



Protocol interface manual

NetLink-MPI

FDL Layer 2 access

Hilscher Gesellschaft für Systemautomation mbH
Rheinstraße 15
D-65795 Hattersheim
Germany

Tel. +49 (6190) 9907-0
Fax. +49 (6190) 9907-50

Hotline and Support: +49 (6190) 9907-99

Sales email: sales@hilscher.com
Hotline and Support email: hotline@hilscher.com

Web: <http://www.hilscher.com>

Index	Date	Version	Chapter	Revision
1	18.03.02	1.000	all	created

Although this software has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this software for any purpose not confirmed by us in writing.

Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this software or its documentation shall be limited to cases of intent.

We reserve the right to modify our products and their specifications at any time in as far as this contributes to technical progress. The version of the manual supplied with the software applies.

1 Introduction	4
1.1 The Layer 2 FDL access	4
1.1.1 FDL_Send_Data_Ack_Req/Con / SDA	5
1.1.2 FDL_Send_Data_Ack_Ind / SDA	7
1.1.3 FDL_Send_Data_No_Ack_Req / SDN	8
1.1.4 FDL_Send_Data_No_Ack_Ind / SDN	10
1.1.5 FDL_Send_Data_With_Reply_Req/Con / SRD	11
1.1.6 FDL_Send_Data_With_Reply_Ind / SRD	13
1.1.7 FDL_Reply_Update	14
1.1.8 FDL_Sap_Activate	16

1 Introduction

This manual serves as extension to the Netlink-MPI Protocol interface manual and covers the message communication channel of the layer 2 PROFIBUS.

1.1 The Layer 2 FDL access

The following subsections abstractly describe the data transfer (FDL = Fieldbus Data Link) services of the EN 50170 Volume 2 bus. The FDL services are made available to the HOST via Layer 2.

The HOST of Layer 2 is provided with the following data transfer services:

- Send Data with Acknowledge (SDA)
- Send Data with No Acknowledge (SDN)
- Send and Request Data with Reply (SRD)

Send Data with Acknowledge (SDA)

This service allows the HOST of the FDL (Layer 2) in the DEVICE (called local User in the following), to send user data (L_sdu) to a single remote station. At the remote station the L_sdu, if received error-free, is delivered by its FDL to the HSOT (called Remote User in the following). The Local HOST receives a confirmation concerning the receipt or non-receipt of the user data. If an error occurred during the transfer, the FDL of the Local HOST shall repeat the data transfer.

Send Data with No Acknowledge (SDN)

This service allows the Local HOST to transfer data (L_sdu) to a single remote station, to many remote stations (Multicast), or to all remote stations (Broadcast) at the same time. The Local HOST receives a confirmation acknowledging just the end of transfer, but not whether the data was duly received. At the remote stations this L_sdu, if received error-free, is passed to the Remote HOST. There is no confirmation, however, that such a transfer has taken place.

Send and Request with Reply (SRD)

This service allows a Local HOST to transfer data (L_sdu) to a single remote station and at the same time to request data (L_sdu) that was made available by the Remote HOST at an earlier time. At the remote station the received L_sdu, if error-free, is passed to the Remote HOST. The service also allows a Local HOST to request data from the Remote HOST without sending data (L_sdu = Null) to the Remote HOST.

The Local HOST receives either the requested data or an indication that the requested data was not available or a confirmation of the non-receipt of the transmitted data. The first two reactions also confirm the receipt of the transferred data.

Sending all these 3 FDL commands as local station does not need any activation within the DEVICE. But to activate and to configure the DEVICE for reception of an SDA and SDN service as a remote station, use the service SAP activate.

1.1.1 FDL_Send_Data_Ack_Req/Con / SDA

This command allows to send a SDA frame to the PROFIBUS network with variable data length field transparently. At the remote station the user data is delivered to the user. The local confirmation message of this command informs about the receipt or non-receipt of the user data in the remote station.

command message				
	variable	type	value	signification
Message Header	msg.rx	byte	7	receiver = FDL-Task
	msg.tx	byte	255	transmitter = HOST
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0	no answer
	msg.f	byte	0	error, state
	msg.b	byte	0x80	command = FDL_Send_Data_Ack_Req
	msg.e	byte	0	extension
FDL_DATA_ACK_REQ_STU	msg.d[0...(x-1)]	struct		

Here is the definition of the

```
typedef struct {
    unsigned char bLen; /*length of data field 0-244 */
    unsigned char bValue[ DATA_STU_SIZE ]; /* data */
} octet_str;

#define DATA_STU_SIZE  244

typedef struct FDL_DATA_ACK_REQ_STUtag{
    unsigned char bSsap; /*Src.Serv.Acc.Pt.0-62,255 */
    unsigned char bDsap; /*Dest.Serv.Acc.Pt.0-63,255 */
    unsigned char bRem_add_da; /*Remote Address 0-126 */
    unsigned char bServ_class; /* Prio Low=0,high=1*/
    octet_str tL_sdu; /* data field structure */
} FDL_DATA_ACK_REQ_STU;
```

The parameter bSsap defines the service access point of the Local User. A value bSsap of 63 which is the global access address is not permissible in this command. Value 255 is the NIL SAP.

The parameter bDsap defines the service access point of the Remote User. Value 255 is the NIL SAP.

The parameter bRem_add_da defines the FDL address of the remote station. The remote address bRem_add_da 127 which is the global address is not permissible in this command.

The parameter bServ_class defines the FDL priority for the data transfer. Two priorities are possible LOW = 0 and HIGH = 1.

response message				
	variable	type	value	signification
Message Header	msg.rx	byte	255	receiver = HOST
	msg.tx	byte	7	transmitter = FDL-Task
	msg.ln	byte	5	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0x80	response = FDL_Send_Data_Ack_Con
	msg.f	byte	f	error, see table below
	msg.b	byte	0	no command
	msg.e	byte	0	extension
FDL_DATA_ACK_CON_STU	msg.d[0...4]	struct		

```
typedef struct {
    unsigned char bSsap; /*Src.Serv.Acc.Pt.0-62,255 */
    unsigned char bDsap; /*Dest.Serv.Acc.Pt.0-63,255 */
    unsigned char bRem_add_da; /*Remote Address 0-126 */
    unsigned char bServ_class; /* Prio Low=0,high=1 */
    unsigned char bL_status; /* error code */
} FDL_DATA_ACK_CON_STU;
```

Here is the definition of the possible bL_status error codes that can be received.

bL_status	signification	error source	help
0 = CON_OK	service could be executed without an error		
2 = CON_RR	resource unavailable	slave	slave has no left buffer space for the requested service
3 = CON_RS	requested function of master is not activated within the slave	slave	slave is not supporting the requested service
9 = CON_NR	no answer-data, although the slave has to reponse with data	slave	slave hasn't sent back any amount of data
17 = CON_NA	no response of the station	slave	check network wiring, check bus address of slave or baud rate support
18 = CON_DS	master not into the logical token ring	network in general	check master DP-Address or highest-station-Address of other masters. Examine bus wiring to bus short circuits.

1.1.2 FDL_Send_Data_Ack_Ind / SDA

This command is passed from the remote FDL controller to the local HOST after receipt of a SDA frame, if the acknowledgement frame was sent. The reception of the indication message needs no acknowledgment message to the local FDL.

command message				
	variable	type	value	signification
Message Header	msg.rx	byte	255	receiver = HOST
	msg.tx	byte	7	transmitter = FDL-Task
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0	no answer
	msg.f	byte	0	error, state
	msg.b	byte	0xC0	command = FDL_Send_Data_Ack_Ind
	msg.e	byte	0	extension
FDL_DATA_ACK_IND_STU	msg.d[0...(x-1)]	struct		

Here is the definition of the

```
typedef struct {
    unsigned char bLen; /*length of data field 0-244 */
    unsigned char bValue[ DATA_STU_SIZE ]; /* data */
} octet_str;

#define DATA_STU_SIZE 244

typedef struct FDL_DATA_ACK_IND_STUtag{
    unsigned char bDsap; /*Dest.Serv.Acc.Pt.0-63,255*/
    unsigned char bSsap; /*Src.Serv.Acc.Pt.0-62,255 */
    unsigned char bRem_add_da; /*Remote Address 0-126 */
    unsigned char bLoc_add_sa; /*Local Address 0-126 */
    unsigned char bServ_class; /*Prio Low=0,high=1*/
    octet_str tL_sdu; /*data field structure */
} FDL_DATA_ACK_IND_STU;
```

The parameter bSsap and bDsap specify the source and destination Service Access Points of the received SDA frame.

The parameter bRem_add_da defines the FDL address of the remote station which has sent this service.

The parameter bServ_class specifies the FDL priority of the received SDA frame.

1.1.3 FDL_Send_Data_No_Ack_Req / SDN

This command allows to send a SDN frame to the PROFIBUS network with variable data length field transparently to a single remote station, to many remote stations (multicast) or to all remote stations (broadcast) at the same time. The local confirmation message of this command informs just about the end of the transfer, but not that the data was duly received.

command message				
	variable	type	value	signification
Message Header	msg.rx	byte	7	receiver = FDL-Task
	msg.tx	byte	255	transmitter = HOST
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0	no answer
	msg.f	byte	0	no error
	msg.b	byte	0x81	command = FDL_Send_Data_No_Ack_Req
	msg.e	byte	0	extension
FDL_DATA_ACK_REQ_STU	msg.d[0...(x-1)]	struct		

Here is the definition of the

```
typedef struct octet_strtag{
    unsigned char bLen; /*length of data field 0-244*/
    unsigned char bValue[ DATA_STU_SIZE ]; /*data*/
} octet_str;

#define DATA_STU_SIZE  244

typedef struct FDL_DATA_ACK_REQ_STUtag {
    unsigned char bSsap; /* Src.Serv.Acc.Pt. 0-62,255 */
    unsigned char bDsap; /*Dest.Serv.Acc.Pt. 0-63,255 */
    unsigned char bRem_add_da; /*Remote Address 0-126*/
    unsigned char bServ_class; /*Prio Low=0,high=1*/
    octet_str tL_sdu; /*data field structure*/
} FDL_DATA_ACK_REQ_STU;
```

The parameter bSsap defines the service access point of the Local User. A value bSsap of 63 which is the global access address is not permissible in this command. Value 255 is the NIL SAP.

The parameter bDsap defines the service access point of the Remote User. For broadcast messages a bDsap of 63 shall be chosen. In case of multicast messages the selection (group of stations) is performed by means of a dedicated bDsap. Value 255 is the NIL SAP.

The parameter bRem_add_da defines the FDL address of the remote station.

The parameter bServ_class defines the FDL priority for the data transfer. Two priorities are possible LOW = 0 and HIGH = 1.

response message				
	variable	type	value	signification
Message Header	msg.rx	byte	255	receiver = HOST
	msg.tx	byte	7	transmitter = FDL-Task
	msg.ln	byte	5	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0x81	response = FDL_Send_Data_No_Ack
	msg.f	byte	f	error, see table below
	msg.b	byte	0	no command
	msg.e	byte	0	extension
FDL_DATA_ACK_CON_STU	msg.d[0...4]	struct		

```
typedef struct FDL_DATA_ACK_CON_STUtag {
    unsigned char bSsap; /* Src.Serv.Acc.Pt. 0-62 */
    unsigned char bDsap; /* Dest.Serv.Acc.Pt. 0-63 */
    unsigned char bRem_add_da; /* Remote Address 0-126 */
    unsigned char bServ_class; /* Prio Low=0, high=1 */
    unsigned char bL_status /* error code */
} FDL_DATA_ACK_CON_STU;
```

Here is the definition of the possible bL_status error codes that can be received.

bL_status	signification	error source	help
0 = CON_OK	service could be executed without an error		
18 = CON_DS	master not into the logical token ring	network in general	check master DP-Address or highest-station-Address of other masters. Examine bus wiring to bus short circuits.

1.1.4 FDL_Send_Data_No_Ack_Ind / SDN

This command is passed from the remote FDL controller to the local HOST after receipt of a SDN frame, if the acknowledgement frame was sent. The reception of the indication message needs no acknowledgment message to the local FDL.

command message				
	variable	type	value	signification
Message Header	msg.rx	byte	255	receiver = HOST
	msg.tx	byte	7	transmitter = FDL-Task
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0	no answer
	msg.f	byte	0	error, state
	msg.b	byte	0xC1	command = FDL_Send_Data_No_Ack_Ind
	msg.e	byte	0	extension
FDL_DATA_ACK_IND_STU	msg.d[0...(x-1)]	struct		

Here is the definition of the

```
typedef struct {
    unsigned char bLen; /*length of data field 0-244*/
    unsigned char bValue[ DATA_STU_SIZE ]; /*data*/
} octet_str;

#define DATA_STU_SIZE 244

typedef struct FDL_DATA_ACK_IND_STUtag{
    unsigned char bDsap; /*Dest.Serv.Acc.Pt.0-63,255*/
    unsigned char bSsap; /*Src.Serv.Acc.Pt.0-62,255*/
    unsigned char bRem_add_da; /*Remote Address 0-126*/
    unsigned char bLoc_add_sa; /* Local Address 0-126*/
    unsigned char bServ_class; /* Prio Low=0,high=1*/
    octet_str tL_sdu; /*data field structure*/
} FDL_DATA_ACK_IND_STU;
```

The parameter bSsap and bDsap specify the source and destination Service Access Points of the received SDN frame.

The parameter bRem_add_da defines the FDL address of the remote station which has sent this service.

The parameter bServ_class specifies the FDL priority of the received SDN frame.

1.1.5 FDL_Send_Data_With_Reply_Req/Con / SRD

This command allows to transfer data to a single remote station and at the same time to request data that was made available by the remote user at an earlier time. The data of the response frame is sent back in the response message transparently. The service allows also to request data from the remote station without sending data to the remote station.(L_sdu = empty).

command message				
	variable	type	value	signification
Message Header	msg.rx	byte	7	receiver = FDL-Task
	msg.tx	byte	255	transmitter = HOST
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0	no answer
	msg.f	byte	0	no error
	msg.b	byte	0x82	command = FDL_Send_Data_Reply
	msg.e	byte	0	extension
FDL_DATA_REPLY_REQ_STU	msg.d[0...(x-1)]	struct		

Here is the definition of the structures:

```
typedef struct octet_strtag{
    unsigned char bLen; /*length of data field 0-244*/
    unsigned char bValue[ DATA_STU_SIZE ]; /*data*/
} octet_str;

#define DATA_STU_SIZE 244

typedef struct FDL_DATA_REPLY_REQ_STUtag {
    unsigned char bSsap; /*Src.Serv.Acc.Pt. 0-62,255*/
    unsigned char bDsap; /*Dest.Serv.Acc.Pt. 0-62,255*/
    unsigned char bRem_add_da; /*Remote Address 0-126*/
    unsigned char bServ_class; /*Prio Low=0,high=1*/
    octet_str tL_sdu; /*data field structure*/
} FDL_DATA_REPLY_REQ_STU;
```

The parameter bDsap defines the service access point of the Remote User. Value 255 is the NIL SAP.

The parameter bSsap defines the service access point of the Local User. A value bSsap of 63 which is the global access address is not permissible in this command. Value 255 is the NIL SAP.

The parameter bRem_add_da defines the FDL address of the remote station.

The parameter bServ_class defines the FDL priority for the data transfer. Two priorities are possible LOW = 0 and HIGH = 1.

response message				
	variable	type	value	signification
Message Header	msg.rx	byte	255	receiver = HOST
	msg.tx	byte	7	transmitter = FDL-Task
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0x82	response = FDL_Send_Data_Reply
	msg.f	byte	f	error, see table below
	msg.b	byte	0	no command
	msg.e	byte	0	extension
FDL_DATA_REPLY_CON_STU	msg.d[0...(x-1)]	struct		

```
typedef struct FDL_DATA_REPLY_CON_STUtag {
    unsigned char bSsap; /*Src.Serv.Acc.Pt. 0-62,255 */
    unsigned char bDsap; /*Dest.Serv.Acc.Pt. 0-62,255 */
    unsigned char bRem_add_da; /*Remote Address 0-126 */
    unsigned char bServ_class; /*Prio Low=0,high=1*/
    unsigned char bL_status /*error code*/
    octet_str tL_sdu; /*data field structure*/
} FDL_DATA_REPLY_CON_STU;
```

The tL_sdu data field contains only valid data, if the bL_status variable is 0. Here is the definition of the possible bL_status error codes that can be received.

bL_status	signification	error source	help
0 = CON_OK	service could be executed without an error		
2 = CON_RR	resource unavailable	slave	slave has no left buffer space for the requested service
3 = CON_RS	requested function of master is not activated within the slave	slave	slave is not supporting the requested service
8 = CON_DL	response data available, sent as low telegram	no error	the slave has responded with data in a low priority telegram
9 = CON_NR	no answer-data, although the slave has to reponse with data	slave	slave hasn't sent back any amount of data
17 = CON_NA	no response of the station	slave	check network wiring, check bus address of slave or baud rate support
18 = CON_DS	master not into the logical token ring	network in general	check master DP-Address or highest-station- Address of other masters. Examine bus wiring to bus short circuits.

1.1.6 FDL_Send_Data_With_Reply_Ind / SRD

command message				
	variable	type	value	signification
Message Header	msg.rx	byte	255	receiver = HOST
	msg.tx	byte	7	transmitter = FDL-Task
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0	no answer
	msg.f	byte	f	error, see table below
	msg.b	byte	0xC2	command = FDL_Send_Data_Reply_Ind
	msg.e	byte	0	extension
FDL_DATA_REPLY_IND_STU	msg.d[0...(x-1)]	struct		

```
typedef struct FDL_DATA_REPLY_IND_STUtag {
    unsigned char bDsap; /*Dest.Serv.Acc.Pt.0-62,255*/
    unsigned char bSsap; /*Src.Serv.Acc.Pt. 0-62,255*/
    unsigned char bRem_add_da; /*Remote Address 0-126*/
    unsigned char bLoc_add_da; /*local Address 0-126*/
    unsigned char bServ_class; /* Prio Low=0,high=1*/
    unsigned char bUpdate_Status; /*status reply dat*/
    octet_str tL_sdu; /*data field structure*/
} FDL_DATA_REPLY_IND_STU;
```

The parameter bUpdate_Status specifies, whether or not the response data has been passed to the local FDL controller. The response data can be set up to with the FDL_Reply_Update primitive. The following parameter values are defined:

```
#define CON_LO    0x20 /* low priority data transmitted
in response */
#define CON_HI    0x21 /* high priority data transmitted
in response */
#define CON_NO    0x22 /* no data transmitted in re-
sponse */
```

1.1.7 FDL_Reply_Update

The HOST is responsible for valid data in the DEVICE if any data is shall be requested from a remote station. By this command the HOST may initiate the loading of this data to a specific SAP. Upon completion of loading the data area, the DEVICE informs the HOST by the confirmation message.

command message				
	variable	type	value	signification
Message Header	msg.rx	byte	7	receiver = FDL-Task
	msg.tx	byte	255	transmitter = HOST
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0	no answer
	msg.f	byte	0	no error
	msg.b	byte	0x83	command = FDL_Reply_Update_Req
	msg.e	byte	0	extension
FDL_REPLY_UPDATE_REQ_STU	msg.d[0...(x-1)]	struct		

Here is the definition of the structures:

```
typedef struct octet_strtag{
    unsigned char bLen; /*length of data field 0-244*/
    unsigned char bValue[ DATA_STU_SIZE ]; /*data*/
} octet_str;

#define DATA_STU_SIZE 244

typedef struct FDL_REPLY_UPDATE_REQ_STUtag {
    unsigned char bSsap; /*Src.Serv.Acc.Point 0-62,255*/
    unsigned char bDsap; /* not used */
    unsigned char bRem_add_da; /* not used */
    unsigned char bServ_class; /* Prio Low=0,high=1*/
    unsigned char bTransmit; /* Single=1/Multiple=2*/
    octet_str tL_sdu; /* data field structure */
} FDL_REPLY_UPDATE_REQ_STU;
```

The parameter bSsap specifies the service access point of the remote User that carries out this request and which data area shall be updated by the tL_sdu. A bSsap value of 63 is not permissible.

The parameter bDsap defines the service access point of the Remote User. Value 255 is the NIL SAP.

The parameter bRem_add_da defines the FDL address of the remote station.

The parameter bServ_class defines the FDL priority for the data transfer. Two priorities are possible LOW = 0 and HIGH = 1.

The parameter bTransmit specifies whether the update is transmitted only once SINGLE = 1 or many times MULTIPLE = 2, in the case of "multiple" the data area is transferred again for each subsequent SRD.

response message				
	variable	type	value	signification
Message Header	msg.rx	byte	255	receiver = HOST
	msg.tx	byte	7	transmitter = FDL-Task
	msg.ln	byte	5	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0x83	response = FDL_Reply_Update_Con
	msg.f	byte	f	error, see table below
	msg.b	byte	0	no command
	msg.e	byte	0	extension
FDL_REPLY_UPDATE_CON_STU	msg.d[0...4]	struct		

```
typedef struct FDL_REPLY_UPDATE_CON_STUtag{
    unsigned char bSsap; /*Src.Serv.Acc.Point 0-62*/
    unsigned char bDsap; /*Dest.Serv.Acc.Point 0-63*/
    unsigned char bRem_add_da; /*Remote Address 0-126*/
    unsigned char bServ_class; /*Prio Low=0,high=1*/
    unsigned char bL_status /*error code*/
} FDL_REPLY_UPDATE_CON_STU;
```

Here is the definition of the possible bL_status error codes that can be received.

bL_status	signification	error source	help
0 = CON_OK	service could be executed without an error		
18 = CON_DS	master not into the logical token ring	network in general	check master DP-Address or highest-station-Address of other masters. Examine bus wiring to bus short circuits.

1.1.8 FDL_Sap_Activate

command message				
	variable	type	value	signification
Message Header	msg.rx	byte	7	receiver = FDL-Task
	msg.tx	byte	255	transmitter = HOST
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0	no answer
	msg.f	byte	0	error, state
	msg.b	byte	0x97	command = FDL_Sap_Activate
	msg.e	byte	0	extension
FDL_SAP_ACTIVATE_REQ_STU	msg.d[0...(x-1)]	struct		

```
typedef struct FMA_SAP_ACTIVATE_REQ_STUtag {
    unsigned char bSsap; /*Src.Serv.Acc.Point 0-63,65*/
    unsigned char bAccess; /*Accessright = 255*/
    unsigned char bService_list_length; /*Length = 4*/
    #define SERVICE_SDA 0
    #define SERVICE_SDN 1
    #define SERVICE_SRD 2
    unsigned char abService_activate[4]; /*Service*/
    #define ROLE_IN_BOTH 1
    #define ROLE_IN_RESPONDER 2
    #define ROLE_IN_INITIATOR 3
    unsigned char abRole_in_service [4]; /*Cfg.Srv.*/
    struct {
        unsigned char bReq_low; /* bufferlength = 244 */
        unsigned char bReq_high; /* bufferlength = 244 */
        unsigned char bInd_low; /* bufferlength = 244 */
        unsigned char bInd_high; /* bufferlength = 244 */
    } tNon_cyclic_L_sdu_length_list[3];
} FMA_SAP_ACTIVATE_REQ_STU;
```

This service provides the HOST with the possibility to activate and to configure a local LSAP for the individual FDL services. An exception to this is the responder function for the Reply services (SRD,CSRD) that are activated by means of the RSAP activate service.

The parameter bSsap specifies the local LSAP that is to be activated and configured. The values 0 to 63 and NIL = 65.

The parameter bAccess with the values ALL = 255 or 0 to 126 is used for access protection and specifies whether all remote stations (all) or only an individual remote station (0 to 126) may have access to the LSAP. The parameter is only valid for responder functions, i.e. abRole_in_service[...] = ROLE_IN_RESPONDER / ROLE_IN_BOTH. If the access value is ALL, the DEVICE automatically sets the buffer length to the maximum value 244 and the services to all supported and the role in service to both independent what the rest command message parameter are.

The parameters `abService_activate[...]` contains the FDL service that shall be activated for this service (SDA, SDN, SRD).

The parameters `abRole_in_service[...]` specifies the configuration for the service to be activated. The following values are specified:

ROLE_IN_INITIATOR: The DEVICE initiates the respective service exclusively

ROLE_IN_RESPONDER: The DEVICE responds to the service exclusively. Not permissible for SRD.

ROLE_IN_BOTH: The DEVICE initiates and responds to the service. Not permissible for SRD.

The value `tNon_cyclic_L_sdu_length_list[...]` is a list of `Max_L_sdu` lengths. It specifies the maximum length low or high priority `L_sdu` for the requested primitive in `bReq_Low/High`. For the indication of the service SDA and SDN and for the confirmation of the service SRD, the maximum length is specified by the parameter `bInd_Low/High`.

response message				
	variable	type	value	signification
Message Header	msg.rx	byte	255	receiver = HOST
	msg.tx	byte	7	transmitter = FDL-Task
	msg.ln	byte	x	length of message
	msg.nr	byte	j	number of the message
	msg.a	byte	0x97	response = FDL_Sap_Activate
	msg.f	byte	0	no error
	msg.b	byte	0	no command
	msg.e	byte	0	extension
FDL_SAP_ACTIVATE_CON_STU	msg.d[0...(x-1)]	struct		

```
typedef struct FMA_SAP_ACTIVATE_CON_STUtag {
    unsigned char  bSsap; /*Src.Serv.Acc.Point 0-63,65*/
    unsigned char  bM_status; /* Error Code */
} FMA_SAP_ACTIVATE_CON_STU;
```