appHandle,

TRDP_SUB_T subHandle,

UINT32 etbTopoCnt,

UINT32 opTrnTopoCnt,

TRDP_IP_ADDR_T srcIpAddr1,

TRDP_IP_ADDR_T srcIpAddr2,

TRDP_IP_ADDR_T destIpAddr )
```

Reprepare for receiving PD messages.

Resubscribe to a specific PD ComID and source IP

## **Parameters**

| in | appHandle    | the handle returned by tlc_openSession  |
|----|--------------|---|
| in | subHandle    | handle for this subscription  |
| in | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                          |
| in | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication   |
| in | srclpAddr1   | Source IP address, lower address in case of address range, set to 0 if not used |
| in | srclpAddr2   | upper address in case of address range, set to 0 if not used                    |
| in | destlpAddr   | IP address to join  |

#### Return values

| TRDP_NO_ERR     | no error   |
|-----------------|--|
| TRDP_PARAM_ERR  | parameter error  |
| TRDP_MEM_ERR    | could not reserve memory (out of memory)               |
| TRDP_NOINIT_ERR | handle invalid   |
| TRDP_SOCK_ERR   | Resource (socket) not available, subscription canceled |

# 5.17.2.23 tlp\_setRedundant()

Do not send non-redundant PDs when we are follower.

Do not send redundant PD's when we are follower.

# **Parameters**

| i | n | appHandle | the handle returned by tlc_openSession                                      |
|---|---|-----------|---|
| i | n | redId     | will be set for all ComID's with the given redld, 0 to change for all redld |
| i | n | leader    | TRUE if we send   |

# Return values

| TRDP_NO_ERR     | no error                             |
|-----------------|--------------------------------------|
| TRDP_PARAM_ERR  | parameter error / redld not existing |
| TRDP_NOINIT_ERR | handle invalid                       |

# 5.17.2.24 tlp\_subscribe()

```
TRDP_SUB_T * pSubHandle,
const void * pUserRef,
TRDP_PD_CALLBACK_T pfCbFunction,
UINT32 comId,
UINT32 etbTopoCnt,
UINT32 opTrnTopoCnt,
TRDP_IP_ADDR_T srcIpAddr1,
TRDP_IP_ADDR_T srcIpAddr2,
TRDP_IP_ADDR_T destIpAddr,
TRDP_FLAGS_T pktFlags,
UINT32 timeout,
TRDP_TO_BEHAVIOR_T toBehavior)
```

Prepare for receiving PD messages.

Subscribe to a specific PD ComID and source IP.

#### **Parameters**

| in  | appHandle    | the handle returned by tlc_openSession  |  |
|-----|--------------|---|--|
| out | pSubHandle   | return a handle for this subscription   |  |
| in  | pUserRef     | user supplied value returned within the info structure                          |  |
| in  | pfCbFunction | Pointer to subscriber specific callback function, NULL to use default function  |  |
| in  | comld        | comld of packet to receive  |  |
| in  | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                          |  |
| in  | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication   |  |
| in  | srclpAddr1   | Source IP address, lower address in case of address range, set to 0 if not used |  |
| in  | srclpAddr2   | upper address in case of address range, set to 0 if not used                    |  |
| in  | pktFlags     | OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE, TRDP_FLAGS_MARSHALL,               |  |
|     |              | TRDP_FLAGS_CALLBACK   |  |
| in  | destlpAddr   | IP address to join  |  |
| in  | timeout      | timeout (>= 10ms) in usec   |  |
| in  | toBehavior   | timeout behavior  |  |

#### Return values

| TRDP_NO_ERR     | no error                                 |
|-----------------|--|
| TRDP_PARAM_ERR  | parameter error                          |
| TRDP_MEM_ERR    | could not reserve memory (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                           |

```
5.17.2.25 tlp_unpublish()
```

Stop sending PD messages.

## **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | pubHandle | the handle returned by prepare         |

#### Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_NOPUB_ERR  | not published   |
| TRDP_NOINIT_ERR | handle invalid  |

## 5.17.2.26 tlp\_unsubscribe()

```
EXT_DECL TRDP_ERR_T tlp_unsubscribe (
          TRDP_APP_SESSION_T appHandle,
          TRDP_SUB_T subHandle )
```

Stop receiving PD messages.

Unsubscribe to a specific PD ComID

## **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | subHandle | the handle for this subscription       |

# Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_NOSUB_ERR  | not subscribed  |
| TRDP_NOINIT_ERR | handle invalid  |

# 5.17.2.27 trdp\_isValidSession()

Check if the session handle is valid.

#### **Parameters**

| in | pSessionHandle | pointer to packet data (dataset) |
|----|----------------|----------------------------------|

# Return values

| TRUE  | is valid   |
|-------|------------|
| FALSE | is invalid |

# 5.17.2.28 trdp\_sessionQueue()

Get the session queue head pointer.

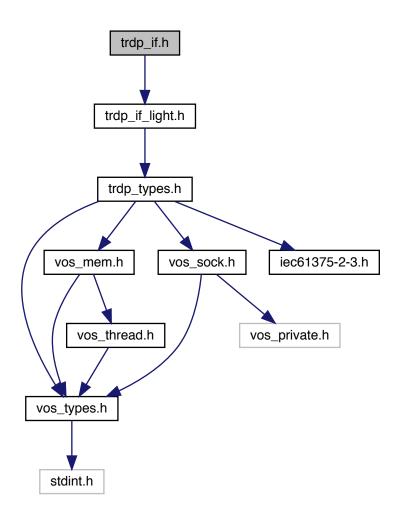
# Return values

&sSession

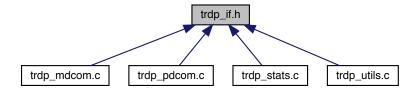
# 5.18 trdp\_if.h File Reference

Typedefs for TRDP communication.

#include "trdp\_if\_light.h"
Include dependency graph for trdp\_if.h:



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



# **Functions**

• BOOL8 trdp\_isValidSession (TRDP\_APP\_SESSION\_T pSessionHandle)

Check if the session handle is valid.

• TRDP\_APP\_SESSION\_T \* trdp\_sessionQueue (void)

Get the session queue head pointer.

# 5.18.1 Detailed Description

Typedefs for TRDP communication.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

trdp\_if.h 1264 2014-07-14 15:54:26Z bloehr

```
BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed
```

### 5.18.2 Function Documentation

#### 5.18.2.1 trdp\_isValidSession()

Check if the session handle is valid.

# **Parameters**

| in | pSessionHandle | pointer to packet data (dataset) |  |
|----|----------------|----------------------------------|--|

| TRUE  | is valid   |
|-------|------------|
| FALSE | is invalid |

#### 5.18.2.2 trdp\_sessionQueue()

Get the session queue head pointer.

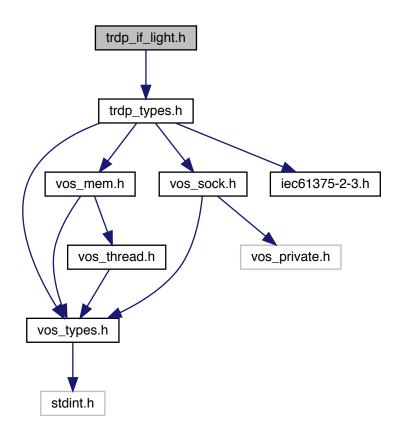
Return values

&sSession

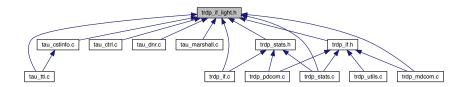
# 5.19 trdp\_if\_light.h File Reference

TRDP Light interface functions (API)

#include "trdp\_types.h"
Include dependency graph for trdp\_if\_light.h:



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



#### **Functions**

• EXT\_DECL TRDP\_ERR\_T tlc\_init (const TRDP\_PRINT\_DBG\_T pPrintDebugString, void \*pRefCon, const TRDP MEM CONFIG T \*pMemConfig)

Support for message data can only be excluded during compile time!

• EXT\_DECL TRDP\_ERR\_T tlc\_openSession (TRDP\_APP\_SESSION\_T \*pAppHandle, TRDP\_IP\_ADDR ← \_ T ownlpAddr, TRDP\_IP\_ADDR\_T leaderlpAddr, const TRDP\_MARSHALL\_CONFIG\_T \*pMarshall, const TRDP\_PD\_CONFIG\_T \*pPdDefault, const TRDP\_MD\_CONFIG\_T \*pMdDefault, const TRDP\_PROCES ← S CONFIG\_T \*pProcessConfig)

Open a session with the TRDP stack.

- EXT\_DECL TRDP\_ERR\_T tlc\_reinitSession (TRDP\_APP\_SESSION\_T appHandle) Re-Initialize.
- EXT\_DECL TRDP\_ERR\_T tlc\_configSession (TRDP\_APP\_SESSION\_T appHandle, const TRDP\_MAR← SHALL\_CONFIG\_T \*pMarshall, const TRDP\_PD\_CONFIG\_T \*pPdDefault, const TRDP\_MD\_CONFIG\_T \*pMdDefault, const TRDP\_PROCESS\_CONFIG\_T \*pProcessConfig)

(Re-)configure a session.

• EXT\_DECL TRDP\_ERR\_T tlc\_closeSession (TRDP\_APP\_SESSION\_T appHandle)

Close a session.

• EXT DECL TRDP ERR T tlc terminate (void)

Un-Initialize.

EXT\_DECL TRDP\_ERR\_T tlc\_setETBTopoCount (TRDP\_APP\_SESSION\_T appHandle, UINT32 etbTopo

 Cnt)

Set new topocount for trainwide communication.

- EXT DECL UINT32 tlc getETBTopoCount (TRDP APP SESSION T appHandle)
  - Set new topocount for trainwide communication.
- EXT\_DECL TRDP\_ERR\_T tlc\_setOpTrainTopoCount (TRDP\_APP\_SESSION\_T appHandle, UINT32 op 

  TrnTopoCnt)

Set new operational train topocount for direction/orientation sensitive communication.

EXT\_DECL\_UINT32 tlc\_getOpTrainTopoCount (TRDP\_APP\_SESSION\_T appHandle)

Set new operational train topocount for direction/orientation sensitive communication.

• EXT\_DECL TRDP\_ERR\_T tlc\_freeBuf (TRDP\_APP\_SESSION\_T appHandle, char \*pBuf)

Frees the buffer reserved by the TRDP layer.

• EXT\_DECL TRDP\_ERR\_T tlc\_getInterval (TRDP\_APP\_SESSION\_T appHandle, TRDP\_TIME\_T \*pInterval, TRDP\_FDS\_T \*pFileDesc, INT32 \*pNoDesc)

Get the lowest time interval for PDs.

• EXT\_DECL TRDP\_ERR\_T tlc\_process (TRDP\_APP\_SESSION\_T appHandle, TRDP\_FDS\_T \*pRfds, INT32 \*pCount)

Work loop of the TRDP handler.

EXT\_DECL TRDP\_IP\_ADDR\_T tlc\_getOwnlpAddress (TRDP\_APP\_SESSION\_T appHandle)

Get the interface address.

• EXT\_DECL TRDP\_ERR\_T tlp\_publish (TRDP\_APP\_SESSION\_T appHandle, TRDP\_PUB\_T \*pPubHandle, const void \*pUserRef, TRDP\_PD\_CALLBACK\_T pfCbFunction, UINT32 comId, UINT32 etbTopoCnt, UIN← T32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srcIpAddr, TRDP\_IP\_ADDR\_T destlpAddr, UINT32 interval, UINT32 redId, TRDP\_FLAGS\_T pktFlags, const TRDP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, U← INT32 dataSize)

Prepare for sending PD messages.

- EXT\_DECL TRDP\_ERR\_T tlp\_republish (TRDP\_APP\_SESSION\_T appHandle, TRDP\_PUB\_T pubHandle, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr, TRDP\_IP\_ADDR\_T destlpAddr)

  Prepare for sending PD messages.
- EXT\_DECL TRDP\_ERR\_T tlp\_unpublish (TRDP\_APP\_SESSION\_T appHandle, TRDP\_PUB\_T pubHandle) Stop sending PD messages.
- EXT\_DECL TRDP\_ERR\_T tlp\_put (TRDP\_APP\_SESSION\_T appHandle, TRDP\_PUB\_T pubHandle, const UINT8 \*pData, UINT32 dataSize)

Update the process data to send.

• EXT\_DECL TRDP\_ERR\_T tlp\_setRedundant (TRDP\_APP\_SESSION\_T appHandle, UINT32 redId, BOOL8 leader)

Do not send redundant PD's when we are follower.

• EXT\_DECL TRDP\_ERR\_T tlp\_getRedundant (TRDP\_APP\_SESSION\_T appHandle, UINT32 redId, BOOL8 \*pLeader)

Get status of redundant Comlds.

• EXT\_DECL TRDP\_ERR\_T tlp\_request (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T subHandle, UINT32 comId, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr, TRDP\_IP\_AD → DR\_T destIpAddr, UINT32 redId, TRDP\_FLAGS\_T pktFlags, const TRDP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize, UINT32 replyComId, TRDP\_IP\_ADDR\_T replyIpAddr)

Initiate sending PD messages (PULL).

• EXT\_DECL TRDP\_ERR\_T tlp\_subscribe (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T \*pSub↔ Handle, const void \*pUserRef, TRDP\_PD\_CALLBACK\_T pfCbFunction, UINT32 comId, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr1, TRDP\_IP\_ADDR\_T srclpAddr2, TRDP\_IP\_ADD↔ R\_T destlpAddr, TRDP\_FLAGS\_T pktFlags, UINT32 timeout, TRDP\_TO\_BEHAVIOR\_T toBehavior)

Prepare for receiving PD messages.

• EXT\_DECL\_TRDP\_ERR\_T\_tlp\_resubscribe (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T sub ← Handle, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr1, TRDP\_IP\_ADDR\_T srclpAddr2, TRDP\_IP\_ADDR\_T destlpAddr)

Reprepare for receiving PD messages.

• EXT\_DECL TRDP\_ERR\_T tlp\_unsubscribe (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T sub ← Handle)

Stop receiving PD messages.

• EXT\_DECL TRDP\_ERR\_T tlp\_get (TRDP\_APP\_SESSION\_T appHandle, TRDP\_SUB\_T subHandle, TR

DP\_PD\_INFO\_T \*pPdInfo, UINT8 \*pData, UINT32 \*pDataSize)

Get the last valid PD message.

EXT\_DECL TRDP\_ERR\_T tlm\_notify (TRDP\_APP\_SESSION\_T appHandle, const void \*pUserRef, TRD← P\_MD\_CALLBACK\_T pfCbFunction, UINT32 comld, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_← IP\_ADDR\_T srclpAddr, TRDP\_IP\_ADDR\_T destlpAddr, TRDP\_FLAGS\_T pktFlags, const TRDP\_SEND\_← PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize, const TRDP\_URI\_USER\_T sourceURI, const TRDP\_URI\_USER\_T destURI)

Initiate sending MD notification message.

• EXT\_DECL TRDP\_ERR\_T tlm\_request (TRDP\_APP\_SESSION\_T appHandle, const void \*pUserRef, T ← RDP\_MD\_CALLBACK\_T pfCbFunction, TRDP\_UUID\_T \*pSessionId, UINT32 comId, UINT32 etbTopo ← Cnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srcIpAddr, TRDP\_IP\_ADDR\_T destIpAddr, TRDP\_FLA ← GS\_T pktFlags, UINT32 numReplies, UINT32 replyTimeout, const TRDP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize, const TRDP\_URI\_USER\_T sourceURI, const TRDP\_URI\_USER\_T destURI)

Initiate sending MD request message.

EXT\_DECL TRDP\_ERR\_T tlm\_confirm (TRDP\_APP\_SESSION\_T appHandle, const TRDP\_UUID\_T \*p↔
 SessionId, UINT16 userStatus, const TRDP\_SEND\_PARAM\_T \*pSendParam)

Initiate sending MD confirm message.

 EXT\_DECL TRDP\_ERR\_T tlm\_abortSession (TRDP\_APP\_SESSION\_T appHandle, const TRDP\_UUID\_T \*pSessionId)

Cancel an open session.

• EXT\_DECL TRDP\_ERR\_T tlm\_addListener (TRDP\_APP\_SESSION\_T appHandle, TRDP\_LIS\_T \*pListen← Handle, const void \*pUserRef, TRDP\_MD\_CALLBACK\_T pfCbFunction, BOOL8 comldListener, UINT32 comld, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr1, TRDP\_IP\_ADDR\_← T srclpAddr2, TRDP\_IP\_ADDR\_T mcDestlpAddr, TRDP\_FLAGS\_T pktFlags, const TRDP\_URI\_USER\_T srcURI, const TRDP\_URI\_USER\_T destURI)

Subscribe to MD messages.

• EXT\_DECL TRDP\_ERR\_T tlm\_readdListener (TRDP\_APP\_SESSION\_T appHandle, TRDP\_LIS\_T listen ← Handle, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr, TRDP\_IP\_ADDR\_← T srclpAddr2, TRDP\_IP\_ADDR\_T mcDestlpAddr)

Resubscribe to MD messages.

• EXT\_DECL TRDP\_ERR\_T tlm\_delListener (TRDP\_APP\_SESSION\_T appHandle, TRDP\_LIS\_T listen ← Handle)

Remove Listener.

TRDP\_ERR\_T tlm\_reply (TRDP\_APP\_SESSION\_T appHandle, const TRDP\_UUID\_T \*pSessionId, UINT32 comId, UINT16 userStatus, const TRDP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize)

Send a MD reply message.

• TRDP\_ERR\_T tlm\_replyQuery (TRDP\_APP\_SESSION\_T appHandle, const TRDP\_UUID\_T \*pSessionId, UINT32 comId, UINT32 comId, UINT32 confirmTimeout, const TRDP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize)

Send a MD reply query message.

EXT\_DECL const CHAR8 \* tlc\_getVersionString (void)

Return a human readable version representation.

EXT\_DECL const TRDP\_VERSION\_T \* tlc\_getVersion (void)

Return version.

• EXT\_DECL TRDP\_ERR\_T tlc\_getStatistics (TRDP\_APP\_SESSION\_T appHandle, TRDP\_STATISTICS\_T \*pStatistics)

Return statistics.

 EXT\_DECL TRDP\_ERR\_T tlc\_getSubsStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNum← Subs, TRDP\_SUBS\_STATISTICS\_T \*pStatistics)

Return PD subscription statistics.

• EXT\_DECL TRDP\_ERR\_T tlc\_getPubStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNumPub, TRDP\_PUB\_STATISTICS\_T \*pStatistics)

Return PD publish statistics.

EXT\_DECL\_TRDP\_ERR\_T\_tlc\_getUdpListStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*p
 — NumList, TRDP\_LIST\_STATISTICS\_T \*pStatistics)

Return UDP MD listener statistics.

• EXT\_DECL TRDP\_ERR\_T tlc\_getTcpListStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNum← List, TRDP\_LIST\_STATISTICS\_T \*pStatistics)

Return TCP MD listener statistics.

 EXT\_DECL TRDP\_ERR\_T tlc\_getRedStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNumRed, TRDP\_RED\_STATISTICS\_T \*pStatistics)

Return redundancy group statistics.

EXT\_DECL TRDP\_ERR\_T tlc\_getJoinStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNumJoin, UINT32 \*plpAddr)

Return join statistics.

• EXT\_DECL TRDP\_ERR\_T tlc\_resetStatistics (TRDP\_APP\_SESSION\_T appHandle)

Reset statistics.

## 5.19.1 Detailed Description

TRDP Light interface functions (API)

Low level functions for communicating using the TRDP protocol

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

#### trdp\_if\_light.h 1711 2018-03-06 16:11:32Z bloehr

```
BL 2018-03-06: Ticket #101 Optional callback function on PD send
BL 2018-02-03: Ticket #190 Source filtering (IP-range) for PD subscribe
BL 2017-11-28: Ticket #180 Filtering rules for DestinationURI does not follow the standard
AHW 2017-11-08: Ticket #179 Max. number of retries (part of sendParam) of a MD request needs to be checked
AHW 2017-05-30: Ticket #143 tlm_replyErr() only at TRDP level allowed
BL 2015-11-24: Accessor for IP address of session
BL 2015-09-04: Ticket #99: refCon for tlc_init()
BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed
```

# 5.19.2 Function Documentation

# 5.19.2.1 tlc\_closeSession()

Close a session.

Clean up and release all resources of that session

#### **Parameters**

| in | appHandle | The handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

#### Return values

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |
| TRDP_PARAM_ERR  | handle NULL    |

#### 5.19.2.2 tlc\_configSession()

```
EXT_DECL TRDP_ERR_T tlc_configSession (

TRDP_APP_SESSION_T appHandle,

const TRDP_MARSHALL_CONFIG_T * pMarshall,

const TRDP_PD_CONFIG_T * pPdDefault,

const TRDP_MD_CONFIG_T * pMdDefault,

const TRDP_PROCESS_CONFIG_T * pProcessConfig )
```

(Re-)configure a session.

tlc\_configSession is called by openSession, but may also be called later on to change the defaults.

## **Parameters**

| in | appHandle      | A handle for further calls to the trdp stack  |
|----|----------------|---|
| in | pMarshall      | Pointer to marshalling configuration  |
| in | pPdDefault     | Pointer to default PD configuration   |
| in | pMdDefault     | Pointer to default MD configuration   |
| in | pProcessConfig | Pointer to process configuration only option parameter is used here to define session |
|    |                | behavior all other parameters are only used to feed statistics                        |

## Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_INIT_ERR  | not yet inited  |
| TRDP_PARAM_ERR | parameter error |

tlc\_configSession is called by openSession, but may also be called later on to change the defaults. Only the supplied settings (pointer != NULL) will be evaluated.

# Parameters

| in | appHandle      | A handle for further calls to the trdp stack  |
|----|----------------|---|
| in | pMarshall      | Pointer to marshalling configuration  |
| in | pPdDefault     | Pointer to default PD configuration   |
| in | pMdDefault     | Pointer to default MD configuration   |
| in | pProcessConfig | Pointer to process configuration only option parameter is used here to define session |
|    |                | behavior all other parameters are only used to feed statistics                        |

| TRDP_NO_ERR | no error |
|-------------|----------|
|-------------|----------|

## Return values

| TRDP_INIT_ERR  | not yet inited  |
|----------------|-----------------|
| TRDP_PARAM_ERR | parameter error |

# 5.19.2.3 tlc\_freeBuf()

Frees the buffer reserved by the TRDP layer.

#### **Parameters**

|   | in | appHandle | The handle returned by tlc_openSession |
|---|----|-----------|--|
| Ī | in | pBuf      | pointer to the buffer to be freed      |

## Return values

| TRDP_NO_ERR     | no error               |
|-----------------|------------------------|
| TRDP_NOINIT_ERR | handle invalid         |
| TRDP_PARAM_ERR  | buffer pointer invalid |

# 5.19.2.4 tlc\_getETBTopoCount()

Set new topocount for trainwide communication.

This value is used for validating outgoing and incoming packets only!

# **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

## Return values

etbTopoCnt

#### 5.19.2.5 tlc\_getInterval()

Get the lowest time interval for PDs.

Return the maximum time interval suitable for 'select()' so that we can send due PD packets in time. If the PD send queue is empty, return zero time

#### **Parameters**

| in     | appHandle The handle returned by tlc_openSession |  |
|--------|--|--|
| out    | pInterval  | pointer to needed interval                           |
| in,out | pFileDesc  | pointer to file descriptor set                       |
| out    | pNoDesc  | pointer to put no of used descriptors (for select()) |

#### **Return values**

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

Return the maximum time interval suitable for 'select()' so that we can send due PD packets in time. If the PD send queue is empty, return zero time

#### **Parameters**

| in     | appHandle The handle returned by tlc_openSession |  |
|--------|--|--|
| out    | pInterval  | pointer to needed interval                                   |
| in,out | pFileDesc  | pointer to file descriptor set                               |
| out    | pNoDesc  | pointer to put no of highest used descriptors (for select()) |

#### **Return values**

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

# 5.19.2.6 tlc\_getJoinStatistics()

## Return join statistics.

Memory for statistics information must be provided by the user. The reserved length is given via pNumJoin implicitely.

## **Parameters**

| in                 | appHandle | the handle returned by tlc_openSession        |
|--------------------|-----------|---|
| in,out             | pNumJoin  | Pointer to the number of joined IP Adresses   |
| out <i>plpAddr</i> |           | Pointer to a list with the joined IP adresses |

#### Return values

| TRDP_NO_ERR     | no error                            |
|-----------------|-------------------------------------|
| TRDP_NOINIT_ERR | handle invalid                      |
| TRDP_PARAM_ERR  | parameter error                     |
| TRDP_MEM_ERR    | there are more items than requested |

Memory for statistics information must be provided by the user.

#### **Parameters**

| in  |      | appHandle | the handle returned by tlc_openSession        |
|-----|------|-----------|---|
| in, | out, | pNumJoin  | Pointer to the number of joined IP Adresses   |
| out | Ī    | plpAddr   | Pointer to a list with the joined IP adresses |

#### Return values

| TRDP_NO_ERR     | no error                            |
|-----------------|-------------------------------------|
| TRDP_NOINIT_ERR | handle invalid                      |
| TRDP_PARAM_ERR  | parameter error                     |
| TRDP_MEM_ERR    | there are more items than requested |

# 5.19.2.7 tlc\_getOpTrainTopoCount()

Set new operational train topocount for direction/orientation sensitive communication.

This value is used for validating outgoing and incoming packets only!

## **Parameters**

| in | appHandle | The handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

| opTrnTopoCnt | New operational topocount value |
|--------------|---------------------------------|
|--------------|---------------------------------|

# 5.19.2.8 tlc\_getOwnlpAddress()

Get the interface address.

#### **Parameters**

| out | appHandle | A handle for further calls to the trdp stack |
|-----|-----------|--|
|-----|-----------|--|

## Return values

| real⇔ |  |
|-------|--|
| IP    |  |

## 5.19.2.9 tlc\_getPubStatistics()

Return PD publish statistics.

Memory for statistics information must be provided by the user. The reserved length is given via pNumPub implicitely.

#### **Parameters**

| in      | appHandle   | the handle returned by tlc_openSession                    |
|---------|-------------|---|
| in, out | pNumPub     | Pointer to the number of publishers                       |
| out     | pStatistics | pointer to a list with the publish statistics information |

## Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

Memory for statistics information must be provided by the user.

#### **Parameters**

| in     | appHandle   | the handle returned by tlc_openSession                    |
|--------|-------------|---|
| in,out | pNumPub     | Pointer to the number of publishers                       |
| out    | pStatistics | Pointer to a list with the publish statistics information |

## Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

# 5.19.2.10 tlc\_getRedStatistics()

Return redundancy group statistics.

Memory for statistics information must be provided by the user. The reserved length is given via pNumRed implicitely.

## **Parameters**

| in     | appHandle   | the handle returned by tlc_openSession                  |
|--------|-------------|---|
| in,out | pNumRed     | Pointer to the number of redundancy groups              |
| out    | pStatistics | Pointer to a list with the redundancy group information |

# Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

Memory for statistics information must be provided by the user.

# **Parameters**

| in     | appHandle   | the handle returned by tlc_openSession                  |
|--------|-------------|---|
| in,out | pNumRed     | Pointer to the number of redundancy groups              |
| out    | pStatistics | Pointer to a list with the redundancy group information |

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

# 5.19.2.11 tlc\_getStatistics()

```
EXT_DECL TRDP_ERR_T tlc_getStatistics (
            TRDP_APP_SESSION_T appHandle,
            TRDP_STATISTICS_T * pStatistics )
```

Return statistics.

Memory for statistics information must be preserved by the user.

#### **Parameters**

| in | 1  | appHandle   | the handle returned by tlc_openSession             |
|----|----|-------------|--|
| ou | ıt | pStatistics | Pointer to statistics for this application session |

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_NOINIT_ERR | handle invalid  |
| TRDP_PARAM_ERR  | parameter error |

Memory for statistics information must be provided by the user.

#### **Parameters**

| in  | appHandle   | the handle returned by tlc_openSession             |
|-----|-------------|--|
| out | pStatistics | Pointer to statistics for this application session |

# Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_NOINIT_ERR | handle invalid  |
| TRDP_PARAM_ERR  | parameter error |

# 5.19.2.12 tlc\_getSubsStatistics()

```
EXT_DECL TRDP_ERR_T tlc_getSubsStatistics (
             TRDP_APP_SESSION_T appHandle,
             UINT16 * pNumSubs,
             TRDP_SUBS_STATISTICS_T * pStatistics )
```

Return PD subscription statistics.

Memory for statistics information must be provided by the user. The reserved length is given via pNumSub implicitely.

# **Parameters**

| ın     | аррнапаіе   | the handle returned by tlc_openSession  |  |
|--------|-------------|---|--|
| in,out | pNumSubs    | In: The number of subscriptions requested Out: Number of subscriptions returned |  |
| in,out | pStatistics | Pointer to an array with the subscription statistics information                |  |

## Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

Memory for statistics information must be provided by the user.

## **Parameters**

| in     | appHandle   | the handle returned by tlc_openSession  |
|--------|-------------|---|
| in,out | pNumSubs    | In: The number of subscriptions requested Out: Number of subscriptions returned |
| in,out | pStatistics | Pointer to an array with the subscription statistics information                |

## Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

# 5.19.2.13 tlc\_getTcpListStatistics()

Return TCP MD listener statistics.

Memory for statistics information must be provided by the user. The reserved length is given via pNumLis implicitely.

#### **Parameters**

| in     | appHandle   | the handle returned by tlc_openSession                     |
|--------|-------------|--|
| in,out | pNumList    | Pointer to the number of listeners                         |
| out    | pStatistics | Pointer to a list with the listener statistics information |

| i i o tai i i vai a o o |   |
|-------------------------|---|
| TRDP_NO_ERR             | no error                                    |
| TRDP_NOINIT_ERR         | handle invalid                              |
| TRDP_PARAM_ERR          | parameter error                             |
| TRDP_MEM_ERR            | there are more subscriptions than requested |

# 5.19.2.14 tlc\_getUdpListStatistics()

Return UDP MD listener statistics.

Memory for statistics information must be provided by the user. The reserved length is given via pNumLis implicitely.

#### **Parameters**

| in     | appHandle   | the handle returned by tlc_openSession                     |
|--------|-------------|--|
| in,out | pNumList    | Pointer to the number of listeners                         |
| out    | pStatistics | Pointer to a list with the listener statistics information |

## Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

# 5.19.2.15 tlc\_getVersion()

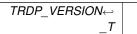
Return version.

Return pointer to version structure

## Return values

| const | TRDP_VERSION← |
|-------|---------------|
|       | _T            |

Return pointer to version structure



### 5.19.2.16 tlc\_getVersionString()

Return a human readable version representation.

Return string in the form 'v.r.u.b'

#### Return values

| const | string |
|-------|--------|
|-------|--------|

# 5.19.2.17 tlc\_init()

Support for message data can only be excluded during compile time!

Initialize the TRDP stack.

tlc\_init initializes the memory subsystem and takes a function pointer to an output function for logging.

#### **Parameters**

| in | pPrintDebugString | Pointer to debug print function |
|----|-------------------|---------------------------------|
| in | pRefCon           | user context                    |
| in | pMemConfig        | Pointer to memory configuration |

# Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | memory allocation failed |
| TRDP_PARAM_ERR | initialization error     |

Support for message data can only be excluded during compile time!

tlc\_init initializes the memory subsystem and takes a function pointer to an output function for logging.

### **Parameters**

| in | pPrintDebugString | Pointer to debug print function |
|----|-------------------|---------------------------------|
| in | pRefCon           | user context                    |
| in | pMemConfig        | Pointer to memory configuration |

#### Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | memory allocation failed |
| TRDP_PARAM_ERR | initialization error     |

# 5.19.2.18 tlc\_openSession()

```
EXT_DECL TRDP_ERR_T tlc_openSession (

TRDP_APP_SESSION_T * pAppHandle,

TRDP_IP_ADDR_T ownIpAddr,

TRDP_IP_ADDR_T leaderIpAddr,

const TRDP_MARSHALL_CONFIG_T * pMarshall,

const TRDP_PD_CONFIG_T * pPdDefault,

const TRDP_MD_CONFIG_T * pMdDefault,

const TRDP_PROCESS_CONFIG_T * pProcessConfig )
```

Open a session with the TRDP stack.

tlc\_openSession returns in pAppHandle a unique handle to be used in further calls to the stack.

#### **Parameters**

| out | pAppHandle     | A handle for further calls to the trdp stack  |
|-----|----------------|---|
| in  | ownlpAddr      | Own IP address, can be different for each process in multihoming systems, if zero,    |
|     |                | the default interface / IP will be used.  |
| in  | leaderlpAddr   | IP address of redundancy leader   |
| in  | pMarshall      | Pointer to marshalling configuration  |
| in  | pPdDefault     | Pointer to default PD configuration   |
| in  | pMdDefault     | Pointer to default MD configuration   |
| in  | pProcessConfig | Pointer to process configuration only option parameter is used here to define session |
|     |                | behavior all other parameters are only used to feed statistics                        |

# Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_INIT_ERR  | not yet inited  |
| TRDP_PARAM_ERR | parameter error |
| TRDP_SOCK_ERR  | socket error    |

## 5.19.2.19 tlc\_process()

Work loop of the TRDP handler.

Search the queue for pending PDs to be sent Search the receive queue for pending PDs (time out)

### **Parameters**

| in     | appHandle | The handle returned by tlc_openSession |
|--------|-----------|--|
| in     | pRfds     | pointer to set of ready descriptors    |
| in,out | pCount    | pointer to number of ready descriptors |

### Return values

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

## 5.19.2.20 tlc\_reinitSession()

#### Re-Initialize.

Should be called by the application when a link-down/link-up event has occured during normal operation. We need to re-join the multicast groups...

#### **Parameters**

| i | .n | appHandle | The handle returned by tlc_openSession |
|---|----|-----------|--|
|---|----|-----------|--|

## Return values

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |
| TRDP_PARAM_ERR  | handle NULL    |

## 5.19.2.21 tlc\_resetStatistics()

## Reset statistics.

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

#### **Return values**

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_NOINIT_ERR | handle invalid  |
| TRDP_PARAM_ERR  | parameter error |

## 5.19.2.22 tlc\_setETBTopoCount()

Set new topocount for trainwide communication.

This value is used for validating outgoing and incoming packets only!

#### **Parameters**

| in | appHandle  | The handle returned by tlc_openSession |
|----|------------|--|
| in | etbTopoCnt | New topocount value                    |

This value is used for validating outgoing and incoming packets only!

## **Parameters**

| in | appHandle  | the handle returned by tlc_openSession |
|----|------------|--|
| in | etbTopoCnt | New etbTopoCnt value                   |

### Return values

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

#### 5.19.2.23 tlc\_setOpTrainTopoCount()

Set new operational train topocount for direction/orientation sensitive communication.

This value is used for validating outgoing and incoming packets only!

| in | appHandle    | The handle returned by tlc_openSession |
|----|--------------|--|
| in | opTrnTopoCnt | New operational topocount value        |

This value is used for validating outgoing and incoming packets only!

#### **Parameters**

| in | appHandle    | The handle returned by tlc_openSession |
|----|--------------|--|
| in | opTrnTopoCnt | New operational topocount value        |

### Return values

| TRDP_NO_ERR     | no error       |
|-----------------|----------------|
| TRDP_NOINIT_ERR | handle invalid |

## 5.19.2.24 tlc\_terminate()

Un-Initialize.

Clean up and close all sessions. Mainly used for debugging/test runs. No further calls to library allowed

#### Return values

```
TRDP_NO_ERR no error
```

Clean up and close all sessions. Mainly used for debugging/test runs. No further calls to library allowed

#### **Return values**

| TRDP_NO_ERR    | no error               |
|----------------|------------------------|
| TRDP_INIT_ERR  | no error               |
| TRDP_MEM_ERR   | TrafficStore nothing   |
| TRDP_MUTEX_ERR | TrafficStore mutex err |

## 5.19.2.25 tlm\_abortSession()

Cancel an open session.

Abort an open session; any pending messages will be dropped

| in            | appHandle                        | the handle returned by tlc_openSession |
|---------------|----------------------------------|--|
| in<br>Generat | p⇔<br>ed by Doxygen<br>SessionId | Session ID returned by request         |

### Return values

| TRDP_NO_ERR         | no error        |
|---------------------|-----------------|
| TRDP_NO_SESSION_ERR | no such session |
| TRDP_NOINIT_ERR     | handle invalid  |

#### 5.19.2.26 tlm\_addListener()

```
EXT_DECL TRDP_ERR_T tlm_addListener (

TRDP_APP_SESSION_T appHandle,

TRDP_LIS_T * pListenHandle,

const void * pUserRef,

TRDP_MD_CALLBACK_T pfCbFunction,

BOOL8 comIdListener,

UINT32 comId,

UINT32 etbTopoCnt,

UINT32 opTrnTopoCnt,

TRDP_IP_ADDR_T srcIpAddr1,

TRDP_IP_ADDR_T srcIpAddr2,

TRDP_IP_ADDR_T mcDestIpAddr,

TRDP_FLAGS_T pktFlags,

const TRDP_URI_USER_T srcURI,

const TRDP_URI_USER_T destURI)
```

## Subscribe to MD messages.

Add a listener to TRDP to get notified when messages are received

#### **Parameters**

| in  | appHandle   | the handle returned by tlc openSession  |  |
|-----|---|---|--|
| out | pListenHandle   | Handle for this listener returned   |  |
| in  | pUserRef  | user supplied value returned with received message                            |  |
| in  | pfCbFunction  | Pointer to listener specific callback function, NULL to use default function  |  |
| in  | comldListener   | set TRUE if comld shall be observed   |  |
| in  | comId   | comld to be observed  |  |
| in  | etbTopoCnt  | ETB topocount to use, 0 if consist local communication                        |  |
| in  | opTrnTopoCnt  | operational topocount, != 0 for orientation/direction sensitive communication |  |
| in  | srclpAddr1  |   |  |
| in  | srclpAddr2  |   |  |
| in  | mcDestlpAddr multicast group to listen on                             |   |  |
| in  | pktFlags  | oktFlags OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_MARSHALL, TRDP_PLAGS_TC       |  |
| in  | srcURI only functional group of source URI, set 0 if not used         |   |  |
| in  | n destURI only functional group of destination URI, set 0 if not used |   |  |

#### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | parameter error |
| TRDP_MEM_ERR   | out of memory   |

### Return values

| TRDP_NOINIT_ERR | handle invalid |
|-----------------|----------------|
|-----------------|----------------|

## 5.19.2.27 tlm\_confirm()

Initiate sending MD confirm message.

Send a MD confirmation message User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session

#### **Parameters**

| in | appHandle  | the handle returned by tlc_openSession                          |  |
|----|--|---|--|
| in | pSessionId   | SessionId Session ID returned by request                        |  |
| in | userStatus Info for requester about application errors |   |  |
| in | pSendParam   | Pointer to send parameters, NULL to use default send parameters |  |

## Return values

| TRDP_NO_ERR         | no error        |
|---------------------|-----------------|
| TRDP_PARAM_ERR      | parameter error |
| TRDP_MEM_ERR        | out of memory   |
| TRDP_NO_SESSION_ERR | no such session |
| TRDP_NOINIT_ERR     | handle invalid  |

## 5.19.2.28 tlm\_delListener()

### Remove Listener.

| in                      | appHandle | the handle returned by tlc_openSession |  |
|-------------------------|-----------|--|--|
| out <i>listenHandle</i> |           | Handle for this listener               |  |

### Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_NOINIT_ERR | handle invalid  |

## 5.19.2.29 tlm\_notify()

```
EXT_DECL TRDP_ERR_T tlm_notify (

TRDP_APP_SESSION_T appHandle,
const void * pUserRef,

TRDP_MD_CALLBACK_T pfCbFunction,
UINT32 comId,
UINT32 etbTopoCnt,
UINT32 opTrnTopoCnt,
TRDP_IP_ADDR_T srcIpAddr,
TRDP_IP_ADDR_T destIpAddr,
TRDP_FLAGS_T pktFlags,
const TRDP_SEND_PARAM_T * pSendParam,
const UINT8 * pData,
UINT32 dataSize,
const TRDP_URI_USER_T sourceURI,
const TRDP_URI_USER_T destURI)
```

Initiate sending MD notification message.

### Send a MD notification message

## Parameters

| in | n appHandle the handle returned by tlc_openSession                                |   |  |  |
|----|---|---|--|--|
| in | pUserRef user supplied value returned with reply                                  |   |  |  |
| in | pfCbFunction  | Pointer to listener specific callback function, NULL to use default function  |  |  |
| in | comld comld of packet to be sent  |   |  |  |
| in | etbTopoCnt  | ETB topocount to use, 0 if consist local communication                        |  |  |
| in | opTrnTopoCnt  | operational topocount, != 0 for orientation/direction sensitive communication |  |  |
| in | srclpAddr   | own IP address, 0 - srcIP will be set by the stack                            |  |  |
| in | in destlpAddr where to send the packet to   |   |  |  |
| in | in pktFlags OPTIONS: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_MARSHALL, TRDP_PLAGS          |   |  |  |
| in | pSendParam optional pointer to send parameter, NULL - default parameters are used |   |  |  |
| in | pData   | pointer to packet data / dataset  |  |  |
| in | n dataSize size of packet data  |   |  |  |
| in | sourceURI   | only functional group of source URI   |  |  |
| in | destURI   | only functional group of destination URI                                      |  |  |

#### Return values

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | parameter error |
| TRDP_MEM_ERR   | out of memory   |

### Return values

| TRDP_NOINIT_ERR | handle invalid |
|-----------------|----------------|
|-----------------|----------------|

## 5.19.2.30 tlm\_readdListener()

```
EXT_DECL TRDP_ERR_T tlm_readdListener (

TRDP_APP_SESSION_T appHandle,

TRDP_LIS_T listenHandle,

UINT32 etbTopoCnt,

UINT32 opTrnTopoCnt,

TRDP_IP_ADDR_T srcIpAddr,

TRDP_IP_ADDR_T srcIpAddr2,

TRDP_IP_ADDR_T mcDestIpAddr)
```

Resubscribe to MD messages.

Readd a listener after topoCount changes to get notified when messages are received

#### **Parameters**

| in  | appHandle   | the handle returned by tlc_openSession   |  |
|-----|---|--|--|
| out | listenHandle  | tenHandle Handle for this listener   |  |
| in  | etbTopoCnt  | nt ETB topocount to use, 0 if consist local communication                        |  |
| in  | opTrnTopoCnt  | nt operational topocount, != 0 for orientation/direction sensitive communication |  |
| in  | srclpAddr Source IP address, lower address in case of address range, set 0 if not use |  |  |
| in  | in srclpAddr2 upper address in case of address range, set 0 if not used               |  |  |
| in  | mcDestIpAddr multicast group to listen on   |  |  |

### Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_MEM_ERR    | out of memory   |
| TRDP_NOINIT_ERR | handle invalid  |

## 5.19.2.31 tlm\_reply()

Send a MD reply message.

Send a MD reply message after receiving an request User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session

#### **Parameters**

| in  | appHandle   | the handle returned by tlc_openSession |  |
|---|---|--|--|
| in pSessionId Session ID returned by indication |   | Session ID returned by indication      |  |
| in  | n comld of packet to be sent  |  |  |
| in  | userStatus Info for requester about application errors                    |  |  |
| in  | pSendParam Pointer to send parameters, NULL to use default send parameter |  |  |
| in  | pData   | pointer to packet data / dataset       |  |
| in  | dataSize  | size of packet data                    |  |

## Return values

| TRDP_NO_ERR         | no error        |
|---------------------|-----------------|
| TRDP_PARAM_ERR      | parameter error |
| TRDP_MEM_ERR        | Out of memory   |
| TRDP_NO_SESSION_ERR | no such session |
| TRDP_NOINIT_ERR     | handle invalid  |

## 5.19.2.32 tlm\_replyQuery()

Send a MD reply query message.

Send a MD reply query message after receiving a request and ask for confirmation. User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session

|   | in   | appHandle                                      | the handle returned by tlc_openSession                          |  |
|---|--|--|---|--|
| Ì | in   | n pSessionId Session ID returned by indication |   |  |
|   | in comld comld of packet to be sent        |  | comld of packet to be sent                                      |  |
|   | in   | userStatus                                     | Info for requester about application errors                     |  |
|   | in confirmTimeout timeout for confirmation |  | timeout for confirmation  |  |
|   | in   | pSendParam                                     | Pointer to send parameters, NULL to use default send parameters |  |
|   | in   | pData  | pointer to packet data / dataset                                |  |
| ĺ | in   | dataSize                                       | size of packet data   |  |
|   |  |  |   |  |

#### Return values

| TRDP_NO_ERR         | no error        |
|---------------------|-----------------|
| TRDP_PARAM_ERR      | parameter error |
| TRDP_MEM_ERR        | out of memory   |
| TRDP_NO_SESSION_ERR | no such session |
| TRDP_NOINIT_ERR     | handle invalid  |

## 5.19.2.33 tlm\_request()

```
EXT_DECL TRDP_ERR_T tlm_request (
           TRDP_APP_SESSION_T appHandle,
            const void * pUserRef,
             TRDP_MD_CALLBACK_T pfCbFunction,
             TRDP_UUID_T * pSessionId,
             UINT32 comId,
             UINT32 etbTopoCnt,
            UINT32 opTrnTopoCnt,
             TRDP_IP_ADDR_T srcIpAddr,
             TRDP_IP_ADDR_T destIpAddr,
             TRDP_FLAGS_T pktFlags,
             UINT32 numReplies,
             UINT32 replyTimeout,
             const TRDP_SEND_PARAM_T * pSendParam,
             const UINT8 * pData,
             UINT32 dataSize,
             const TRDP_URI_USER_T sourceURI,
             const TRDP_URI_USER_T destURI )
```

Initiate sending MD request message.

## Send a MD request message

| Parameters  |  |  |  |
|---|--|--|--|
| appHandle   | the handle returned by tlc_openSession   |  |  |
| pUserRef  | user supplied value returned with reply  |  |  |
| pfCbFunction Pointer to listener specific callback function, NULL to use default function |  |  |  |
| ut pSessionId return session ID   |  |  |  |
| comld   | comld of packet to be sent   |  |  |
| etbTopoCnt  | ETB topocount to use, 0 if consist local communication   |  |  |
| opTrnTopoCnt  | operational topocount, != 0 for orientation/direction sensitive communication  |  |  |
| srclpAddr   | own IP address, 0 - srcIP will be set by the stack   |  |  |
| destlpAddr  | where to send the packet to  |  |  |
| pktFlags  | OPTIONS: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_MARSHALL, TRDP_PLAGS_TCP   |  |  |
| numReplies  | number of expected replies, 0 if unknown   |  |  |
| replyTimeout  | timeout for reply  |  |  |
| pSendParam  | Pointer to send parameters, NULL to use default send parameters  |  |  |
| pData   | pointer to packet data / dataset   |  |  |
| dataSize  | size of packet data  |  |  |
| sourceURI   | only functional group of source URI  |  |  |
| destURI   | only functional group of destination URI   |  |  |
|   | appHandle pUserRef pfCbFunction pSessionId comId etbTopoCnt opTrnTopoCnt srclpAddr destlpAddr pktFlags numReplies replyTimeout pSendParam pData dataSize |  |  |

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_MEM_ERR    | out of memory   |
| TRDP_NOINIT_ERR | handle invalid  |

## 5.19.2.34 tlp\_get()

```
EXT_DECL TRDP_ERR_T tlp_get (
          TRDP_APP_SESSION_T appHandle,
          TRDP_SUB_T subHandle,
          TRDP_PD_INFO_T * pPdInfo,
          UINT8 * pData,
          UINT32 * pDataSize )
```

Get the last valid PD message.

This allows polling of PDs instead of event driven handling by callback

### **Parameters**

| in     | appHandle | the handle returned by tlc_openSession |
|--------|-----------|--|
| in     | subHandle | the handle returned by subscription    |
| in,out | pPdInfo   | pointer to application's info buffer   |
| in,out | pData     | pointer to application's data buffer   |
| in,out | pDataSize | in: size of buffer, out: size of data  |

## Return values

| TRDP_NO_ERR      | no error                         |
|------------------|----------------------------------|
| TRDP_PARAM_ERR   | parameter error                  |
| TRDP_SUB_ERR     | not subscribed                   |
| TRDP_TIMEOUT_ERR | packet timed out                 |
| TRDP_NOINIT_ERR  | handle invalid                   |
| TRDP_COMID_ERR   | ComID not found when marshalling |

This allows polling of PDs instead of event driven handling by callbacks

| in     | appHandle | the handle returned by tlc_openSession |
|--------|-----------|--|
| in     | subHandle | the handle returned by subscription    |
| in,out | pPdInfo   | pointer to application's info buffer   |
| in,out | pData     | pointer to application's data buffer   |
| in,out | pDataSize | in: size of buffer, out: size of data  |

## Return values

| TRDP_NO_ERR      | no error                         |
|------------------|----------------------------------|
| TRDP_PARAM_ERR   | parameter error                  |
| TRDP_SUB_ERR     | not subscribed                   |
| TRDP_TIMEOUT_ERR | packet timed out                 |
| TRDP_NOINIT_ERR  | handle invalid                   |
| TRDP_COMID_ERR   | ComID not found when marshalling |

## 5.19.2.35 tlp\_getRedundant()

Get status of redundant Comlds.

#### **Parameters**

| in      | appHandle | the handle returned by tlc_openSession                            |
|---------|-----------|---|
| in      | redId     | will be set for all ComID's with the given redId, 0 for all redId |
| in, out | pLeader   | TRUE if we send (leader)  |

## Return values

| TRDP_NO_ERR     | no error                             |
|-----------------|--------------------------------------|
| TRDP_PARAM_ERR  | parameter error / redld not existing |
| TRDP_NOINIT_ERR | handle invalid                       |

Only the status of the first found redundancy group entry will be returned!

### **Parameters**

| in     | appHandle | the handle returned by tlc_openSession                |
|--------|-----------|---|
| in     | redId     | will be returned for all ComID's with the given redId |
| in,out | pLeader   | TRUE if we're sending this redundancy group (leader)  |

#### Return values

| TRDP_NO_ERR     | no error                      |
|-----------------|-------------------------------|
| TRDP_PARAM_ERR  | redld invalid or not existing |
| TRDP_NOINIT_ERR | handle invalid                |

## 5.19.2.36 tlp\_publish()

```
EXT_DECL TRDP_ERR_T tlp_publish (
            TRDP_APP_SESSION_T appHandle,
             TRDP_PUB_T * pPubHandle,
             const void * pUserRef,
             TRDP_PD_CALLBACK_T pfCbFunction,
             UINT32 comId,
             UINT32 etbTopoCnt,
             UINT32 opTrnTopoCnt,
             TRDP_IP_ADDR_T srcIpAddr,
             TRDP_IP_ADDR_T destIpAddr,
             UINT32 interval,
             UINT32 redId,
             TRDP_FLAGS_T pktFlags,
             const TRDP_SEND_PARAM_T * pSendParam,
             const UINT8 * pData,
             UINT32 dataSize )
```

Prepare for sending PD messages.

Queue a PD message, it will be send when tlc\_publish has been called

### **Parameters**

| in  | appHandle  | the handle returned by tlc_openSession  |  |
|-----|--|---|--|
| out | pPubHandle   | returned handle for related re/unpublish                                      |  |
| in  | pUserRef user supplied value returned within the info structure of callback function       |   |  |
| in  | pfCbFunction   | Pointer to pre-send callback function, NULL if not used                       |  |
| in  | comld  | comld of packet to send   |  |
| in  | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                        |  |
| in  | opTrnTopoCnt   | operational topocount, != 0 for orientation/direction sensitive communication |  |
| in  | srclpAddr  | own IP address, 0 - srcIP will be set by the stack                            |  |
| in  | destlpAddr   | destIpAddr where to send the packet to  |  |
| in  | interval   | val frequency of PD packet (>= 10ms) in usec                                  |  |
| in  | redId  | 0 - Non-redundant, > 0 valid redundancy group                                 |  |
| in  | pktFlags OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE, TRDP_FLAGS_MARSHATRDP_FLAGS_CALLBACK |   |  |
| in  | pSendParam optional pointer to send parameter, NULL - default parameters are used          |   |  |
| in  | pData  | pointer to data packet / dataset, NULL if sending starts later with tlp_put() |  |
| in  | dataSize size of data packet >= 0 and <= TRDP_MAX_PD_DATA_SIZE                             |   |  |

## Return values

| TRDP_NO_ERR     | no error                         |
|-----------------|----------------------------------|
| TRDP_PARAM_ERR  | parameter error                  |
| TRDP_MEM_ERR    | could not insert (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                   |

## 5.19.2.37 tlp\_put()

Update the process data to send.

Update previously published data. The new telegram will be sent earliest when tlc\_process is called.

#### **Parameters**

| in     | appHandle | the handle returned by tlc_openSession |
|--------|-----------|--|
| in     | pubHandle | the handle returned by publish         |
| in,out | pData     | pointer to application's data buffer   |
| in,out | dataSize  | size of data                           |

#### Return values

| TRDP_NO_ERR     | no error   |
|-----------------|--|
| TRDP_PARAM_ERR  | parameter error on uninitialized parameter or changed dataSize compared to published one |
| TRDP_PUB_ERR    | not published  |
| TRDP_NOINIT_ERR | handle invalid   |
| TRDP_COMID_ERR  | ComID not found when marshalling   |

Update previously published data. The new telegram will be sent earliest when tlc\_process is called.

## Parameters

| in     | appHandle | the handle returned by tlc_openSession |
|--------|-----------|--|
| in     | pubHandle | the handle returned by publish         |
| in,out | pData     | pointer to application's data buffer   |
| in,out | dataSize  | size of data                           |

#### Return values

| TRDP_NO_ERR     | no error   |
|-----------------|--|
| TRDP_PARAM_ERR  | parameter error on uninitialized parameter or changed dataSize compared to |
|                 | published one  |
| TRDP_NOPUB_ERR  | not published  |
| TRDP_NOINIT_ERR | handle invalid   |
| TRDP_COMID_ERR  | ComID not found when marshalling   |

## 5.19.2.38 tlp\_republish()

```
TRDP_PUB_T pubHandle,
UINT32 etbTopoCnt,
UINT32 opTrnTopoCnt,
TRDP_IP_ADDR_T srcIpAddr,
TRDP_IP_ADDR_T destIpAddr)
```

Prepare for sending PD messages.

Reinitialize and queue a PD message, it will be send when tlc\_publish has been called

#### **Parameters**

| in | appHandle    | the handle returned by tlc_openSession  |  |
|----|--------------|---|--|
| in | pubHandle    | handle for related unpublish  |  |
| in | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                        |  |
| in | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication |  |
| in | srclpAddr    | own IP address, 0 - srcIP will be set by the stack                            |  |
| in | destlpAddr   | where to send the packet to   |  |

#### Return values

| TRDP_NO_ERR     | no error                         |
|-----------------|----------------------------------|
| TRDP_PARAM_ERR  | parameter error                  |
| TRDP_MEM_ERR    | could not insert (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                   |

#### 5.19.2.39 tlp\_request()

```
EXT_DECL TRDP_ERR_T tlp_request (

TRDP_APP_SESSION_T appHandle,

TRDP_SUB_T subHandle,

UINT32 comId,

UINT32 etbTopoCnt,

UINT32 opTrnTopoCnt,

TRDP_IP_ADDR_T srcIpAddr,

TRDP_IP_ADDR_T destIpAddr,

UINT32 redId,

TRDP_FLAGS_T pktFlags,

const TRDP_SEND_PARAM_T * pSendParam,

const UINT8 * pData,

UINT32 replyComId,

TRDP_IP_ADDR_T replyIpAddr)
```

Initiate sending PD messages (PULL).

### Send a PD request message

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | subHandle | handle from related subscribe          |

## **Parameters**

| in | comId        | comld of packet to be sent  |
|----|--------------|---|
| in | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                        |
| in | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication |
| in | srclpAddr    | own IP address, 0 - srcIP will be set by the stack                            |
| in | destlpAddr   | where to send the packet to   |
| in | redId        | 0 - Non-redundant, > 0 valid redundancy group                                 |
| in | pktFlags     | OPTIONS: TTRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE,                                |
|    |              | TRDP_FLAGS_MARSHALL, TRDP_FLAGS_CALLBACK                                      |
| in | pSendParam   | optional pointer to send parameter, NULL - default parameters are used        |
| in | pData        | pointer to packet data / dataset  |
| in | dataSize     | size of packet data   |
| in | replyComId   | comld of reply  |
| in | replylpAddr  | IP for reply  |
|    |              |   |

### Return values

| TRDP_NO_ERR     | no error                         |
|-----------------|----------------------------------|
| TRDP_PARAM_ERR  | parameter error                  |
| TRDP_MEM_ERR    | could not insert (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                   |

## Send a PD request message

## **Parameters**

| in | appHandle    | the handle returned by tlc_openSession  |
|----|--------------|---|
| in | subHandle    | handle from related subscribe   |
| in | comld        | comld of packet to be sent  |
| in | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                                |
| in | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication         |
| in | srclpAddr    | own IP address, 0 - srcIP will be set by the stack                                    |
| in | destlpAddr   | where to send the packet to   |
| in | redId        | 0 - Non-redundant, > 0 valid redundancy group   |
| in | pktFlags     | OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE, TRDP_FLAGS_MARSHALL, TRDP_FLAGS_CALLBACK |
| in | pSendParam   | optional pointer to send parameter, NULL - default parameters are used                |
| in | pData        | pointer to packet data / dataset  |
| in | dataSize     | size of packet data   |
| in | replyComId   | comld of reply (default comID of subscription)  |
| in | replylpAddr  | IP for reply  |

### Return values

| TRDP_NO_ERR     | no error                         |
|-----------------|----------------------------------|
| TRDP_PARAM_ERR  | parameter error                  |
| TRDP_MEM_ERR    | could not insert (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                   |
| TRDP_NOSUB_ERR  | no matching subscription found   |

## 5.19.2.40 tlp\_resubscribe()

```
EXT_DECL TRDP_ERR_T tlp_resubscribe (

TRDP_APP_SESSION_T appHandle,

TRDP_SUB_T subHandle,

UINT32 etbTopoCnt,

UINT32 opTrnTopoCnt,

TRDP_IP_ADDR_T srcIpAddr1,

TRDP_IP_ADDR_T srcIpAddr2,

TRDP_IP_ADDR_T destIpAddr )
```

Reprepare for receiving PD messages.

Resubscribe to a specific PD ComID and source IP

#### **Parameters**

| in | appHandle    | the handle returned by tlc_openSession  |
|----|--------------|---|
| in | subHandle    | handle for this subscription  |
| in | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                        |
| in | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication |
| in | srclpAddr1   | IP for source filtering, set 0 if not used                                    |
| in | srclpAddr2   | IP for source filtering range, set 0 if not used                              |
| in | destlpAddr   | IP address to join  |

## Return values

| TRDP_NO_ERR     | no error                                 |
|-----------------|--|
| TRDP_PARAM_ERR  | parameter error                          |
| TRDP_MEM_ERR    | could not reserve memory (out of memory) |
| TRDP_NOINIT_ERR | handle invalid                           |

Resubscribe to a specific PD ComID and source IP

### **Parameters**

| in | appHandle    | the handle returned by tlc_openSession  |
|----|--------------|---|
| in | subHandle    | handle for this subscription  |
| in | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                          |
| in | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication   |
| in | srclpAddr1   | Source IP address, lower address in case of address range, set to 0 if not used |
| in | srclpAddr2   | upper address in case of address range, set to 0 if not used                    |
| in | destlpAddr   | IP address to join  |

#### Return values

| TRDP_NO_ERR    | no error                                 |
|----------------|--|
| TRDP_PARAM_ERR | parameter error                          |
| TRDP_MEM_ERR   | could not reserve memory (out of memory) |

### Return values

| TRDP_NOINIT_ERR | handle invalid   |
|-----------------|--|
| TRDP_SOCK_ERR   | Resource (socket) not available, subscription canceled |

## 5.19.2.41 tlp\_setRedundant()

Do not send redundant PD's when we are follower.

#### **Parameters**

| in | appHandle | the handle returned by tlc_openSession                                      |
|----|-----------|---|
| in | redId     | will be set for all ComID's with the given redId, 0 to change for all redId |
| in | leader    | TRUE if we send   |

### Return values

| TRDP_NO_ERR     | no error                             |
|-----------------|--------------------------------------|
| TRDP_PARAM_ERR  | parameter error / redld not existing |
| TRDP_NOINIT_ERR | handle invalid                       |

Do not send redundant PD's when we are follower.

#### **Parameters**

| in | appHandle | the handle returned by tlc_openSession                                      |
|----|-----------|---|
| in | redId     | will be set for all ComID's with the given redId, 0 to change for all redId |
| in | leader    | TRUE if we send   |

#### Return values

| TRDP_NO_ERR     | no error                             |
|-----------------|--------------------------------------|
| TRDP_PARAM_ERR  | parameter error / redld not existing |
| TRDP_NOINIT_ERR | handle invalid                       |

## 5.19.2.42 tlp\_subscribe()

```
EXT_DECL TRDP_ERR_T tlp_subscribe (
          TRDP_APP_SESSION_T appHandle,
          TRDP_SUB_T * pSubHandle,
```

```
const void * pUserRef,
TRDP_PD_CALLBACK_T pfCbFunction,
UINT32 comId,
UINT32 etbTopoCnt,
UINT32 opTrnTopoCnt,
TRDP_IP_ADDR_T srcIpAddr1,
TRDP_IP_ADDR_T srcIpAddr2,
TRDP_IP_ADDR_T destIpAddr,
TRDP_FLAGS_T pktFlags,
UINT32 timeout,
TRDP_TO_BEHAVIOR_T toBehavior)
```

Prepare for receiving PD messages.

Subscribe to a specific PD ComID and source IP

#### **Parameters**

| in  | appHandle    | the handle returned by tlc_openSession  |  |
|-----|--------------|---|--|
| out | pSubHandle   | return a handle for this subscription   |  |
| in  | pUserRef     | user supplied value returned within the info structure                                |  |
| in  | pfCbFunction | Pointer to subscriber specific callback function, NULL to use default function        |  |
| in  | comld        | comld of packet to receive  |  |
| in  | etbTopoCnt   | ETB topocount to use, 0 if consist local communication                                |  |
| in  | opTrnTopoCnt | operational topocount, != 0 for orientation/direction sensitive communication         |  |
| in  | srclpAddr1   | Source IP address, lower address in case of address range, set to 0 if not used       |  |
| in  | srclpAddr2   | upper address in case of address range, set to 0 if not used                          |  |
| in  | destlpAddr   | IP address to join  |  |
| in  | pktFlags     | OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE, TRDP_FLAGS_MARSHALL, TRDP_FLAGS_CALLBACK |  |
| in  | timeout      | timeout (>= 10ms) in usec   |  |
| in  | toBehavior   | OPTION: TRDP_TO_DEFAULT, TRDP_TO_SET_TO_ZERO, TRDP_TO_KEEP_LAST_VALUE                 |  |

#### Return values

| TRDP_NO_ERR     | no error                                |  |
|-----------------|---|--|
| TRDP_PARAM_ERR  | parameter error                         |  |
| TRDP_MEM_ERR    | could not reserve memory (out of memory |  |
| TRDP_NOINIT_ERR | handle invalid                          |  |

Subscribe to a specific PD ComID and source IP.

| in  | appHandle  | the handle returned by tlc_openSession  |  |
|-----|--|---|--|
| out | pSubHandle   | return a handle for this subscription   |  |
| in  | pUserRef   | user supplied value returned within the info structure                                  |  |
| in  | pfCbFunction   | Function Pointer to subscriber specific callback function, NULL to use default function |  |
| in  | comld  | comld of packet to receive  |  |
| in  | etbTopoCnt   | poCnt ETB topocount to use, 0 if consist local communication                            |  |
| in  | opTrnTopoCnt operational topocount, != 0 for orientation/direction sensitive communication |   |  |
| in  | srclpAddr1 Source IP address, lower address in case of address range, set to 0 if not used |   |  |

## **Parameters**

| in | srclpAddr2 | upper address in case of address range, set to 0 if not used      |  |
|----|------------|---|--|
| in | pktFlags   | OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE, TRDP_FLAGS_MARSHALL, |  |
|    |            | TRDP_FLAGS_CALLBACK   |  |
| in | destlpAddr | IP address to join  |  |
| in | timeout    | timeout (>= 10ms) in usec   |  |
| in | toBehavior | timeout behavior  |  |

## Return values

| TRDP_NO_ERR     | no error                                |  |
|-----------------|---|--|
| TRDP_PARAM_ERR  | parameter error                         |  |
| TRDP_MEM_ERR    | could not reserve memory (out of memory |  |
| TRDP_NOINIT_ERR | handle invalid                          |  |

## 5.19.2.43 tlp\_unpublish()

## Stop sending PD messages.

## **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | pubHandle | the handle returned by publish         |

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_NOPUB_ERR  | not published   |
| TRDP_NOINIT_ERR | handle invalid  |

## **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | pubHandle | the handle returned by prepare         |

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_NOPUB_ERR  | not published   |
| TRDP_NOINIT_ERR | handle invalid  |

#### 5.19.2.44 tlp\_unsubscribe()

Stop receiving PD messages.

Unsubscribe to a specific PD ComID

### **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | subHandle | the handle for this subscription       |

### Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_SUB_ERR    | not subscribed  |
| TRDP_NOINIT_ERR | handle invalid  |

Unsubscribe to a specific PD ComID

### **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | subHandle | the handle for this subscription       |

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_PARAM_ERR  | parameter error |
| TRDP_NOSUB_ERR  | not subscribed  |
| TRDP_NOINIT_ERR | handle invalid  |

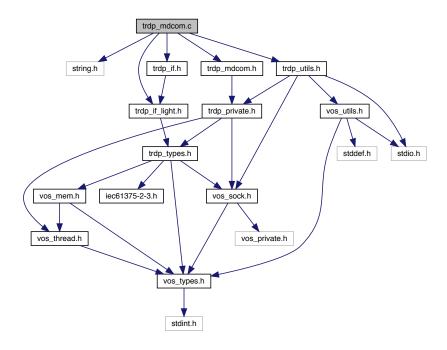
# 5.20 trdp\_mdcom.c File Reference

Functions for MD communication.

```
#include <string.h>
#include "trdp_if_light.h"
#include "trdp_if.h"
#include "trdp_utils.h"
```

#include "trdp\_mdcom.h"

Include dependency graph for trdp mdcom.c:



#### **Functions**

TRDP\_ERR\_T trdp\_mdGetTCPSocket (TRDP\_SESSION\_PT pSession)

Initialize the specific parameters for message data Open a listening socket.

• void trdp\_mdFreeSession (MD\_ELE\_T \*pMDSession)

Free memory of session.

• TRDP\_ERR\_T trdp\_mdSend (TRDP\_SESSION\_PT appHandle)

Sending MD messages Send the messages stored in the sendQueue Call user's callback if needed.

void trdp\_mdCheckPending (TRDP\_APP\_SESSION\_T appHandle, TRDP\_FDS\_T \*pFileDesc, INT32 \*p← NoDesc)

Check for pending packets, set FD if non blocking.

void trdp\_mdCheckListenSocks (const TRDP\_SESSION\_PT appHandle, TRDP\_FDS\_T \*pRfds, INT32 \*p
 Count)

Checking receive connection requests and data Call user's callback if needed.

• void trdp\_mdCheckTimeouts (TRDP\_SESSION\_PT appHandle)

Checking message data timeouts Call user's callback if needed.

TRDP\_ERR\_T trdp\_mdReply (const TRDP\_MSG\_T msgType, TRDP\_APP\_SESSION\_T appHandle, TRD→P\_UUID\_T pSessionId, UINT32 comId, UINT32 timeout, INT32 replyStatus, const TRDP\_SEND\_PARAM\_T\*
 \*pSendParam, const UINT8 \*pData, UINT32 dataSize)

Send a MD reply/reply query message.

• TRDP\_ERR\_T trdp\_mdCall (const TRDP\_MSG\_T msgType, TRDP\_APP\_SESSION\_T appHandle, const void \*pUserRef, TRDP\_MD\_CALLBACK\_T pfCbFunction, TRDP\_UUID\_T \*pSessionId, UINT32 comId, U ← INT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr, TRDP\_IP\_ADDR\_T destIpAddr, TRDP\_FLAGS\_T pktFlags, UINT32 numExpReplies, UINT32 replyTimeout, INT32 replyStatus, const TR← DP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize, const TRDP\_URI\_USER\_T srcURI, const TRDP\_URI\_USER\_T destURI)

Initiate sending MD request message - private SW level Send a MD request message.

• TRDP\_ERR\_T trdp\_mdConfirm (TRDP\_APP\_SESSION\_T appHandle, const TRDP\_UUID\_T \*pSessionId, UINT16 userStatus, const TRDP\_SEND\_PARAM\_T \*pSendParam)

Initiate sending MD confirm message - private SW level Send a MD confirmation message User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session.

#### 5.20.1 Detailed Description

Functions for MD communication.

Note

Project: TCNOpen TRDP prototype stack

#### **Author**

Simone Pachera, FARsystems Gari Oiarbide, CAF Michael Koch, Bombardier Transportations Bernd Loehr, NewTec

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

## trdp\_mdcom.c 1779 2018-11-07 09:49:55Z bloehr

```
BL 2018-11-07: Ticket #185 MD reply: Infinite timeout wrong handled
BL 2018-11-07: Ticket #220 Message Data - Different behaviour UDP & TCP
BL 2018-11-06: for-loops limited to sCurrentMaxSocketCnt instead VOS_MAX_SOCKET_CNT
BL 2018-06-27: Ticket #206 Message data transmission fails for several test cases (revisited, size handling r
SB 2018-10-29: Ticket #216: Message data size and padding wrong if marshalling is used
BL 2018-06-27: Ticket #206 Message data transmission fails for several test cases
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2018-05-14: Ticket #200 Notify 'sender element' fields, set twice
BL 2018-01-29: Ticket #188 Typo in the TRDP_VAR_SIZE definition
BL 2017-11-28: Ticket #180 Filtering rules for DestinationURI does not follow the standard
BL 2017-11-15: Ticket #1
                           Unjoin on unsubscribe/delListener (finally ;-)
BL 2017-11-09: Ticket #174: Receiving fragmented TCP packets
AHW 2017-11-08: Ticket #179 Max. number of retries (part of sendParam) of a MD request needs to be checked
BL 2017-06-28: Ticket #160: Receiving fragmented TCP packets
BL 2017-05-22: Ticket #122: Addendum for 64Bit compatibility (VOS_TIME_T -> VOS_TIMEVAL_T)
AHW 2017-05-22: Ticket #158 Infinit timeout at TRDB level is 0 acc. standard
BL 2017-05-08: Compiler warnings, doxygen comment errors
BL 2017-03-01: Ticket #149 SourceUri and DestinationUri don't with 32 characters
BL 2017-02-27: Ticket #148 Wrong element used in trdp_mdCheckTimeouts() to invoke the callback
BL 2017-02-10: Ticket #138 Erroneous closing of receive md socket
BL 2017-02-10: Ticket #142 Compiler warnings / MISRA-C 2012 issues
BL 2016-07-09: Ticket #127 MD notify message: Invalid session identifier
BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings)
BL 2016-03-10: Ticket #115 MD: Missing parameter pktFlags in tlm_reply() and tlm_replyQuery()
BL 2016-02-04: Ticket #110: Handling of optional marshalling on sending
BL 2015-12-22: Mutex removed
BL 2015-08-31: Ticket #94: TRDP_REDUNDANT flag is evaluated, beQuiet removed
BL 2014-08-28: Ticket #62: Failing TCP communication fixed,
                           Do not read if there's nothing to read ('Mc' has no data!)
BL 2014-08-25: Ticket #57+58: Padding / zero bytes trailing MD & PD packets fixed
BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed
               Ticket #47: Protocol change: no FCS for data part of telegrams
BL 2014-02-28: Ticket #25: CRC32 calculation is not according to IEEE802.3
```

### 5.20.2 Function Documentation

## 5.20.2.1 trdp\_mdCall()

```
TRDP_ERR_T trdp_mdCall (
            const TRDP_MSG_T msgType,
             TRDP_APP_SESSION_T appHandle,
             const void * pUserRef,
             TRDP_MD_CALLBACK_T pfCbFunction,
             TRDP_UUID_T * pSessionId,
             UINT32 comId,
             UINT32 etbTopoCnt,
             UINT32 opTrnTopoCnt,
             TRDP_IP_ADDR_T srcIpAddr,
             TRDP_IP_ADDR_T destIpAddr,
             TRDP_FLAGS_T pktFlags,
             UINT32 numExpReplies,
             UINT32 replyTimeout,
             INT32 replyStatus,
             const TRDP_SEND_PARAM_T * pSendParam,
             const UINT8 * pData,
             UINT32 dataSize,
             \verb"const TRDP_URI_USER_T" srcURI",
             const TRDP_URI_USER_T destURI )
```

Initiate sending MD request message - private SW level Send a MD request message.

| in  | msgType       | TRDP_MSG_MN or TRDP_MSG_MR  |  |
|-----|---------------|---|--|
| in  | appHandle     | the handle returned by tlc_init   |  |
| in  | pUserRef      | user supplied value returned with reply                                       |  |
| in  | pfCbFunction  | Pointer to listener specific callback function, NULL to use default function  |  |
| out | pSessionId    | return session ID   |  |
| in  | comld         | comld of packet to be sent  |  |
| in  | etbTopoCnt    | ETB topocount to use, 0 if consist local communication                        |  |
| in  | opTrnTopoCnt  | operational topocount, != 0 for orientation/direction sensitive communication |  |
| in  | srclpAddr     | own IP address, 0 - srcIP will be set by the stack                            |  |
| in  | destlpAddr    | where to send the packet to   |  |
| in  | pktFlags      | OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE,                                  |  |
|     |               | TRDP_FLAGS_MARSHALL   |  |
| in  | numExpReplies | number of expected replies, 0 if unknown                                      |  |
| in  | replyTimeout  | timeout for reply   |  |
| in  | replyStatus   | status to be returned   |  |
| in  | pSendParam    | Pointer to send parameters, NULL to use default send parameters               |  |
| in  | pData         | pointer to packet data / dataset  |  |
| in  | dataSize      | size of packet data   |  |
| in  | srcURI        | only functional group of source URI   |  |
| in  | destURI       | only functional group of destination URI                                      |  |

#### **Return values**

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | parameter error |
| TRDP_MEM_ERR   | out of memory   |

## 5.20.2.2 trdp\_mdCheckListenSocks()

Checking receive connection requests and data Call user's callback if needed.

### **Parameters**

| in     | appHandle | session pointer                        |
|--------|-----------|--|
| in     | pRfds     | pointer to set of ready descriptors    |
| in,out | pCount    | pointer to number of ready descriptors |

## 5.20.2.3 trdp\_mdCheckPending()

Check for pending packets, set FD if non blocking.

## **Parameters**

| in  | appHandle session pointer |  |
|---|---------------------------|--|
| in,out  | pFileDesc                 | pointer to set of ready descriptors    |
| in, out <i>pNoDesc</i> pointer to number of ready des |                           | pointer to number of ready descriptors |

## 5.20.2.4 trdp\_mdCheckTimeouts()

```
void trdp_mdCheckTimeouts ( \label{trdp_mdCheckTimeouts} \texttt{TRDP\_SESSION\_PT} \ app{\tt Handle} \ )
```

Checking message data timeouts Call user's callback if needed.

### **Parameters**

| in a | pHandle | session | pointer |
|------|---------|---------|---------|
|------|---------|---------|---------|

## 5.20.2.5 trdp\_mdConfirm()

Initiate sending MD confirm message - private SW level Send a MD confirmation message User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session.

#### **Parameters**

| in | appHandle  | the handle returned by tlc_init                                 |  |
|----|------------|---|--|
| in | pSessionId | Session ID returned by request                                  |  |
| in | userStatus | Info for requester about application errors                     |  |
| in | pSendParam | Pointer to send parameters, NULL to use default send parameters |  |

#### **Return values**

| TRDP_NO_ERR        | no error        |
|--------------------|-----------------|
| TRDP_PARAM_ERR     | parameter error |
| TRDP_MEM_ERR       | out of memory   |
| TRDP_NOSESSION_ERR | no such session |

## 5.20.2.6 trdp\_mdFreeSession()

```
void trdp_mdFreeSession ( \label{eq:mdFreeSession} \texttt{MD\_ELE\_T} * \textit{pMDSession} \; )
```

Free memory of session.

| in | pMDSession | session pointer |
|----|------------|-----------------|
|----|------------|-----------------|

Here is the call graph for this function:



## 5.20.2.7 trdp\_mdGetTCPSocket()

Initialize the specific parameters for message data Open a listening socket.

#### **Parameters**

| in | pSession | session parameters |
|----|----------|--------------------|
|----|----------|--------------------|

#### Return values

| TRDP_NO_ERR    | no error             |
|----------------|----------------------|
| TRDP_PARAM_ERR | initialization error |

## 5.20.2.8 trdp\_mdReply()

```
TRDP_ERR_T trdp_mdReply (

const TRDP_MSG_T msgType,

TRDP_APP_SESSION_T appHandle,

TRDP_UUID_T pSessionId,

UINT32 comId,

UINT32 timeout,

INT32 replyStatus,

const TRDP_SEND_PARAM_T * pSendParam,

const UINT8 * pData,

UINT32 dataSize )
```

Send a MD reply/reply query message.

Send either a MD reply message or a MD reply query message after receiving a request and ask for confirmation. User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session

### **Parameters**

| in | msgType     | TRDP_MSG_MP or TRDP_MSG_MQ                                      |
|----|-------------|---|
| in | appHandle   | the handle returned by tlc_init                                 |
| in | pSessionId  | Session ID returned by indication                               |
| in | comId       | comld of packet to be sent                                      |
| in | timeout     | time out for confirmations (zero for TRDP_MSG_MP)               |
| in | replyStatus | Info for requester about application errors                     |
| in | pSendParam  | Pointer to send parameters, NULL to use default send parameters |
| in | pData       | pointer to packet data / dataset                                |
| in | dataSize    | size of packet data   |

### Return values

| TRDP_NO_ERR         | no error        |
|---------------------|-----------------|
| TRDP_PARAM_ERR      | parameter error |
| TRDP_MEM_ERR        | out of memory   |
| TRDP_NO_SESSION_ERR | no such session |

## 5.20.2.9 trdp\_mdSend()

Sending MD messages Send the messages stored in the sendQueue Call user's callback if needed.

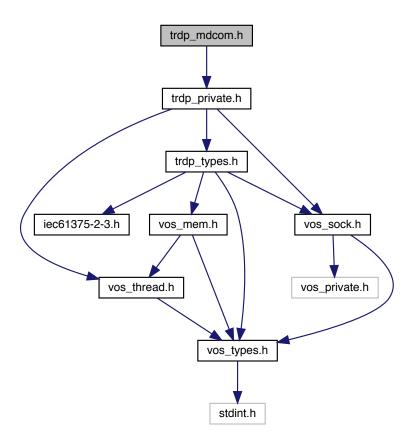
### **Parameters**

| in | appHandle | session pointer |
|----|-----------|-----------------|
|----|-----------|-----------------|

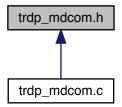
# 5.21 trdp\_mdcom.h File Reference

Functions for MD communication.

#include "trdp\_private.h"
Include dependency graph for trdp\_mdcom.h:



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



## **Functions**

• TRDP\_ERR\_T trdp\_mdGetTCPSocket (TRDP\_SESSION\_PT pSession)

Initialize the specific parameters for message data Open a listening socket.

void trdp mdFreeSession (MD ELE T\*pMDSession)

Free memory of session.

TRDP ERR T trdp mdSend (TRDP SESSION PT appHandle)

Sending MD messages Send the messages stored in the sendQueue Call user's callback if needed.

 void trdp\_mdCheckPending (TRDP\_APP\_SESSION\_T appHandle, TRDP\_FDS\_T \*pFileDesc, INT32 \*p↔ NoDesc)

Check for pending packets, set FD if non blocking.

void trdp\_mdCheckListenSocks (const TRDP\_SESSION\_PT appHandle, TRDP\_FDS\_T \*pRfds, INT32 \*p← Count)

Checking receive connection requests and data Call user's callback if needed.

void trdp\_mdCheckTimeouts (TRDP\_SESSION\_PT appHandle)

Checking message data timeouts Call user's callback if needed.

• TRDP\_ERR\_T trdp\_mdConfirm (TRDP\_APP\_SESSION\_T appHandle, const TRDP\_UUID\_T \*pSessionId, UINT16 userStatus, const TRDP\_SEND\_PARAM\_T \*pSendParam)

Initiate sending MD confirm message - private SW level Send a MD confirmation message User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session.

 TRDP\_ERR\_T trdp\_mdReply (const TRDP\_MSG\_T msgType, TRDP\_APP\_SESSION\_T appHandle, TRD← P\_UUID\_T pSessionId, UINT32 comId, UINT32 timeout, INT32 replyStatus, const TRDP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize)

Send a MD reply/reply query message.

TRDP\_ERR\_T trdp\_mdCall (const TRDP\_MSG\_T msgType, TRDP\_APP\_SESSION\_T appHandle, const void \*pUserRef, TRDP\_MD\_CALLBACK\_T pfCbFunction, TRDP\_UUID\_T \*pSessionId, UINT32 comId, U← INT32 etbTopoCnt, UINT32 opTrnTopoCnt, TRDP\_IP\_ADDR\_T srclpAddr, TRDP\_IP\_ADDR\_T destlpAddr, TRDP\_FLAGS\_T pktFlags, UINT32 numExpReplies, UINT32 replyTimeout, INT32 replyStatus, const TR← DP\_SEND\_PARAM\_T \*pSendParam, const UINT8 \*pData, UINT32 dataSize, const TRDP\_URI\_USER\_T srcURI, const TRDP\_URI\_USER\_T destURI)

Initiate sending MD request message - private SW level Send a MD request message.

## 5.21.1 Detailed Description

Functions for MD communication.

Note

Project: TCNOpen TRDP prototype stack

Author

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

trdp\_mdcom.h 1681 2017-11-09 10:37:16Z ahweiss

```
AHW 2017-11-08: Ticket #179 Max. number of retries (part of sendParam) of a MD request needs to be checked BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed

Ticket #47: Protocol change: no FCS for data part of telegrams
```

## 5.21.2 Function Documentation

## 5.21.2.1 trdp\_mdCall()

```
TRDP_ERR_T trdp_mdCall (
            const TRDP_MSG_T msgType,
             TRDP_APP_SESSION_T appHandle,
             const void * pUserRef,
             TRDP_MD_CALLBACK_T pfCbFunction,
             TRDP_UUID_T * pSessionId,
             UINT32 comId,
             UINT32 etbTopoCnt,
             UINT32 opTrnTopoCnt,
             TRDP_IP_ADDR_T srcIpAddr,
             TRDP_IP_ADDR_T destIpAddr,
             TRDP_FLAGS_T pktFlags,
             UINT32 numExpReplies,
             UINT32 replyTimeout,
             INT32 replyStatus,
             const TRDP_SEND_PARAM_T * pSendParam,
             const UINT8 * pData,
             UINT32 dataSize,
             \verb"const TRDP_URI_USER_T" srcURI",
             const TRDP_URI_USER_T destURI )
```

Initiate sending MD request message - private SW level Send a MD request message.

|     | 0.0           |   |
|-----|---------------|---|
| in  | msgType       | TRDP_MSG_MN or TRDP_MSG_MR  |
| in  | appHandle     | the handle returned by tlc_init   |
| in  | pUserRef      | user supplied value returned with reply                                       |
| in  | pfCbFunction  | Pointer to listener specific callback function, NULL to use default function  |
| out | pSessionId    | return session ID   |
| in  | comld         | comld of packet to be sent  |
| in  | etbTopoCnt    | ETB topocount to use, 0 if consist local communication                        |
| in  | opTrnTopoCnt  | operational topocount, != 0 for orientation/direction sensitive communication |
| in  | srclpAddr     | own IP address, 0 - srcIP will be set by the stack                            |
| in  | destlpAddr    | where to send the packet to   |
| in  | pktFlags      | OPTION: TRDP_FLAGS_DEFAULT, TRDP_FLAGS_NONE,                                  |
|     |               | TRDP_FLAGS_MARSHALL   |
| in  | numExpReplies | number of expected replies, 0 if unknown                                      |
| in  | replyTimeout  | timeout for reply   |
| in  | replyStatus   | status to be returned   |
| in  | pSendParam    | Pointer to send parameters, NULL to use default send parameters               |
| in  | pData         | pointer to packet data / dataset  |
| in  | dataSize      | size of packet data   |
| in  | srcURI        | only functional group of source URI   |
| in  | destURI       | only functional group of destination URI                                      |
|     |               |   |

#### **Return values**

| TRDP_NO_ERR    | no error        |
|----------------|-----------------|
| TRDP_PARAM_ERR | parameter error |
| TRDP_MEM_ERR   | out of memory   |

## 5.21.2.2 trdp\_mdCheckListenSocks()

Checking receive connection requests and data Call user's callback if needed.

### **Parameters**

| in     | appHandle | session pointer                        |
|--------|-----------|--|
| in     | pRfds     | pointer to set of ready descriptors    |
| in,out | pCount    | pointer to number of ready descriptors |

## 5.21.2.3 trdp\_mdCheckPending()

Check for pending packets, set FD if non blocking.

## **Parameters**

| in     | appHandle | appHandle session pointer              |  |
|--------|-----------|--|--|
| in,out | pFileDesc | pointer to set of ready descriptors    |  |
| in,out | pNoDesc   | pointer to number of ready descriptors |  |

## 5.21.2.4 trdp\_mdCheckTimeouts()

```
void trdp_mdCheckTimeouts ( \label{trdp_mdCheckTimeouts} \texttt{TRDP\_SESSION\_PT} \ app{\tt Handle} \ )
```

Checking message data timeouts Call user's callback if needed.

#### **Parameters**

| in | appHandle | session pointer |
|----|-----------|-----------------|
|----|-----------|-----------------|

## 5.21.2.5 trdp\_mdConfirm()

Initiate sending MD confirm message - private SW level Send a MD confirmation message User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session.

#### **Parameters**

| in | appHandle  | the handle returned by tlc_init                                 |  |
|----|------------|---|--|
| in | pSessionId | Session ID returned by request                                  |  |
| in | userStatus | Info for requester about application errors                     |  |
| in | pSendParam | Pointer to send parameters, NULL to use default send parameters |  |

#### **Return values**

| TRDP_NO_ERR        | no error        |
|--------------------|-----------------|
| TRDP_PARAM_ERR     | parameter error |
| TRDP_MEM_ERR       | out of memory   |
| TRDP_NOSESSION_ERR | no such session |

## 5.21.2.6 trdp\_mdFreeSession()

```
void trdp_mdFreeSession ( \label{eq:mdFreeSession} \texttt{MD\_ELE\_T} * \textit{pMDSession} \; )
```

Free memory of session.

| in | pMDSession | session pointer |
|----|------------|-----------------|
|----|------------|-----------------|

Here is the call graph for this function:



## 5.21.2.7 trdp\_mdGetTCPSocket()

Initialize the specific parameters for message data Open a listening socket.

#### **Parameters**

| in | pSession | session parameters |
|----|----------|--------------------|
|----|----------|--------------------|

#### Return values

| TRDP_NO_ERR    | no error             |
|----------------|----------------------|
| TRDP_PARAM_ERR | initialization error |

## 5.21.2.8 trdp\_mdReply()

```
TRDP_ERR_T trdp_mdReply (

const TRDP_MSG_T msgType,

TRDP_APP_SESSION_T appHandle,

TRDP_UUID_T pSessionId,

UINT32 comId,

UINT32 timeout,

INT32 replyStatus,

const TRDP_SEND_PARAM_T * pSendParam,

const UINT8 * pData,

UINT32 dataSize )
```

Send a MD reply/reply query message.

Send either a MD reply message or a MD reply query message after receiving a request and ask for confirmation. User reference, source and destination IP addresses as well as topo counts and packet flags are taken from the session

### **Parameters**

| in | msgType  | msgType TRDP_MSG_MP or TRDP_MSG_MQ                      |  |
|----|--|---|--|
| in | appHandle the handle returned by tlc_init                                  |   |  |
| in | pSessionId   | Session ID returned by indication                       |  |
| in | comId  | comld of packet to be sent                              |  |
| in | timeout  | time out for confirmations (zero for TRDP_MSG_MP)       |  |
| in | replyStatus  | replyStatus Info for requester about application errors |  |
| in | pSendParam Pointer to send parameters, NULL to use default send parameters |   |  |
| in | pData pointer to packet data / dataset                                     |   |  |
| in | dataSize   | size of packet data                                     |  |

#### Return values

| TRDP_NO_ERR         | no error        |
|---------------------|-----------------|
| TRDP_PARAM_ERR      | parameter error |
| TRDP_MEM_ERR        | out of memory   |
| TRDP_NO_SESSION_ERR | no such session |

## 5.21.2.9 trdp\_mdSend()

Sending MD messages Send the messages stored in the sendQueue Call user's callback if needed.

### **Parameters**

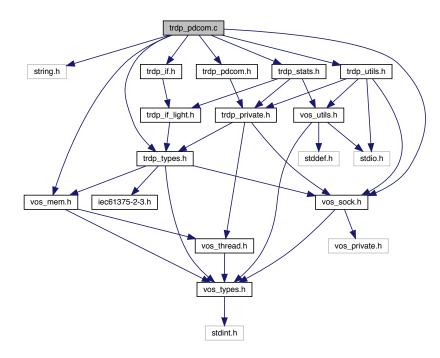
| in | appHandle | session pointer |
|----|-----------|-----------------|
|----|-----------|-----------------|

# 5.22 trdp\_pdcom.c File Reference

Functions for PD communication.

```
#include <string.h>
#include "trdp_types.h"
#include "trdp_utils.h"
#include "trdp_pdcom.h"
#include "trdp_if.h"
#include "trdp_stats.h"
#include "vos_sock.h"
#include "vos_mem.h"
```

Include dependency graph for trdp\_pdcom.c:



#### **Functions**

 void trdp\_pdInit (PD\_ELE\_T \*pPacket, TRDP\_MSG\_T type, UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, UINT32 replyComId, UINT32 replyIpAddress)

Initialize/construct the packet Set the header infos.

TRDP\_ERR\_T trdp\_pdPut (PD\_ELE\_T \*pPacket, TRDP\_MARSHALL\_T marshall, void \*refCon, const UI

NT8 \*pData, UINT32 dataSize)

Copy data Update the data to be sent.

• TRDP\_ERR\_T trdp\_pdGet (PD\_ELE\_T \*pPacket, TRDP\_UNMARSHALL\_T unmarshall, void \*refCon, const UINT8 \*pData, UINT32 \*pDataSize)

Copy data Set the header infos.

• TRDP\_ERR\_T trdp\_pdSendQueued (TRDP\_SESSION\_PT appHandle)

Send all due PD messages.

TRDP\_ERR\_T trdp\_pdReceive (TRDP\_SESSION\_PT appHandle, SOCKET sock)

Receiving PD messages Read the receive socket for arriving PDs, copy the packet to a new PD\_ELE\_T Check for protocol errors and compare the received data to the data in our receive queue.

void trdp\_pdCheckPending (TRDP\_APP\_SESSION\_T appHandle, TRDP\_FDS\_T \*pFileDesc, INT32 \*p
 — NoDesc)

Check for pending packets, set FD if non blocking.

• void trdp\_pdHandleTimeOuts (TRDP\_SESSION\_PT appHandle)

Check for time outs.

TRDP\_ERR\_T trdp\_pdCheckListenSocks (TRDP\_SESSION\_PT appHandle, TRDP\_FDS\_T \*pRfds, INT32 \*pCount)

Checking receive connection requests and data Call user's callback if needed.

void trdp pdUpdate (PD ELE T\*pPacket)

Update the header values.

```
• TRDP_ERR_T trdp_pdCheck (PD_HEADER_T *pPacket, UINT32 packetSize)
```

Check if the PD header values and the CRCs are sane.

• TRDP ERR T trdp pdSend (SOCKET pdSock, PD ELE T \*pPacket, UINT16 port)

Send one PD packet.

TRDP\_ERR\_T trdp\_pdDistribute (PD\_ELE\_T \*pSndQueue)

Distribute send time of PD packets over time.

#### 5.22.1 Detailed Description

Functions for PD communication.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2015. All rights reserved.

ld

#### trdp\_pdcom.c 1772 2018-10-30 12:18:12Z bloehr

```
BL 2018-10-29: Ticket #217 PD Pull requests must be subscribed for
 BL 2018-08-07: Ticket #207 tlp_put() and variable dataSize
 BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
 BL 2018-01-29: Ticket #186 Potential SEGFAULT in case of PD timeout
 BL 2017-11-28: Ticket #180 Filtering rules for DestinationURI does not follow the standard
 BL 2017-11-15: Ticket #1 Unjoin on unsubscribe/delListener (finally ;-)
 BL 2017-11-10: Ticket #172 Infinite loop of message sending after PD Pull Request when registered in multicas
 BL 2017-07-24: Ticket #166 Bug in trdp_pdReceive for "if data has changed"
 BL 2017-03-01: Ticket #136 PD topography counter with faulty behavior
 BL 2017-02-27: Ticket #146 On Timeout, PD Callback is always called with no data/datasize == 0
 BL 2017-02-10: Ticket #132: tlp_publish: Check of datasize wrong if using marshaller
 BL 2017-02-08: Ticket #142: Compiler warnings / MISRA-C 2012 issues
 BL 2017-02-08: Ticket #133: Accelerate PD packet reception
 BL 2016-06-24: Ticket #121: Callback on first packet after time out
BL 2016-06-08: Ticket #120: ComIds for statistics changed to proposed 61375 errata
 BL 2016-06-01: Ticket #119: tlc_getInterval() repeatedly returns 0 after timeout
BL 2016-03-04: Ticket #112: Marshalling sets wrong datasetLength (PD)
IBO 2016-02-03: Ticket #109: vos_ntohs -> vos_ntohl for datasetlength when unmarshalling
 BL 2016-01-25: Ticket #106: User needs to be informed on every received PD packet
 BL 2015-12-14: Ticket #33: source size check for marshalling
BL 2015-11-24: Ticket #104: PD telegrams with no data is never sent
BL 2015-08-31: Ticket #94: TRDP_REDUNDANT flag is evaluated, beQuiet removed
BL 2015-08-05: Ticket #81: Counts for packet loss
AHW 2015-04-10: Ticket #76: Wrong initialisation of frame pointer in trdp_pdReceive()
AHW 2015-04-10: Ticket #79: handling for dataSize==0/pData== NULL fixed in in trdp_pdPut()
BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed
                Ticket #47: Protocol change: no FCS for data part of telegrams
                Ticket #43: Usage of memset() in the trdp_pdReceive() function
BL 2014-06-02: Ticket #41: Sequence counter handling fixed
                Ticket #42: memcmp only if callback enabled
BL 2014-02-28: Ticket #25: CRC32 calculation is not according IEEE802.3
 BL 2014-02-27: Ticket #23: tlc_getInterval() always returning 10ms
 BL 2014-01-09: Ticket #14: Wrong error return in trdp_pdDistribute()
 BL 2013-06-24: ID 125: Time-out handling and ready descriptors fixed
 BL 2013-04-09: ID 92: Pull request led to reset of push message type
 BL 2013-01-25: ID 20: Redundancy handling fixed
```

## 5.22.2 Function Documentation

## 5.22.2.1 trdp\_pdCheck()

Check if the PD header values and the CRCs are sane.

## **Parameters**

| in | pPacket    | pointer to the packet to check |
|----|------------|--------------------------------|
| in | packetSize | max size to check              |

## Return values

| TRDP_NO_ERR  |  |
|--------------|--|
| TRDP_CRC_ERR |  |

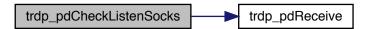
# 5.22.2.2 trdp\_pdCheckListenSocks()

Checking receive connection requests and data Call user's callback if needed.

#### **Parameters**

| in     | appHandle | session pointer                        |
|--------|-----------|--|
| in     | pRfds     | pointer to set of ready descriptors    |
| in,out | pCount    | pointer to number of ready descriptors |

Here is the call graph for this function:



#### 5.22.2.3 trdp\_pdCheckPending()

Check for pending packets, set FD if non blocking.

#### **Parameters**

| in     | appHandle | session pointer                        |
|--------|-----------|--|
| in,out | pFileDesc | pointer to set of ready descriptors    |
| in,out | pNoDesc   | pointer to number of ready descriptors |

#### 5.22.2.4 trdp\_pdDistribute()

```
TRDP_ERR_T trdp_pdDistribute ( PD\_ELE\_T \ * \ pSndQueue \ )
```

Distribute send time of PD packets over time.

The duration of PD packets on a 100MBit/s network ranges from 3us to 150us max. Because a cyclic thread scheduling below 5ms would put a too heavy load on the system, and PD packets cannot get larger than 1432 (+ UDP header), we will not account for differences in packet size. Another factor is the differences in intervals for different packets: We should only change the starting times of the packets within 1/2 the interval time. Otherwise a late addition of packets could lead to timeouts of already queued packets. Scheduling will be computed based on the smallest interval time.

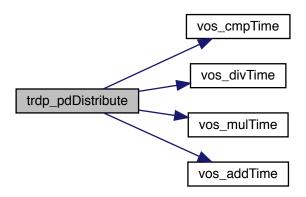
## **Parameters**

| in | pSndQueue | pointer to send queue |
|----|-----------|-----------------------|
|----|-----------|-----------------------|

#### Return values

TRDP\_NO\_ERR

Here is the call graph for this function:



## 5.22.2.5 trdp\_pdHandleTimeOuts()

Check for time outs.

#### **Parameters**

| in | appHandle | application handle |
|----|-----------|--------------------|

Here is the call graph for this function:



# 5.22.2.6 trdp\_pdInit()

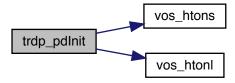
```
TRDP_MSG_T type,
UINT32 etbTopoCnt,
UINT32 opTrnTopoCnt,
UINT32 replyComId,
UINT32 replyIpAddress )
```

Initialize/construct the packet Set the header infos.

#### **Parameters**

| in | pPacket        | pointer to the packet element to init |
|----|----------------|---------------------------------------|
| in | type           | type the packet                       |
| in | etbTopoCnt     | topocount to use for PD frame         |
| in | opTrnTopoCnt   | topocount to use for PD frame         |
| in | replyComId     | Pull request comId                    |
| in | replylpAddress | Pull request lp                       |

Here is the call graph for this function:



## 5.22.2.7 trdp\_pdPut()

Copy data Update the data to be sent.

| in | pPacket  | pointer to the packet element to send |
|----|----------|---------------------------------------|
| in | marshall | pointer to marshalling function       |
| in | refCon   | reference for marshalling function    |
| in | pData    | pointer to data                       |
| in | dataSize | size of data                          |

## Return values

| TRDP_NO_ERR | no error other errors |
|-------------|-----------------------|
|-------------|-----------------------|

## 5.22.2.8 trdp\_pdReceive()

Receiving PD messages Read the receive socket for arriving PDs, copy the packet to a new PD\_ELE\_T Check for protocol errors and compare the received data to the data in our receive queue.

If it is a new packet, check if it is a PD Request (PULL). If it is an update, exchange the existing entry with the new one Call user's callback if needed

#### **Parameters**

| in | appHandle | session pointer         |
|----|-----------|-------------------------|
| in | sock      | the socket to read from |

## Return values

| TRDP_NO_ERR        | no error                                       |
|--------------------|--|
| TRDP_PARAM_ERR     | parameter error                                |
| TRDP_WIRE_ERR      | protocol error (late packet, version mismatch) |
| TRDP_QUEUE_ERR     | not in queue                                   |
| TRDP_CRC_ERR       | header checksum                                |
| TRDP_TOPOCOUNT_ERR | invalid topocount                              |

## 5.22.2.9 trdp\_pdSend()

Send one PD packet.

| in | pdSock  | socket descriptor            |
|----|---------|------------------------------|
| in | pPacket | pointer to packet to be sent |
| in | port    | port on which to send        |

## **Return values**

| TRDP_NO_ERR |  |
|-------------|--|
| TRDP_IO_ERR |  |

# 5.22.2.10 trdp\_pdSendQueued()

Send all due PD messages.

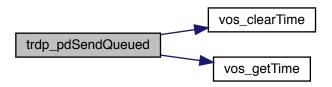
#### **Parameters**

| ter |
|-----|
|     |

## Return values

| TRDP_NO_ERR | no error         |
|-------------|------------------|
| TRDP_IO_ERR | socket I/O error |

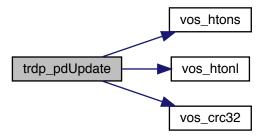
Here is the call graph for this function:



# 5.22.2.11 trdp\_pdUpdate()

Update the header values.

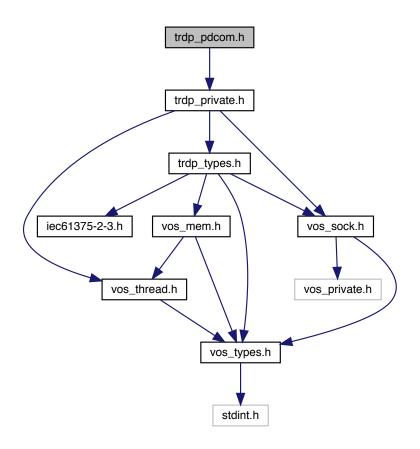
Here is the call graph for this function:



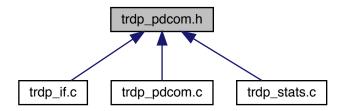
# 5.23 trdp\_pdcom.h File Reference

Functions for PD communication.

#include "trdp\_private.h"
Include dependency graph for trdp\_pdcom.h:



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



#### **Functions**

void trdp\_pdInit (PD\_ELE\_T \*, TRDP\_MSG\_T, UINT32 topoCount, UINT32 optopoCount, UINT32 reply
 — ComId, UINT32 replyIpAddress)

Initialize/construct the packet Set the header infos.

void trdp\_pdUpdate (PD\_ELE\_T \*)

Update the header values.

TRDP\_ERR\_T trdp\_pdPut (PD\_ELE\_T \*, TRDP\_MARSHALL\_T func, void \*refCon, const UINT8 \*pData, UINT32 dataSize)

Copy data Update the data to be sent.

• TRDP ERR T trdp pdCheck (PD HEADER T \*pPacket, UINT32 packetSize)

Check if the PD header values and the CRCs are sane.

• TRDP ERR T trdp pdSend (SOCKET pdSock, PD ELE T \*pPacket, UINT16 port)

Send one PD packet.

• TRDP\_ERR\_T trdp\_pdGet (PD\_ELE\_T \*pPacket, TRDP\_UNMARSHALL\_T unmarshall, void \*refCon, const UINT8 \*pData, UINT32 \*pDataSize)

Copy data Set the header infos.

• TRDP\_ERR\_T trdp\_pdSendQueued (TRDP\_SESSION\_PT appHandle)

Send all due PD messages.

TRDP\_ERR\_T trdp\_pdReceive (TRDP\_SESSION\_PT pSessionHandle, SOCKET sock)

Receiving PD messages Read the receive socket for arriving PDs, copy the packet to a new PD\_ELE\_T Check for protocol errors and compare the received data to the data in our receive queue.

void trdp\_pdCheckPending (TRDP\_APP\_SESSION\_T appHandle, TRDP\_FDS\_T \*pFileDesc, INT32 \*p
 — NoDesc)

Check for pending packets, set FD if non blocking.

void trdp\_pdHandleTimeOuts (TRDP\_SESSION\_PT appHandle)

Check for time outs.

TRDP\_ERR\_T trdp\_pdCheckListenSocks (TRDP\_SESSION\_PT appHandle, TRDP\_FDS\_T \*pRfds, INT32 \*pCount)

Checking receive connection requests and data Call user's callback if needed.

• TRDP ERR T trdp pdDistribute (PD ELE T\*pSndQueue)

Distribute send time of PD packets over time.

## 5.23.1 Detailed Description

Functions for PD communication.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

trdp\_pdcom.h 1740 2018-06-20 16:03:12Z bloehr

```
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed
Ticket #47: Protocol change: no FCS for data part of telegrams
```

## 5.23.2 Function Documentation

# 5.23.2.1 trdp\_pdCheck()

```
TRDP_ERR_T trdp_pdCheck (

PD_HEADER_T * pPacket,

UINT32 packetSize )
```

Check if the PD header values and the CRCs are sane.

#### **Parameters**

| in | pPacket    | pointer to the packet to check |
|----|------------|--------------------------------|
| in | packetSize | max size to check              |

#### Return values

| TRDP_NO_ERR  |  |
|--------------|--|
| TRDP_CRC_ERR |  |

## 5.23.2.2 trdp\_pdCheckListenSocks()

Checking receive connection requests and data Call user's callback if needed.

#### **Parameters**

| in     | appHandle | session pointer                        |
|--------|-----------|--|
| in     | pRfds     | pointer to set of ready descriptors    |
| in,out | pCount    | pointer to number of ready descriptors |

Here is the call graph for this function:



## 5.23.2.3 trdp\_pdCheckPending()

Check for pending packets, set FD if non blocking.

## **Parameters**

| in     | appHandle | session pointer                        |
|--------|-----------|--|
| in,out | pFileDesc | pointer to set of ready descriptors    |
| in,out | pNoDesc   | pointer to number of ready descriptors |

# 5.23.2.4 trdp\_pdDistribute()

```
TRDP_ERR_T trdp_pdDistribute ( {\tt PD\_ELE\_T} \ * \ pSndQueue \ )
```

Distribute send time of PD packets over time.

The duration of PD packets on a 100MBit/s network ranges from 3us to 150us max. Because a cyclic thread scheduling below 5ms would put a too heavy load on the system, and PD packets cannot get larger than 1432 (+ UDP header), we will not account for differences in packet size. Another factor is the differences in intervals for different packets: We should only change the starting times of the packets within 1/2 the interval time. Otherwise a late addition of packets could lead to timeouts of already queued packets. Scheduling will be computed based on the smallest interval time.

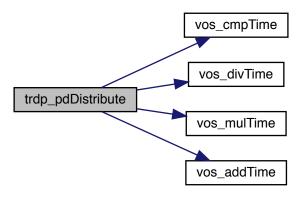
#### **Parameters**

| in | pSndQueue | pointer to send queue |
|----|-----------|-----------------------|
|----|-----------|-----------------------|

#### Return values

```
TRDP_NO_ERR
```

Here is the call graph for this function:



## 5.23.2.5 trdp\_pdHandleTimeOuts()

Check for time outs.

| in | appHandle | application handle |  |
|----|-----------|--------------------|--|

Here is the call graph for this function:



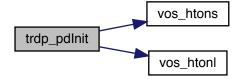
# 5.23.2.6 trdp\_pdInit()

Initialize/construct the packet Set the header infos.

## **Parameters**

| in | pPacket        | pointer to the packet element to init |
|----|----------------|---------------------------------------|
| in | type           | type the packet                       |
| in | etbTopoCnt     | topocount to use for PD frame         |
| in | opTrnTopoCnt   | topocount to use for PD frame         |
| in | replyComId     | Pull request comId                    |
| in | replylpAddress | Pull request lp                       |

Here is the call graph for this function:



#### 5.23.2.7 trdp\_pdPut()

Copy data Update the data to be sent.

#### **Parameters**

| in | pPacket  | pointer to the packet element to send |
|----|----------|---------------------------------------|
| in | marshall | pointer to marshalling function       |
| in | refCon   | reference for marshalling function    |
| in | pData    | pointer to data                       |
| in | dataSize | size of data                          |

#### Return values

| TRDP_NO_ERR | no error other errors |
|-------------|-----------------------|
|-------------|-----------------------|

## 5.23.2.8 trdp\_pdReceive()

Receiving PD messages Read the receive socket for arriving PDs, copy the packet to a new PD\_ELE\_T Check for protocol errors and compare the received data to the data in our receive queue.

If it is a new packet, check if it is a PD Request (PULL). If it is an update, exchange the existing entry with the new one Call user's callback if needed

#### **Parameters**

| in             | appHandle | session pointer         |
|----------------|-----------|-------------------------|
| in <b>sock</b> |           | the socket to read from |

## Return values

| TRDP_NO_ERR        | no error                                       |
|--------------------|--|
| TRDP_PARAM_ERR     | parameter error                                |
| TRDP_WIRE_ERR      | protocol error (late packet, version mismatch) |
| TRDP_QUEUE_ERR     | not in queue                                   |
| TRDP_CRC_ERR       | header checksum                                |
| TRDP_TOPOCOUNT_ERR | invalid topocount                              |

# 5.23.2.9 trdp\_pdSend()

Send one PD packet.

## **Parameters**

| in | pdSock  | socket descriptor            |
|----|---------|------------------------------|
| in | pPacket | pointer to packet to be sent |
| in | port    | port on which to send        |

## Return values

| TRDP_NO_ERR |  |
|-------------|--|
| TRDP_IO_ERR |  |

# 5.23.2.10 trdp\_pdSendQueued()

Send all due PD messages.

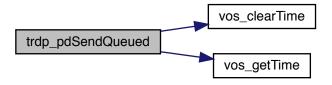
## **Parameters**

| in | appHandle | session pointer |
|----|-----------|-----------------|
|----|-----------|-----------------|

## Return values

| TRDP_NO_ERR | no error         |
|-------------|------------------|
| TRDP_IO_ERR | socket I/O error |

Here is the call graph for this function:



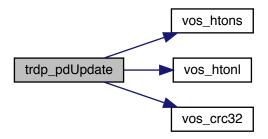
# 5.23.2.11 trdp\_pdUpdate()

Update the header values.

#### **Parameters**

| in | pPacket | pointer to the packet to update |
|----|---------|---------------------------------|
|----|---------|---------------------------------|

Here is the call graph for this function:



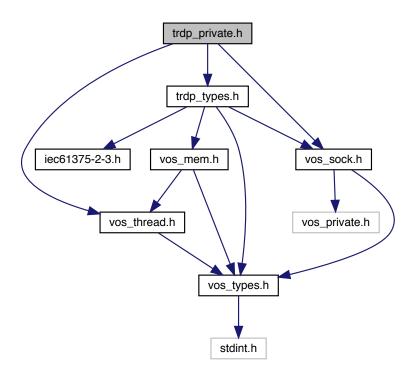
# 5.24 trdp\_private.h File Reference

Typedefs for TRDP communication.

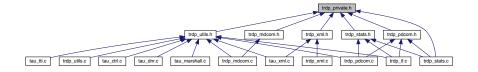
```
#include "trdp_types.h"
#include "vos_thread.h"
```

#include "vos\_sock.h"

Include dependency graph for trdp\_private.h:



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



## **Data Structures**

struct TRDP\_HANDLE

Hidden handle definition, used as unique addressing item.

struct TRDP\_SEQ\_CNT\_ENTRY\_T

Tuples of last received sequence counter per comld.

• struct TRDP\_SOCKET\_TCP

TCP parameters.

• struct TRDP\_SOCKETS

Socket item.

struct GNU PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

struct GNU\_PACKED

Types for ETB control.

struct PD\_ELE

Queue element for PD packets to send or receive.

struct TRDP SESSION

Session/application variables store.

#### **Macros**

• #define TRDP\_TIMER\_GRANULARITY 10000u

granularity in us

• #define TRDP DEBUG DEFAULT FILE SIZE 65536u

Default maximum size of log file.

• #define TRDP\_SEQ\_CNT\_START\_ARRAY\_SIZE 64u

This should be enough for the start.

#define TRDP\_IF\_WAIT\_FOR\_READY 120u

120 seconds (120 tries each second to bind to an IP address)

• #define TRDP\_PRIV\_NONE 0u

Internal flags for packets.

• #define TRDP TIMED OUT 0x2u

if set, inform the user

#define TRDP\_INVALID\_DATA 0x4u

if set, inform the user

• #define TRDP\_REQ\_2B\_SENT 0x8u

if set, the request needs to be sent

• #define TRDP PULL SUB 0x10u

if set, its a PULL subscription

• #define TRDP\_REDUNDANT 0x20u

if set, packet should not be sent (redundant)

#define TRDP\_CHECK\_COMID 0x40u

if set, do filter comId (addListener)

## **Typedefs**

typedef struct TRDP\_HANDLE TRDP\_ADDRESSES\_T

Hidden handle definition, used as unique addressing item.

typedef struct TRDP\_SOCKET\_TCP TRDP\_SOCKET\_TCP\_T

TCP parameters.

typedef struct TRDP\_SOCKETS\_T

Socket item.

typedef struct PD\_ELE PD\_ELE\_T

Queue element for PD packets to send or receive.

typedef struct TRDP\_SESSION TRDP\_SESSION\_T

Session/application variables store.

#### **Enumerations**

```
• enum TRDP MD ELE ST T{
 TRDP ST NONE = 0u,
 TRDP_ST_TX_NOTIFY_ARM = 1u,
 TRDP_ST_TX_REQUEST_ARM = 2u,
 TRDP ST TX REPLY ARM = 3u,
 TRDP_ST_TX_REPLYQUERY_ARM = 4u,
 TRDP_ST_TX_CONFIRM_ARM = 5u,
 TRDP ST RX READY = 6,
 TRDP ST TX REQUEST W4REPLY = 7u,
 TRDP ST RX REPLYQUERY W4C = 8u,
 TRDP ST RX REQ W4AP REPLY = 9u,
 TRDP_ST_TX_REQ_W4AP_CONFIRM = 10u,
 TRDP ST RX REPLY SENT = 11u,
 TRDP_ST_RX_NOTIFY_RECEIVED = 12u,
 TRDP_ST_TX_REPLY_RECEIVED = 13u,
 TRDP_ST_RX_CONF_RECEIVED = 14u }
    Internal MD state.
enum TRDP_SOCK_TYPE_T {
 TRDP SOCK PD = 0u,
 TRDP SOCK MD UDP = 1u,
 TRDP SOCK MD TCP = 2u }
    Socket usage.
```

#### 5.24.1 Detailed Description

Typedefs for TRDP communication.

TRDP internal type definitions

Note

Project: TCNOpen TRDP prototype stack

#### **Author**

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

#### trdp\_private.h 1780 2018-11-07 09:51:13Z bloehr

```
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2017-11-28: Ticket #180 Filtering rules for DestinationURI does not follow the standard
BL 2017-11-17: superfluous session->redID replaced by sndQueue->redId
BL 2017-11-17: undone: Ticket #169 Encapsulate declaration of packed structures within a macro
BL 2017-05-08: Compiler warnings: enum flags to #defines
BL 2017-05-08: Ticket #155: Kill trdp_proto.h - move definitions to iec61375-2-3.h and here
BL 2017-02-28: Ticket #140 TRDP_TIMER_FOREVER ->
BL 2017-02-28: Ticket #142 Compiler warnings / MISRA-C 2012 issues
BL 2015-08-31: Ticket #94: "beQuiet" removed
BL 2015-08-05: Ticket #81: Counts for packet loss
BL 2014-06-02: Ticket #41: Sequence counter handling fixed
```

# 5.24.2 Enumeration Type Documentation

# 5.24.2.1 TRDP\_MD\_ELE\_ST\_T

enum TRDP\_MD\_ELE\_ST\_T

Internal MD state.

#### Enumerator

| TRDP_ST_NONE                | neutral value   |
|-----------------------------|---|
| TRDP_ST_TX_NOTIFY_ARM       | ready to send notify MD                               |
| TRDP_ST_TX_REQUEST_ARM      | ready to send request MD                              |
| TRDP_ST_TX_REPLY_ARM        | ready to send reply MD                                |
| TRDP_ST_TX_REPLYQUERY_ARM   | ready to send reply with confirm request MD           |
| TRDP_ST_TX_CONFIRM_ARM      | ready to send confirm MD                              |
| TRDP_ST_RX_READY            | armed listener  |
| TRDP_ST_TX_REQUEST_W4REPLY  | request sent, wait for reply                          |
| TRDP_ST_RX_REPLYQUERY_W4C   | reply send, with confirm request MD                   |
| TRDP_ST_RX_REQ_W4AP_REPLY   | request received, wait for application reply send     |
| TRDP_ST_TX_REQ_W4AP_CONFIRM | reply conf. rq. tx, wait for application conf send    |
| TRDP_ST_RX_REPLY_SENT       | reply sent  |
| TRDP_ST_RX_NOTIFY_RECEIVED  | notification received, wait for application to accept |
| TRDP_ST_TX_REPLY_RECEIVED   | reply received  |
| TRDP_ST_RX_CONF_RECEIVED    | confirmation received                                 |

## 5.24.2.2 TRDP\_SOCK\_TYPE\_T

enum TRDP\_SOCK\_TYPE\_T

## Socket usage.

## Enumerator

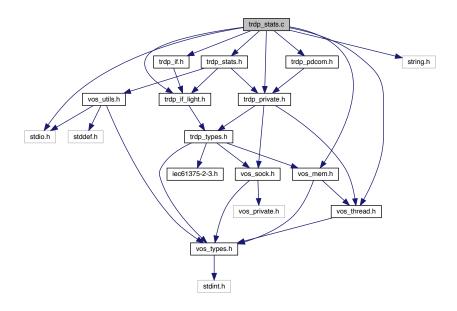
| TRDP_SOCK_PD     | Socket is used for UDP process data. |
|------------------|--------------------------------------|
| TRDP_SOCK_MD_UDP | Socket is used for UDP message data. |
| TRDP_SOCK_MD_TCP | Socket is used for TCP message data. |

# 5.25 trdp\_stats.c File Reference

Statistics functions for TRDP communication.

```
#include <stdio.h>
#include <string.h>
#include "trdp_stats.h"
#include "trdp_if_light.h"
#include "trdp_if.h"
#include "trdp_private.h"
#include "trdp_pdcom.h"
#include "vos_mem.h"
#include "vos_thread.h"
```

Include dependency graph for trdp stats.c:



## **Functions**

void trdp\_UpdateStats (TRDP\_APP\_SESSION\_T appHandle)

Update the statistics.

void trdp\_initStats (TRDP\_APP\_SESSION\_T appHandle)

Init statistics.

• EXT\_DECL TRDP\_ERR\_T tlc\_resetStatistics (TRDP\_APP\_SESSION\_T appHandle)

• EXT\_DECL TRDP\_ERR\_T tlc\_getStatistics (TRDP\_APP\_SESSION\_T appHandle, TRDP\_STATISTICS\_T \*pStatistics)

Return statistics.

 EXT\_DECL TRDP\_ERR\_T tlc\_getSubsStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNum← Subs, TRDP\_SUBS\_STATISTICS\_T \*pStatistics)

Return PD subscription statistics.

• EXT\_DECL TRDP\_ERR\_T tlc\_getPubStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNumPub, TRDP PUB STATISTICS T \*pStatistics)

Return PD publish statistics.

• EXT\_DECL TRDP\_ERR\_T tlc\_getRedStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNumRed, TRDP RED STATISTICS T \*pStatistics)

Return redundancy group statistics.

 EXT\_DECL TRDP\_ERR\_T tlc\_getJoinStatistics (TRDP\_APP\_SESSION\_T appHandle, UINT16 \*pNumJoin, UINT32 \*plpAddr)

Return join statistics.

void trdp\_pdPrepareStats (TRDP\_APP\_SESSION\_T appHandle, PD\_ELE\_T \*pPacket)
 Fill the statistics packet.

## 5.25.1 Detailed Description

Statistics functions for TRDP communication.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

trdp stats.c 1740 2018-06-20 16:03:12Z bloehr

```
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2017-11-17: superfluous session->redID replaced by sndQueue->redId
BL 2017-05-22: Ticket #122: Addendum for 64Bit compatibility (VOS_TIME_T -> VOS_TIMEVAL_T)
BL 2017-05-08: Compiler warnings
BL 2017-03-01: Ticket #149 SourceUri and DestinationUri don't with 32 characters
BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings)
BL 2016-05-04: Ticket #117: PD Status packet is not sent on request
BL 2015-08-05: Ticket #81: Counts for packet loss
```

## 5.25.2 Function Documentation

#### 5.25.2.1 tlc\_getJoinStatistics()

Return join statistics.

Memory for statistics information must be provided by the user.

## **Parameters**

| in     | appHandle | the handle returned by tlc_openSession        |
|--------|-----------|---|
| in,out | pNumJoin  | Pointer to the number of joined IP Adresses   |
| out    | plpAddr   | Pointer to a list with the joined IP adresses |

#### Return values

| TRDP_NO_ERR     | no error                            |
|-----------------|-------------------------------------|
| TRDP_NOINIT_ERR | handle invalid                      |
| TRDP_PARAM_ERR  | parameter error                     |
| TRDP_MEM_ERR    | there are more items than requested |

## 5.25.2.2 tlc\_getPubStatistics()

Return PD publish statistics.

Memory for statistics information must be provided by the user.

#### Parameters

| in     | appHandle the handle returned by tlc_openSession |   |  |
|--------|--|---|--|
| in,out | pNumPub Pointer to the number of publishers      |   |  |
| out    | pStatistics                                      | tatistics Pointer to a list with the publish statistics information |  |

#### Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

## 5.25.2.3 tlc\_getRedStatistics()

Return redundancy group statistics.

Memory for statistics information must be provided by the user.

#### **Parameters**

| in     | appHandle  | the handle returned by tlc_openSession     |
|--------|--|--|
| in,out | pNumRed  | Pointer to the number of redundancy groups |
| out    | out pStatistics Pointer to a list with the redundancy group inform |  |

#### Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

## 5.25.2.4 tlc\_getStatistics()

Return statistics.

Memory for statistics information must be provided by the user.

#### **Parameters**

| in  | appHandle              | the handle returned by tlc_openSession             |  |
|-----|------------------------|--|--|
| out | out <i>pStatistics</i> | Pointer to statistics for this application session |  |

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_NOINIT_ERR | handle invalid  |
| TRDP_PARAM_ERR  | parameter error |

## 5.25.2.5 tlc\_getSubsStatistics()

Return PD subscription statistics.

Memory for statistics information must be provided by the user.

## **Parameters**

| in     | appHandle  | the handle returned by tlc_openSession  |
|--------|--|---|
| in,out | pNumSubs   | In: The number of subscriptions requested Out: Number of subscriptions returned |
| in,out | pStatistics Pointer to an array with the subscription statistics information |   |

## Return values

| TRDP_NO_ERR     | no error                                    |
|-----------------|---|
| TRDP_NOINIT_ERR | handle invalid                              |
| TRDP_PARAM_ERR  | parameter error                             |
| TRDP_MEM_ERR    | there are more subscriptions than requested |

## 5.25.2.6 tlc\_resetStatistics()

Reset statistics.

#### **Parameters**

| iı | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
|----|-----------|--|

## Return values

| TRDP_NO_ERR     | no error        |
|-----------------|-----------------|
| TRDP_NOINIT_ERR | handle invalid  |
| TRDP_PARAM_ERR  | parameter error |

# 5.25.2.7 trdp\_initStats()

Init statistics.

Clear the stats structure for a session.

## **Parameters**

|  | in | appHandle | the handle returned by tlc_openSession |
|--|----|-----------|--|
|--|----|-----------|--|

## < host name

< leader host name Here is the call graph for this function:



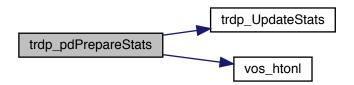
# 5.25.2.8 trdp\_pdPrepareStats()

Fill the statistics packet.

#### **Parameters**

| in     | appHandle | the handle returned by tlc_openSession |
|--------|-----------|--|
| in,out | pPacket   | pointer to the packet to fill          |

Here is the call graph for this function:



# 5.25.2.9 trdp\_UpdateStats()

Update the statistics.

#### **Parameters**

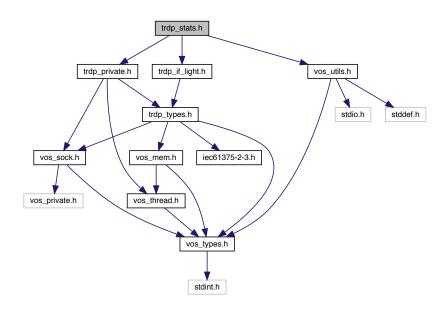
|  | n <b>appHandle</b> | the handle returned by tlc_openSession | ] |
|--|--------------------|--|---|
|--|--------------------|--|---|

# 5.26 trdp\_stats.h File Reference

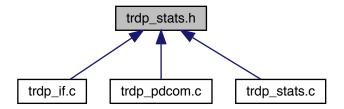
Statistics for TRDP communication.

```
#include "trdp_if_light.h"
#include "trdp_private.h"
#include "vos_utils.h"
```

Include dependency graph for trdp\_stats.h:



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



## **Functions**

void trdp\_initStats (TRDP\_APP\_SESSION\_T appHandle)

Init statistics.

• void trdp\_pdPrepareStats (TRDP\_APP\_SESSION\_T appHandle, PD\_ELE\_T \*pPacket)

Fill the statistics packet.

## 5.26.1 Detailed Description

Statistics for TRDP communication.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

trdp\_stats.h 1065 2013-09-06 08:12:09Z aweiss

## 5.26.2 Function Documentation

#### 5.26.2.1 trdp\_initStats()

Init statistics.

Clear the stats structure for a session.

#### **Parameters**

|  | in | appHandle | the handle returned by tlc_openSession |
|--|----|-----------|--|
|--|----|-----------|--|

< host name

< leader host name Here is the call graph for this function:



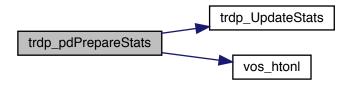
## 5.26.2.2 trdp\_pdPrepareStats()

Fill the statistics packet.

#### **Parameters**

| in     | appHandle | the handle returned by tlc_openSession |
|--------|-----------|--|
| in,out | pPacket   | pointer to the packet to fill          |

Here is the call graph for this function:

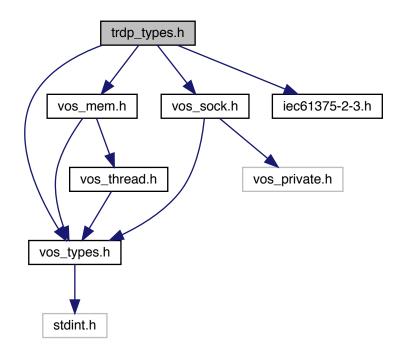


# 5.27 trdp\_types.h File Reference

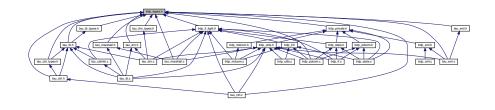
Typedefs for TRDP communication.

```
#include "vos_types.h"
#include "vos_mem.h"
#include "vos_sock.h"
```

#include "iec61375-2-3.h"
Include dependency graph for trdp\_types.h:



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



## **Data Structures**

• struct TRDP PD INFO T

Process data info from received telegram; allows the application to generate responses.

struct TRDP\_MD\_INFO\_T

Message data info from received telegram; allows the application to generate responses.

• struct TRDP\_SEND\_PARAM\_T

Quality/type of service and time to live.

• struct TRDP\_DATASET\_ELEMENT\_T

Dataset element definition.

struct TRDP\_DATASET

Dataset definition.

struct TRDP\_COMID\_DSID\_MAP\_T

Comld - data set mapping element definition.

• struct TRDP\_STATISTICS\_REQUEST\_T

TRDP statistics type definitions.

struct TRDP\_MEM\_STATISTICS\_T

Structure containing all general memory statistics information.

struct TRDP PD STATISTICS T

Structure containing all general PD statistics information.

struct TRDP\_MD\_STATISTICS\_T

Structure containing all general MD statistics information.

struct TRDP STATISTICS T

Structure containing all general memory, PD and MD statistics information.

struct TRDP SUBS STATISTICS T

Table containing particular PD subscription information.

struct TRDP\_PUB\_STATISTICS\_T

Table containing particular PD publishing information.

struct TRDP\_LIST\_STATISTICS\_T

Information about a particular MD listener.

struct TRDP RED STATISTICS T

A table containing PD redundant group information.

struct TRDP\_MARSHALL\_CONFIG\_T

Marshaling/unmarshalling configuration.

struct TRDP\_PD\_CONFIG\_T

Default PD configuration.

struct TRDP\_MD\_CONFIG\_T

Default MD configuration.

· struct TRDP MEM CONFIG T

Enumeration type for memory pre-fragmentation, reuse of VOS definition.

struct TRDP\_PROCESS\_CONFIG\_T

Various flags/general TRDP options for library initialization.

## **Macros**

• #define USE HEAP 0

If this is set, we can allocate dynamically memory.

• #define TRDP\_FLAGS\_DEFAULT 0u

Various flags for PD and MD packets.

#define TRDP\_FLAGS\_NONE 0x01u

No flags set.

• #define TRDP FLAGS MARSHALL 0x02u

Optional marshalling/unmarshalling in TRDP stack.

• #define TRDP FLAGS CALLBACK 0x04u

Use of callback function.

• #define TRDP FLAGS TCP 0x08u

Use TCP for message data.

#define TRDP\_FLAGS\_FORCE\_CB 0x10u

Force a callback for every received packet.

#define TRDP INFINITE TIMEOUT 0xfffffffu

Infinite reply timeout.

#define TRDP\_TIMER\_FOREVER 0xffffffff

No time out for subscription.

#define TRDP BOOL8 TRDP BITSET8

1 bit relevant (equal to zero = false, not equal to zero = true)

#define TRDP ANTIVALENT8 TRDP BITSET8

2 bit relevant (0x0 = errror, 0x01 = false, 0x02 = true, 0x03 undefined)

#define TRDP\_OPTION\_NONE 0u

Various flags/general TRDP options for library initialization.

#define TRDP OPTION BLOCK 0x01u

Default: Use nonblocking I/O calls, polling necessary Set: Read calls will block, use select()

#define TRDP OPTION TRAFFIC SHAPING 0x02u

Use traffic shaping - distribute packet sending Default: OFF.

• #define TRDP OPTION NO REUSE ADDR 0x04u

Do not allow re-use of address/port (-> no multihoming) Default: Allow.

#define TRDP OPTION NO MC LOOP BACK 0x08u

Do not allow loop back of multicast traffic Default: Allow.

#define TRDP OPTION NO UDP CHK 0x10u

Suppress UDP CRC generation Default: Compute UDP CRC.

#### **Typedefs**

typedef VOS IP4 ADDR T TRDP IP ADDR T

TRDP general type definitions.

typedef CHAR8 TRDP\_NET\_LABEL\_T[TRDP\_MAX\_LABEL\_LEN]

Definition for usage in network packets, not necessarily \0 terminated!

typedef VOS\_VERSION\_T TRDP\_VERSION\_T

Version information.

typedef VOS TIMEVAL T TRDP TIME T

Timer value compatible with timeval / select.

typedef VOS\_FDS\_T TRDP\_FDS\_T

File descriptor set compatible with fd\_set / select.

typedef VOS\_UUID\_T TRDP\_UUID\_T

UUID definition reuses the VOS definition.

typedef struct TRDP\_DATASET\_T

Dataset definition.

typedef TRDP DATASET T \* pTRDP DATASET T

Array of pointers to dataset.

typedef VOS\_PRINT\_DBG\_T TRDP\_PRINT\_DBG\_T

TRDP configuration type definitions.

typedef VOS\_LOG\_T TRDP\_LOG\_T

Categories for logging, reuse of the VOS definition.

typedef TRDP\_ERR\_T(\* TRDP\_MARSHALL\_T) (void \*pRefCon, UINT32 comId, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDst, UINT32 \*pDstSize, TRDP\_DATASET\_T \*\*ppCachedDS)

Function type for marshalling .

• typedef TRDP\_ERR\_T(\* TRDP\_UNMARSHALL\_T) (void \*pRefCon, UINT32 comId, UINT8 \*pSrc, UINT32 srcSize, UINT8 \*pDst, UINT32 \*pDstSize, TRDP\_DATASET\_T \*\*ppCachedDS)

Function type for unmarshalling.

• typedef void(\* TRDP\_PD\_CALLBACK\_T) (void \*pRefCon, TRDP\_APP\_SESSION\_T appHandle, const T← RDP\_PD\_INFO\_T \*pMsg, UINT8 \*pData, UINT32 dataSize)

Callback for receiving indications, timeouts, releases, responses.

 typedef void(\* TRDP\_MD\_CALLBACK\_T) (void \*pRefCon, TRDP\_APP\_SESSION\_T appHandle, const T← RDP\_MD\_INFO\_T \*pMsg, UINT8 \*pData, UINT32 dataSize)

Callback for receiving indications, timeouts, releases, responses.

#### **Enumerations**

```
• enum TRDP ERR T {
 TRDP NO ERR = 0,
 TRDP PARAM ERR = -1,
 TRDP INIT ERR = -2,
 TRDP_NOINIT_ERR = -3,
 TRDP_TIMEOUT_ERR = -4,
 TRDP NODATA ERR = -5,
 TRDP SOCK ERR = -6,
 TRDP IO ERR = -7,
 TRDP\_MEM\_ERR = -8,
 TRDP_SEMA_ERR = -9,
 TRDP_QUEUE_ERR = -10,
 TRDP_QUEUE_FULL_ERR = -11,
 TRDP_MUTEX_ERR = -12,
 TRDP THREAD ERR = -13,
 TRDP BLOCK ERR = -14,
 TRDP INTEGRATION ERR = -15,
 TRDP_NOCONN_ERR = -16,
 TRDP_NOSESSION_ERR = -30,
 TRDP SESSION ABORT ERR = -31,
 TRDP_NOSUB_ERR = -32,
 TRDP_NOPUB_ERR = -33,
 TRDP NOLIST ERR = -34,
 TRDP CRC ERR = -35,
 TRDP WIRE ERR = -36,
 TRDP TOPO ERR = -37,
 TRDP COMID ERR = -38,
 TRDP\_STATE\_ERR = -39,
 TRDP_APP_TIMEOUT_ERR = -40,
 TRDP_APP_REPLYTO_ERR = -41,
 TRDP_APP_CONFIRMTO_ERR = -42,
 TRDP REPLYTO ERR = -43,
 TRDP_CONFIRMTO_ERR = -44,
 TRDP_REQCONFIRMTO_ERR = -45,
 TRDP PACKET ERR = -46,
 TRDP_UNRESOLVED_ERR = -47,
 TRDP_XML_PARSER_ERR = -48,
 TRDP_INUSE_ERR = -49,
 TRDP MARSHALLING ERR = -50,
 TRDP UNKNOWN ERR = -99 }
    Return codes for all API functions, -1..-29 taken over from vos.

    enum TRDP REPLY STATUS T

    TRDP data transfer type definitions.
enum TRDP_RED_STATE_T {
 TRDP_RED_FOLLOWER = 0u,
 TRDP_RED_LEADER = 1u }
    Redundancy states.
enum TRDP_TO_BEHAVIOR_T {
 TRDP_TO_DEFAULT = 0u,
 TRDP_TO_SET_TO_ZERO = 1u,
 TRDP TO KEEP LAST VALUE = 2u }
    How invalid PD shall be handled.
• enum TRDP DATA TYPE T {
 TRDP_INVALID = 0u,
 TRDP_BITSET8 = 1u,
```

```
TRDP_CHAR8 = 2u,
TRDP_UTF16 = 3u,
TRDP_INT8 = 4u,
TRDP_INT16 = 5u,
TRDP INT32 = 6u,
TRDP INT64 = 7u,
TRDP UINT8 = 8u.
TRDP UINT16 = 9u,
TRDP UINT32 = 10u,
TRDP\_UINT64 = 11u,
TRDP_REAL32 = 12u,
TRDP_REAL64 = 13u,
TRDP_TIMEDATE32 = 14u
TRDP TIMEDATE48 = 15u,
TRDP_TIMEDATE64 = 16u,
TRDP_TYPE_MAX = 30u }
```

TRDP dataset description definitions.

## 5.27.1 Detailed Description

Typedefs for TRDP communication.

F

Note

Project: TCNOpen TRDP prototype stack

## Author

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2015. All rights reserved.

```
BL 2018-09-05: Ticket #211 XML handling: Dataset Name should be stored in TRDP_DATASET_ELEMENT_T
BL 2018-05-02: Ticket #188 Typo in the TRDP_VAR_SIZE definition
BL 2017-11-13: Ticket #176 TRDP_LABEL_T breaks field alignment -> TRDP_NET_LABEL_T
BL 2017-05-22: Ticket #122: Addendum for 64Bit compatibility (VOS_TIME_T -> VOS_TIMEVAL_T)
AHW 2017-05-22: Ticket #158 Infinit timeout at TRDB level is 0 acc. standard
BL 2017-05-08: Compiler warnings, doxygen comment errors
BL 2017-04-28: Ticket #155: Kill trdp_proto.h - move definitions to iec61375-2-3.h
BL 2017-03-01: Ticket #149 SourceUri and DestinationUri don't with 32 characters
BL 2017-02-27: Ticket #142 Compiler warnings / MISRA-C 2012 issues
BL 2016-06-08: Ticket #120: ComIds for statistics changed to proposed 61375 errata
BL 2016-02-11: Ticket #111: 'unit', 'scale', 'offset' attributes added to TRDP_DATASET_ELEMENT
BL 2016-01-25: Ticket #106: User needs to be informed on every received PD packet
BL 2015-12-14: Ticket #33: source size check for marshalling
BL 2015-08-05: Ticket #81: Counts for packet loss
BL 2014-07-14: Ticket #46: Protocol change: operational topocount needed
BL 2014-02-27: Ticket #17: tlp_subscribe() returns wrong *pSubHandle
```

## 5.27.2 Macro Definition Documentation

## 5.27.2.1 TRDP\_FLAGS\_DEFAULT

```
#define TRDP_FLAGS_DEFAULT Ou
```

Various flags for PD and MD packets.

Default value defined in tlc\_openDession will be taken

# 5.27.3 Typedef Documentation

#### 5.27.3.1 TRDP\_IP\_ADDR\_T

```
typedef VOS_IP4_ADDR_T TRDP_IP_ADDR_T
```

TRDP general type definitions.

## 5.27.3.2 TRDP\_MARSHALL\_T

```
typedef TRDP_ERR_T(* TRDP_MARSHALL_T) (void *pRefCon, UINT32 comId, UINT8 *pSrc, UINT32 src← Size, UINT8 *pDst, UINT32 *pDstSize, TRDP_DATASET_T **ppCachedDS)
```

Function type for marshalling .

The function must know about the dataset's alignment etc.

#### **Parameters**

| in     | pRefCon    | pointer to user context                                  |
|--------|------------|--|
| in     | comId      | ComId to identify the structure out of a configuration   |
| in     | pSrc       | pointer to received original message                     |
| in     | srcSize    | size of the source buffer                                |
| in     | pDst       | pointer to a buffer for the treated message              |
| in,out | pDstSize   | size of the provide buffer / size of the treated message |
| in,out | ppCachedDS | pointer to pointer of cached dataset                     |

## Return values

| TRDP_NO_ERR    | no error                 |
|----------------|--------------------------|
| TRDP_MEM_ERR   | provided buffer to small |
| TRDP_COMID_ERR | comid not existing       |

#### 5.27.3.3 TRDP\_MD\_CALLBACK\_T

```
typedef void(* TRDP_MD_CALLBACK_T) (void *pRefCon, TRDP_APP_SESSION_T appHandle, const TRDP_M← D_INFO_T *pMsg, UINT8 *pData, UINT32 dataSize)
```

Callback for receiving indications, timeouts, releases, responses.

#### **Parameters**

| in | appHandle | handle returned also by tlc_init               |
|----|-----------|--|
| in | pRefCon   | pointer to user context                        |
| in | pMsg      | pointer to received message information        |
| in | pData     | pointer to received data                       |
| in | dataSize  | size of received data pointer to received data |

## 5.27.3.4 TRDP\_PD\_CALLBACK\_T

```
typedef void(* TRDP_PD_CALLBACK_T) (void *pRefCon, TRDP_APP_SESSION_T appHandle, const TRDP_P \leftarrow D_INFO_T *pMsg, UINT8 *pData, UINT32 dataSize)
```

Callback for receiving indications, timeouts, releases, responses.

#### **Parameters**

| in | pRefCon   | pointer to user context                        |
|----|-----------|--|
| in | appHandle | application handle returned by tlc_openSession |
| in | pMsg      | pointer to received message information        |
| in | pData     | pointer to received data                       |
| in | dataSize  | size of received data pointer to received data |

#### 5.27.3.5 TRDP\_PRINT\_DBG\_T

```
typedef VOS_PRINT_DBG_T TRDP_PRINT_DBG_T
```

TRDP configuration type definitions.

Callback function definition for error/debug output, reuse of the VOS defined function.

#### 5.27.3.6 TRDP\_TIME\_T

```
typedef VOS_TIMEVAL_T TRDP_TIME_T
```

Timer value compatible with timeval / select.

Relative or absolute date, depending on usage

## 5.27.3.7 TRDP\_UNMARSHALL\_T

typedef TRDP\_ERR\_T(\* TRDP\_UNMARSHALL\_T) (void \*pRefCon, UINT32 comId, UINT8 \*pSrc, UINT32 src $\leftarrow$  Size, UINT8 \*pDst, UINT32 \*pDstSize, TRDP\_DATASET\_T \*\*ppCachedDS)

Function type for unmarshalling.

The function must know about the dataset's alignment etc.

#### **Parameters**

| in     | pRefCon    | pointer to user context                                  |
|--------|------------|--|
| in     | comId      | ComId to identify the structure out of a configuration   |
| in     | pSrc       | pointer to received original message                     |
| in     | srcSize    | data length from TRDP packet header                      |
| in     | pDst       | pointer to a buffer for the treated message              |
| in,out | pDstSize   | size of the provide buffer / size of the treated message |
| in,out | ppCachedDS | pointer to pointer of cached dataset                     |

## **Return values**

| TRDP_NO_ERR    | no error                |
|----------------|-------------------------|
| TRDP_MEM_ERR   | provide buffer to small |
| TRDP_COMID_ERR | comid not existing      |

# 5.27.4 Enumeration Type Documentation

# 5.27.4.1 TRDP\_DATA\_TYPE\_T

enum TRDP\_DATA\_TYPE\_T

TRDP dataset description definitions.

Dataset element definition

## Enumerator

| TRDP_INVALID | Invalid/unknown.               |
|--------------|--------------------------------|
| TRDP_BITSET8 | =UINT8                         |
| TRDP_CHAR8   | char, can be used also as UTF8 |
| TRDP_UTF16   | Unicode UTF-16 character.      |
| TRDP_INT8    | Signed integer, 8 bit.         |
| TRDP_INT16   | Signed integer, 16 bit.        |
| TRDP_INT32   | Signed integer, 32 bit.        |
| TRDP_INT64   | Signed integer, 64 bit.        |
| TRDP_UINT8   | Unsigned integer, 8 bit.       |
| TRDP_UINT16  | Unsigned integer, 16 bit.      |

## Enumerator

| TRDP_UINT32     | Unsigned integer, 32 bit.                           |
|-----------------|---|
| TRDP_UINT64     | Unsigned integer, 64 bit.                           |
| TRDP_REAL32     | Floating point real, 32 bit.                        |
| TRDP_REAL64     | Floating point real, 64 bit.                        |
| TRDP_TIMEDATE32 | 32 bit UNIX time                                    |
| TRDP_TIMEDATE48 | 48 bit TCN time (32 bit UNIX time and 16 bit ticks) |
| TRDP_TIMEDATE64 | 32 bit UNIX time + 32 bit microseconds              |
| TRDP_TYPE_MAX   | Values greater are considered nested datasets.      |

### 5.27.4.2 TRDP\_ERR\_T

enum TRDP\_ERR\_T

Return codes for all API functions, -1..-29 taken over from vos.

### Enumerator

| Ellullierator          |   |
|------------------------|---|
| TRDP_NO_ERR            | No error.   |
| TRDP_PARAM_ERR         | Parameter missing or out of range.                |
| TRDP_INIT_ERR          | Call without valid initialization.                |
| TRDP_NOINIT_ERR        | Call with invalid handle.                         |
| TRDP_TIMEOUT_ERR       | Timout.   |
| TRDP_NODATA_ERR        | Non blocking mode: no data received.              |
| TRDP_SOCK_ERR          | Socket error / option not supported.              |
| TRDP_IO_ERR            | Socket IO error, data can't be received/sent.     |
| TRDP_MEM_ERR           | No more memory available.                         |
| TRDP_SEMA_ERR          | Semaphore not available.                          |
| TRDP_QUEUE_ERR         | Queue empty.                                      |
| TRDP_QUEUE_FULL_ERR    | Queue full.                                       |
| TRDP_MUTEX_ERR         | Mutex not available.                              |
| TRDP_THREAD_ERR        | Thread error.                                     |
| TRDP_BLOCK_ERR         | System call would have blocked in blocking mode.  |
| TRDP_INTEGRATION_ERR   | Alignment or endianess for selected target wrong. |
| TRDP_NOCONN_ERR        | No TCP connection.                                |
| TRDP_NOSESSION_ERR     | No such session.                                  |
| TRDP_SESSION_ABORT_ERR | Session aborted.                                  |
| TRDP_NOSUB_ERR         | No subscriber.                                    |
| TRDP_NOPUB_ERR         | No publisher.                                     |
| TRDP_NOLIST_ERR        | No listener.                                      |
| TRDP_CRC_ERR           | Wrong CRC.  |
| TRDP_WIRE_ERR          | Wire.   |
| TRDP_TOPO_ERR          | Invalid topo count.                               |
| TRDP_COMID_ERR         | Unknown Comld.                                    |
| TRDP_STATE_ERR         | Call in wrong state.                              |
| TRDP_APP_TIMEOUT_ERR   | Application Timeout.                              |
| TRDP_APP_REPLYTO_ERR   | Application Reply Sent Timeout.                   |
|                        |   |

### Enumerator

| TRDP_APP_CONFIRMTO_ERR | Application Confirm Sent Timeout.         |
|------------------------|---|
| TRDP_REPLYTO_ERR       | Protocol Reply Timeout.                   |
| TRDP_CONFIRMTO_ERR     | Protocol Confirm Timeout.                 |
| TRDP_REQCONFIRMTO_ERR  | Protocol Confirm Timeout (Request sender) |
| TRDP_PACKET_ERR        | Incomplete message data packet.           |
| TRDP_UNRESOLVED_ERR    | DNR: address could not be resolved.       |
| TRDP_XML_PARSER_ERR    | Returned by the tau_xml subsystem.        |
| TRDP_INUSE_ERR         | Resource is still in use.                 |
| TRDP_MARSHALLING_ERR   | Source size exceeded, dataset mismatch.   |
| TRDP_UNKNOWN_ERR       | Unspecified error.                        |

## 5.27.4.3 TRDP\_RED\_STATE\_T

enum TRDP\_RED\_STATE\_T

Redundancy states.

### Enumerator

| TRDP_RED_FOLLOWER | Redundancy follower - redundant PD will be not sent out. |
|-------------------|--|
| TRDP_RED_LEADER   | Redundancy leader - redundant PD will be sent out.       |

## 5.27.4.4 TRDP\_REPLY\_STATUS\_T

enum TRDP\_REPLY\_STATUS\_T

TRDP data transfer type definitions.

Reply status messages

5.27.4.5 TRDP\_TO\_BEHAVIOR\_T

enum TRDP\_TO\_BEHAVIOR\_T

How invalid PD shall be handled.

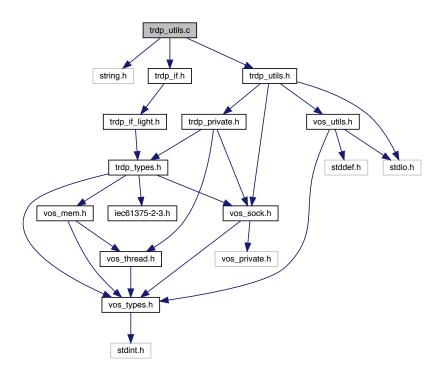
#### Enumerator

| TRDP TO DEFAULT         | Default value defined in tlc openDession will be taken. |
|-------------------------|---|
| TRDP_TO_SET_TO_ZERO     | If set, data will be reset to zero on time out.         |
| TRDP_TO_KEEP_LAST_VALUE | If set, last received values will be returned.          |

## 5.28 trdp\_utils.c File Reference

Helper functions for TRDP communication.

```
#include <string.h>
#include "trdp_if.h"
#include "trdp_utils.h"
Include dependency graph for trdp_utils.c:
```



#### **Functions**

• INT32 trdp\_getCurrentMaxSocketCnt ()

Return the largest number of the socket index.

- TRDP\_IP\_ADDR\_T trdp\_findMCjoins (TRDP\_APP\_SESSION\_T appHandle, TRDP\_IP\_ADDR\_T mcGroup)

  Check an MC group not used by other sockets / subscribers/ listeners.
- UINT32 trdp\_packetSizePD (UINT32 dataSize)

Get the packet size from the raw data size.

• UINT32 trdp\_packetSizeMD (UINT32 dataSize)

Get the packet size from the raw data size.

• PD\_ELE\_T \* trdp\_queueFindComId (PD\_ELE\_T \*pHead, UINT32 comId)

Return the element with same comld.

• PD\_ELE\_T \* trdp\_queueFindPubAddr (PD\_ELE\_T \*pHead, TRDP\_ADDRESSES\_T \*addr)

Return the element with same comld and IP addresses.

• PD\_ELE\_T \* trdp\_queueFindSubAddr (PD\_ELE\_T \*pHead, TRDP\_ADDRESSES\_T \*addr)

Return the element with same comld and IP addresses.

• void trdp\_queueDelElement (PD\_ELE\_T \*\*ppHead, PD\_ELE\_T \*pDelete)

Delete an element.

BOOL8 trdp\_validTopoCounters (UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, UINT32 etbTopoCntFilter, U
 INT32 opTrnTopoCntFilter)

Check topography counters The applied conformance pattern follows Table A.5/A.21 (positive match): Telegram to be sent Locally stored value (appSession) Case etbTopoCnt opTrnTopoCnt etbTopoCntFilter opTrnTopoCntFilter 1 any any 0 0 2 any equal 0 equal 3 equal any equal 0 4 equal equal equal equal.

void trdp\_queueAppLast (PD\_ELE\_T \*\*ppHead, PD\_ELE\_T \*pNew)

Append an element at end of queue.

void trdp\_queueInsFirst (PD\_ELE\_T \*\*ppHead, PD\_ELE\_T \*pNew)

Insert an element at front of queue.

void trdp initSockets (TRDP SOCKETS T iface[])

Handle the socket pool: Initialize it.

TRDP\_ERR\_T trdp\_requestSocket (TRDP\_SOCKETS\_T iface[], UINT16 port, const TRDP\_SEND\_PA
 RAM\_T \*params, TRDP\_IP\_ADDR\_T srcIP, TRDP\_IP\_ADDR\_T mcGroup, TRDP\_SOCK\_TYPE\_T type,
 TRDP\_OPTION\_T options, BOOL8 rcvMostly, SOCKET useSocket, INT32 \*pIndex, TRDP\_IP\_ADDR\_←
 T cornerlp)

Handle the socket pool: Request a socket from our socket pool First we loop through the socket pool and check if there is already a socket which would suit us.

void trdp\_releaseSocket (TRDP\_SOCKETS\_T iface[], INT32 IIndex, UINT32 connectTimeout, BOOL8 checkAll, TRDP\_IP\_ADDR\_T mcGroupUsed)

Handle the socket pool: if a received TCP socket is unused, the socket connection timeout is started.

- UINT32 trdp\_getSeqCnt (UINT32 comId, TRDP\_MSG\_T msgType, TRDP\_IP\_ADDR\_T srclpAddr)
  - Get the initial sequence counter for the comID/message type and subnet (source IP).

remove the sequence counter for the comID/source IP.

 int trdp\_checkSequenceCounter (PD\_ELE\_T \*pElement, UINT32 sequenceCounter, TRDP\_IP\_ADDR\_← T srcIP, TRDP MSG T msgType)

check and update the sequence counter for the comID/source IP.

- BOOL8 trdp\_isAddressed (const TRDP\_URI\_USER\_T listUri, const TRDP\_URI\_USER\_T destUri)
  - Check if listener URI is in addressing range of destination URI.
- BOOL8 trdp\_isInIPrange (TRDP\_IP\_ADDR\_T receivedSrcIP, TRDP\_IP\_ADDR\_T listenedSourceIPlow, T

   —
   RDP\_IP\_ADDR\_T listenedSourceIPhigh)

Check if received IP is in addressing range of listener's IPs.

### 5.28.1 Detailed Description

Helper functions for TRDP communication.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

#### trdp\_utils.c 1779 2018-11-07 09:49:55Z bloehr

```
BL 2018-11-06: for-loops limited to sCurrentMaxSocketCnt instead VOS_MAX_SOCKET_CNT
BL 2018-11-06: Ticket #219: PD Sequence Counter is not synched correctly
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2018-02-03: Ticket #190 Source filtering (IP-range) for PD subscribe
BL 2017-11-28: Ticket #180 Filtering rules for DestinationURI does not follow the standard
BL 2017-11-15: Ticket #1
                          Unjoin on unsubscribe/delListener (finally ;-)
BL 2017-11-15: Ticket #175 PD: Handling of sequence counter
BL 2017-11-09: Ticket \#181/182 Missing padding bytes in user dataset of PD/MD-PDU
BL 2017-11-06: Ticket #178 trdp_releaseSocket does not cleanup tcpParams
BL 2017-11-06: Ticket #174 Socket is closed, even if in use
BL 2017-06-07: Undoing setting of usage (came in with #126 fix!)
BL 2017-05-08: Ticket #126 Opened UDP socket is not released if join or bind failed in trdp_requestSocket()
BL 2017-05-08: Compiler warnings, static definitions
BL 2017-03-01: Ticket #136 PD topography counter with faulty behavior
BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings)
BL 2016-03-01: Setting correct multicast TTL for PDs
BL 2014-08-25: Ticket #57+58: Padding / zero bytes trailing MD & PD packets fixed
BL 2014-06-02: Ticket #41: Sequence counter handling fixed
```

#### 5.28.2 Function Documentation

#### 5.28.2.1 trdp\_checkSequenceCounter()

check and update the sequence counter for the comID/source IP.

If the comID/srcIP is not found, update it and return 0 - else if already received, return 1 On memory error, return -1

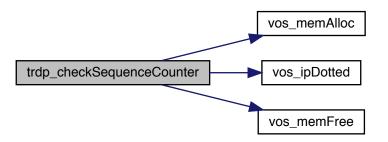
#### **Parameters**

| in | pElement        | subscription element      |
|----|-----------------|---------------------------|
| in | sequenceCounter | sequence counter to check |
| in | srcIP           | Source IP address         |
| in | msgType         | type of the message       |

#### Return values

```
0 - no duplicate 1 - duplicate or old sequence counter -1 - memory error
```

Here is the call graph for this function:



### 5.28.2.2 trdp\_findMCjoins()

Check an MC group not used by other sockets / subscribers/ listeners.

#### **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | mcGroup   | multicast group to look for            |

#### Return values

|  | multi | cast group if unused VOS_INADDR_ANY if used |
|--|-------|---|
|--|-------|---|

## 5.28.2.3 trdp\_getCurrentMaxSocketCnt()

```
INT32 trdp_getCurrentMaxSocketCnt ( void \quad )
```

Return the largest number of the socket index.

#### **Returns**

maxSocketCount

#### 5.28.2.4 trdp\_getSeqCnt()

Get the initial sequence counter for the comID/message type and subnet (source IP).

If the comID/srcIP is not found elsewhere, return 0 - else return its current sequence number (the redundant packet needs the same seqNo)

Note: The standard demands that sequenceCounter is managed per comID/msgType at each publisher, but shall be the same for redundant telegrams (subnet/srcIP).

#### **Parameters**

| in | comId     | comID to look for |
|----|-----------|-------------------|
| in | msgType   | PD/MD type        |
| in | srclpAddr | Source IP address |

#### Return values

| return | the sequence number |
|--------|---------------------|
|--------|---------------------|

### 5.28.2.5 trdp\_initSockets()

Handle the socket pool: Initialize it.

### **Parameters**

| in | iface | pointer to the socket pool |
|----|-------|----------------------------|
|----|-------|----------------------------|

### 5.28.2.6 trdp\_isAddressed()

```
BOOL8 trdp_isAddressed (

const TRDP_URI_USER_T listUri,

const TRDP_URI_USER_T destUri)
```

Check if listener URI is in addressing range of destination URI.

#### **Parameters**

| in | listUri | Null terminated listener URI string to compare    |
|----|---------|---|
| in | destUri | Null terminated destination URI string to compare |

### Return values

| FALSE | - not in addressing range                                |
|-------|--|
| TRUE  | - listener URI is in addressing range of destination URI |

### 5.28.2.7 trdp\_isInIPrange()

```
BOOL8 trdp_isInIPrange (

TRDP_IP_ADDR_T receivedSrcIP,

TRDP_IP_ADDR_T listenedSourceIPlow,

TRDP_IP_ADDR_T listenedSourceIPhigh )
```

Check if received IP is in addressing range of listener's IPs.

#### **Parameters**

| in | receivedSrcIP        | Received IP address |
|----|----------------------|---------------------|
| in | listenedSourceIPlow  | Lower bound IP      |
| in | listenedSourceIPhigh | Upper bound IP      |

#### **Return values**

| FALSE | - not in addressing range                        |  |
|-------|--|--|
| TRUE  | - received IP is in addressing range of listener |  |

### 5.28.2.8 trdp\_packetSizeMD()

Get the packet size from the raw data size.

### **Parameters**

| in | dataSize | net data size (without padding) |
|----|----------|---------------------------------|
|----|----------|---------------------------------|

### Return values

| packet   size the size of the complete packet to be sent or receive |
|---|
|---|

### 5.28.2.9 trdp\_packetSizePD()

```
UINT32 trdp_packetSizePD (
```

```
UINT32 dataSize )
```

Get the packet size from the raw data size.

### **Parameters**

| in dataSize net data size (without pad | ng) |
|--|-----|
|--|-----|

#### **Return values**

### 5.28.2.10 trdp\_queueAppLast()

Append an element at end of queue.

### **Parameters**

| in | ppHead | pointer to pointer to head of queue |
|----|--------|-------------------------------------|
| in | pNew   | pointer to element to append        |

### 5.28.2.11 trdp\_queueDelElement()

Delete an element.

### **Parameters**

| in                | ppHead | pointer to pointer to head of queue |
|-------------------|--------|-------------------------------------|
| in <i>pDelete</i> |        | pointer to element to delete        |

### 5.28.2.12 trdp\_queueFindComId()

Return the element with same comld.

#### **Parameters**

| in | pHead | pointer to head of queue |
|----|-------|--------------------------|
| in | comld | ComID to search for      |

### Return values

|      | NULL pointer to PD element |
|------|----------------------------|
| NULL | No PD element found        |

### 5.28.2.13 trdp\_queueFindPubAddr()

Return the element with same comld and IP addresses.

#### **Parameters**

| in             | pHead | pointer to head of queue                                       |  |
|----------------|-------|--|--|
| in <i>addr</i> |       | Pub/Sub handle (Address, ComID, srcIP & dest IP) to search for |  |

### Return values

| != NULL pointer to PD eleme |  |
|-----------------------------|--|
| NULL No PD element found    |  |

## 5.28.2.14 trdp\_queueFindSubAddr()

Return the element with same comld and IP addresses.

### **Parameters**

| in | pHead | pointer to head of queue                                       |
|----|-------|--|
| in | addr  | Pub/Sub handle (Address, ComID, srcIP & dest IP) to search for |

#### Return values

| != NULL |      |                            |
|---------|------|----------------------------|
|         |      | NULL pointer to PD element |
|         | NULL | No PD element found        |

### 5.28.2.15 trdp\_queueInsFirst()

Insert an element at front of queue.

#### **Parameters**

| in | ppHead | pointer to pointer to head of queue |
|----|--------|-------------------------------------|
| in | pNew   | pointer to element to insert        |

### 5.28.2.16 trdp\_releaseSocket()

Handle the socket pool: if a received TCP socket is unused, the socket connection timeout is started.

Handle the socket pool: Release a socket from our socket pool.

In Udp, Release a socket from our socket pool

### Parameters

| in,out | iface          | socket pool                     |
|--------|----------------|---------------------------------|
| in     | IIndex         | index of socket to release      |
| in     | connectTimeout | time out                        |
| in     | checkAll       | release all TCP pending sockets |
| in     | mcGroupUsed    | release MC group subscription   |

### 5.28.2.17 trdp\_requestSocket()

```
TRDP_ERR_T trdp_requestSocket (
          TRDP_SOCKETS_T iface[],
          UINT16 port,
          const TRDP_SEND_PARAM_T * params,
          TRDP_IP_ADDR_T srcIP,
```

```
TRDP_IP_ADDR_T mcGroup,
TRDP_SOCK_TYPE_T type,
TRDP_OPTION_T options,
BOOL8 rcvMostly,
SOCKET useSocket,
INT32 * pIndex,
TRDP_IP_ADDR_T cornerIp)
```

Handle the socket pool: Request a socket from our socket pool First we loop through the socket pool and check if there is already a socket which would suit us.

If a multicast group should be joined, we do that on an otherwise suitable socket - up to 20 multicast goups can be joined per socket. If a socket for multicast publishing is requested, we also use the source IP to determine the interface for outgoing multicast traffic.

#### **Parameters**

| in,out | iface     | socket pool  |
|--------|-----------|--|
| in     | port      | port to use  |
| in     | params    | parameters to use  |
| in     | srcIP     | IP to bind to (0 = any address)                          |
| in     | mcGroup   | MC group to join (0 = do not join)                       |
| in     | type      | type determines port to bind to (PD, MD/UDP, MD/TCP)     |
| in     | options   | blocking/nonblocking                                     |
| in     | rcvMostly | primarily used for receiving (tbd: bind on sender, too?) |
| out    | useSocket | socket to use, do not open a new one                     |
| out    | pIndex    | returned index of socket pool                            |
| in     | cornerlp  | only used for receiving                                  |

#### **Return values**

| TRDP_NO_ERR    |  |
|----------------|--|
| TRDP_PARAM_ERR |  |

#### 5.28.2.18 trdp\_resetSequenceCounter()

remove the sequence counter for the comID/source IP.

The sequence counter should be reset if there was a packet time out.

### **Parameters**

| in | pElement | subscription element |
|----|----------|----------------------|
| in | srcIP    | Source IP address    |
| in | msgType  | message type         |

### Return values

none

#### 5.28.2.19 trdp\_validTopoCounters()

Check topography counters The applied conformance pattern follows Table A.5/A.21 (positive match): Telegram to be sent Locally stored value (appSession) Case etbTopoCnt opTrnTopoCnt etbTopoCntFilter opTrnTopoCntFilter 1 any any 0 0 2 any equal 0 equal 3 equal any equal 0 4 equal equal equal equal.

#### **Parameters**

| in | etbTopoCnt ETB topography counter to be checked |  |
|----|---|--|
| in | opTrnTopoCnt                                    | Operational topography counter to be checked |
| in | etbTopoCntFilter                                | ETB topography counter filter value          |
| in | opTrnTopoCntFilter                              | Operational topography counter filter value  |

#### Return values

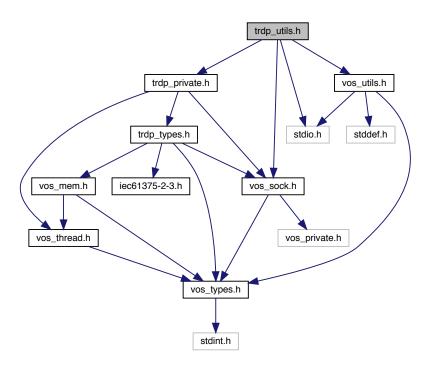
|  | Filter criteria matched FALSE Filter criteria not matched |
|--|---|
|  |   |
|  |   |
|  |   |

# 5.29 trdp\_utils.h File Reference

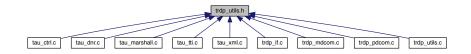
Common utilities for TRDP communication.

```
#include <stdio.h>
#include "trdp_private.h"
#include "vos_utils.h"
#include "vos_sock.h"
```

Include dependency graph for trdp\_utils.h:



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



### **Functions**

- PD\_ELE\_T \* trdp\_queueFindComId (PD\_ELE\_T \*pHead, UINT32 comId)

  Return the element with same comId.
- PD\_ELE\_T \* trdp\_queueFindSubAddr (PD\_ELE\_T \*pHead, TRDP\_ADDRESSES\_T \*pAddr)

  Return the element with same comld and IP addresses.
- PD\_ELE\_T \* trdp\_queueFindPubAddr (PD\_ELE\_T \*pHead, TRDP\_ADDRESSES\_T \*addr)

  Return the element with same comld and IP addresses.
- void trdp\_queueDelElement (PD\_ELE\_T \*\*pHead, PD\_ELE\_T \*pDelete)

  Delete an element.
- void trdp\_queueAppLast (PD\_ELE\_T \*\*pHead, PD\_ELE\_T \*pNew)
  - Append an element at end of queue.
- void trdp\_queueInsFirst (PD\_ELE\_T \*\*pHead, PD\_ELE\_T \*pNew)
   Insert an element at front of queue.
- INT32 trdp\_getCurrentMaxSocketCnt (void)

Return the largest number of the socket index.

void trdp\_initSockets (TRDP\_SOCKETS\_T iface[])

Handle the socket pool: Initialize it.

void trdp\_initUncompletedTCP (TRDP\_APP\_SESSION\_T appHandle)

void trdp\_resetSequenceCounter (PD\_ELE\_T \*pElement, TRDP\_IP\_ADDR\_T srcIP, TRDP\_MSG\_T msg
 — Type)

remove the sequence counter for the comID/source IP.

- TRDP\_IP\_ADDR\_T trdp\_findMCjoins (TRDP\_APP\_SESSION\_T appHandle, TRDP\_IP\_ADDR\_T mcGroup)

  Check an MC group not used by other sockets / subscribers/ listeners.
- TRDP\_ERR\_T trdp\_requestSocket (TRDP\_SOCKETS\_T iface[], UINT16 port, const TRDP\_SEND\_PA

  RAM\_T \*params, TRDP\_IP\_ADDR\_T srcIP, TRDP\_IP\_ADDR\_T mcGroup, TRDP\_SOCK\_TYPE\_T type,
  TRDP\_OPTION\_T options, BOOL8 rcvMostly, SOCKET useSocket, INT32 \*pIndex, TRDP\_IP\_ADDR\_←
  T cornerlp)

Handle the socket pool: Request a socket from our socket pool First we loop through the socket pool and check if there is already a socket which would suit us.

void trdp\_releaseSocket (TRDP\_SOCKETS\_T iface[], INT32 IIndex, UINT32 connectTimeout, BOOL8 checkAll, TRDP\_IP\_ADDR\_T mcGroupUsed)

Handle the socket pool: Release a socket from our socket pool.

UINT32 trdp\_packetSizePD (UINT32 dataSize)

Get the packet size from the raw data size.

• UINT32 trdp\_packetSizeMD (UINT32 dataSize)

Get the packet size from the raw data size.

• UINT32 trdp\_getSeqCnt (UINT32 comID, TRDP\_MSG\_T msgType, TRDP\_IP\_ADDR\_T srcIP)

Get the initial sequence counter for the comID/message type and subnet (source IP).

check and update the sequence counter for the comID/source IP.

- BOOL8 trdp\_isAddressed (const TRDP\_URI\_USER\_T listUri, const TRDP\_URI\_USER\_T destUri)

  Check if listener URI is in addressing range of destination URI.
- BOOL8 trdp\_validTopoCounters (UINT32 etbTopoCnt, UINT32 opTrnTopoCnt, UINT32 etbTopoCntFilter, U
   INT32 opTrnTopoCntFilter)

Check topography counters The applied conformance pattern follows Table A.5/A.21 (positive match): Telegram to be sent Locally stored value (appSession) Case etbTopoCnt opTrnTopoCnt etbTopoCntFilter opTrnTopoCntFilter 1 any any 0 0 2 any equal 0 equal 3 equal any equal 0 4 equal equal equal equal.

BOOL8 trdp\_isInIPrange (TRDP\_IP\_ADDR\_T receivedSrcIP, TRDP\_IP\_ADDR\_T listenedSourceIPlow, T
 —
 RDP IP ADDR T listenedSourceIPhigh)

Check if received IP is in addressing range of listener's IPs.

#### 5.29.1 Detailed Description

Common utilities for TRDP communication.

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

trdp\_utils.h 1779 2018-11-07 09:49:55Z bloehr

```
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2017-11-28: Ticket #180 Filtering rules for DestinationURI does not follow the standard
BL 2017-11-15: Ticket #1 Unjoin on unsubscribe/delListener (finally;-)
BL 2017-05-08: Doxygen comment errors
BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings)
```

#### 5.29.2 Function Documentation

#### 5.29.2.1 trdp\_checkSequenceCounter()

check and update the sequence counter for the comID/source IP.

If the comID/srcIP is not found, update it and return 0 - else if already received, return 1 On memory error, return -1

### **Parameters**

| in | pElement        | subscription element      |
|----|-----------------|---------------------------|
| in | sequenceCounter | sequence counter to check |
| in | srcIP           | Source IP address         |
| in | msgType         | type of the message       |

#### Return values

```
0 - no duplicate 1 - duplicate sequence counter -1 - memory error
```

If the comID/srcIP is not found, update it and return 0 - else if already received, return 1 On memory error, return -1

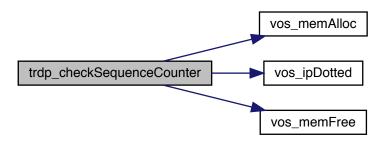
#### **Parameters**

| in | pElement        | subscription element      |
|----|-----------------|---------------------------|
| in | sequenceCounter | sequence counter to check |
| in | srcIP           | Source IP address         |
| in | msgType         | type of the message       |

### Return values

```
0 - no duplicate 1 - duplicate or old sequence counter -1 - memory error
```

Here is the call graph for this function:



### 5.29.2.2 trdp\_findMCjoins()

Check an MC group not used by other sockets / subscribers/ listeners.

### **Parameters**

| in | appHandle | the handle returned by tlc_openSession |
|----|-----------|--|
| in | mcGroup   | multicast group to look for            |

### Return values

```
multi cast group if unused VOS_INADDR_ANY if used
```

### 5.29.2.3 trdp\_getCurrentMaxSocketCnt()

```
\label{eq:int32} INT32 \ trdp\_getCurrentMaxSocketCnt \ ( \\ void \ )
```

Return the largest number of the socket index.

#### Returns

maxSocketCount

### 5.29.2.4 trdp\_getSeqCnt()

Get the initial sequence counter for the comID/message type and subnet (source IP).

If the comID/srcIP is not found elsewhere, return 0 - else return its current sequence number (the redundant packet needs the same seqNo)

Note: The standard demands that sequenceCounter is managed per comID/msgType at each publisher, but shall be the same for redundant telegrams (subnet/srcIP).

#### **Parameters**

| in | comID   | comID to look for |
|----|---------|-------------------|
| in | msgType | PD/MD type        |
| in | srcIP   | Source IP address |

#### Return values

| return | the sequence number |
|--------|---------------------|
|--------|---------------------|

If the comID/srcIP is not found elsewhere, return 0 - else return its current sequence number (the redundant packet needs the same seqNo)

Note: The standard demands that sequenceCounter is managed per comID/msgType at each publisher, but shall be the same for redundant telegrams (subnet/srcIP).

#### **Parameters**

| in | comld     | comID to look for |
|----|-----------|-------------------|
| in | msgType   | PD/MD type        |
| in | srclpAddr | Source IP address |

#### Return values

| return | the sequence number |
|--------|---------------------|

## 5.29.2.5 trdp\_initSockets()

```
void trdp_initSockets (
```

```
TRDP_SOCKETS_T iface[] )
```

Handle the socket pool: Initialize it.

### **Parameters**

| in | iface | pointer to the socket pool |
|----|-------|----------------------------|
|----|-------|----------------------------|

### 5.29.2.6 trdp\_initUncompletedTCP()

???

#### **Parameters**

| in | appHandle | session handle |
|----|-----------|----------------|
|    | app.rana  | 0000.0         |

### 5.29.2.7 trdp\_isAddressed()

```
BOOL8 trdp_isAddressed (

const TRDP_URI_USER_T listUri,

const TRDP_URI_USER_T destUri)
```

Check if listener URI is in addressing range of destination URI.

### **Parameters**

| in | listUri | Null terminated listener URI string to compare    |
|----|---------|---|
| in | destUri | Null terminated destination URI string to compare |

#### Return values

| FALSE | - not in addressing range                                |  |
|-------|--|--|
| TRUE  | - listener URI is in addressing range of destination URI |  |

### 5.29.2.8 trdp\_isInIPrange()

```
BOOL8 trdp_isInIPrange (

TRDP_IP_ADDR_T receivedSrcIP,

TRDP_IP_ADDR_T listenedSourceIPlow,

TRDP_IP_ADDR_T listenedSourceIPhigh )
```

Check if received IP is in addressing range of listener's IPs.

### **Parameters**

| in | receivedSrcIP        | Received IP address |
|----|----------------------|---------------------|
| in | listenedSourceIPlow  | Lower bound IP      |
| in | listenedSourceIPhigh | Upper bound IP      |

### Return values

| FALSE | - not in addressing range                        |  |
|-------|--|--|
| TRUE  | - received IP is in addressing range of listener |  |

### 5.29.2.9 trdp\_packetSizeMD()

Get the packet size from the raw data size.

### **Parameters**

| in | dataSize | net data size |
|----|----------|---------------|
|----|----------|---------------|

### Return values

| packet | size the size of the complete packet to be sent or received |
|--------|---|
|--------|---|

### **Parameters**

| in | dataSize | net data size (without padding) |
|----|----------|---------------------------------|
|----|----------|---------------------------------|

### Return values

| packet | size the size of the complete packet to be sent or received |
|--------|---|
|--------|---|

## 5.29.2.10 trdp\_packetSizePD()

Get the packet size from the raw data size.

### **Parameters**

| in dataSize ne | t data size |
|----------------|-------------|
|----------------|-------------|

### Return values

| packet | size the size of the complete packet to be sent or received |
|--------|---|
|--------|---|

#### **Parameters**

| i | n | dataSize | net data size (without padding) |
|---|---|----------|---------------------------------|
|---|---|----------|---------------------------------|

### Return values

```
packet size the size of the complete packet to be sent or received
```

#### 5.29.2.11 trdp\_queueAppLast()

Append an element at end of queue.

#### **Parameters**

| in | ppHead | pointer to pointer to head of queue |
|----|--------|-------------------------------------|
| in | pNew   | pointer to element to append        |

### 5.29.2.12 trdp\_queueDelElement()

### Delete an element.

### Parameters

| in | ppHead  | pointer to pointer to head of queue |
|----|---------|-------------------------------------|
| in | pDelete | pointer to element to delete        |

### 5.29.2.13 trdp\_queueFindComId()

Return the element with same comld.

#### **Parameters**

| in | pHead | pointer to head of queue |
|----|-------|--------------------------|
| in | comld | ComID to search for      |

### Return values

| !=   | NULL pointer to PD element |
|------|----------------------------|
| NULL | No PD element found        |

## 5.29.2.14 trdp\_queueFindPubAddr()

```
PD_ELE_T* trdp_queueFindPubAddr ( \label{eq:pd_ele} \texttt{PD_ELE_T} * \textit{pHead}, \\ \texttt{TRDP\_ADDRESSES\_T} * \textit{addr} )
```

Return the element with same comld and IP addresses.

### **Parameters**

| in | pHead | pointer to head of queue                                       |
|----|-------|--|
| in | addr  | Pub/Sub handle (Address, ComID, srcIP & dest IP) to search for |

### **Return values**

| !=   | NULL pointer to PD element |
|------|----------------------------|
| NULL | No PD element found        |

### 5.29.2.15 trdp\_queueFindSubAddr()

Return the element with same comld and IP addresses.

#### **Parameters**

| l | in | pHead | pointer to head of queue                                       |  |
|---|----|-------|--|--|
|   | in | addr  | Pub/Sub handle (Address, ComID, srcIP & dest IP) to search for |  |

#### Return values

| != NULL pointer to PD element |
|-------------------------------|
|-------------------------------|

### Return values

|  | NULL | No PD element found |
|--|------|---------------------|
|--|------|---------------------|

### 5.29.2.16 trdp\_queueInsFirst()

Insert an element at front of queue.

#### **Parameters**

| in | ppHead | pointer to pointer to head of queue |
|----|--------|-------------------------------------|
| in | pNew   | pointer to element to insert        |

## 5.29.2.17 trdp\_releaseSocket()

Handle the socket pool: Release a socket from our socket pool.

### Parameters

| in,out | iface          | socket pool                     |
|--------|----------------|---------------------------------|
| in     | IIndex         | index of socket to release      |
| in     | connectTimeout | timeout value                   |
| in     | checkAll       | release all TCP pending sockets |
| in     | mcGroupUsed    | release MC group subscription   |

Handle the socket pool: Release a socket from our socket pool.

In Udp, Release a socket from our socket pool

## Parameters

| in,out | iface          | socket pool                     |  |  |
|--------|----------------|---------------------------------|--|--|
| in     | IIndex         | index of socket to release      |  |  |
| in     | connectTimeout | time out                        |  |  |
| in     | checkAll       | release all TCP pending sockets |  |  |
| in     | mcGroupUsed    | release MC group subscription   |  |  |

#### 5.29.2.18 trdp\_requestSocket()

Handle the socket pool: Request a socket from our socket pool First we loop through the socket pool and check if there is already a socket which would suit us.

If a multicast group should be joined, we do that on an otherwise suitable socket - up to 20 multicast goups can be joined per socket. If a socket for multicast publishing is requested, we also use the source IP to determine the interface for outgoing multicast traffic.

#### **Parameters**

| in,out               | iface socket pool |  |  |
|----------------------|-------------------|--|--|
| in <i>port</i>       |                   | port to use  |  |
| in                   | params            | parameters to use                                    |  |
| in                   | srcIP             | IP to bind to (0 = any address)                      |  |
| in                   | mcGroup           | MC group to join (0 = do not join)                   |  |
| in <i>type</i>       |                   | type determines port to bind to (PD, MD/UDP, MD/TCP) |  |
| in <i>options</i>    |                   | blocking/nonblocking                                 |  |
| in                   | rcvMostly         | only used for receiving                              |  |
| out <i>useSocket</i> |                   | socket to use, do not open a new one                 |  |
| out <i>plndex</i>    |                   | returned index of socket pool                        |  |
| in cornerlp          |                   | only used for receiving                              |  |

#### Return values

| TRDP_NO_ERR    |  |
|----------------|--|
| TRDP_PARAM_ERR | If a multicast group should be joined, we do that on an otherwise suitable socket - up to 20 multicast goups can be joined per socket. If a socket for multicast publishing is requested, we also use the source IP to determine the interface for outgoing multicast traffic. |

#### **Parameters**

| in,out  | iface                                 | socket pool                        |
|---|---------------------------------------|------------------------------------|
| in  | port port to use                      |                                    |
| in  | params parameters to use              |                                    |
| in  | srcIP IP to bind to (0 = any address) |                                    |
| in mcGroup MC group to join (0 = do not join) |                                       | MC group to join (0 = do not join) |

#### **Parameters**

| in  | type  | type determines port to bind to (PD, MD/UDP, MD/TCP) |  |
|---|---|--|--|
| in  | in options blocking/nonblocking                           |  |  |
| in  | n rcvMostly primarily used for receiving (tbd: bind on se |  |  |
| out useSocket socket to use, do not open a new or |   | socket to use, do not open a new one                 |  |
| out plndex returned index of socket pool          |   | returned index of socket pool                        |  |
| in cornerlp only used for receiving               |   | only used for receiving                              |  |

#### **Return values**

| TRDP_NO_ERR    |  |
|----------------|--|
| TRDP_PARAM_ERR |  |

## 5.29.2.19 trdp\_resetSequenceCounter()

remove the sequence counter for the comID/source IP.

The sequence counter should be reset if there was a packet time out.

### **Parameters**

| in | pElement | subscription element |
|----|----------|----------------------|
| in | srcIP    | Source IP address    |
| in | msgType  | message type         |

#### **Return values**

```
none
```

### 5.29.2.20 trdp\_validTopoCounters()

Check topography counters The applied conformance pattern follows Table A.5/A.21 (positive match): Telegram to be sent Locally stored value (appSession) Case etbTopoCnt opTrnTopoCnt etbTopoCntFilter opTrnTopoCntFilter 1 any any 0 0 2 any equal 0 equal 3 equal any equal 0 4 equal equal equal equal.

### **Parameters**

| in | etbTopoCnt         | ETB topography counter to be checked         |
|----|--------------------|--|
| in | opTrnTopoCnt       | Operational topography counter to be checked |
| in | etbTopoCntFilter   | ETB topography counter filter value          |
| in | opTrnTopoCntFilter | Operational topography counter filter value  |

### Return values

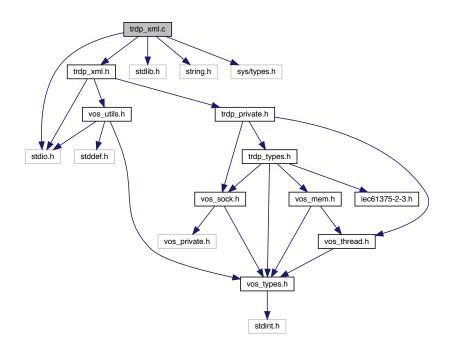
| TRUE | Filter criteria matched FALSE Filter criteria not matched |
|------|---|
|------|---|

# 5.30 trdp\_xml.c File Reference

## Simple XML parser.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include "trdp_xml.h"
```

Include dependency graph for trdp\_xml.c:



## **Functions**

• TRDP\_ERR\_T trdp\_XMLOpen (XML\_HANDLE\_T \*pXML, const char \*file)

Opens the XML parsing.

void trdp\_XMLRewind (XML\_HANDLE\_T \*pXML)

Rewind to start.

• void trdp\_XMLClose (XML\_HANDLE\_T \*pXML)

Closes the XML parsng.

int trdp\_XMLSeekStartTagAny (XML\_HANDLE\_T \*pXML, char \*tag, int maxlen)

Seek next tag on starting depth and return it in provided buffer.

int trdp\_XMLSeekStartTag (XML\_HANDLE\_T \*pXML, const char \*tag)

Seek a specific tag.

int trdp\_XMLCountStartTag (XML\_HANDLE\_T \*pXML, const char \*tag)

Count a specific tag.

void trdp\_XMLEnter (XML\_HANDLE\_T \*pXML)

Enter level in XML file.

void trdp XMLLeave (XML HANDLE T \*pXML)

Leave level in XML file.

XML\_TOKEN\_T trdp\_XMLGetAttribute (XML\_HANDLE\_T \*pXML, CHAR8 \*attribute, UINT32 \*pValueInt, CHAR8 \*value)

Get value of next attribute, as string and if possible as integer.

### 5.30.1 Detailed Description

Simple XML parser.

Hint: Missing optional elements must be handled using the count-function, otherwise following elements will be following ignored!

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH; based on code by Peter Brander, Bombardier

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

ld

trdp\_xml.c 1778 2018-11-07 08:31:57Z s-bender

```
SB 2018-11-07: Ticket #221 readXmlDatasets failed BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings) BL 2016-02-24: missing include (thanks to Robert) BL 2016-02-11: Ticket #102: Replacing libxml2
```

#### 5.30.2 Function Documentation

```
5.30.2.1 trdp_XMLClose()
```

Closes the XML parsng.

### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
|----|------|-----------------------|

#### Return values

| none |  |
|------|--|
|------|--|

### 5.30.2.2 trdp\_XMLCountStartTag()

### Count a specific tag.

#### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
| in | tag  | Tag to count          |

### Return values

```
0 if found !=0 if not found
```

## 5.30.2.3 trdp\_XMLEnter()

Enter level in XML file.

### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|

### Return values

none

## 5.30.2.4 trdp\_XMLGetAttribute()

 $XML\_TOKEN\_T$  trdp $\_XMLGetAttribute$  (

```
XML_HANDLE_T * pXML,
CHAR8 * attribute,
UINT32 * pValueInt,
CHAR8 * value )
```

Get value of next attribute, as string and if possible as integer.

#### **Parameters**

| in  | pXML      | Pointer to local data              |
|-----|-----------|------------------------------------|
| in  | attribute | Pointer to attribute               |
| out | pValueInt | Pointer to resulting integer value |
| out | value     | Pointer to resulting string value  |

#### **Return values**

| ı | TOK ATTRIBLITE | if found token if not found |
|---|----------------|-----------------------------|
|   | ION ALIMBULE   | ii louna token ii not louna |

### 5.30.2.5 trdp\_XMLLeave()

```
void trdp_XMLLeave ( \label{eq:mlleave} {\tt XML\_HANDLE\_T\ *\ pXML\ )}
```

Leave level in XML file.

#### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|

## Return values

none

### 5.30.2.6 trdp\_XMLOpen()

Opens the XML parsing.

#### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
| in | file | Pathname of XML file  |

### Return values

none

### 5.30.2.7 trdp\_XMLRewind()

### Rewind to start.

#### **Parameters**

| in $pXML$ Pointer to local of |
|-------------------------------|
|-------------------------------|

### Return values

none

## 5.30.2.8 trdp\_XMLSeekStartTag()

### Seek a specific tag.

### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
| in | tag  | Tag to be found       |

### Return values

```
0 | if found !=0 if not found
```

## 5.30.2.9 trdp\_XMLSeekStartTagAny()

Seek next tag on starting depth and return it in provided buffer.

Start tags on deeper depths are ignored.

### **Parameters**

| in     | pXML   | Pointer to local data |
|--------|--------|-----------------------|
| in,out | tag    | Buffer for found tag  |
| in     | maxlen | Length of buffer      |

### Return values

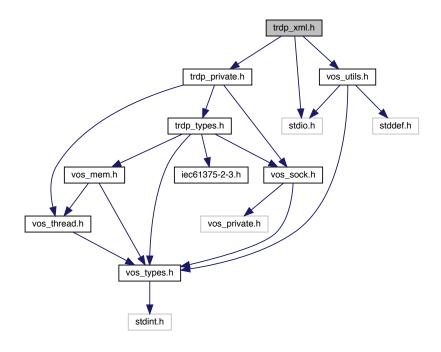
| if found !=0 if not fo | ound |
|------------------------|------|
|------------------------|------|

# 5.31 trdp\_xml.h File Reference

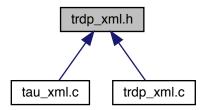
Simple XML parser.

```
#include <stdio.h>
#include "trdp_private.h"
#include "vos_utils.h"
```

Include dependency graph for trdp\_xml.h:



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



#### **Functions**

• TRDP\_ERR\_T trdp\_XMLOpen (XML\_HANDLE\_T \*pXML, const char \*file)

Opens the XML parsing.

void trdp\_XMLClose (XML\_HANDLE\_T \*pXML)

Closes the XML parsng.

int trdp\_XMLCountStartTag (XML\_HANDLE\_T \*pXML, const char \*tag)

Count a specific tag.

• int trdp\_XMLSeekStartTagAny (XML\_HANDLE\_T \*pXML, char \*tag, int maxlen)

Seek next tag on starting depth and return it in provided buffer.

int trdp XMLSeekStartTag (XML HANDLE T \*pXML, const char \*tag)

Seek a specific tag.

XML\_TOKEN\_T trdp\_XMLGetAttribute (XML\_HANDLE\_T \*pXML, CHAR8 \*attribute, UINT32 \*pValueInt, CHAR8 \*value)

Get value of next attribute, as string and if possible as integer.

void trdp\_XMLRewind (XML\_HANDLE\_T \*pXML)

Rewind to start.

void trdp\_XMLEnter (XML\_HANDLE\_T \*pXML)

Enter level in XML file.

void trdp\_XMLLeave (XML\_HANDLE\_T \*pXML)

Leave level in XML file.

#### 5.31.1 Detailed Description

Simple XML parser.

Note

Project: TCNOpen TRDP prototype stack

Author

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright NewTec GmbH or its subsidiaries and others, 2016. All rights reserved.

ld

trdp\_xml.h 1581 2017-02-08 17:39:38Z bloehr

```
BL 2016-02-11: Ticket #102: Replacing libxml2
```

### 5.31.2 Function Documentation

### 5.31.2.1 trdp\_XMLClose()

### Closes the XML parsng.

#### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
|----|------|-----------------------|

#### Return values

```
none
```

### 5.31.2.2 trdp\_XMLCountStartTag()

#### Count a specific tag.

#### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
| in | tag  | Tag to count          |

#### Return values

```
0 if found !=0 if not found
```

### 5.31.2.3 trdp\_XMLEnter()

Enter level in XML file.

#### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
|----|------|-----------------------|

### Return values

```
none
```

### 5.31.2.4 trdp\_XMLGetAttribute()

Get value of next attribute, as string and if possible as integer.

#### **Parameters**

| in  | pXML      | Pointer to local data              |
|-----|-----------|------------------------------------|
| in  | attribute | Pointer to attribute               |
| out | pValueInt | Pointer to resulting integer value |
| out | value     | Pointer to resulting string value  |

### Return values

| TOK_ATTRIBUTE | if found token if not found |
|---------------|-----------------------------|
|---------------|-----------------------------|

### 5.31.2.5 trdp\_XMLLeave()

```
void trdp_XMLLeave ( \label{eq:mlleave} {\tt XML\_HANDLE\_T\ *\ pXML\ )}
```

Leave level in XML file.

### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
|----|------|-----------------------|

#### Return values

```
none
```

## 5.31.2.6 trdp\_XMLOpen()

## Opens the XML parsing.

#### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
| in | file | Pathname of XML file  |

### **Return values**

```
none
```

## 5.31.2.7 trdp\_XMLRewind()

## Rewind to start.

### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
|----|------|-----------------------|

### Return values

none

## 5.31.2.8 trdp\_XMLSeekStartTag()

```
int trdp\_XMLSeekStartTag (
```

```
XML_HANDLE_T * pXML,
const char * tag )
```

## Seek a specific tag.

### **Parameters**

| in | pXML | Pointer to local data |
|----|------|-----------------------|
| in | tag  | Tag to be found       |

#### Return values

```
0 if found !=0 if not found
```

## 5.31.2.9 trdp\_XMLSeekStartTagAny()

Seek next tag on starting depth and return it in provided buffer.

Start tags on deeper depths are ignored.

### **Parameters**

| in     | pXML   | Pointer to local data |
|--------|--------|-----------------------|
| in,out | tag    | Buffer for found tag  |
| in     | maxlen | Length of buffer      |

#### Return values

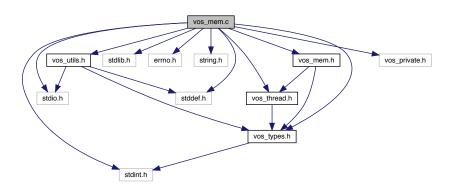
```
0 if found !=0 if not found
```

# 5.32 vos\_mem.c File Reference

### Memory functions.

```
#include <stdio.h>
#include <stddef.h>
#include <stdint.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include "vos_types.h"
#include "vos_utils.h"
```

```
#include "vos_mem.h"
#include "vos_thread.h"
#include "vos_private.h"
Include dependency graph for vos_mem.c:
```



#### **Functions**

• EXT\_DECL VOS\_ERR\_T vos\_memInit (UINT8 \*pMemoryArea, UINT32 size, const UINT32 fragMem[VO↔ S\_MEM\_NBLOCKSIZES])

Initialize the memory unit.

• EXT DECL void vos memDelete (UINT8 \*pMemoryArea)

Delete the memory area.

EXT\_DECL UINT8 \* vos\_memAlloc (UINT32 size)

Allocate a block of memory (from memory area above).

• EXT DECL void vos memFree (void \*pMemBlock)

Deallocate a block of memory (from memory area above).

• EXT\_DECL VOS\_ERR\_T vos\_memCount (UINT32 \*pAllocatedMemory, UINT32 \*pFreeMemory, UINT32 \*pMinFree, UINT32 \*pNumAllocBlocks, UINT32 \*pNumAllocErr, UINT32 \*pNumFreeErr, UINT32 block← Size[VOS\_MEM\_NBLOCKSIZES], UINT32 usedBlockSize[VOS\_MEM\_NBLOCKSIZES])

Return used and available memory (of memory area above).

EXT\_DECL void vos\_qsort (void \*pBuf, UINT32 num, UINT32 size, int(\*compare)(const void \*, const void \*))

Sort an array.

• EXT\_DECL void \* vos\_bsearch (const void \*pKey, const void \*pBuf, UINT32 num, UINT32 size, int(\*compare)(const void \*, const void \*))

Binary search in a sorted array.

EXT\_DECL INT32 vos\_strnicmp (const CHAR8 \*pStr1, const CHAR8 \*pStr2, UINT32 count)

Case insensitive string compare.

• EXT\_DECL void vos\_strncpy (CHAR8 \*pStrDst, const CHAR8 \*pStrSrc, UINT32 count)

String copy with length limitation.

String concatenation with length limitation.

 $\bullet \ \ \mathsf{EXT\_DECL} \ void \ vos\_strncat \ (\mathsf{CHAR8} \ *pStrDst, \ \mathsf{UINT32} \ count, \ const \ \mathsf{CHAR8} \ *pStrSrc)$ 

• EXT\_DECL VOS\_ERR\_T vos\_queueCreate (VOS\_QUEUE\_POLICY\_T queueType, UINT32 maxNoOfMsg, VOS\_QUEUE\_T \*pQueueHandle)

Initialize a message queue.

• EXT\_DECL VOS\_ERR\_T vos\_queueSend (VOS\_QUEUE\_T queueHandle, UINT8 \*pData, UINT32 size) Send a message.

• EXT\_DECL VOS\_ERR\_T vos\_queueReceive (VOS\_QUEUE\_T queueHandle, UINT8 \*\*ppData, UINT32 \*pSize, UINT32 usTimeout)

Get a message.

• EXT\_DECL VOS\_ERR\_T vos\_queueDestroy (VOS\_QUEUE\_T queueHandle)

Destroy a message queue.

## 5.32.1 Detailed Description

Memory functions.

OS abstraction of memory access and control

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

```
vos_mem.c 1768 2018-10-25 09:59:45Z ahweiss
```

Changes: BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32) BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings) BL 2016-02-10: Debug print: tabs before size output BL 2012-12-03: ID 1: "using uninitialized PD\_ELE\_T.pulllpAddress variable" ID 2: "uninitialized PD\_ELE\_T newPD->pNext in tlp\_subscribe()"

#### 5.32.2 Function Documentation

### 5.32.2.1 vos\_bsearch()

Binary search in a sorted array.

This is just a wrapper for the standard bsearch function.

#### **Parameters**

| in | pKey    | Key to search for   |
|----|---------|---|
| in | pBuf    | Pointer to the array to search  |
| in | num     | number of elements  |
| in | size    | size of one element   |
| in | compare | Pointer to compare function return -n if arg1 < arg2, return 0 if arg1 == arg2, return +n if arg1 |
|    |         | > arg2 where n is an integer != 0   |

### Return values

### 5.32.2.2 vos\_memAlloc()

Allocate a block of memory (from memory area above).

#### **Parameters**

|  | in | size | Size of requested block |  |
|--|----|------|-------------------------|--|
|--|----|------|-------------------------|--|

### Return values

| Pointer | to memory area         |
|---------|------------------------|
| NULL    | if no memory available |

## 5.32.2.3 vos\_memCount()

```
EXT_DECL VOS_ERR_T vos_memCount (

UINT32 * pAllocatedMemory,

UINT32 * pFreeMemory,

UINT32 * pMinFree,

UINT32 * pNumAllocBlocks,

UINT32 * pNumAllocErr,

UINT32 * pNumFreeErr,

UINT32 blockSize[VOS_MEM_NBLOCKSIZES],

UINT32 usedBlockSize[VOS_MEM_NBLOCKSIZES])
```

Return used and available memory (of memory area above).

### **Parameters**

| T didinotoro |                  |                                  |  |  |
|--------------|------------------|----------------------------------|--|--|
| out          | pAllocatedMemory | Pointer to allocated memory size |  |  |

#### **Parameters**

| out | pFreeMemory     | Pointer to free memory size                                |
|-----|-----------------|--|
| out | pMinFree        | Pointer to minimal free memory size in statistics interval |
| out | pNumAllocBlocks | Pointer to number of allocated memory blocks               |
| out | pNumAllocErr    | Pointer to number of allocation errors                     |
| out | pNumFreeErr     | Pointer to number of free errors                           |
| out | blockSize       | Pointer to list of memory block sizes                      |
| out | usedBlockSize   | Pointer to list of used memoryblocks                       |

#### Return values

| VOS_NO_ERR   | no error               |
|--------------|------------------------|
| VOS_INIT_ERR | module not initialised |

#### 5.32.2.4 vos\_memDelete()

```
EXT_DECL void vos_memDelete ( {\tt UINT8*pMemoryArea} \ )
```

Delete the memory area.

This will eventually invalidate any previously allocated memory blocks! It should be called last before the application quits. No further access to the memory blocks is allowed after this call.

### **Parameters**

|    |             | Daintanta mananana ana a    |
|----|-------------|-----------------------------|
| ın | рметогулгеа | Pointer to memory area used |

## 5.32.2.5 vos\_memFree()

Deallocate a block of memory (from memory area above).

## **Parameters**

|  | in <i>pMemBlock</i> | Pointer to memory block to be freed |  |
|--|---------------------|-------------------------------------|--|
|--|---------------------|-------------------------------------|--|

## 5.32.2.6 vos\_memInit()

```
EXT_DECL VOS_ERR_T vos_memInit (
```

```
UINT8 * pMemoryArea,
UINT32 size,
const UINT32 fragMem[VOS_MEM_NBLOCKSIZES] )
```

Initialize the memory unit.

Init a supplied block of memory and prepare it for use with vos\_memAlloc and vos\_memFree. The used block sizes can be supplied and will be preallocated. If half of the overall size of the requested memory area would be pre-allocated, either by the default pre-allocation table or a provided one, no pre-allocation takes place.

#### **Parameters**

|   | in | pMemoryArea Pointer to memory area to use |   |
|---|----|---|---|
| Ī | in | size                                      | Size of provided memory area  |
|   | in | fragMem                                   | Pointer to list of preallocated block sizes, used to fragment memory for large blocks |

#### Return values

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |
| VOS_MEM_ERR   | no memory available            |
| VOS_MUTEX_ERR | no mutex available             |

### 5.32.2.7 vos\_qsort()

#### Sort an array.

This is just a wrapper for the standard qsort function.

### **Parameters**

| in,out | pBuf    | Pointer to the array to sort  |  |
|--------|---------|---|--|
| in     | num     | number of elements  |  |
| in     | size    | size of one element   |  |
| in     | compare | Pointer to compare function return -n if $arg1 < arg2$ , return 0 if $arg1 == arg2$ , return +n if $arg1 > arg2$ where n is an integer != 0 |  |

```
none
```

# 5.32.2.8 vos\_queueCreate()

Initialize a message queue.

Returns a handle for further calls

#### **Parameters**

| in  | queueType    | Define queue type (1 = FIFO, 2 = LIFO, 3 = PRIO) |
|-----|--------------|--|
| in  | maxNoOfMsg   | Maximum number of messages                       |
| out | pQueueHandle | Handle of created queue                          |

#### Return values

| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |
| VOS_INIT_ERR   | not supported                  |
| VOS_QUEUE_ERR  | error creating queue           |

## 5.32.2.9 vos\_queueDestroy()

Destroy a message queue.

Free all resources used by this queue

## **Parameters**

| in | queueHandle | Queue handle |
|----|-------------|--------------|
|----|-------------|--------------|

| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |

## 5.32.2.10 vos\_queueReceive()

### Get a message.

#### **Parameters**

| in  | queueHandle | Queue handle                                 |
|-----|-------------|--|
| out | ppData      | Pointer to data pointer to be received       |
| out | pSize       | Size of receive data                         |
| in  | usTimeout   | Maximum time to wait for a message (in usec) |

#### Return values

| VOSNO_ERR      | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |
| VOS_QUEUE_ERR  | queue is empty                 |

## 5.32.2.11 vos\_queueSend()

## Send a message.

### **Parameters**

| in | queueHandle | Queue handle               |
|----|-------------|----------------------------|
| in | pData       | Pointer to data to be sent |
| in | size        | Size of data to be sent    |

| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |
| VOS_INIT_ERR   | not supported                  |
| VOS_QUEUE_ERR  | error creating queue           |

## 5.32.2.12 vos\_strncat()

String concatenation with length limitation.

#### **Parameters**

| in | pStrDst | Destination string               |
|----|---------|----------------------------------|
| in | count   | Size of destination buffer       |
| in | pStrSrc | Null terminated string to append |

#### Return values

```
none
```

## 5.32.2.13 vos\_strncpy()

String copy with length limitation.

#### **Parameters**

| in | pStrDst | Destination string                   |
|----|---------|--------------------------------------|
| in | pStrSrc | Null terminated string to copy       |
| in | count   | Maximum number of characters to copy |

## **Return values**

none

## 5.32.2.14 vos\_strnicmp()

```
EXT_DECL INT32 vos_strnicmp (
const CHAR8 * pStr1,
```

```
const CHAR8 * pStr2,
UINT32 count )
```

Case insensitive string compare.

### **Parameters**

| in | pStr1 | Null terminated string to compare       |  |
|----|-------|---|--|
| in | pStr2 | Null terminated string to compare       |  |
| in | count | Maximum number of characters to compare |  |

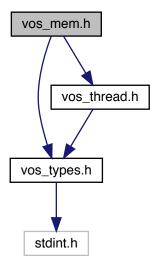
#### Return values

| 0                                   | - equal                      |
|-------------------------------------|------------------------------|
| <0                                  | - string1 less than string 2 |
| >0 - string 1 greater than string 2 |                              |

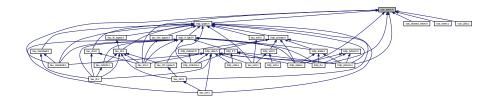
# 5.33 vos\_mem.h File Reference

Memory and queue functions for OS abstraction.

```
#include "vos_types.h"
#include "vos_thread.h"
Include dependency graph for vos_mem.h:
```



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



#### **Macros**

#define VOS MEM MAX PREALLOCATE 10u

Max blocks to pre-allocate.

#define VOS MEM NBLOCKSIZES 15u

No of pre-defined block sizes.

• #define VOS\_MEM\_BLOCKSIZES

We internally allocate memory always by these block sizes.

Default pre-allocation of free memory blocks.

### **Typedefs**

typedef struct VOS\_QUEUE \* VOS\_QUEUE\_T
 Opaque queue define.

## **Enumerations**

enum VOS\_QUEUE\_POLICY\_T
 Queue policy matching pthread/Posix defines.

## **Functions**

Initialize the memory unit.

EXT\_DECL void vos\_memDelete (UINT8 \*pMemoryArea)

Delete the memory area.

• EXT DECL UINT8 \* vos memAlloc (UINT32 size)

Allocate a block of memory (from memory area above).

EXT\_DECL void vos\_memFree (void \*pMemBlock)

Deallocate a block of memory (from memory area above).

• EXT\_DECL VOS\_ERR\_T vos\_memCount (UINT32 \*pAllocatedMemory, UINT32 \*pFreeMemory, UINT32 \*pMinFree, UINT32 \*pNumAllocBlocks, UINT32 \*pNumAllocErr, UINT32 \*pNumFreeErr, UINT32 block← Size[VOS\_MEM\_NBLOCKSIZES], UINT32 usedBlockSize[VOS\_MEM\_NBLOCKSIZES])

Return used and available memory (of memory area above).

EXT\_DECL void vos\_qsort (void \*pBuf, UINT32 num, UINT32 size, int(\*compare)(const void \*, const void \*))

Sort an array.

EXT\_DECL void \* vos\_bsearch (const void \*pKey, const void \*pBuf, UINT32 num, UINT32 size, int(\*compare)(const void \*, const void \*))

Binary search in a sorted array.

EXT\_DECL INT32 vos\_strnicmp (const CHAR8 \*pStr1, const CHAR8 \*pStr2, UINT32 count)
 Case insensitive string compare.

• EXT\_DECL void vos\_strncpy (CHAR8 \*pStrDst, const CHAR8 \*pStrSrc, UINT32 count) String copy with length limitation.

• EXT\_DECL void vos\_strncat (CHAR8 \*pStrDst, UINT32 count, const CHAR8 \*pStrSrc) String concatenation with length limitation.

 EXT\_DECL VOS\_ERR\_T vos\_queueCreate (VOS\_QUEUE\_POLICY\_T queueType, UINT32 maxNoOfMsg, VOS\_QUEUE\_T \*pQueueHandle)

Initialize a message queue.

- EXT\_DECL VOS\_ERR\_T vos\_queueSend (VOS\_QUEUE\_T queueHandle, UINT8 \*pData, UINT32 size)
   Send a message.
- EXT\_DECL VOS\_ERR\_T vos\_queueReceive (VOS\_QUEUE\_T queueHandle, UINT8 \*\*ppData, UINT32 \*pSize, UINT32 usTimeout)

Get a message.

EXT\_DECL VOS\_ERR\_T vos\_queueDestroy (VOS\_QUEUE\_T queueHandle)
 Destroy a message queue.

### 5.33.1 Detailed Description

Memory and queue functions for OS abstraction.

This module provides memory control supervison

Note

Project: TCNOpen TRDP prototype stack

Author

Bernd Loehr, NewTec GmbH Peter Brander (Memory scheme)

### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

vos\_mem.h 1631 2017-05-31 12:03:26Z bloehr

```
BL 2017-05-08: Compiler warnings, doxygen comment errors
```

## 5.33.2 Macro Definition Documentation

#### 5.33.2.1 VOS\_MEM\_BLOCKSIZES

```
#define VOS_MEM_BLOCKSIZES
```

#### Value:

```
 \{ 34\text{u},\ 48\text{u},\ 128\text{u},\ 180\text{u},\ 256\text{u},\ 512\text{u},\ 1024\text{u},\ 1480\text{u},\ 2048\text{u},\ \backslash \\  \qquad \qquad 4096\text{u},\ 11520\text{u},\ 16384\text{u},\ 32768\text{u},\ 65536\text{u},\ 131072\text{u} \}
```

We internally allocate memory always by these block sizes.

The largest available block is 524288 Bytes, provided the overal size of the used memory allocation area is larger.

### 5.33.2.2 VOS\_MEM\_PREALLOCATE

```
#define VOS_MEM_PREALLOCATE {0u, 0u, 0u, 0u, 0u, 0u, 0u, 4u, 0u, 0u, 0u, 0u, 0u, 0u, 0u, 0u}
```

Default pre-allocation of free memory blocks.

To avoid problems with too many small blocks and no large one. Specify how many of each block size that should be pre-allocated (and freed!) to pre-segment the memory area.

### 5.33.3 Function Documentation

## 5.33.3.1 vos\_bsearch()

Binary search in a sorted array.

This is just a wrapper for the standard bsearch function.

### **Parameters**

| in | pKey    | Key to search for   |
|----|---------|---|
| in | pBuf    | Pointer to the array to search  |
| in | num     | number of elements  |
| in | size    | size of one element   |
| in | compare | Pointer to compare function return -n if arg1 < arg2, return 0 if arg1 == arg2, return +n if arg1 > arg2 where n is an integer != 0 |

### Return values

| Pointer | to found element or NULL |
|---------|--------------------------|
|---------|--------------------------|

## 5.33.3.2 vos\_memAlloc()

Allocate a block of memory (from memory area above).

#### **Parameters**

| in | size | Size of requested block |
|----|------|-------------------------|
|----|------|-------------------------|

#### Return values

| Pointer | to memory area         |  |
|---------|------------------------|--|
| NULL    | if no memory available |  |

## 5.33.3.3 vos\_memCount()

```
EXT_DECL VOS_ERR_T vos_memCount (

UINT32 * pAllocatedMemory,

UINT32 * pFreeMemory,

UINT32 * pMinFree,

UINT32 * pNumAllocBlocks,

UINT32 * pNumAllocErr,

UINT32 * pNumFreeErr,

UINT32 blockSize[VOS_MEM_NBLOCKSIZES],

UINT32 usedBlockSize[VOS_MEM_NBLOCKSIZES])
```

Return used and available memory (of memory area above).

## Parameters

| out | pAllocatedMemory | Pointer to allocated memory size                           |
|-----|------------------|--|
| out | pFreeMemory      | Pointer to free memory size                                |
| out | pMinFree         | Pointer to minimal free memory size in statistics interval |
| out | pNumAllocBlocks  | Pointer to number of allocated memory blocks               |
| out | pNumAllocErr     | Pointer to number of allocation errors                     |
| out | pNumFreeErr      | Pointer to number of free errors                           |
| out | blockSize        | Pointer to list of memory block sizes                      |
| out | usedBlockSize    | Pointer to list of used memoryblocks                       |

### Return values

| VOS_NO_ERR   | no error               |
|--------------|------------------------|
| VOS_INIT_ERR | module not initialised |

## 5.33.3.4 vos\_memDelete()

```
EXT_DECL void vos_memDelete ( {\tt UINT8*pMemoryArea} \ )
```

Delete the memory area.

This will eventually invalidate any previously allocated memory blocks! It should be called last before the application quits. No further access to the memory blocks is allowed after this call.

#### **Parameters**

| in | pMemoryArea | Pointer to memory area to use |
|----|-------------|-------------------------------|
|----|-------------|-------------------------------|

This will eventually invalidate any previously allocated memory blocks! It should be called last before the application quits. No further access to the memory blocks is allowed after this call.

#### **Parameters**

| ir | 1 | pMemoryArea | Pointer to memory area used |
|----|---|-------------|-----------------------------|
|----|---|-------------|-----------------------------|

### 5.33.3.5 vos\_memFree()

Deallocate a block of memory (from memory area above).

#### **Parameters**

| in | pMemBlock | Pointer to memory block to be freed |
|----|-----------|-------------------------------------|
|----|-----------|-------------------------------------|

## 5.33.3.6 vos\_memInit()

Initialize the memory unit.

Init a supplied block of memory and prepare it for use with vos\_alloc and vos\_dealloc. The used block sizes can be supplied and will be preallocated.

#### **Parameters**

| in   | pMemoryArea | TemoryArea Pointer to memory area to use |  |
|--|-------------|--|--|
| in   | size        | Size of provided memory area             |  |
| in fragMem Pointer to list of preallocate block sizes, used to fragment memory for large b |             |  |  |

### Return values

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |
| VOS_MEM_ERR   | no memory available            |

Init a supplied block of memory and prepare it for use with vos\_memAlloc and vos\_memFree. The used block sizes can be supplied and will be preallocated. If half of the overall size of the requested memory area would be pre-allocated, either by the default pre-allocation table or a provided one, no pre-allocation takes place.

#### **Parameters**

| ĺ | in  | n pMemoryArea Pointer to memory area to use |  |
|---|---|---|--|
| Ī | in  | size Size of provided memory area           |  |
| Ī | in fragMem Pointer to list of preallocated block sizes, used to fragment memory for large block |   |  |

### **Return values**

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |
| VOS_MEM_ERR   | no memory available            |
| VOS_MUTEX_ERR | no mutex available             |

### 5.33.3.7 vos\_qsort()

```
EXT_DECL void vos_qsort (
            void * pBuf,
            UINT32 num,
            UINT32 size,
            int(*) (const void *, const void *) compare )
```

### Sort an array.

This is just a wrapper for the standard qsort function.

## Parameters

| in,out          | pBuf    | Pointer to the array to sort  |  |
|-----------------|---------|---|--|
| in              | num     | number of elements  |  |
| in              | size    | size of one element   |  |
| Generated by Do | compare | Pointer to compare function return -n if arg1 < arg2, return 0 if arg1 == arg2, return +n |  |
|                 |         | if arg1 > arg2 where n is an integer != 0   |  |

## Return values

none

## 5.33.3.8 vos\_queueCreate()

Initialize a message queue.

Returns a handle for further calls

#### **Parameters**

| in  | queueType    | Define queue type (1 = FIFO, 2 = LIFO, 3 = PRIO) |
|-----|--------------|--|
| in  | maxNoOfMsg   | Maximum number of messages                       |
| out | pQueueHandle | Handle of created queue                          |

### Return values

| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |
| VOS_INIT_ERR   | not supported                  |
| VOS_QUEUE_ERR  | error creating queue           |

# 5.33.3.9 vos\_queueDestroy()

Destroy a message queue.

Free all resources used by this queue

## **Parameters**

| in | queueHandle | Queue handle |
|----|-------------|--------------|
|----|-------------|--------------|

| VOS_NO_ERR | no error |
|------------|----------|

### Return values

| VOS_INIT_ERR   | module not initialised         |
|----------------|--------------------------------|
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |

## 5.33.3.10 vos\_queueReceive()

## Get a message.

### **Parameters**

| in  | queueHandle | Queue handle                                 |
|-----|-------------|--|
| out | ppData      | Pointer to data pointer to be received       |
| out | pSize       | Size of receive data                         |
| in  | usTimeout   | Maximum time to wait for a message (in usec) |

### Return values

| VOSNO_ERR      | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |
| VOS_QUEUE_ERR  | queue is empty                 |

# 5.33.3.11 vos\_queueSend()

### Send a message.

### **Parameters**

| in | queueHandle | Queue handle               |
|----|-------------|----------------------------|
| in | pData       | Pointer to data to be sent |
| in | size        | Size of data to be sent    |

## Return values

| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |
| VOS_INIT_ERR   | not supported                  |
| VOS_QUEUE_ERR  | error creating queue           |

## 5.33.3.12 vos\_strncat()

String concatenation with length limitation.

#### **Parameters**

| in | pStrDst | Destination string               |
|----|---------|----------------------------------|
| in | count   | Size of destination buffer       |
| in | pStrSrc | Null terminated string to append |

### Return values

```
none
```

### 5.33.3.13 vos\_strncpy()

String copy with length limitation.

### **Parameters**

| in | pStrDst | Destination string                   |
|----|---------|--------------------------------------|
| in | pStrSrc | Null terminated string to copy       |
| in | count   | Maximum number of characters to copy |

| none |  |
|------|--|
|      |  |

### 5.33.3.14 vos\_strnicmp()

Case insensitive string compare.

#### **Parameters**

| in | pStr1 | Null terminated string to compare       |
|----|-------|---|
| in | pStr2 | Null terminated string to compare       |
| in | count | Maximum number of characters to compare |

### Return values

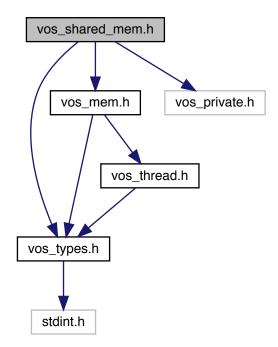
| 0  | - equal                          |
|----|----------------------------------|
| <0 | - string1 less than string 2     |
| >0 | - string 1 greater than string 2 |

# 5.34 vos\_shared\_mem.h File Reference

Shared Memory functions for OS abstraction.

```
#include "vos_types.h"
#include "vos_mem.h"
#include "vos_private.h"
```

Include dependency graph for vos\_shared\_mem.h:



## **Functions**

• EXT\_DECL VOS\_ERR\_T vos\_sharedOpen (const CHAR8 \*pKey, VOS\_SHRD\_T \*pHandle, UINT8 \*\*pp↔ MemoryArea, UINT32 \*pSize)

Create a shared memory area or attach to existing one.

• EXT\_DECL VOS\_ERR\_T vos\_sharedClose (VOS\_SHRD\_T handle, const UINT8 \*pMemoryArea)

Close connection to the shared memory area.

## 5.34.1 Detailed Description

Shared Memory functions for OS abstraction.

This module provides shared memory control supervison

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Kazumasa Aiba, TOSHIBA

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright TOSHIBA, Japan, 2013.

ld

```
vos_mem.h 282 2013-01-11 07:08:44Z 97029
```

#### 5.34.2 Function Documentation

#### 5.34.2.1 vos\_sharedClose()

Close connection to the shared memory area.

If the area was created by the calling process, the area will be closed (freed). If the area was attached, it will be detached. This function is not available in each target implementation.

#### **Parameters**

| in | handle      | Returned handle        |
|----|-------------|------------------------|
| in | pMemoryArea | Pointer to memory area |

#### Return values

| VOS_NO_ERR  | no error            |
|-------------|---------------------|
| VOS_MEM_ERR | no memory available |

## 5.34.2.2 vos\_sharedOpen()

Create a shared memory area or attach to existing one.

The first call with the a specified key will create a shared memory area with the supplied size and will return a handle and a pointer to that area. If the area already exists, the area will be opened. This function is not available in each target implementation.

### **Parameters**

| in     | pKey         | Unique identifier (file name)   |
|--------|--------------|---|
| out    | pHandle      | Pointer to returned handle  |
| out    | ppMemoryArea | Pointer to pointer to memory area                                       |
| in,out | pSize        | Pointer to size of area to allocate, on return actual size after attach |

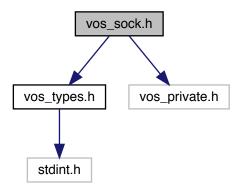
### Return values

| VOS_NO_ERR  | no error            |
|-------------|---------------------|
| VOS_MEM_ERR | no memory available |

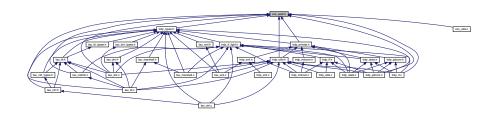
# 5.35 vos\_sock.h File Reference

Typedefs for OS abstraction.

```
#include "vos_types.h"
#include "vos_private.h"
Include dependency graph for vos_sock.h:
```



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



#### **Data Structures**

struct VOS\_SOCK\_OPT\_T

Common socket options.

#### **Macros**

• #define VOS\_MAX\_SOCKET\_CNT 4

The maximum number of sockets influences memory usage; for small systems we should define a smaller set.

#define VOS MAX MULTICAST CNT 5

The maximum number of multicast groups one socket can join.

#define VOS\_TTL\_MULTICAST 64

The maximum number of hops a multicast packet can take.

#define VOS\_MAX\_IF\_NAME\_SIZE 16

The maximum number of IP interface adapters that can be handled by VOS.

#define VOS MAX NUM IF 8

The maximum number of unicast addresses that can be handled by VOS.

#define VOS\_MAX\_NUM\_UNICAST 10

The MAC size supported by VOS.

• #define VOS MAC SIZE 6

Size of socket send and receive buffer.

#define VOS INVALID SOCKET -1

Invalid socket number.

#### **Functions**

• EXT\_DECL UINT16 vos\_htons (UINT16 val)

Byte swapping 2 Bytes.

• EXT DECL UINT16 vos ntohs (UINT16 val)

Byte swapping 2 Bytes.

• EXT\_DECL UINT32 vos\_htonl (UINT32 val)

Byte swapping 4 Bytes.

EXT\_DECL UINT32 vos\_ntohl (UINT32 val)

Byte swapping 4 Bytes.

EXT\_DECL UINT64 vos\_htonll (UINT64 val)

Byte swapping 8å Bytes.

• EXT DECL UINT64 vos ntohll (UINT64 val)

Byte swapping 8 Bytes.

EXT\_DECL UINT32 vos\_dottedIP (const CHAR8 \*pDottedIP)

Convert IP address from dotted dec.

EXT DECL const CHAR8 \* vos ipDotted (UINT32 ipAddress)

Convert IP address to dotted dec.

EXT\_DECL BOOL8 vos\_isMulticast (UINT32 ipAddress)

Check if the supplied address is a multicast group address.

EXT\_DECL VOS\_ERR\_T vos\_getInterfaces (UINT32 \*pAddrCnt, VOS\_IF\_REC\_T ifAddrs[])

Get a list of interface addresses The caller has to provide an array of interface records to be filled.

• EXT DECL BOOL8 vos netIfUp (VOS IP4 ADDR T ifAddress)

Get the state of an interface.

EXT\_DECL\_INT32 vos\_select (SOCKET highDesc, VOS\_FDS\_T \*pReadableFD, VOS\_FDS\_T \*p
 — WriteableFD, VOS\_FDS\_T \*pErrorFD, VOS\_TIMEVAL\_T \*pTimeOut)

select function.

• EXT\_DECL VOS\_ERR\_T vos\_sockInit (void)

Initialize the socket library.

EXT\_DECL void vos\_sockTerm (void)

De-Initialize the socket library.

EXT\_DECL VOS\_ERR\_T vos\_sockGetMAC (UINT8 pMAC[VOS\_MAC\_SIZE])

Return the MAC address of the default adapter.

- EXT\_DECL VOS\_ERR\_T vos\_sockOpenUDP (SOCKET \*pSock, const VOS\_SOCK\_OPT\_T \*pOptions)
   Create an UDP socket.
- EXT\_DECL VOS\_ERR\_T vos\_sockOpenTCP (SOCKET \*pSock, const VOS\_SOCK\_OPT\_T \*pOptions)

  Create a TCP socket.
- EXT\_DECL VOS\_ERR\_T vos\_sockClose (SOCKET sock)

Close a socket.

- EXT\_DECL VOS\_ERR\_T vos\_sockSetOptions (SOCKET sock, const VOS\_SOCK\_OPT\_T \*pOptions)
   Set socket options.
- EXT\_DECL VOS\_ERR\_T vos\_sockJoinMC (SOCKET sock, UINT32 mcAddress, UINT32 ipAddress) Join a multicast group.
- EXT\_DECL VOS\_ERR\_T vos\_sockLeaveMC (SOCKET sock, UINT32 mcAddress, UINT32 ipAddress)
   Leave a multicast group.
- EXT\_DECL VOS\_ERR\_T vos\_sockSendUDP (SOCKET sock, const UINT8 \*pBuffer, UINT32 \*pSize, UIN← T32 ipAddress, UINT16 port)

Send UDP data.

EXT\_DECL VOS\_ERR\_T vos\_sockReceiveUDP (SOCKET sock, UINT8 \*pBuffer, UINT32 \*pSize, UINT32 \*pSrcIPAddr, UINT16 \*pSrcIPPort, UINT32 \*pDstIPAddr, BOOL8 peek)

Receive UDP data.

• EXT\_DECL VOS\_ERR\_T vos\_sockBind (SOCKET sock, UINT32 ipAddress, UINT16 port)

Bind a socket to an address and port.

EXT\_DECL VOS\_ERR\_T vos\_sockListen (SOCKET sock, UINT32 backlog)

Listen for incoming TCP connections.

EXT\_DECL VOS\_ERR\_T vos\_sockAccept (SOCKET sock, SOCKET \*pSock, UINT32 \*pIPAddress, UINT16 \*pPort)

Accept an incoming TCP connection.

- EXT\_DECL VOS\_ERR\_T vos\_sockConnect (SOCKET sock, UINT32 ipAddress, UINT16 port)
   Open a TCP connection.
- EXT\_DECL VOS\_ERR\_T vos\_sockSendTCP (SOCKET sock, const UINT8 \*pBuffer, UINT32 \*pSize)
   Send TCP data.
- EXT\_DECL VOS\_ERR\_T vos\_sockReceiveTCP (SOCKET sock, UINT8 \*pBuffer, UINT32 \*pSize)

  \*\*Receive TCP data.\*\*
- EXT\_DECL VOS\_ERR\_T vos\_sockSetMulticastlf (SOCKET sock, UINT32 mclfAddress)
   Set Using Multicast I/F.
- EXT\_DECL VOS\_IP4\_ADDR\_T vos\_determineBindAddr (VOS\_IP4\_ADDR\_T srcIP, VOS\_IP4\_ADDR\_ T mcGroup, VOS\_IP4\_ADDR\_T rcvMostly)

Determines the address to bind to since the behaviour in the different OS is different.

#### 5.35.1 Detailed Description

Typedefs for OS abstraction.

This is the declaration for the OS independend socket interface

Note

Project: TCNOpen TRDP prototype stack

**Author** 

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

vos sock.h 1765 2018-10-04 12:18:54Z ahweiss

```
BL 2018-06-20: Ticket #184: Building with VS 2015: WIN64 and Windows threads (SOCKET instead of INT32)
BL 2018-03-06: 64Bit endian swap added
BL 2017-05-22: Ticket #122: Addendum for 64Bit compatibility (VOS_TIME_T -> VOS_TIMEVAL_T)
```

### 5.35.2 Macro Definition Documentation

```
5.35.2.1 VOS_MAX_SOCKET_CNT
```

```
#define VOS_MAX_SOCKET_CNT 4
```

The maximum number of sockets influences memory usage; for small systems we should define a smaller set.

The maximum number of concurrent usable sockets per application session

```
5.35.2.2 VOS_TTL_MULTICAST
```

```
#define VOS_TTL_MULTICAST 64
```

The maximum number of hops a multicast packet can take.

The maximum size for the interface name

## 5.35.3 Function Documentation

## 5.35.3.1 vos\_determineBindAddr()

Determines the address to bind to since the behaviour in the different OS is different.

### **Parameters**

|   | in | srcIP     | IP to bind to (0 = any address)                          |
|---|----|-----------|--|
|   | in | mcGroup   | MC group to join (0 = do not join)                       |
| Ī | in | rcvMostly | primarily used for receiving (tbd: bind on sender, too?) |

### Return values

| Address | to bind to |
|---------|------------|
|---------|------------|

## 5.35.3.2 vos\_dottedIP()

Convert IP address from dotted dec.

to !host! endianess

#### **Parameters**

| in | p⇔       | IP address as dotted decimal. |
|----|----------|-------------------------------|
|    | DottedIP |                               |

## Return values

|  | address | in UINT32 in host endianess |
|--|---------|-----------------------------|
|--|---------|-----------------------------|

## 5.35.3.3 vos\_getInterfaces()

Get a list of interface addresses The caller has to provide an array of interface records to be filled.

## **Parameters**

| in,out | pAddrCnt | in: pointer to array size of interface record out: pointer to number of interface records read |
|--------|----------|--|
| in,out | ifAddrs  | array of interface records   |

| VOS_NO_ERR    | no error                        |
|---------------|---------------------------------|
| VOS_PARAM_ERR | pAddrCnt and/or ifAddrs == NULL |
| VOS_MEM_ERR   | memory allocation error         |

## Return values

| VOS_SOCK_ERR | GetAdaptersInfo() error |
|--------------|-------------------------|
|--------------|-------------------------|

## 5.35.3.4 vos\_htonl()

```
EXT_DECL UINT32 vos_htonl ( UINT32 val )
```

## Byte swapping 4 Bytes.

#### **Parameters**

| in <i>val</i> | Initial value. |
|---------------|----------------|
|---------------|----------------|

### Return values

```
swapped value
```

## 5.35.3.5 vos\_htonll()

### Byte swapping 8å Bytes.

# **Parameters**

| in <i>val</i> | Initial value. |
|---------------|----------------|
|---------------|----------------|

### **Return values**

```
swapped value
```

## 5.35.3.6 vos\_htons()

## Byte swapping 2 Bytes.

## **Parameters**

## Return values

```
swapped value
```

## 5.35.3.7 vos\_ipDotted()

Convert IP address to dotted dec.

from !host! endianess

#### **Parameters**

|  | in | ipAddress | address in UINT32 in host endianess |  |
|--|----|-----------|-------------------------------------|--|
|--|----|-----------|-------------------------------------|--|

#### **Return values**

IP address as dotted decimal.

## 5.35.3.8 vos\_isMulticast()

```
EXT_DECL BOOL8 vos_isMulticast ( {\tt UINT32~\it ipAddress~)}
```

Check if the supplied address is a multicast group address.

## **Parameters**

| in | ipAddress | IP address to check. |
|----|-----------|----------------------|
|----|-----------|----------------------|

| TRUE  | address is a multicast address     |  |
|-------|------------------------------------|--|
| FALSE | address is not a multicast address |  |

## 5.35.3.9 vos\_netIfUp()

```
EXT_DECL BOOL8 vos_netIfUp ( {\tt VOS\_IP4\_ADDR\_T~\it ifAddress~)}
```

Get the state of an interface.

### **Parameters**

| in | ifAddress | address of interface to check |
|----|-----------|-------------------------------|
|----|-----------|-------------------------------|

### Return values

TRUE interface is up and ready FALSE interface is down / not ready

### 5.35.3.10 vos\_ntohl()

Byte swapping 4 Bytes.

### **Parameters**

| in | val | Initial value. |
|----|-----|----------------|
|----|-----|----------------|

#### **Return values**

```
swapped value
```

## 5.35.3.11 vos\_ntohll()

Byte swapping 8 Bytes.

## **Parameters**

| in | val | Initial value. |
|----|-----|----------------|

| swapped | value |
|---------|-------|

#### 5.35.3.12 vos\_ntohs()

Byte swapping 2 Bytes.

### **Parameters**

| in <i>val</i> | Initial value. |
|---------------|----------------|
|---------------|----------------|

### Return values

```
swapped value
```

#### 5.35.3.13 vos\_select()

```
EXT_DECL INT32 vos_select (

SOCKET highDesc,

VOS_FDS_T * pReadableFD,

VOS_FDS_T * pWriteableFD,

VOS_FDS_T * pErrorFD,

VOS_TIMEVAL_T * pTimeOut )
```

select function.

Set the ready sockets in the supplied sets. Note: Some target systems might define this function as NOP.

### **Parameters**

| in     | highDesc     | max. socket descriptor + 1      |
|--------|--------------|---------------------------------|
| in,out | pReadableFD  | pointer to readable socket set  |
| in,out | pWriteableFD | pointer to writeable socket set |
| in,out | pErrorFD     | pointer to error socket set     |
| in     | pTimeOut     | pointer to time out value       |

## Return values

```
number of ready file descriptors
```

## 5.35.3.14 vos\_sockAccept()

```
EXT_DECL VOS_ERR_T vos_sockAccept ( SOCKET sock,
```

```
SOCKET * pSock,
UINT32 * pIPAddress,
UINT16 * pPort )
```

Accept an incoming TCP connection.

Accept incoming connections on the provided socket. May block and will return a new socket descriptor when accepting a connection. The original socket \*pSock, remains open.

#### **Parameters**

| in  | sock       | Socket descriptor                                |
|-----|------------|--|
| out | pSock      | Pointer to socket descriptor, on exit new socket |
| out | pIPAddress | source IP to receive on, 0 for any               |
| out | pPort      | port to receive on, 17224 for PD                 |

#### Return values

| VOS_NO_ERR      | no error                        |
|-----------------|---------------------------------|
| VOS_PARAM_ERR   | NULL parameter, parameter error |
| VOS_UNKNOWN_ERR | sock descriptor unknown error   |

### 5.35.3.15 vos\_sockBind()

Bind a socket to an address and port.

### **Parameters**

| in | sock      | socket descriptor                    |
|----|-----------|--------------------------------------|
| in | ipAddress | source IP to receive from, 0 for any |
| in | port      | port to receive from                 |

### Return values

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |
| VOS_IO_ERR    | Input/Output error             |
| VOS_MEM_ERR   | resource error                 |

#### 5.35.3.16 vos\_sockClose()

```
{\tt EXT\_DECL~VOS\_ERR\_T~vos\_sockClose~(}
```

```
SOCKET sock )
```

Close a socket.

Release any resources aquired by this socket

### **Parameters**

| in sock socket descriptor |
|---------------------------|
|---------------------------|

#### Return values

| VOS_NO_ERR    | no error      |
|---------------|---------------|
| VOS_PARAM_ERR | pSock == NULL |

## 5.35.3.17 vos\_sockConnect()

```
EXT_DECL VOS_ERR_T vos_sockConnect (

SOCKET sock,

UINT32 ipAddress,

UINT16 port )
```

Open a TCP connection.

#### **Parameters**

|   | in | sock      | socket descriptor |
|---|----|-----------|-------------------|
|   | in | ipAddress | destination IP    |
| ĺ | in | port      | destination port  |

### Return values

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |
| VOS_IO_ERR    | Input/Output error             |

## 5.35.3.18 vos\_sockGetMAC()

Return the MAC address of the default adapter.

## **Parameters**

| out | рМАС | return MAC address. |
|-----|------|---------------------|
|-----|------|---------------------|

### **Return values**

| VOS_NO_ERR    | no error                                     |
|---------------|--|
| VOS_PARAM_ERR | pMAC == NULL                                 |
| VOS_SOCK_ERR  | socket not available or option not supported |

## 5.35.3.19 vos\_socklnit()

Initialize the socket library.

Must be called once before any other call

### Return values

| VOS_NO_ERR   | no error              |
|--------------|-----------------------|
| VOS_SOCK_ERR | sockets not supported |

## 5.35.3.20 vos\_sockJoinMC()

```
EXT_DECL VOS_ERR_T vos_sockJoinMC (

SOCKET sock,

UINT32 mcAddress,

UINT32 ipAddress)
```

Join a multicast group.

Note: Some target systems might not support this option.

### **Parameters**

| in | sock      | socket descriptor                                     |
|----|-----------|---|
| in | mcAddress | multicast group to join                               |
| in | ipAddress | depicts interface on which to join, default 0 for any |

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |
| VOS_SOCK_ERR  | option not supported           |

## 5.35.3.21 vos\_sockLeaveMC()

Leave a multicast group.

Note: Some target systems might not support this option.

### **Parameters**

| in | sock      | socket descriptor                                      |
|----|-----------|--|
| in | mcAddress | multicast group to join                                |
| in | ipAddress | depicts interface on which to leave, default 0 for any |

### Return values

| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |
| VOS_SOCK_ERR   | option not supported           |

## 5.35.3.22 vos\_sockListen()

Listen for incoming TCP connections.

## **Parameters**

| in | sock    | socket descriptor                             |
|----|---------|---|
| in | backlog | maximum connection attempts if system is busy |

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |
| VOS_IO_ERR    | Input/Output error             |
| VOS_MEM_ERR   | resource error                 |

### 5.35.3.23 vos\_sockOpenTCP()

# Create a TCP socket.

Return a socket descriptor for further calls. The socket options are optional and can be applied later.

#### **Parameters**

| out | pSock    | pointer to socket descriptor returned |
|-----|----------|---------------------------------------|
| in  | pOptions | pointer to socket options (optional)  |

# Return values

| VOS_NO_ERR    | no error                                     |
|---------------|--|
| VOS_PARAM_ERR | pSock == NULL                                |
| VOS_SOCK_ERR  | socket not available or option not supported |

# 5.35.3.24 vos\_sockOpenUDP()

#### Create an UDP socket.

Return a socket descriptor for further calls. The socket options are optional and can be applied later. Note: Some target systems might not support every option.

# **Parameters**

| out | pSock    | pointer to socket descriptor returned |
|-----|----------|---------------------------------------|
| in  | pOptions | pointer to socket options (optional)  |

### Return values

| VOS_NO_ERR    | no error                                     |
|---------------|--|
| VOS_PARAM_ERR | pSock == NULL                                |
| VOS_SOCK_ERR  | socket not available or option not supported |

# 5.35.3.25 vos\_sockReceiveTCP()

```
EXT_DECL VOS_ERR_T vos_sockReceiveTCP (
SOCKET sock,
```

```
UINT8 * pBuffer,
UINT32 * pSize )
```

### Receive TCP data.

The caller must provide a sufficient sized buffer. If the supplied buffer is smaller than the bytes received, \*pSize will reflect the number of copied bytes and the call should be repeated until \*pSize is 0 (zero). If the socket was created in blocking-mode (default), then this call will block and will only return if data has been received or the socket was closed or an error occured. If called in non-blocking mode, and no data is available, VOS\_NODATA\_ERR will be returned.

#### **Parameters**

| in     | sock    | socket descriptor                   |
|--------|---------|-------------------------------------|
| out    | pBuffer | pointer to applications data buffer |
| in,out | pSize   | pointer to the received data size   |

#### Return values

| VOS_NO_ERR     | no error                                 |
|----------------|--|
| VOS_PARAM_ERR  | sock descriptor unknown, parameter error |
| VOS_IO_ERR     | data could not be read                   |
| VOS_NODATA_ERR | no data in non-blocking                  |
| VOS_BLOCK_ERR  | call would have blocked in blocking mode |

### 5.35.3.26 vos\_sockReceiveUDP()

#### Receive UDP data.

The caller must provide a sufficient sized buffer. If the supplied buffer is smaller than the bytes received, \*pSize will reflect the number of copied bytes and the call should be repeated until \*pSize is 0 (zero). If the socket was created in blocking-mode (default), then this call will block and will only return if data has been received or the socket was closed or an error occured. If called in non-blocking mode, and no data is available, VOS\_NODATA\_ERR will be returned. If pointers are provided, source IP, source port and destination IP will be reported on return.

### **Parameters**

| in     | sock       | socket descriptor                   |
|--------|------------|-------------------------------------|
| out    | pBuffer    | pointer to applications data buffer |
| in,out | pSize      | pointer to the received data size   |
| out    | pSrcIPAddr | pointer to source IP                |
| out    | pSrcIPPort | pointer to source port              |
| out    | pDstIPAddr | pointer to dest IP                  |
| in     | peek       | if true, leave data in queue        |

# Return values

| VOS_NO_ERR     | no error                                 |
|----------------|--|
| VOS_PARAM_ERR  | sock descriptor unknown, parameter error |
| VOS_IO_ERR     | data could not be read                   |
| VOS_NODATA_ERR | no data                                  |
| VOS_BLOCK_ERR  | Call would have blocked in blocking mode |

# 5.35.3.27 vos\_sockSendTCP()

Send TCP data.

Send data to the supplied address and port.

# **Parameters**

| in     | sock    | socket descriptor                                   |
|--------|---------|---|
| in     | pBuffer | pointer to data to send                             |
| in,out | pSize   | In: size of the data to send, Out: no of bytes sent |

# Return values

| VOS_NO_ERR     | no error  |
|----------------|---|
| VOS_PARAM_ERR  | sock descriptor unknown, parameter error                      |
| VOS_IO_ERR     | data could not be sent  |
| VOS_NOCONN_ERR | no TCP connection   |
| VOS_BLOCK_ERR  | call would have blocked in blocking mode, data partially sent |

# 5.35.3.28 vos\_sockSendUDP()

Send UDP data.

Send data to the given address and port.

# **Parameters**

| in     | sock      | socket descriptor                                   |
|--------|-----------|---|
| in     | pBuffer   | pointer to data to send                             |
| in,out | pSize     | In: size of the data to send, Out: no of bytes sent |
| in     | ipAddress | destination IP                                      |
| in     | port      | destination port                                    |

# Return values

| VOS_NO_ERR    | no error                                 |
|---------------|--|
| VOS_PARAM_ERR | parameter out of range/invalid           |
| VOS_IO_ERR    | data could not be sent                   |
| VOS_BLOCK_ERR | Call would have blocked in blocking mode |

# 5.35.3.29 vos\_sockSetMulticastIf()

# Set Using Multicast I/F.

# **Parameters**

| Ī | in | sock        | socket descriptor           |
|---|----|-------------|-----------------------------|
| Ī | in | mclfAddress | using Multicast I/F Address |

# Return values

| VOS_NO_ERR    | no error                                 |
|---------------|--|
| VOS_PARAM_ERR | sock descriptor unknown, parameter error |

# 5.35.3.30 vos\_sockSetOptions()

```
EXT_DECL VOS_ERR_T vos_sockSetOptions ( {\tt SOCKET}\ sock, {\tt const}\ {\tt VOS\_SOCK\_OPT\_T}\ *\ pOptions\ )
```

# Set socket options.

Note: Some target systems might not support each option.

# **Parameters**

| in | sock     | socket descriptor                    |
|----|----------|--------------------------------------|
| in | pOptions | pointer to socket options (optional) |

# Return values

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |

# 5.35.3.31 vos\_sockTerm()

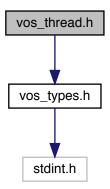
De-Initialize the socket library.

Must be called after last socket call

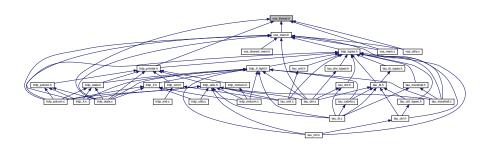
# 5.36 vos\_thread.h File Reference

Threading functions for OS abstraction.

```
#include "vos_types.h"
Include dependency graph for vos_thread.h:
```



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



#### **Macros**

#define VOS\_MAX\_THREAD\_CNT 100

The maximum number of concurrent usable threads.

#define VOS SEMA WAIT FOREVER 0xFFFFFFFU

Timeout value to wait forever for a semaphore.

# **Typedefs**

typedef UINT8 VOS THREAD PRIORITY T

Thread priority range from 1 (highest) to 255 (lowest), 0 default of the target system.

typedef void(\_\_cdecl \* VOS\_THREAD\_FUNC\_T) (void \*pArg)

Thread function definition.

typedef struct VOS\_MUTEX \* VOS\_MUTEX\_T

Hidden mutex handle definition.

typedef struct VOS\_SEMA \* VOS\_SEMA\_T

Hidden semaphore handle definition.

typedef void \* VOS\_THREAD\_T

Hidden thread handle definition.

# **Enumerations**

enum VOS\_THREAD\_POLICY\_T

Thread policy matching pthread/Posix defines.

enum VOS\_SEMA\_STATE\_T

State of the semaphore.

# **Functions**

· EXT DECL VOS ERR T vos threadInit (void)

Initialize the thread library.

EXT\_DECL void vos\_threadTerm (void)

De-Initialize the thread library.

• EXT\_DECL VOS\_ERR\_T vos\_threadCreate (VOS\_THREAD\_T \*pThread, const CHAR8 \*pName, VOS\_← THREAD\_POLICY\_T policy, VOS\_THREAD\_PRIORITY\_T priority, UINT32 interval, UINT32 stackSize, V← OS\_THREAD\_FUNC\_T pFunction, void \*pArguments)

Create a thread.

- EXT\_DECL void vos\_cyclicThread (UINT32 interval, VOS\_THREAD\_FUNC\_T pFunction, void \*pArguments) Cyclic thread functions.
- EXT\_DECL VOS\_ERR\_T vos\_threadTerminate (VOS\_THREAD\_T thread)

Terminate a thread.

• EXT DECL VOS ERR T vos threadlsActive (VOS THREAD T thread)

Is the thread still active? This call will return VOS\_NO\_ERR if the thread is still active, VOS\_PARAM\_ERR in case it ran out

EXT\_DECL VOS\_ERR\_T vos\_threadDelay (UINT32 delay)

Delay the execution of the current thread by the given delay in us.

EXT\_DECL\_VOS\_ERR\_T vos\_threadSelf (VOS\_THREAD\_T\*pThread)

Return thread handle of calling task.

EXT\_DECL void vos\_getTime (VOS\_TIMEVAL\_T \*pTime)

Return the current time in sec and us.

EXT\_DECL const CHAR8 \* vos\_getTimeStamp (void)

Get a time-stamp string.

EXT\_DECL void vos\_clearTime (VOS\_TIMEVAL\_T \*pTime)

Clear the time stamp.

EXT\_DECL void vos\_addTime (VOS\_TIMEVAL\_T \*pTime, const VOS\_TIMEVAL\_T \*pAdd)

Add the second to the first time stamp, return sum in first.

EXT\_DECL void vos\_subTime (VOS\_TIMEVAL\_T \*pTime, const VOS\_TIMEVAL\_T \*pSub)

Subtract the second from the first time stamp, return diff in first.

EXT\_DECL INT32 vos\_cmpTime (const VOS\_TIMEVAL\_T \*pTime, const VOS\_TIMEVAL\_T \*pCmp)

Compare the second from the first time stamp, return diff in first.

EXT\_DECL void vos\_divTime (VOS\_TIMEVAL\_T \*pTime, UINT32 divisor)

Divide the first time by the second, return quotient in first.

EXT\_DECL void vos\_mulTime (VOS\_TIMEVAL\_T \*pTime, UINT32 mul)

Multiply the first time by the second, return product in first.

EXT\_DECL void vos\_getUuid (VOS\_UUID\_T pUuID)

Get a universal unique identifier according to RFC 4122 time based version.

EXT\_DECL VOS\_ERR\_T vos\_mutexCreate (VOS\_MUTEX\_T \*pMutex)

Create a mutex.

EXT\_DECL void vos\_mutexDelete (VOS\_MUTEX\_T pMutex)

Delete a mutex.

EXT\_DECL VOS\_ERR\_T vos\_mutexLock (VOS\_MUTEX\_T pMutex)

Take a mutex.

• EXT\_DECL VOS\_ERR\_T vos\_mutexTryLock (VOS\_MUTEX\_T pMutex)

Try to take a mutex.

EXT\_DECL VOS\_ERR\_T vos\_mutexUnlock (VOS\_MUTEX\_T pMutex)

Release a mutex.

• EXT\_DECL VOS\_ERR\_T vos\_semaCreate (VOS\_SEMA\_T \*pSema, VOS\_SEMA\_STATE\_T initialState)

Create a semaphore.

EXT\_DECL void vos\_semaDelete (VOS\_SEMA\_T sema)

Delete a semaphore.

• EXT DECL VOS ERR T vos semaTake (VOS SEMA T sema, UINT32 timeout)

Take a semaphore.

EXT\_DECL void vos\_semaGive (VOS\_SEMA\_T sema)

Give a semaphore.

#### 5.36.1 Detailed Description

Threading functions for OS abstraction.

Thread-, semaphore- and time-handling functions

Note

Project: TCNOpen TRDP prototype stack

Author

Bernd Loehr, NewTec GmbH

#### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2014. All rights reserved.

ld

vos\_thread.h 1763 2018-09-21 16:03:13Z ahweiss

```
BL 2017-05-22: Ticket #122: Addendum for 64Bit compatibility (VOS_TIME_T -> VOS_TIMEVAL_T)
```

# 5.36.2 Function Documentation

# 5.36.2.1 vos\_addTime()

Add the second to the first time stamp, return sum in first.

#### **Parameters**

| in,out | pTime | Pointer to time value |
|--------|-------|-----------------------|
| in     | pAdd  | Pointer to time value |

# 5.36.2.2 vos\_clearTime()

Clear the time stamp.

### **Parameters**

| out | pTime | Pointer to time value |  |  |  |
|-----|-------|-----------------------|--|--|--|

# 5.36.2.3 vos\_cmpTime()

Compare the second from the first time stamp, return diff in first.

# **Parameters**

| in,out | pTime | Pointer to time value            |
|--------|-------|----------------------------------|
| in     | рСтр  | Pointer to time value to compare |

# Return values

| 0  | pTime == pCmp |
|----|---------------|
| -1 | pTime < pCmp  |
| 1  | pTime > pCmp  |

# 5.36.2.4 vos\_cyclicThread()

Cyclic thread functions.

Wrapper for cyclic threads. The thread function will be called cyclically with interval.

# **Parameters**

| in interval Interval for cyclic threads in us (incl. |  | Interval for cyclic threads in us (incl. runtime) |
|--|--|---|
| in <i>pFunction</i> Pointer to the thread function   |  | Pointer to the thread function                    |
| in <i>pArguments</i>                                 |  | Pointer to the thread function parameters         |

# Return values

```
void
```

# 5.36.2.5 vos\_divTime()

Divide the first time by the second, return quotient in first.

### **Parameters**

| in,out | pTime   | Pointer to time value |
|--------|---------|-----------------------|
| in     | divisor | Divisor               |

# 5.36.2.6 vos\_getTime()

Return the current time in sec and us.

#### **Parameters**

| out | pTime | Pointer to time value |
|-----|-------|-----------------------|
|-----|-------|-----------------------|

# 5.36.2.7 vos\_getTimeStamp()

Get a time-stamp string.

Get a time-stamp string for debugging in the form "yyyymmdd-hh:mm:ss.ms" Depending on the used OS / hardware the time might not be a real-time stamp but relative from start of system.

### **Return values**

```
timestamp "yyyymmdd-hh:mm:ss.ms"
```

### 5.36.2.8 vos\_getUuid()

Get a universal unique identifier according to RFC 4122 time based version.

# **Parameters**

```
out | pUuID | Pointer to a universal unique identifier
```

# 5.36.2.9 vos\_mulTime()

Multiply the first time by the second, return product in first.

### **Parameters**

| in,out | pTime | Pointer to time value |
|--------|-------|-----------------------|
| in     | mul   | Factor                |

# 5.36.2.10 vos\_mutexCreate()

Create a mutex.

Return a mutex handle. The mutex will be available at creation.

# **Parameters**

| out | pMutex | Pointer to mutex handle |
|-----|--------|-------------------------|
|-----|--------|-------------------------|

### Return values

| VOS_NO_ERR    | no error               |
|---------------|------------------------|
| VOS_INIT_ERR  | module not initialised |
| VOS_PARAM_ERR | pMutex == NULL         |
| VOS_MUTEX_ERR | no mutex available     |

# 5.36.2.11 vos\_mutexDelete()

Delete a mutex.

Release the resources taken by the mutex.

# **Parameters**

| in | pMutex | mutex handle |  |
|----|--------|--------------|--|

# **Return values**

| VOS_NO_ERR | no error |
|------------|----------|
|------------|----------|

# 5.36.2.12 vos\_mutexLock()

Take a mutex.

Wait for the mutex to become available (lock).

### **Parameters**

| in | pMutex | mutex handle |
|----|--------|--------------|
|----|--------|--------------|

# Return values

| VOS_NO_ERR     | no error               |
|----------------|------------------------|
| VOS_INIT_ERR   | module not initialised |
| VOS_NOINIT_ERR | invalid handle         |

# 5.36.2.13 vos\_mutexTryLock()

Try to take a mutex.

If mutex is can't be taken VOS\_MUTEX\_ERR is returned.

### **Parameters**

| in | pMutex | mutex handle |
|----|--------|--------------|

### Return values

| VOS_NO_ERR     | no error               |
|----------------|------------------------|
| VOS_INIT_ERR   | module not initialised |
| VOS_NOINIT_ERR | invalid handle         |
| VOS_MUTEX_ERR  | no mutex available     |

# 5.36.2.14 vos\_mutexUnlock()

Release a mutex.

Unlock the mutex.

### **Parameters**

| in | pMutex | mutex handle |
|----|--------|--------------|
|----|--------|--------------|

# 5.36.2.15 vos\_semaCreate()

Create a semaphore.

Return a semaphore handle. Depending on the initial state the semaphore will be available on creation or not.

### **Parameters**

| out | pSema        | Pointer to semaphore handle        |
|-----|--------------|------------------------------------|
| in  | initialState | The initial state of the sempahore |

#### Return values

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_INIT_ERR  | module not initialised         |
| VOS_PARAM_ERR | parameter out of range/invalid |
| VOS_SEMA_ERR  | no semaphore available         |

# 5.36.2.16 vos\_semaDelete()

```
EXT_DECL void vos_semaDelete ( \label{eq:vos_semaDelete} VOS\_SEMA\_T \ \textit{sema} \ )
```

Delete a semaphore.

This will eventually release any processes waiting for the semaphore.

# **Parameters**

| in | sema | semaphore handle |
|----|------|------------------|
|    | ooma | oomaphoro nanalo |

# 5.36.2.17 vos\_semaGive()

Give a semaphore.

Release (increase) a semaphore.

# **Parameters**

| in sema semaphore handle | i | n | sema | semaphore handle |
|--------------------------|---|---|------|------------------|
|--------------------------|---|---|------|------------------|

# 5.36.2.18 vos\_semaTake()

Take a semaphore.

Try to get (decrease) a semaphore.

# **Parameters**

| in | sema    | semaphore handle                         |
|----|---------|--|
| in | timeout | Max. time in us to wait, 0 means no wait |

# Return values

| VOS_NO_ERR     | no error                        |
|----------------|---------------------------------|
| VOS_INIT_ERR   | module not initialised          |
| VOS_NOINIT_ERR | invalid handle                  |
| VOS_PARAM_ERR  | parameter out of range/invalid  |
| VOS_SEMA_ERR   | could not get semaphore in time |

# 5.36.2.19 vos\_subTime()

Subtract the second from the first time stamp, return diff in first.

### **Parameters**

| in,out | pTime | Pointer to time value |
|--------|-------|-----------------------|
| in     | pSub  | Pointer to time value |

### 5.36.2.20 vos\_threadCreate()

### Create a thread.

Create a thread and return a thread handle for further requests. Not each parameter may be supported by all target systems!

# **Parameters**

| out | pThread    | Pointer to returned thread handle               |
|-----|------------|---|
| in  | pName      | Pointer to name of the thread (optional)        |
| in  | policy     | Scheduling policy (FIFO, Round Robin or other)  |
| in  | priority   | Scheduling priority (1255 (highest), default 0) |
| in  | interval   | Interval for cyclic threads in us (optional)    |
| in  | stackSize  | Minimum stacksize, default 0: 16kB              |
| in  | pFunction  | Pointer to the thread function                  |
| in  | pArguments | Pointer to the thread function parameters       |

### Return values

| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |

### 5.36.2.21 vos\_threadDelay()

Delay the execution of the current thread by the given delay in us.

# **Parameters**

| ĺ |    |       |             |
|---|----|-------|-------------|
|   | in | delay | Delay in us |

### Return values

| VOS_NO_ERR   | no error               |
|--------------|------------------------|
| VOS_INIT_ERR | module not initialised |

### 5.36.2.22 vos\_threadInit()

Initialize the thread library.

Must be called once before any other call

### Return values

| VOS_NO_ERR   | no error                |
|--------------|-------------------------|
| VOS_INIT_ERR | threading not supported |

# 5.36.2.23 vos\_threadlsActive()

Is the thread still active? This call will return VOS\_NO\_ERR if the thread is still active, VOS\_PARAM\_ERR in case it ran out.

### **Parameters**

| in thread Threa |
|-----------------|
|-----------------|

# Return values

| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |

# 5.36.2.24 vos\_threadSelf()

Return thread handle of calling task.

#### **Parameters**

|  | out | pThread | pointer to thread handle |
|--|-----|---------|--------------------------|

# Return values

| VOS_NO_ERR    | no error                       |
|---------------|--------------------------------|
| VOS_PARAM_ERR | parameter out of range/invalid |

# 5.36.2.25 vos\_threadTerm()

De-Initialize the thread library.

Must be called after last thread/timer call

# 5.36.2.26 vos\_threadTerminate()

Terminate a thread.

This call will terminate the thread with the given threadld and release all resources. Depending on the underlying architectures, it may just block until the thread ran out.

# **Parameters**

|  | in | thread | Thread handle (or NULL if current thread) |  |
|--|----|--------|---|--|
|--|----|--------|---|--|

### Return values

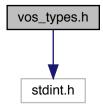
| VOS_NO_ERR     | no error                       |
|----------------|--------------------------------|
| VOS_INIT_ERR   | module not initialised         |
| VOS_NOINIT_ERR | invalid handle                 |
| VOS_PARAM_ERR  | parameter out of range/invalid |

# 5.37 vos\_types.h File Reference

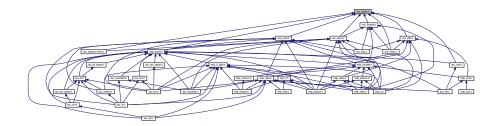
Typedefs for OS abstraction.

#include <stdint.h>

Include dependency graph for vos\_types.h:



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



# **Data Structures**

• struct VOS\_VERSION\_T

Version information.

# Macros

• #define INLINE inline

inline macros

• #define AV\_ERROR 0x00

ANTIVALENT8 values.

• #define TR\_DIR1 0x01

Directions/Orientations.

# **Typedefs**

typedef UINT8 VOS\_UUID\_T[16]

universal unique identifier according to RFC 4122, time based version

typedef struct timeval VOS\_TIMEVAL\_T

Timer value compatible with timeval / select.

• typedef void(\* VOS\_PRINT\_DBG\_T) (void \*pRefCon, VOS\_LOG\_T category, const CHAR8 \*pTime, const CHAR8 \*pFile, UINT16 LineNumber, const CHAR8 \*pMsgStr)

Function definition for error/debug output.

### **Enumerations**

```
enum VOS ERR T {
 VOS NO ERR = 0,
 VOS PARAM ERR = -1,
 VOS_INIT_ERR = -2,
 VOS_NOINIT_ERR = -3,
 VOS_TIMEOUT_ERR = -4,
 VOS NODATA ERR = -5,
 VOS_SOCK_ERR = -6,
 VOS_IO_ERR = -7,
 VOS MEM ERR = -8,
 VOS SEMA ERR = -9,
 VOS_QUEUE_ERR = -10,
 VOS_QUEUE_FULL_ERR = -11,
 VOS MUTEX ERR = -12,
 VOS THREAD ERR = -13,
 VOS_BLOCK_ERR = -14,
 VOS_INTEGRATION_ERR = -15,
 VOS NOCONN ERR = -16,
 VOS INUSE ERR = -49,
 VOS UNKNOWN ERR = -99 }
    Return codes for all VOS API functions.
enum VOS LOG T {
 VOS LOG ERROR = 0,
 VOS_LOG_WARNING = 1,
 VOS\_LOG\_INFO = 2,
 VOS LOG DBG = 3,
 VOS_LOG_USR = 4 }
    Categories for logging.
```

### 5.37.1 Detailed Description

Typedefs for OS abstraction.

Note

Project: TCNOpen TRDP prototype stack

Author

Bernd Loehr, NewTec GmbH

# Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

### vos types.h 1763 2018-09-21 16:03:13Z ahweiss

```
BL 2018-06-25: Ticket #202: vos_mutexTrylock return value
BL 2018-05-03: no inline if < C99
BL 2017-11-17: Undone: Ticket #169 Encapsulate declaration of packed structures within a macro
BL 2017-11-10: Additional log type: VOS_LOG_USR
BL 2017-05-22: Ticket #122: Addendum for 64Bit compatibility (VOS_TIME_T -> VOS_TIMEVAL_T)
BL 2017-05-08: Doxygen comment errors
BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings)
```

# 5.37.2 Typedef Documentation

# 5.37.2.1 VOS\_PRINT\_DBG\_T

typedef void(\* VOS\_PRINT\_DBG\_T) (void \*pRefCon, VOS\_LOG\_T category, const CHAR8 \*pTime, const CHAR8 \*pFile, UINT16 LineNumber, const CHAR8 \*pMsgStr)

Function definition for error/debug output.

The function will be called for logging and error message output. The user can decide, what kind of info will be logged by filtering the category.

### **Parameters**

| in | pRefCon   | pointer to user context           |  |
|----|---|-----------------------------------|--|
| in | n category Log category (Error, Warning, Info etc.)   |                                   |  |
| in | pTime pointer to NULL-terminated string of time stamp |                                   |  |
| in | pointer to NULL-terminated string of source mod       |                                   |  |
| in | LineNumber  | Line number                       |  |
| in | pMsgStr   | pointer to NULL-terminated string |  |

### 5.37.2.2 VOS\_TIMEVAL\_T

 $\verb|typedef| struct timeval VOS\_TIMEVAL\_T| \\$ 

Timer value compatible with timeval / select.

Relative or absolute date, depending on usage Assume 32 Bit system, if not defined

# 5.37.3 Enumeration Type Documentation

# 5.37.3.1 VOS\_ERR\_T

enum VOS\_ERR\_T

Return codes for all VOS API functions.

#### Enumerator

| VOS_NO_ERR    | No error.                                    |
|---------------|--|
| VOS_PARAM_ERR | Necessary parameter missing or out of range. |
| VOS_INIT_ERR  | Call without valid initialization.           |

# Enumerator

| VOS_NOINIT_ERR      | The supplied handle/reference is not valid.       |
|---------------------|---|
| VOS_TIMEOUT_ERR     | Timout.   |
| VOS_NODATA_ERR      | Non blocking mode: no data received.              |
| VOS_SOCK_ERR        | Socket option not supported.                      |
| VOS_IO_ERR          | Socket IO error, data can't be received/sent.     |
| VOS_MEM_ERR         | No more memory available.                         |
| VOS_SEMA_ERR        | Semaphore not available.                          |
| VOS_QUEUE_ERR       | Queue empty.                                      |
| VOS_QUEUE_FULL_ERR  | Queue full.                                       |
| VOS_MUTEX_ERR       | Mutex not available.                              |
| VOS_THREAD_ERR      | Thread creation error.                            |
| VOS_BLOCK_ERR       | System call would have blocked in blocking mode.  |
| VOS_INTEGRATION_ERR | Alignment or endianess for selected target wrong. |
| VOS_NOCONN_ERR      | No TCP connection.                                |
| VOS_INUSE_ERR       | Resource is still in use.                         |
| VOS_UNKNOWN_ERR     | Unknown error.                                    |
|                     |   |

5.37.3.2 VOS\_LOG\_T

enum VOS\_LOG\_T

Categories for logging.

# Enumerator

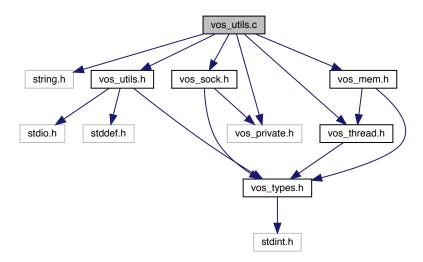
| VOS_LOG_ERROR   | This is a critical error. |
|-----------------|---------------------------|
| VOS_LOG_WARNING | This is a warning.        |
| VOS_LOG_INFO    | This is an info.          |
| VOS_LOG_DBG     | This is a debug info.     |
| VOS_LOG_USR     | This is a user info.      |

# 5.38 vos\_utils.c File Reference

# Common functions for VOS.

```
#include <string.h>
#include "vos_utils.h"
#include "vos_sock.h"
#include "vos_thread.h"
#include "vos_mem.h"
#include "vos_private.h"
```

Include dependency graph for vos\_utils.c:



### **Functions**

- VOS\_ERR\_T vos\_init (void \*pRefCon, VOS\_PRINT\_DBG\_T pDebugOutput)
   Initialize the virtual operating system.
- EXT\_DECL void vos\_terminate (void)

Delnitialize the vos library.

- UINT32 vos\_crc32 (UINT32 crc, const UINT8 \*pData, UINT32 dataLen)

  Compute crc32 according to IEEE802.3.
- UINT32 vos\_sc32 (UINT32 crc, const UINT8 \*pData, UINT32 dataLen)
   Compute crc32 according to IEC 61375-2-3 B.7.
- const char \* vos\_getVersionString (void)

Return a human readable version representation.

• EXT\_DECL const VOS\_VERSION\_T \* vos\_getVersion (void)

Return version.

• EXT\_DECL const CHAR8 \* vos\_getErrorString (VOS\_ERR\_T error)

Return a human readable error representation.

# 5.38.1 Detailed Description

Common functions for VOS.

Common functions of the abstraction layer. Mainly debugging support.

Note

Project: TCNOpen TRDP prototype stack

#### Author

Bernd Loehr, NewTec GmbH

### Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

vos utils.c 1761 2018-09-18 15:54:02Z bloehr

```
BL 2017-05-08: Compiler warnings
BL 2017-02-27: #142 Compiler warnings / MISRA-C 2012 issues
BL 2016-08-17: parentheses added (compiler warning)
BL 2016-07-06: Ticket #122 64Bit compatibility (+ compiler warnings)
BL 2016-03-10: Ticket #114 SC-32
BL 2016-02-10: ifdef DEBUG for some functions
BL 2014-02-28: Ticket #25: CRC32 calculation is not according IEEE802.3
```

# 5.38.2 Function Documentation

```
5.38.2.1 vos_crc32()
```

Compute crc32 according to IEEE802.3.

Calculate CRC for the given buffer and length.

/ to IEC 61375-2-3 A.3 Note: Returned CRC is inverted

#### **Parameters**

| in     | crc     | Initial value.           |
|--------|---------|--------------------------|
| in,out | pData   | Pointer to data.         |
| in     | dataLen | length in bytes of data. |

### Return values

| crc32 | according to |  |
|-------|--------------|--|
|       | IEEE802.3    |  |

# 5.38.2.2 vos\_getErrorString()

Return a human readable error representation.

# **Parameters**

| in | error | The TRDP or VOS error code |
|----|-------|----------------------------|
|----|-------|----------------------------|

# Return values

| const | string pointer to error string |
|-------|--------------------------------|
|-------|--------------------------------|

# 5.38.2.3 vos\_getVersion()

Return version.

Return pointer to version structure

# Return values

```
VOS_VERSION↔
T
```

# 5.38.2.4 vos\_getVersionString()

Return a human readable version representation.

Return string in the form 'v.r.u.b'

# Return values

const string

### 5.38.2.5 vos\_init()

Initialize the virtual operating system.

Initialize the vos library.

#### **Parameters**

| i | .n | pRefCon      | context for debug output function |
|---|----|--------------|-----------------------------------|
| i | n  | pDebugOutput | Pointer to debug output function. |

# Return values

| VOS_NO_ERR | no error VOS_INTEGRATION_ERR if endianess/alignment mismatch VOS_SOCK_ERR |  |
|------------|---|--|
|            | sockets not supported VOS_UNKNOWN_ERR initialisation error                |  |

### 5.38.2.6 vos\_sc32()

Compute crc32 according to IEC 61375-2-3 B.7.

Compute crc32 according to IEC 61375-2-3 B.7 Note: Returned CRC is inverted.

# **Parameters**

| in     | crc     | Initial value.           |
|--------|---------|--------------------------|
| in,out | pData   | Pointer to data.         |
| in     | dataLen | length in bytes of data. |

# Return values

| sc32 | according to IEC 61375-2-3 |
|------|----------------------------|
|------|----------------------------|

# 5.38.2.7 vos\_terminate()

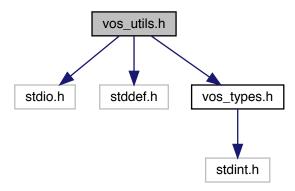
Delnitialize the vos library.

Should be called last after TRDP stack/application does not use any VOS function anymore.

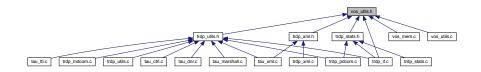
# 5.39 vos\_utils.h File Reference

# Typedefs for OS abstraction.

```
#include <stdio.h>
#include <stddef.h>
#include "vos_types.h"
Include dependency graph for vos_utils.h:
```



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



### **Macros**

- #define VOS\_MAX\_PRNT\_STR\_SIZE 256u
  - String size definitions for the debug output functions.
- #define VOS\_MAX\_FRMT\_SIZE 64u

Max

#define VOS\_MAX\_ERR\_STR\_SIZE (VOS\_MAX\_PRNT\_STR\_SIZE - VOS\_MAX\_FRMT\_SIZE)

Мах.

• #define VOS DIR SEP '/'

This is a helper define for separating a path in debug output.

#define vos\_snprintf(str, size, format, args ...) snprintf(str, size, format, ## args) /\*lint !e586 logging output needed \*/

Safe printf function.

• #define vos\_printLogStr(level, string)

Debug output macro without formatting options.

#define vos\_printLog(level, format, args ...)

Debug output macro with formatting options.

• #define ALIGNOF(type) ((UINT32)offsetof(struct { char c; type member; }, member))

Alignment macros.

#define INITFCS 0xffffffffu

CRC/FCS constants.

• #define SIZE\_OF\_FCS 4u

for better understanding of address calculations

#define L ENDIAN

Define endianess if not already done by compiler.

#### **Functions**

- EXT\_DECL UINT32 vos\_crc32 (UINT32 crc, const UINT8 \*pData, UINT32 dataLen) Calculate CRC for the given buffer and length.
- EXT\_DECL UINT32 vos\_sc32 (UINT32 crc, const UINT8 \*pData, UINT32 dataLen)

Compute crc32 according to IEC 61375-2-3 B.7 Note: Returned CRC is inverted.

- EXT\_DECL VOS\_ERR\_T vos\_init (void \*pRefCon, VOS\_PRINT\_DBG\_T pDebugOutput)

  Initialize the vos library.
- EXT DECL void vos terminate (void)

Delnitialize the vos library.

EXT\_DECL const CHAR8 \* vos\_getVersionString (void)

Return a human readable version representation.

• EXT\_DECL const VOS\_VERSION\_T \* vos\_getVersion (void)

Return version.

EXT\_DECL const CHAR8 \* vos\_getErrorString (VOS\_ERR\_T error)

Return a human readable error representation.

### 5.39.1 Detailed Description

Typedefs for OS abstraction.

Note

Project: TCNOpen TRDP prototype stack

Author

Bernd Loehr, NewTec GmbH

# Remarks

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/. Copyright Bombardier Transportation Inc. or its subsidiaries and others, 2013. All rights reserved.

ld

vos\_utils.h 1763 2018-09-21 16:03:13Z ahweiss

```
BL 2017-05-08: Compiler warnings, doxygen comment errors
BL 2017-02-08: Ticket #142: Compiler warnings / MISRA-C 2012 issues
BL 2016-03-10: Ticket #114 SC-32
BL 2014-02-28: Ticket #25: CRC32 calculation is not according IEEE802.3
```

# 5.39.2 Macro Definition Documentation

```
5.39.2.1 INITFCS
```

```
#define INITFCS 0xffffffffu
```

CRC/FCS constants.

Initial FCS value

### 5.39.2.2 VOS\_MAX\_ERR\_STR\_SIZE

```
#define VOS_MAX_ERR_STR_SIZE (VOS_MAX_PRNT_STR_SIZE - VOS_MAX_FRMT_SIZE)
```

Max.

size of the error part

### 5.39.2.3 VOS\_MAX\_FRMT\_SIZE

```
#define VOS_MAX_FRMT_SIZE 64u
```

Max.

size of the 'format' part

# 5.39.2.4 VOS\_MAX\_PRNT\_STR\_SIZE

```
#define VOS_MAX_PRNT_STR_SIZE 256u
```

String size definitions for the debug output functions.

Max. size of the debug/error string of debug function

# 5.39.3 Function Documentation

### 5.39.3.1 vos\_crc32()

Calculate CRC for the given buffer and length.

For TRDP FCS CRC calculation the CRC32 according to IEEE802.3 with start value 0xffffffff is used.

# **Parameters**

| in     | crc     | Initial value.           |
|--------|---------|--------------------------|
| in,out | pData   | Pointer to data.         |
| in     | dataLen | length in bytes of data. |

# Return values

| crc32 | according to |
|-------|--------------|
|       | IEEE802.3    |

Calculate CRC for the given buffer and length.

/ to IEC 61375-2-3 A.3 Note: Returned CRC is inverted

# **Parameters**

| in     | crc     | Initial value.           |
|--------|---------|--------------------------|
| in,out | pData   | Pointer to data.         |
| in     | dataLen | length in bytes of data. |

### **Return values**

| crc32 | according to |
|-------|--------------|
|       | IEEE802.3    |

# 5.39.3.2 vos\_getErrorString()

Return a human readable error representation.

# **Parameters**

| in | error | The TRDP or VOS error code |
|----|-------|----------------------------|
|----|-------|----------------------------|

# Return values

| const | string pointer to error string |
|-------|--------------------------------|
|-------|--------------------------------|

# 5.39.3.3 vos\_getVersion()

Return version.

Return pointer to version structure

# Return values

```
const VOS_VERSION←
_T
```

Return pointer to version structure

### Return values

```
VOS_VERSION←
_T
```

# 5.39.3.4 vos\_getVersionString()

Return a human readable version representation.

Return string in the form 'v.r.u.b'

# Return values

```
const string
```

# 5.39.3.5 vos\_init()

Initialize the vos library.

This is used to set the output function for all VOS error and debug output.

# **Parameters**

| in | pRefCon      | user context                     |
|----|--------------|----------------------------------|
| in | pDebugOutput | pointer to debug output function |

### Return values

| VOS_NO_ERR | no error |  |  |
|------------|----------|--|--|

# Return values

| VOS_INIT_ERR   ui | nsupported |
|-------------------|------------|
|-------------------|------------|

Initialize the vos library.

### **Parameters**

| in | pRefCon      | context for debug output function |
|----|--------------|-----------------------------------|
| in | pDebugOutput | Pointer to debug output function. |

### Return values

| VOS_NO_ERR | no error VOS_INTEGRATION_ERR if endianess/alignment mismatch VOS_SOCK_ERR |  |
|------------|---|--|
|            | sockets not supported VOS_UNKNOWN_ERR initialisation error                |  |

5.39.3.6 vos\_sc32()

Compute crc32 according to IEC 61375-2-3 B.7 Note: Returned CRC is inverted.

# **Parameters**

| in     | crc     | Initial value.           |
|--------|---------|--------------------------|
| in,out | pData   | Pointer to data.         |
| in     | dataLen | length in bytes of data. |

# Return values

| crc32 | according to IEC 61375-2-3 |
|-------|----------------------------|
|-------|----------------------------|

Compute crc32 according to IEC 61375-2-3 B.7 Note: Returned CRC is inverted.

# **Parameters**

| in     | crc     | Initial value.           |
|--------|---------|--------------------------|
| in,out | pData   | Pointer to data.         |
| in     | dataLen | length in bytes of data. |

### **Return values**

| sc32 | according to IEC 61375-2-3 |
|------|----------------------------|
|      |                            |

# 5.39.3.7 vos\_terminate()

```
\begin{tabular}{ll} EXT\_DECL \ void \ vos\_terminate \ ( \\ void \ ) \end{tabular}
```

DeInitialize the vos library.

Should be called last after TRDP stack/application does not use any VOS function anymore.

# Index

| cnCnt                      | etbTopoCnt, 16                     |
|----------------------------|------------------------------------|
| TRDP_ETB_INFO_T, 33        | inhibit, 16                        |
| cnld                       | isLead, 17                         |
| TRDP_FUNCTION_INFO_T, 34   | leadDir, 17                        |
| confVehCnt                 | leadVehOfCst, 17                   |
| GNU_PACKED, 15             | lifesign, 17                       |
| confVehList                | msgType, 17                        |
| GNU_PACKED, 15             | opCstList, 17                      |
| cstld                      | opTrnDirState, 18                  |
| TRDP_CONSIST_INFO_T, 28    | opTrnTopoCnt, 18                   |
| cstInfoGetPropSize         | opVehList, 18                      |
| tau_cstinfo.c, 68          | ownOpCstNo, 18                     |
| cstList                    | protocolVersion, 18                |
| GNU_PACKED, 15             | reserved01, 19                     |
| cstOwner                   | reserved02, 19                     |
| TRDP_CONSIST_INFO_T, 28    | reserved03, 19                     |
| cstUUID                    | reserved04, 19                     |
| GNU_PACKED, 15             | reserved06, 20                     |
| cstVehNo                   | safetyTrail, 20                    |
| TRDP_FUNCTION_INFO_T, 35   | trnCstNo, 20                       |
|                            | trnDirState, 20                    |
| DNS_HEADER, 9              | trnld, 20                          |
| datasetLength              | trnNetDir, 21                      |
| GNU_PACKED, 16             | trnOperator, 21                    |
| destAddr                   | trnTopoCnt, 21                     |
| TRDP_PUB_STATISTICS_T, 47  | trnVehNo, 21                       |
| deviceName                 | vehld, 21                          |
| GNU_PACKED, 16             | vehOrient, 21                      |
|                            | version, 22                        |
| ETB_CTRL_COMID             |                                    |
| iec61375-2-3.h, 64         | INITFCS                            |
| etbld                      | vos_utils.h, 363                   |
| GNU_PACKED, 16             | iec61375-2-3.h, 59                 |
| TRDP_FUNCTION_INFO_T, 35   | ETB_CTRL_COMID, 64                 |
| etbTopoCnt                 | TRDP_ETBCTRL_DSID, 64              |
| GNU_PACKED, 16             | TRDP_MAX_FILE_NAME_LEN, 64         |
|                            | TRDP_MAX_LABEL_LEN, 64             |
| fctld                      | TRDP_MAX_MD_DATA_SIZE, 64          |
| TRDP_FUNCTION_INFO_T, 35   | TRDP_MAX_URI_HOST_LEN, 64          |
| filterAddr                 | TRDP_MAX_URI_LEN, 65               |
| TRDP_SUBS_STATISTICS_T, 55 | TRDP_MAX_URI_USER_LEN, 65          |
|                            | TRDP_MD_DEFAULT_REPLY_TIMEOUT, 65  |
| GNU_PACKED, 9              | TRDP_MD_INFINITE_TIME, 65          |
| confVehCnt, 15             | TRDP_MIN_PD_HEADER_SIZE, 65        |
| confVehList, 15            | TRDP_MSG_PD, 65                    |
| cstList, 15                | TRDP_PD_UDP_PORT, 66               |
| cstUUID, 15                | TRDP_PROCESS_DEFAULT_CYCLE_TIME, 6 |
| datasetLength, 16          | TRDP_USR_URI_SIZE, 66              |
| deviceName, 16             | TTDB_NET_DIR_REQ_COMID, 66         |
| etbld, 16                  | TTDB_OP_DIR_INFO_COMID, 66         |

| TTDB_STAT_CST_REQ_COMID, 66 | TRDP_DBG_CONFIG_T, 30         |
|-----------------------------|-------------------------------|
| TTDB_TRN_DIR_REQ_COMID, 67  | TRDP_DBG_DEFAULT              |
| inhibit                     | tau_xml.h, 148                |
| GNU_PACKED, 16              | TRDP DNS REPLY, 31            |
| isLead                      | tcnUriCnt, 32                 |
| GNU PACKED, 17              | TRDP_DNS_REQUEST, 32          |
|                             | tcnUriCnt, 33                 |
| leadDir                     | TRDP_ERR_T                    |
| GNU PACKED, 17              |                               |
| leadVehOfCst                | trdp_types.h, 263             |
| GNU PACKED, 17              | TRDP_ETB_INFO_T, 33           |
| lifesign                    | cnCnt, 33                     |
| <u> </u>                    | TRDP_ETBCTRL_DSID             |
| GNU_PACKED, 17              | iec61375-2-3.h, 64            |
| msgType                     | TRDP_EXCHG_OPTION_T           |
| GNU PACKED, 17              | tau_xml.h, 148                |
| GNO_FACKED, 17              | TRDP_FLAGS_DEFAULT            |
| opCstList                   | trdp_types.h, 260             |
| •                           | TRDP FUNCTION INFO T, 34      |
| GNU_PACKED, 17              | cnld, 34                      |
| opTrnDirState               | cstVehNo, 35                  |
| GNU_PACKED, 18              | •                             |
| opTrnTopoCnt                | etbld, 35                     |
| GNU_PACKED, 18              | fctld, 35                     |
| opVehList                   | TRDP_HANDLE, 35               |
| GNU_PACKED, 18              | TRDP_IP_ADDR_T                |
| ownOpCstNo                  | trdp_types.h, 260             |
| GNU PACKED, 18              | TRDP_LIST_STATISTICS_T, 36    |
| _ ,                         | TRDP_MARSHALL_CONFIG_T, 37    |
| PD ELE, 22                  | TRDP MARSHALL T               |
| pFrame, 24                  | trdp_types.h, 260             |
| pFrame                      | TRDP_MAX_FILE_NAME_LEN        |
| PD ELE, 24                  | iec61375-2-3.h, 64            |
| protocolVersion             |                               |
| GNU PACKED, 18              | TRDP_MAX_LABEL_LEN            |
| GINO_I AGINED, IO           | iec61375-2-3.h, 64            |
| reserved01                  | TRDP_MAX_MD_DATA_SIZE         |
| GNU_PACKED, 19              | iec61375-2-3.h, 64            |
|                             | TRDP_MAX_URI_HOST_LEN         |
| reserved02                  | iec61375-2-3.h, 64            |
| GNU_PACKED, 19              | TRDP_MAX_URI_LEN              |
| reserved03                  | iec61375-2-3.h, 65            |
| GNU_PACKED, 19              | TRDP_MAX_URI_USER_LEN         |
| reserved04                  | iec61375-2-3.h, 65            |
| GNU_PACKED, 19              | TRDP_MD_CALLBACK_T            |
| reserved06                  | trdp types.h, 261             |
| GNU_PACKED, 20              | TRDP MD CONFIG T, 38          |
|                             | <i>_ ·</i>                    |
| safetyTrail                 | TRDP_MD_DEFAULT_REPLY_TIMEOUT |
| GNU_PACKED, 20              | iec61375-2-3.h, 65            |
|                             | TRDP_MD_ELE_ST_T              |
| TAU_MARSHALL_INFO_T, 24     | trdp_private.h, 245           |
| TCN_URI, 25                 | TRDP_MD_INFINITE_TIME         |
| TRDP_CLTR_CST_INFO_T, 25    | iec61375-2-3.h, 65            |
| TRDP_COMID_DSID_MAP_T, 26   | TRDP_MD_INFO_T, 39            |
| TRDP_CONSIST_INFO_T, 26     | TRDP MD STATISTICS T, 40      |
| cstld, 28                   | TRDP MEM CONFIG T, 41         |
| cstOwner, 28                | TRDP MEM STATISTICS T, 42     |
| TRDP_DATA_TYPE_T            | TRDP_MIN_PD_HEADER_SIZE       |
|                             | iec61375-2-3.h, 65            |
| trdp_types.h, 262           |                               |
| TRDP_DATASET_ELEMENT_T, 29  | TRDP_MSG_PD                   |
| TRDP_DATASET, 28            | iec61375-2-3.h, 65            |

| TRDP_PD_CALLBACK_T              | tau_DNRstatus                            |
|---------------------------------|--|
| trdp_types.h, 261               | tau_dnr.c, 85                            |
| TRDP_PD_CONFIG_T, 42            | tau_dnr.h, 91                            |
| TRDP_PD_INFO_T, 43              | tau_addr2Uri                             |
| TRDP_PD_STATISTICS_T, 44        | tau_dnr.c, 84                            |
| TRDP_PD_UDP_PORT                | tau_dnr.h, 89                            |
| iec61375-2-3.h, 66              | tau_calcDatasetSize                      |
| TRDP_PRINT_DBG_T                | tau_marshall.c, 97                       |
| trdp_types.h, 261               | tau_marshall.h, 104                      |
| TRDP_PROCESS_CONFIG_T, 45       | tau_calcDatasetSizeByComId               |
| TRDP_PROCESS_DEFAULT_CYCLE_TIME | tau marshall.c, 98                       |
| iec61375-2-3.h, 66              | tau_marshall.h, 105                      |
| TRDP_PROP_T, 46                 | tau_cstinfo.c, 67                        |
| TRDP_PUB_STATISTICS_T, 46       | cstInfoGetPropSize, 68                   |
| destAddr, 47                    | tau_ctrl.c, 69                           |
| TRDP_RED_STATE_T                | tau_getEcspStat, 71                      |
| trdp_types.h, 264               | tau_initEcspCtrl, 71                     |
| TRDP RED STATISTICS T, 47       | tau_requestEcspConfirm, 73               |
| TRDP REPLY STATUS T             | tau_setEcspCtrl, 73                      |
| trdp_types.h, 264               | tau_terminateEcspCtrl, 74                |
| TRDP SDT DEFAULT CMTHR          | tau_ctrl.h, 74                           |
| tau xml.c, 141                  | tau_getEcspStat, 77                      |
| TRDP_SDT_PAR_T, 47              | tau_getEcspStat, 77 tau_initEcspCtrl, 77 |
|                                 | <del>-</del> · ·                         |
| TRDP_SEND_PARAM_T, 48           | tau_requestEcspConfirm, 78               |
| TRDP_SEQ_CNT_ENTRY_T, 49        | tau_setEcspCtrl, 78                      |
| TRDP_SESSION, 49                | tau_terminateEcspCtrl, 79                |
| TRDP_SOCK_TYPE_T                | tau_ctrl_types.h, 79                     |
| trdp_private.h, 245             | tau_deInitDnr                            |
| TRDP_SOCKET_TCP, 51             | tau_dnr.c, 84                            |
| TRDP_SOCKETS, 51                | tau_dnr.h, 90                            |
| usage, 52                       | tau_deInitTTI                            |
| TRDP_STATISTICS_REQUEST_T, 52   | tau_tti.c, 114                           |
| TRDP_STATISTICS_T, 53           | tau_tti.h, 126                           |
| TRDP_SUBS_STATISTICS_T, 54      | tau_dnr.c, 82                            |
| filterAddr, 55                  | tau_DNRstatus, 85                        |
| timeout, 55                     | tau_addr2Uri, 84                         |
| toBehav, 55                     | tau_deInitDnr, 84                        |
| TRDP_TIME_T                     | tau_getOwnAddr, 85                       |
| trdp_types.h, 261               | tau_initDnr, 86                          |
| TRDP_TO_BEHAVIOR_T              | tau_uri2Addr, 86                         |
| trdp_types.h, 264               | tau_dnr.h, 87                            |
| TRDP_UNMARSHALL_T               | tau_DNRstatus, 91                        |
| trdp_types.h, 261               | tau_addr2Uri, 89                         |
| TRDP_USR_URI_SIZE               | tau_deInitDnr, 90                        |
| iec61375-2-3.h, 66              | tau_getOwnAddr, 91                       |
| TRDP_VEHICLE_INFO_T, 56         | tau_initDnr, 91                          |
| vehld, 57                       | tau_uri2Addr, 92                         |
| TRDP_XML_DOC_HANDLE_T, 57       | tau_dnr_types.h, 93                      |
| TTDB_NET_DIR_REQ_COMID          | tau_freeTelegrams                        |
| iec61375-2-3.h, 66              | tau_xml.c, 141                           |
| TTDB_OP_DIR_INFO_COMID          | tau_xml.h, 149                           |
| iec61375-2-3.h, 66              | tau_freeXmlDatasetConfig                 |
| TTDB_STAT_CST_REQ_COMID         | tau_xml.c, 141                           |
| iec61375-2-3.h, 66              | tau_xml.h, 149                           |
| TTDB_TRN_DIR_REQ_COMID          | tau_freeXmlDoc                           |
| iec61375-2-3.h, 67              | tau_xml.c, 142                           |
| TTI_CACHED_CONSISTS             | tau xml.h, 150                           |
| tau tti.c, 114                  | tau_getCstFctCnt                         |
| _ ,                             |  |

| tau_tti.c, 115                            | tau_tti.h, 135                            |
|---|---|
| tau_tti.h, 127                            | tau_marshall                              |
| tau_getCstFctInfo                         | tau_marshall.c, 99                        |
| tau_tti.c, 115                            | tau_marshall.h, 107                       |
| tau_tti.h, 127                            | tau_marshall.c, 96                        |
| tau_getCstInfo                            | tau_calcDatasetSize, 97                   |
| tau_tti.c, 116                            | tau_calcDatasetSizeByComld, 98            |
| tau_tti.h, 128                            | tau_initMarshall, 99                      |
| tau_getCstVehCnt                          | tau_marshall, 99                          |
| tau_tti.c, 116                            | tau_marshallDs, 100                       |
| tau_tti.h, 128                            | tau_unmarshall, 101                       |
| tau_getEcspStat<br>tau_ctrl.c, 71         | tau_unmarshallDs, 101 tau_marshall.h, 102 |
| tau_ctrl.h, 77                            | tau_calcDatasetSize, 104                  |
| tau_ctiff, 77 tau_getOpTrDirectory        | tau_calcDatasetSizeByComld, 105           |
| tau_tti.c, 117                            | tau_initMarshall, 106                     |
| tau_tti.h, 129                            | tau_marshall, 107                         |
| tau getOpTrnDirectoryStatusInfo           | tau_marshallDs, 108                       |
| tau_tti.c, 117                            | tau unmarshall, 110                       |
| tau_tti.h, 130                            | tau unmarshallDs, 111                     |
| tau_getOwnAddr                            | tau marshallDs                            |
| tau dnr.c, 85                             | tau_marshall.c, 100                       |
| tau dnr.h, 91                             | tau_marshall.h, 108                       |
| tau_getOwnIds                             | tau_prepareXmlDoc                         |
| tau_tti.c, 117                            | tau xml.c, 142                            |
| tau_tti.h, 130                            | tau_xml.h, 150                            |
| tau_getStaticCstInfo                      | tau_readXmlDatasetConfig                  |
| tau_tti.c, 118                            | tau xml.c, 142                            |
| tau_tti.h, 131                            | tau_xml.h, 151                            |
| tau_getTTI                                | tau_readXmlDeviceConfig                   |
| tau_tti.c, 120                            | tau_xml.c, 144                            |
| tau_tti.h, 133                            | tau_xml.h, 151                            |
| tau_getTrDirectory                        | tau_readXmlInterfaceConfig                |
| tau_tti.c, 118                            | tau_xml.c, 144                            |
| tau_tti.h, 131                            | tau_xml.h, 152                            |
| tau_getTrnCstCnt                          | tau_requestEcspConfirm                    |
| tau_tti.c, 119                            | tau_ctrl.c, 73                            |
| tau_tti.h, 132                            | tau_ctrl.h, 78                            |
| tau_getTrnVehCnt                          | tau_setEcspCtrl                           |
| tau_tti.c, 119                            | tau_ctrl.c, 73                            |
| tau_tti.h, 132                            | tau_ctrl.h, 78                            |
| tau_getVehInfo                            | tau_terminateEcspCtrl                     |
| tau_tti.c, 120                            | tau_ctrl.c, 74                            |
| tau_tti.h, 134                            | tau_ctrl.h, 79                            |
| tau_getVehOrient                          | tau_tti.c, 112                            |
| tau_tti.c, 121                            | TTI_CACHED_CONSISTS, 114                  |
| tau_tti.h, 134                            | tau_delnitTTI, 114                        |
| tau_initDnr                               | tau_getCstFctCnt, 115                     |
| tau_dnr.c, 86                             | tau_getCstFctInfo, 115                    |
| tau_dnr.h, 91                             | tau_getCstInfo, 116                       |
| tau_initEcspCtrl                          | tau_getCstVehCnt, 116                     |
| tau_ctrl.c, 71                            | tau_getOpTrDirectory, 117                 |
| tau_ctrl.h, 77                            | tau_getOpTrnDirectoryStatusInfo, 117      |
| tau_initMarshall                          | tau_getOwnlds, 117                        |
| tau_marshall.c, 99<br>tau_marshall.h, 106 | tau_getStaticCstInfo, 118 tau_getTTI, 120 |
| tau_initTTlaccess                         | tau_getTrDirectory, 118                   |
| tau_mit r raccess tau_tti.c, 121          | tau_getTrnCstCnt, 119                     |
| tau_tii.0, 121                            | tau_got initostoni, 117                   |

| tau_getTrnVehCnt, 119                      | tlc_configSession                      |
|--|--|
| tau_getVehInfo, 120                        | trdp_if.c, 158                         |
| tau_getVehOrient, 121                      | trdp_if_light.h, 180                   |
| tau_initTTlaccess, 121                     | tlc_freeBuf                            |
| tau_tti.h, 123                             | trdp_if_light.h, 181                   |
| tau_deInitTTI, 126                         | tlc_getETBTopoCount                    |
| tau_getCstFctCnt, 127                      | trdp_if.c, 158                         |
| tau_getCstFctInfo, 127                     | trdp_if_light.h, 181                   |
| tau_getCstInfo, 128                        | tlc_getInterval                        |
| tau_getCstVehCnt, 128                      | trdp_if.c, 159                         |
| tau_getOpTrDirectory, 129                  | trdp_if_light.h, 181                   |
| tau_getOpTrnDirectoryStatusInfo, 130       | tlc_getJoinStatistics                  |
| tau_getOwnlds, 130                         | trdp_if_light.h, 182                   |
| tau_getStaticCstInfo, 131                  | trdp_stats.c, 247                      |
| tau_getTTI, 133                            | tlc_getOpTrainTopoCount                |
| tau_getTrDirectory, 131                    | trdp_if.c, 159                         |
| tau_getTrnCstCnt, 132                      | trdp_if_light.h, 183                   |
| tau_getTrnVehCnt, 132                      | tlc_getOwnlpAddress                    |
| tau_getVehInfo, 134                        | trdp_if.c, 159                         |
| tau_getVehOrient, 134                      | trdp_if_light.h, 183                   |
| tau_initTTlaccess, 135                     | tlc_getPubStatistics                   |
| tau_tti_types.h, 136 tau_unmarshall        | trdp_if_light.h, 184                   |
|  | trdp_stats.c, 248 tlc getRedStatistics |
| tau_marshall.c, 101<br>tau_marshall.h, 110 | trdp_if_light.h, 185                   |
| tau unmarshallDs                           | trdp_n_ngnt.n, 765                     |
| tau_marshall.c, 101                        | tlc_getStatistics                      |
| tau_marshall.h, 111                        | trdp_if_light.h, 185                   |
| tau_uri2Addr                               | trdp_n_ngnt.n, 765                     |
| tau_dnr.c, 86                              | tlc_getSubsStatistics                  |
| tau_dnr.h, 92                              | trdp_if_light.h, 186                   |
| tau_xml.c, 139                             | trdp_stats.c, 249                      |
| TRDP_SDT_DEFAULT_CMTHR, 141                | tlc_getTcpListStatistics               |
| tau_freeTelegrams, 141                     | trdp_if_light.h, 187                   |
| tau_freeXmlDatasetConfig, 141              | tlc_getUdpListStatistics               |
| tau_freeXmlDoc, 142                        | trdp_if_light.h, 187                   |
| tau prepareXmlDoc, 142                     | tlc_getVersion                         |
| tau_readXmlDatasetConfig, 142              | trdp_if.c, 160                         |
| tau_readXmlDeviceConfig, 144               | trdp if light.h, 188                   |
| tau_readXmlInterfaceConfig, 144            | tlc_getVersionString                   |
| tau_xml.h, 145                             | trdp_if.c, 160                         |
| TRDP_DBG_DEFAULT, 148                      | trdp_if_light.h, 188                   |
| TRDP_EXCHG_OPTION_T, 148                   | tlc_init                               |
| tau_freeTelegrams, 149                     | trdp_if.c, 160                         |
| tau_freeXmlDatasetConfig, 149              | trdp_if_light.h, 189                   |
| tau_freeXmlDoc, 150                        | tlc_openSession                        |
| tau_prepareXmlDoc, 150                     | trdp_if.c, 161                         |
| tau_readXmlDatasetConfig, 151              | trdp_if_light.h, 190                   |
| tau_readXmlDeviceConfig, 151               | tlc_process                            |
| tau_readXmIInterfaceConfig, 152            | trdp_if.c, 162                         |
| tcnUriCnt                                  | trdp_if_light.h, 190                   |
| TRDP_DNS_REPLY, 32                         | tlc_reinitSession                      |
| TRDP_DNS_REQUEST, 33                       | trdp_if.c, 162                         |
| timeout                                    | trdp_if_light.h, 191                   |
| TRDP_SUBS_STATISTICS_T, 55                 | tlc_resetStatistics                    |
| tlc_closeSession                           | trdp_if_light.h, 191                   |
| trdp_if.c, 157                             | trdp_stats.c, 250                      |
| trdp_if_light.h, 179                       | tlc_setETBTopoCount                    |

| trdp_if.c, 163                         | trdp_if_light.h, 210                        |
|--|---|
| trdp_if_light.h, 192                   | toBehav                                     |
| tlc_setOpTrainTopoCount                | TRDP_SUBS_STATISTICS_T, 55                  |
| trdp_if.c, 163                         | trdp_UpdateStats                            |
| trdp_if_light.h, 192                   | trdp_stats.c, 251                           |
| tlc_terminate                          | trdp_XMLClose                               |
| trdp_if.c, 164                         | trdp_xml.c, 291                             |
| trdp_if_light.h, 193                   | trdp_xml.h, 297                             |
| tlm_abortSession                       | trdp_XMLCountStartTag                       |
| trdp_if_light.h, 193                   | trdp_xml.c, 292                             |
| tlm_addListener                        | trdp_xml.h, 297                             |
| trdp_if_light.h, 194                   | trdp_XMLEnter                               |
| tlm_confirm                            | trdp_xml.c, 292                             |
| trdp_if_light.h, 195                   | trdp_xml.h, 298                             |
| tlm_delListener                        | trdp_XMLGetAttribute                        |
| trdp_if_light.h, 195                   | trdp_xml.c, 292                             |
| tlm_notify                             | trdp_xml.h, 298                             |
| trdp_if_light.h, 196                   | trdp_XMLLeave                               |
| tlm_readdListener                      | trdp_xml.c, 293                             |
| trdp_if_light.h, 197                   | trdp_xml.h, 298                             |
| tlm_reply                              | trdp_XMLOpen                                |
| trdp_if_light.h, 197                   | trdp_xml.c, 293                             |
| tlm_replyQuery                         | trdp_xml.h, 299                             |
| trdp_if_light.h, 198                   | trdp_XMLRewind                              |
| tlm_request                            | trdp_xml.c, 294                             |
| trdp_if_light.h, 199                   | trdp_xml.h, 299                             |
| tlp_get<br>trdp_if.c, 164              | trdp_XMLSeekStartTag<br>trdp_xml.c, 294     |
| trdp_ii.c, 104<br>trdp_if_light.h, 200 | trdp_xml.h, 299                             |
| tlp_getRedundant                       | trdp_XMLSeekStartTagAny                     |
| trdp_if.c, 165                         | trdp_xml.c, 294                             |
| trdp_if.c, 100<br>trdp_if_light.h, 201 | trdp_xml.h, 300                             |
| tlp_publish                            | trdp_xmin, 500<br>trdp_checkSequenceCounter |
| trdp_if.c, 165                         | trdp_utils.c, 267                           |
| trdp_ir.e, 100<br>trdp_if_light.h, 201 | trdp_utils.h, 279                           |
| tlp_put                                | trdp_dllmain.c, 153                         |
| trdp_if.c, 166                         | trdp_findMCjoins                            |
| trdp_if_light.h, 202                   | trdp_utils.c, 268                           |
| tlp_republish                          | trdp_utils.h, 280                           |
| trdp_if.c, 167                         | trdp_getCurrentMaxSocketCnt                 |
| trdp_if_light.h, 203                   | trdp_utils.c, 268                           |
| tlp_request                            | trdp_utils.h, 280                           |
| trdp_if.c, 167                         | trdp getSeqCnt                              |
| trdp_if_light.h, 204                   | trdp utils.c, 268                           |
| tlp resubscribe                        | trdp_utils.h, 281                           |
| trdp if.c, 168                         | trdp_if.c, 154                              |
| trdp_if_light.h, 206                   | tlc_closeSession, 157                       |
| tlp setRedundant                       | tlc_configSession, 158                      |
| trdp_if.c, 169                         | tlc_getETBTopoCount, 158                    |
| trdp_if_light.h, 207                   | tlc getInterval, 159                        |
| tlp subscribe                          | tlc_getOpTrainTopoCount, 159                |
| trdp_if.c, 169                         | tlc_getOwnlpAddress, 159                    |
| trdp_if_light.h, 207                   | tlc_getVersion, 160                         |
| tlp_unpublish                          | tlc_getVersionString, 160                   |
| trdp_if.c, 170                         | tlc_init, 160                               |
| trdp_if_light.h, 209                   | tlc_openSession, 161                        |
| tlp_unsubscribe                        | tlc_process, 162                            |
| trdp_if.c, 171                         | tlc_reinitSession, 162                      |
| • —                                    |   |

|          | tlc_setETBTopoCount, 163      | tlp_request, 204             |
|----------|-------------------------------|------------------------------|
|          | tlc_setOpTrainTopoCount, 163  | tlp_resubscribe, 206         |
|          | tlc_terminate, 164            | tlp_setRedundant, 207        |
|          | tlp_get, 164                  | tlp_subscribe, 207           |
|          | tlp_getRedundant, 165         | tlp_unpublish, 209           |
|          | tlp_publish, 165              | tlp_unsubscribe, 210         |
|          | tlp_put, 166                  | trdp_initSockets             |
|          | tlp republish, 167            | trdp_utils.c, 269            |
|          | tlp request, 167              | trdp_utils.h, 281            |
|          | tlp_resubscribe, 168          | trdp_initStats               |
|          | tlp_setRedundant, 169         | trdp_stats.c, 250            |
|          | tlp_subscribe, 169            | trdp_stats.h, 253            |
|          | tlp_unpublish, 170            | trdp_initUncompletedTCP      |
|          | tlp_unsubscribe, 171          | trdp_utils.h, 282            |
|          | •                             | •                            |
|          | trdp_isValidSession, 171      | trdp_isAddressed             |
| A.c.alca | trdp_sessionQueue, 172        | trdp_utils.c, 269            |
| trap_    | _if.h, 172                    | trdp_utils.h, 282            |
|          | trdp_isValidSession, 174      | trdp_isInIPrange             |
|          | trdp_sessionQueue, 175        | trdp_utils.c, 270            |
| trdp_    | _iflight.h, 175               | trdp_utils.h, 282            |
|          | tlc_closeSession, 179         | trdp_isValidSession          |
|          | tlc_configSession, 180        | trdp_if.c, 171               |
|          | tlc_freeBuf, 181              | trdp_if.h, 174               |
|          | tlc_getETBTopoCount, 181      | trdp_mdCall                  |
|          | tlc_getInterval, 181          | trdp_mdcom.c, 213            |
|          | tlc_getJoinStatistics, 182    | trdp_mdcom.h, 220            |
|          | tlc_getOpTrainTopoCount, 183  | trdp_mdCheckListenSocks      |
|          | tlc_getOwnlpAddress, 183      | trdp_mdcom.c, 214            |
|          | tlc_getPubStatistics, 184     | trdp_mdcom.h, 221            |
|          | tlc_getRedStatistics, 185     | trdp_mdCheckPending          |
|          | tlc_getStatistics, 185        | trdp_mdcom.c, 214            |
|          | tlc_getSubsStatistics, 186    | trdp_mdcom.h, 221            |
|          | tlc_getTcpListStatistics, 187 | trdp_mdCheckTimeouts         |
|          | tlc_getUdpListStatistics, 187 | trdp_mdcom.c, 214            |
|          | tlc getVersion, 188           | trdp_mdcom.h, 221            |
|          | tlc_getVersionString, 188     | trdp_mdConfirm               |
|          |                               | · —                          |
|          | tlc_init, 189                 | trdp_mdcom.c, 215            |
|          | tlc_openSession, 190          | trdp_mdcom.h, 222            |
|          | tlc_process, 190              | trdp_mdFreeSession           |
|          | tlc_reinitSession, 191        | trdp_mdcom.c, 215            |
|          | tlc_resetStatistics, 191      | trdp_mdcom.h, 222            |
|          | tlc_setETBTopoCount, 192      | trdp_mdGetTCPSocket          |
|          | tlc_setOpTrainTopoCount, 192  | trdp_mdcom.c, 216            |
|          | tlc_terminate, 193            | trdp_mdcom.h, 223            |
|          | tlm_abortSession, 193         | trdp_mdReply                 |
|          | tlm_addListener, 194          | trdp_mdcom.c, 216            |
|          | tlm_confirm, 195              | trdp_mdcom.h, 223            |
|          | tlm_delListener, 195          | trdp_mdSend                  |
|          | tlm_notify, 196               | trdp_mdcom.c, 217            |
|          | tlm readdListener, 197        | trdp_mdcom.h, 224            |
|          | tlm_reply, 197                | trdp mdcom.c, 210            |
|          | tlm_replyQuery, 198           | trdp_mdCall, 213             |
|          | tlm_request, 199              | trdp_mdCheckListenSocks, 214 |
|          | tlp_get, 200                  | trdp mdCheckPending, 214     |
|          | tlp_getRedundant, 201         | trdp_mdCheckTimeouts, 214    |
|          | tlp_publish, 201              | trdp_mdConfirm, 215          |
|          | tlp_put, 202                  | trdp_mdGommm, 215            |
|          | • —•                          | . —                          |
|          | tlp_republish, 203            | trdp_mdGetTCPSocket, 216     |

| trdp_mdReply, 216            | trdp_pdDistribute, 228       |
|------------------------------|------------------------------|
| trdp_mdSend, 217             | trdp_pdHandleTimeOuts, 229   |
| trdp_mdcom.h, 217            | trdp_pdInit, 229             |
| trdp_mdCall, 220             | trdp_pdPut, 230              |
| trdp_mdCheckListenSocks, 221 | trdp_pdReceive, 231          |
| trdp_mdCheckPending, 221     | trdp_pdSend, 231             |
| trdp_mdCheckTimeouts, 221    | trdp_pdSendQueued, 232       |
| trdp_mdConfirm, 222          | trdp_pdUpdate, 232           |
| trdp_mdFreeSession, 222      | trdp_pdcom.h, 233            |
| trdp_mdGetTCPSocket, 223     | trdp_pdCheck, 235            |
| trdp_mdReply, 223            | trdp_pdCheckListenSocks, 235 |
| trdp_mdSend, 224             | trdp_pdCheckPending, 236     |
| trdp_packetSizeMD            | trdp_pdDistribute, 236       |
| trdp_utils.c, 270            | trdp_pdHandleTimeOuts, 237   |
| trdp_utils.h, 284            | trdp_pdInit, 238             |
| trdp_packetSizePD            | trdp_pdPut, 238              |
| trdp_utils.c, 270            | trdp_pdReceive, 239          |
| trdp_utils.h, 284            | trdp_pdSend, 240             |
| trdp_pdCheck                 | trdp_pdSendQueued, 240       |
| trdp_pdcom.c, 227            | trdp_pdUpdate, 241           |
| trdp_pdcom.h, 235            | trdp_private.h, 241          |
| trdp_pdCheckListenSocks      | TRDP_MD_ELE_ST_T, 245        |
| trdp_pdcom.c, 227            | TRDP_SOCK_TYPE_T, 245        |
| trdp_pdcom.h, 235            | trdp_queueAppLast            |
| trdp_pdCheckPending          | trdp_utils.c, 271            |
| trdp_pdcom.c, 228            | trdp_utils.h, 285            |
| trdp_pdcom.h, 236            | trdp_queueDelElement         |
| trdp_pdDistribute            | trdp_utils.c, 271            |
| trdp_pdcom.c, 228            | trdp_utils.h, 285            |
| trdp_pdcom.h, 236            | trdp_queueFindComId          |
| trdp_pdHandleTimeOuts        | trdp_utils.c, 271            |
| trdp_pdcom.c, 229            | trdp_utils.h, 285            |
| trdp_pdcom.h, 237            | trdp_queueFindPubAddr        |
| trdp_pdInit                  | trdp utils.c, 273            |
| trdp_pdcom.c, 229            | trdp_utils.h, 286            |
| trdp_pdcom.h, 238            | trdp_queueFindSubAddr        |
| trdp_pdPrepareStats          | trdp_utils.c, 273            |
| trdp_stats.c, 251            | trdp_utils.h, 286            |
| trdp_stats.h, 254            | trdp queueInsFirst           |
| trdp_pdPut                   | trdp_utils.c, 274            |
| trdp_pdcom.c, 230            | trdp_utils.h, 287            |
| trdp_pdcom.h, 238            | trdp_releaseSocket           |
| trdp_pdReceive               | trdp_utils.c, 274            |
| trdp_pdcom.c, 231            | trdp_utils.h, 287            |
| trdp_pdcom.h, 239            | trdp_requestSocket           |
| trdp_pdSend                  | trdp_utils.c, 274            |
| trdp_pdcom.c, 231            | trdp_utils.h, 288            |
| trdp_pdcom.h, 240            | trdp_resetSequenceCounter    |
| trdp_pdSendQueued            | trdp_utils.c, 275            |
| trdp_pdcom.c, 232            | trdp_utils.h, 289            |
| trdp_pdcom.h, 240            | trdp_sessionQueue            |
| trdp_pdUpdate                | trdp_if.c, 172               |
| trdp_pdcom.c, 232            | trdp_if.h, 175               |
| trdp_pdcom.h, 241            | trdp_stats.c, 245            |
| trdp_pdcom.c, 224            | tlc_getJoinStatistics, 247   |
| trdp_pdCheck, 227            | tlc_getPubStatistics, 248    |
| trdp_pdCheckListenSocks, 227 | tlc_getRedStatistics, 248    |
| trdp_pdCheckPending, 228     | tlc_getStatistics, 249       |

|       | tlc_getSubsStatistics, 249       | trdp_queueInsFirst, 287        |
|-------|----------------------------------|--------------------------------|
|       | tlc_resetStatistics, 250         | trdp_releaseSocket, 287        |
|       | trdp UpdateStats, 251            | trdp_requestSocket, 288        |
|       | trdp_initStats, 250              | trdp_resetSequenceCounter, 289 |
|       | trdp_pdPrepareStats, 251         | trdp_validTopoCounters, 289    |
| trdn  | stats.h, 252                     | trdp_validTopoCounters         |
| up_   | trdp_initStats, 253              | trdp_utils.c, 276              |
|       | trdp_pdPrepareStats, 254         | trdp_utils.h, 289              |
| trdn  | types.h, 254                     | trdp_xml.c, 290                |
| trup_ |                                  | trdp_XMLClose, 291             |
|       | TRDP_DATA_TYPE_T, 262            | trdp_XMLCountStartTag, 292     |
|       | TRDP_ERR_T, 263                  | • —                            |
|       | TRDP_FLAGS_DEFAULT, 260          | trdp_XMLEnter, 292             |
|       | TRDP_IP_ADDR_T, 260              | trdp_XMLGetAttribute, 292      |
|       | TRDP_MARSHALL_T, 260             | trdp_XMLLeave, 293             |
|       | TRDP_MD_CALLBACK_T, 261          | trdp_XMLOpen, 293              |
|       | TRDP_PD_CALLBACK_T, 261          | trdp_XMLRewind, 294            |
|       | TRDP_PRINT_DBG_T, 261            | trdp_XMLSeekStartTag, 294      |
|       | TRDP_RED_STATE_T, 264            | trdp_XMLSeekStartTagAny, 294   |
|       | TRDP_REPLY_STATUS_T, 264         | trdp_xml.h, 295                |
|       | TRDP_TIME_T, 261                 | trdp_XMLClose, 297             |
|       | TRDP_TO_BEHAVIOR_T, 264          | trdp_XMLCountStartTag, 297     |
|       | TRDP UNMARSHALL T, 261           | trdp_XMLEnter, 298             |
| trdn  | utils.c, 265                     | trdp_XMLGetAttribute, 298      |
| up_   | trdp_checkSequenceCounter, 267   | trdp_XMLLeave, 298             |
|       | trdp_findMCjoins, 268            | trdp_XMLOpen, 299              |
|       | . —                              | trdp XMLRewind, 299            |
|       | trdp_getCurrentMaxSocketCnt, 268 | trdp_XMLSeekStartTag, 299      |
|       | trdp_getSeqCnt, 268              | trdp_XMLSeekStartTagAny, 300   |
|       | trdp_initSockets, 269            | trnCstNo                       |
|       | trdp_isAddressed, 269            | GNU_PACKED, 20                 |
|       | trdp_isInIPrange, 270            | trnDirState                    |
|       | trdp_packetSizeMD, 270           |                                |
|       | trdp_packetSizePD, 270           | GNU_PACKED, 20                 |
|       | trdp_queueAppLast, 271           | trnld                          |
|       | trdp_queueDelElement, 271        | GNU_PACKED, 20                 |
|       | trdp_queueFindComId, 271         | trnNetDir                      |
|       | trdp_queueFindPubAddr, 273       | GNU_PACKED, 21                 |
|       | trdp_queueFindSubAddr, 273       | trnOperator                    |
|       | trdp queuelnsFirst, 274          | GNU_PACKED, 21                 |
|       | trdp_releaseSocket, 274          | trnTopoCnt                     |
|       | trdp_requestSocket, 274          | GNU_PACKED, 21                 |
|       | trdp_resetSequenceCounter, 275   | trnVehNo                       |
|       | trdp_validTopoCounters, 276      | GNU_PACKED, 21                 |
| trdn  | utils.h, 276                     |                                |
| iiup_ |                                  | usage                          |
|       | trdp_checkSequenceCounter, 279   | TRDP_SOCKETS, 52               |
|       | trdp_findMCjoins, 280            |                                |
|       | trdp_getCurrentMaxSocketCnt, 280 | VOS_ERR_T                      |
|       | trdp_getSeqCnt, 281              | vos_types.h, 355               |
|       | trdp_initSockets, 281            | VOS_LOG_T                      |
|       | trdp_initUncompletedTCP, 282     | vos_types.h, 356               |
|       | trdp_isAddressed, 282            | VOS_MAX_ERR_STR_SIZE           |
|       | trdp_isInIPrange, 282            | vos_utils.h, 363               |
|       | trdp_packetSizeMD, 284           | VOS_MAX_FRMT_SIZE              |
|       | trdp_packetSizePD, 284           | vos_utils.h, 363               |
|       | trdp_queueAppLast, 285           | VOS_MAX_PRNT_STR_SIZE          |
|       | trdp_queueDelElement, 285        | vos_utils.h, 363               |
|       | trdp_queueFindComId, 285         | VOS_MAX_SOCKET_CNT             |
|       | trdp_queueFindPubAddr, 286       | vos_sock.h, 325                |
|       | trdp_queueFindSubAddr, 286       | VOS_MEM_BLOCKSIZES             |
|       | • — •                            |                                |

| vos_mem.h, 311          | vos_sock.h, 327          |
|-------------------------|--------------------------|
| VOS_MEM_PREALLOCATE     | vos_htons                |
| vos_mem.h, 312          | vos_sock.h, 327          |
| VOS_PRINT_DBG_T         | vos_init                 |
| vos_types.h, 355        | vos_utils.c, 359         |
| VOS_SOCK_OPT_T, 57      | vos_utils.h, 365         |
| VOS_TIMEVAL_T           | vos_ipDotted             |
| vos_types.h, 355        | vos_sock.h, 328          |
| VOS_TTL_MULTICAST       | vos_isMulticast          |
| vos_sock.h, 325         | vos_sock.h, 328          |
| VOS_VERSION_T, 58       | vos_mem.c, 300           |
| vehld                   | vos_bsearch, 302         |
| GNU_PACKED, 21          | vos_memAlloc, 303        |
| TRDP_VEHICLE_INFO_T, 57 | vos_memCount, 303        |
| vehOrient               | vos_memDelete, 304       |
| GNU_PACKED, 21          | vos_memFree, 304         |
| version                 | vos_memInit, 304         |
| GNU_PACKED, 22          | vos_qsort, 305           |
| vos_addTime             | vos_queueCreate, 305     |
| vos_thread.h, 342       | vos_queueDestroy, 306    |
| vos_bsearch             | vos_queueReceive, 306    |
| vos_mem.c, 302          | vos_queueSend, 307       |
| vos_mem.h, 312          | vos_strncat, 308         |
| vos_clearTime           | vos_strncpy, 308         |
| vos_thread.h, 342       | vos_strnicmp, 308        |
| vos_cmpTime             | vos_mem.h, 309           |
| vos_thread.h, 342       | VOS_MEM_BLOCKSIZES, 311  |
| vos_crc32               | VOS_MEM_PREALLOCATE, 312 |
| vos_utils.c, 358        | vos_bsearch, 312         |
| vos_utils.h, 363        | vos_memAlloc, 313        |
| vos_cyclicThread        | vos_memCount, 313        |
| vos_thread.h, 344       | vos_memDelete, 314       |
| vos_determineBindAddr   | vos_memFree, 314         |
| vos_sock.h, 325         | vos_memInit, 314         |
| vos_divTime             | vos_qsort, 315           |
| vos_thread.h, 344       | vos_queueCreate, 316     |
| vos_dottedIP            | vos_queueDestroy, 316    |
| vos_sock.h, 326         | vos_queueReceive, 317    |
| vos_getErrorString      | vos_queueSend, 317       |
| vos_utils.c, 358        | vos_strncat, 318         |
| vos_utils.h, 364        | vos_strncpy, 318         |
| vos_getInterfaces       | vos_strnicmp, 319        |
| vos_sock.h, 326         | vos_memAlloc             |
| vos_getTime             | vos_mem.c, 303           |
| vos_thread.h, 345       | vos_mem.h, 313           |
| vos_getTimeStamp        | vos_memCount             |
| vos_thread.h, 345       | vos_mem.c, 303           |
| vos_getUuid             | vos_mem.h, 313           |
| vos_thread.h, 345       | vos_memDelete            |
| vos_getVersion          | vos_mem.c, 304           |
| vos_utils.c, 359        | vos_mem.h, 314           |
| vos_utils.h, 364        | vos_memFree              |
| vos_getVersionString    | vos_mem.c, 304           |
| vos_utils.c, 359        | vos_mem.h, 314           |
| vos_utils.h, 365        | vos_memInit              |
| vos_htonl               | vos_mem.c, 304           |
| vos_sock.h, 327         | vos_mem.h, 314           |
| vos_htonll              | vos_mulTime              |
|                         |                          |

| vos_thread.h, 345                | vos_dottedIP, 326                 |
|----------------------------------|-----------------------------------|
| vos_mutexCreate                  | vos_getInterfaces, 326            |
| vos_thread.h, 346                | vos_htonl, 327                    |
| vos_mutexDelete                  | vos_htonII, 327                   |
| vos_thread.h, 346                | vos_htons, 327                    |
| vos_mutexLock                    | vos_ipDotted, 328                 |
| vos_thread.h, 346                | vos_isMulticast, 328              |
| vos_mutexTryLock                 | vos_netIfUp, 328                  |
| vos_thread.h, 347                | vos_ntohl, 329                    |
| vos_mutexUnlock                  | vos_ntohll, 329                   |
| vos_thread.h, 347                | vos_ntohs, 330                    |
| vos_netIfUp                      | vos_select, 330                   |
| vos_sock.h, 328                  | vos_sockAccept, 330               |
| vos_ntohl                        | vos_sockBind, 331                 |
| vos_sock.h, 329                  | vos_sockClose, 331                |
| vos_ntohll                       | vos_sockConnect, 332              |
| vos_sock.h, 329                  | vos_sockGetMAC, 332               |
| vos_ntohs                        | vos_sockInit, 333                 |
| vos_sock.h, 330                  | vos_sockJoinMC, 333               |
| vos_qsort                        | vos_sockLeaveMC, 333              |
| vos_mem.c, 305                   | vos_sockListen, 334               |
| vos_mem.h, 315                   | vos_sockOpenTCP, 334              |
| vos_queueCreate                  | vos_sockOpenUDP, 335              |
| vos_mem.c, 305                   | vos_sockReceiveTCP, 335           |
| vos_mem.h, 316                   | vos_sockReceiveUDP, 336           |
| vos_queueDestroy                 | vos_sockSendTCP, 337              |
| vos_mem.c, 306                   | vos sockSendUDP, 337              |
| vos mem.h, 316                   | vos sockSetMulticastIf, 338       |
| vos_queueReceive                 | vos_sockSetOptions, 338           |
| vos_mem.c, 306                   | vos_sockTerm, 339                 |
| vos mem.h, 317                   | vos sockAccept                    |
| vos_queueSend                    | vos sock.h, 330                   |
| vos_mem.c, 307                   | vos sockBind                      |
| vos mem.h, 317                   | vos_sock.h, 331                   |
| vos sc32                         | vos sockClose                     |
| vos utils.c, 360                 | vos_sock.h, 331                   |
| vos_utils.h, 366                 | vos_sockConnect                   |
| vos select                       | vos_sock.h, 332                   |
| vos_sock.h, 330                  | vos sockGetMAC                    |
| vos_semaCreate                   | vos_sock.h, 332                   |
| vos_semaoreate vos thread.h, 348 | vos_sockInit                      |
| vos semaDelete                   |                                   |
| vos_thread.h, 348                | vos_sock.h, 333<br>vos sockJoinMC |
|                                  | <del>-</del>                      |
| vos_semaGive                     | vos_sock.h, 333                   |
| vos_thread.h, 348                | vos_sockLeaveMC                   |
| vos_semaTake                     | vos_sock.h, 333                   |
| vos_thread.h, 349                | vos_sockListen                    |
| vos_shared_mem.h, 319            | vos_sock.h, 334                   |
| vos_sharedClose, 321             | vos_sockOpenTCP                   |
| vos_sharedOpen, 321              | vos_sock.h, 334                   |
| vos_sharedClose                  | vos_sockOpenUDP                   |
| vos_shared_mem.h, 321            | vos_sock.h, 335                   |
| vos_sharedOpen                   | vos_sockReceiveTCP                |
| vos_shared_mem.h, 321            | vos_sock.h, 335                   |
| vos_sock.h, 322                  | vos_sockReceiveUDP                |
| VOS_MAX_SOCKET_CNT, 325          | vos_sock.h, 336                   |
| VOS_TTL_MULTICAST, 325           | vos_sockSendTCP                   |
| vos_determineBindAddr, 325       | vos_sock.h, 337                   |
| <del></del>                      | <del>-</del> ·                    |

| vos_sockSendUDP                             | vos_thread.h, 351                              |
|---|--|
| vos_sock.h, 337                             | vos_threadTerm                                 |
| vos_sockSetMulticastIf                      | vos_thread.h, 352                              |
| vos_sock.h, 338                             | vos_threadTerminate                            |
| vos_sockSetOptions                          | vos_thread.h, 352                              |
| vos_sock.h, 338                             | vos_types.h, 352                               |
| vos_sockTerm                                | VOS_ERR_T, 355                                 |
| vos_sock.h, 339                             | VOS_LOG_T, 356                                 |
| vos_strncat                                 | VOS_PRINT_DBG_T, 355                           |
| vos_mem.c, 308                              | VOS_TIMEVAL_T, 355                             |
| vos_mem.h, 318                              | vos_utils.c, 356                               |
| vos_strncpy                                 | vos_crc32, 358                                 |
| vos_mem.c, 308                              | vos_getErrorString, 358                        |
| vos_mem.h, 318                              | vos_getVersion, 359                            |
| vos_strnicmp                                | vos_getVersionString, 359                      |
| vos_mem.c, 308                              | vos_init, 359                                  |
| vos_mem.h, 319                              | vos_sc32, 360                                  |
| vos_subTime                                 | vos_terminate, 360                             |
| vos_thread.h, 349                           | vos_utils.h, 361                               |
| vos_terminate                               | INITFCS, 363                                   |
| vos_utils.c, 360                            | VOS_MAX_ERR_STR_SIZE, 363                      |
| vos_utils.h, 366                            | VOS_MAX_FRMT_SIZE, 363                         |
| vos_thread.h, 339                           | VOS_MAX_PRNT_STR_SIZE, 363                     |
| vos_addTime, 342                            | vos_crc32, 363                                 |
| vos_clearTime, 342                          | vos_getErrorString, 364<br>vos_getVersion, 364 |
| vos_cmpTime, 342                            | vos_getVersionString, 365                      |
| vos_cyclicThread, 344                       | vos_init, 365                                  |
| vos_divTime, 344                            | vos_sc32, 366                                  |
| vos_getTime, 345                            | vos_terminate, 366                             |
| vos_getTimeStamp, 345                       | vos_terminate, 500                             |
| vos_getUuid, 345                            |  |
| vos_mulTime, 345                            |  |
| vos_mutexCreate, 346                        |  |
| vos_mutexDelete, 346                        |  |
| vos_mutexLock, 346                          |  |
| vos_mutexTryLock, 347                       |  |
| vos_mutexUnlock, 347                        |  |
| vos_semaCreate, 348                         |  |
| vos_semaDelete, 348                         |  |
| vos_semaGive, 348                           |  |
| vos_semaTake, 349                           |  |
| vos_subTime, 349                            |  |
| vos_threadCreate, 349                       |  |
| vos_threadDelay, 350<br>vos_threadInit, 351 |  |
| vos_threadlsActive, 351                     |  |
| vos threadSelf, 351                         |  |
| vos_threadTerm, 352                         |  |
| vos_threadTerminate, 352                    |  |
| vos threadCreate                            |  |
| vos thread.h, 349                           |  |
| vos_threadDelay                             |  |
| vos_thread.h, 350                           |  |
| vos threadlnit                              |  |
| vos_thread.h, 351                           |  |
| vos_threadlsActive                          |  |
| vos_thread.h, 351                           |  |
| vos threadSelf                              |  |
| 100_1110440011                              |  |