0.0.1 VDP state machine procedures

0.0.1.1 buildDea()

The buildDea() procedure builds a DEASSOCIATE TLV for the ~~current state machine~~VSI and returns it in the return parameter.

0.0.1.2 resourceCmd(rxCmd, Delay)

This procedure makes a resource request from the Bridge, waits for a response, builds a response TLV and places it in the variable Resp. The response values reflect the requested resource action (PREASSOC, ASSOC, or DEASSOC), conditioned by return variable resourceResult which is set to NULL before calling the resourceCmd procedure and set to one of the following values by the procedure:

a) SUCCESS
b) KEEP
c) DEASSOCIATE
d) TIMEOUT

The response constructed by the procedure in the Resp variable can be PREASSOC, ASSOC, or DEASSOC with a reason, keep indicator and hard error indicator~~success or fail~~. For a successful completion the procedure will copy the rxCmd parameter into the Resp variable. If the Bridge is selecting VIDs based on GroupIDs, then the procedure also replaces zero VIDs with valid VIDs.

The delay parameter specifies how long the procedure should wait for a response ~~from the Bridge~~. If the delay is exceeded, ~~and~~ no response is received and the VSI is not associated, then the procedure returns a value of TIMEOUT~~timeOut~~. ~~otherwise, a value of SUCCESS is returned~~If the delay is exceeded, no response is received and the VSI is associated, then the procedure returns a value of KEEP along with a Resp equal to the rxCmd parameter. If the delay is not exceeded, then the procedure returns SUCCESS, KEEP or DEASSOCIATE depending on the response recieved along with the rxCmd in the Resp and the reason set as follows:.

~~The resourceCmd() procedure decodes each possible command type:~~

a) **DEASSOC**:
   1) The procedure returns DEASSOCIATE along with Resp.reason set to *Success*.
b) **PREASSOC**:
   1) If the resources requested are valid the procedure returns SUCCESS along with Resp.reason set to *Success*. ~~and fail i~~
   2) If the resources are invalid the ~~Otherwise the Bridge~~ procedure returns DEASSOCIATE along with Resp.reason set to a code other than *Success* and the Resp.hard set TRUE if a retry will not change the situation or FALSE if a retry might change the situation.
c) **PREASSOCR**:
   1) If the resources requested are valid, available and reserved for this VSI the procedure returns SUCCESS along with the Resp.reason set to *Success*. ~~and fail i~~
   2) If the resources are invalid, unavailable or not reserved for this VSI the ~~Otherwise the Bridge~~ procedure returns DEASSOCIATE along with the Resp.reason code other than *Success* and the Resp.hard set TRUE if a retry will not change the situation or FALSE if a retry might change the situation.
d) **ASSOC**:
   1) If the resources requested are valid, available and enabled for this VSI the procedure returns SUCCESS along with the Resp.reason set to *Success*.
   2) If the resources are invalid, unavailable or not enabled for this VSI and the VSI is not currently Associated the procedure returns DEASSOCIATE along with the Resp.reason code other than

Success and the Resp.hard set TRUE if a retry will not change the situation or FALSE if a retry might change the situation.

3) If the resources are invalid, unavailable or not enabled for this VSI and the VSI is currently Associated the procedure returns KEEP along with the Resp.reason code other than *Success* and the Resp.hard set TRUE if a retry will not change the situation or FALSE if a retry might change the situation.

0.0.1.3 resourceFree()

The resourceFree() procedure frees all resources associated with this VSI~~state machine instance~~.

0.0.1.4 TxTlv(tlv)

The TxTlv() procedure causes the TLV passed in the tlv parameter to be transmitted.