8.4.2 Shutting down a queue

A connection is closed when every queue within the connection is disconnected. Disconnection is handled per queue basis. **Orderly release** of the connection is initiated on a queue from a sender side, and receiving queue is disconnected from the other end (sender side of the queue).

8.4.2.1 Shutting down a queue by an initiator

When an application client at an initiator desires to close (orderly release) a connection with a target service, the initiator shall signal an ORB of which *final* bit and *notify* bit are set to one and *queue* field is equal to I2T_QUEUE of the connection. The initiator shall not signal any succeeding ORBs of those *queue* field is equal to I2T_QUEUE of the connection, except for the re-queuing ORB which has the same *signature* in case of recovery, until the target allocates the queue number in a future CONNECT request or response. (When target completes the execution of the ORB of which *final* bit is set to one, target may indicate that peer is disconnecting the queue to its client.)

After a request completion status block associated with the ORB of which *final* bit is set to one is received, the initiator shall create a control operation ORB whose buffer contains a DISCONNECT_CONFIRMATION control request. The initiator shall specify the I2T_QUEUE parameter.

The target may allocate the same queue number as the number specified by the parameter in a future CONNECT request or response after receiving DISCONNECT_CONFIRMATION control request.

NOTE - If the connection is bi-directional non-blocking connection, disconnection of the other queue (T2I_QUEUE of the connection) may be initiated by the target. Until the other queue is disconnected, the connection is in the state of “half close”.

8.4.2.2 Shutting down a queue by a target

When an application client at a target desires to close (orderly release) a connection with an initiator service, the target shall, at first, complete all outstanding data delivery to the initiator. After completing to send all outstanding data, the target shall create a buffer that contains a SHUTDOWN_QUEUE control request and signal the initiator to retrieve the control request by asserting the *attention* bit in a status block. The SHUTDOWN_QUEUE control request shall specify the T2I_QUEUE parameter.

After signaling SHUTDOWN_QUEUE control request, the target shall not block the execution of the specifying T2I_QUEUE. (When the initiator receives the SHUTDOWN_QUEUE control request, the initiator may indicate that peer is disconnecting the queue to its client.)

After receiving SHUTDOWN_QUEUE control request from the target, initiator shall signal an ORB of which *final* bit and *notify* bit are set to one and *queue* field is equal to T2I_QUEUE of the connection. The initiator shall not signal any succeeding ORBs of those *queue* field is equal to T2I_QUEUE of the connection, except for the re-queuing ORB which has the same *signature* in case of recovery, until the target allocates the queue number in a future CONNECT request or response.

After a request completion status block associated with the ORB of which *final* bit is set to one is received, the initiator shall create a control operation ORB whose buffer contains a DISCONNECT_CONFIRMATION control request. The initiator shall specify the T2I_QUEUE parameter.

The target may allocate the same queue number as the number specified by the parameter in a future CONNECT request or response after receiving DISCONNECT_CONFIRMATION control request.

NOTE - If the connection is bi-directional non-blocking connection, disconnection of the other queue (I2T_QUEUE of the connection) may be initiated by the initiator. Until the other queue is disconnected, the connection is in the state of “half close”.

8.4.3 Aborting a connection

Abortive release of the connection can be initiated by both ends. Abortive release of the connection is established by aborting every queue in the connection.

8.4.3.1 Aborting a queue by an initiator

When an application client at an initiator desires to close (abortive release) a connection with a target service, the initiator shall create a control operation ORB whose buffer contains a SHUTDOWN_QUEUE control request. The initiator shall specify the active queue number parameter of the connection. The initiator shall also signal an ORB of which *final* bit and *notify* bit are set to one and *queue* field is equal to active queue number of the connection to each active queue of the connection.

Note: In case of non-blocking bi-directional connection and both queue is active, the initiator may specify both I2T_QUEUE and T2I_QUEUE parameters with one SHUTDOWN_QUEUE control information. If the connection is in the state of half close, the initiator shall not specify the queue number that has already been disconnected.

The initiator shall not signal any succeeding ORBs of those *queue* field is equal to the queue number(s) of the connection specified with SHUTDOWN_QUEUE control request, except for the re-queuing ORB which has the same *signature* in case of recovery, until the target allocates the queue number in a future CONNECT request or response.

After receiving SHUTDOWN_QUEUE control request, the target shall not block the execution of the specified queue(s) if specified queue exists. Otherwise, if specified queue does not exist, the target shall ignore the specified queue number parameter(s). (When the target receives the SHUTDOWN_QUEUE control request, the target may indicate that peer is abortively disconnecting the queue to its client.)

After a request completion status block associated with the ORB of which *final* bit is set to one is received, the initiator shall create a control operation ORB whose buffer contains a DISCONNECT_CONFIRMATION control request. The initiator shall specify the corresponding queue number parameter.

Note: In case of non-blocking bi-directional connection, the initiator may specify both I2T_QUEUE and T2I_QUEUE parameters with one DISCONNECT_CONFIRMATION control information.

The target may allocate the same queue number as the number specified by the parameter in a future CONNECT request or response after receiving DISCONNECT_CONFIRMATION control request.

8.4.3.2 Aborting a queue by a target

When an application client at a target desires to close (abortive release) a connection with an initiator service, the target shall create a buffer that contains a SHUTDOWN_QUEUE control request and signal the initiator to retrieve the control request by asserting the *attention* bit in a status block. The SHUTDOWN_QUEUE control request shall specify the active queue number parameter(s) of the connection.

After signaling SHUTDOWN_QUEUE control request, the target shall not block the execution of the specifying queue(s). (When the initiator receives the SHUTDOWN_QUEUE control request, the initiator may indicate that peer is abortively disconnecting the queue to its client.) (Need to distinguish the control request between orderly and abortive release?)

Note: In case of non-blocking bi-directional connection and both queues are active, the target may specify both I2T_QUEUE and T2I_QUEUE parameters with one SHUTDOWN_QUEUE control information. If the connection is in the state of half close, the target shall not specify the queue number that has already been disconnected.

After receiving SHUTDOWN_QUEUE control request from the target, initiator shall signal an ORB of which *final* bit and *notify* bit are set to one and *queue* field is equal to corresponding queue number of the connection to each specified queue if the specified queue exists. Otherwise, if the specified queue does not exist, the initiator shall ignore the specified queue number parameter(s). The initiator shall not signal any succeeding ORBs of those *queue* field is equal to specified queue number parameter(s) of the connection, except for the re-queuing ORB which has the same *signature* in case of recovery, until the target allocates the queue number in a future CONNECT request or response.

After a status block associated with the ORB of which *final* bit is set to one is received, the initiator shall create a control operation ORB whose buffer contains a DISCONNECT_CONFIRMATION control request. The initiator shall specify the corresponding queue number parameter.

NOTE - In case of non-blocking bi-directional connection, the initiator may specify both I2T_QUEUE and T2I_QUEUE parameters with one DISCONNECT_CONFIRMATION control information.

The target may allocate the same queue number as the number specified by the parameter in a future CONNECT request or response after receiving DISCONNECT_CONFIRMATION control request.

8.4.3.3 Simultaneous Close

It is possible for both sides to perform a disconnection of a queue simultaneously. If a initiator has already signaled an ORB of which *final* bit and *notify* bit are set to one and *queue* field is equal to a disconnecting queue of the connection, the initiator shall not signal any ORB of which *queue* field is equal to the disconnecting queue including its *final* bit is set to one even when the initiator receives SHUTDOWN_QUEUE control request which specifies the disconnecting queue from the target, except for the re-queuing ORB which has the same *signature* in case of recovery.