SEIKO EPSON Corp.

Proposal for “Disconnect” command

To: **1394 PWG** [file: fn-99-0319-00]

From: Fumio Nagasaka (EPSON Software Development Laboratory Inc.,)

Date: 99/03/19

1 Background

In Miami 1394 PWG meeting, difficulty to provide “Disconnect” command for PWG profile had been turned out. A disconnect request is handled by the management queue (queue id = 0) which has higher priority of execution. And the 1394 PWG profile uses unordered execution model defined in ANSI NCITS.325-1998 (SBP-2). Thus when one of peers invoked a “disconnect” command to the other peer, the request shall be executed as soon as possible. The “disconnect” command specifies a queue to be disconnected. However both peer hardly know what ORB had been consumed, and the initiator hardly knows whether it may dispose ORBs remained in the current ORB-list.

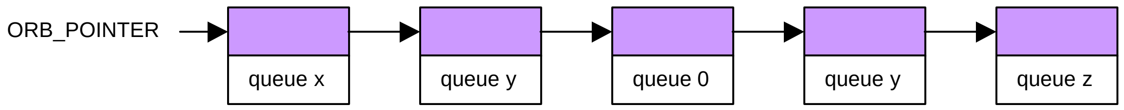


Figure 1 – Multiplexed queues in an SBP-2 task set

The nature of SBP-2 does not allow a target device to have transport facilities as same as an initiator device does. However through discussions of the 1394 PWG, the necessity of the initiator’s provision to receive several commands from the target device has been cleared. In this way, 1394 PWG command set (normally so called “profile”) allows a target device to send a “disconnect” command. When a target requested a “disconnect” command, this command shall be stored in the input buffer of a transport flow ORB associated with queue id set to be zero. (See “*Peter Johansson,* IDT_r01, Feb. 28, 1999”). The initiator have to know what ORB had been consumed by the target, and what ORB had not been consumed yet.

But how does the initiator know this?

(1) May the initiator read ORB_POINTER register of the target?

No. The ORB_POINTER register shows current ORB to be processed and this value may be different from the ORB aligned in specified queue to be disconnected.

(2) May the initiator examine ORB pointer stored in the STATUS_BLOCK replied by the target?

No. The ORB_POINTER stored in the STATUS_BLOCK is the address of a transport flow ORB which was placed by the initiator to receive this request from the target.

2 Purpose

The purpose of this report is to suggest adding some descriptions for the 1394 PWG profile, to provide “Disconnect” command. This report proposes to make a new parameter ID called as “TAGGED_ORB”. The reasons why the new parameter ID required are listed below:

- To facilitate a PWG IDT (Imaging Device Transport) compliant peer to have an opportunity to send a “Disconnect” command.
- To facilitate the fair access for both peers of the connection.
-

3 Proposal

3.1 Provision

Yellow portions are additional portions for IDT_r01 proposal.

[Peter Johansson]

The parameter ID shall specify the parameter format, either immediate or variable-length. The most significant bit of the parameter ID determines the format; parameters whose ID values are in the range zero to $7F_{16}$, inclusive, shall conform to the format specified by Figure 10 while those in the range 80_{16} – FF_{16}, inclusive, shall conform to the format specified by Figure 11. Defined values for parameter ID are given in Table 1; all values not specified are reserved for future standardization.

Table 1 – Parameter ID values

Parameter ID	Parameter name	Description
1	TASK_SLOTS	See *Peter Johansson* IDT_r01, Feb. 28, 1999
82_{16}	SERVICE_ID	See *Peter Johansson* IDT_r01, Feb. 28, 1999
3	I2T_QUEUE	See *Peter Johansson* IDT_r01, Feb. 28, 1999
4	T2I_QUEUE	See *Peter Johansson* IDT_r01, Feb. 28, 1999
85_{16}	TAGGED_ORB	A 48-bit ORB pointer with a prefixed 16-bit signature used to specify a processed ORB to be disconnected within transient phase caused by the transport command.

The format of immediate parameters is shown below.

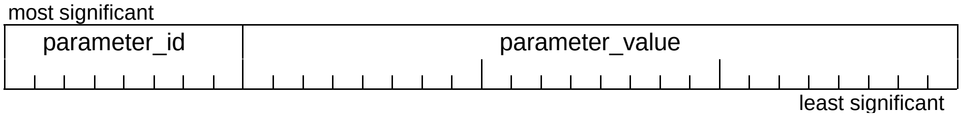
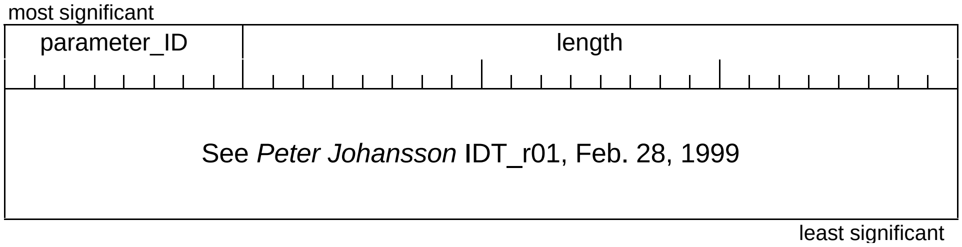


Figure 20 – Immediate parameter format

The *parameter_ID* field shall specify the parameter, as encoded by Table 1.

The *parameter_value* field shall specify the immediate value of the parameter. Unless otherwise specified for a particular value of *parameter_ID*, the *value* field shall contain an unsigned 24-bit number.

The format of variable-length parameters (which are usually ASCII text strings) is shown below.



See *Peter Johansson* IDT_r01, Feb. 28, 1999

Figure 11 – Variable-length parameter format

The *parameter_ID* field shall specify the parameter, as encoded by Table 1.

The *length* field shall specify the parameter length, in bytes.

The *parameter_value* field shall contain the value of the parameter and shall commence with the most significant byte of the parameter value. If the length of the parameter is not a multiple of four, the parameter value shall be padded with trailing bytes of zero. Unless otherwise specified for a particular value of *parameter_ID*, the *value* field shall contain an ASCII text string without leading or trailing blank characters.

An example of parameter Encoding for "TAGGED_ORB" is shown below.

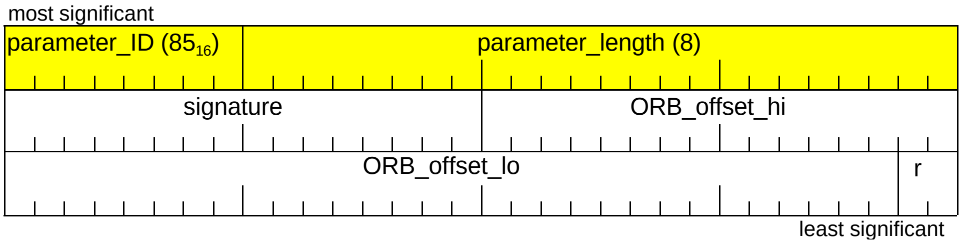
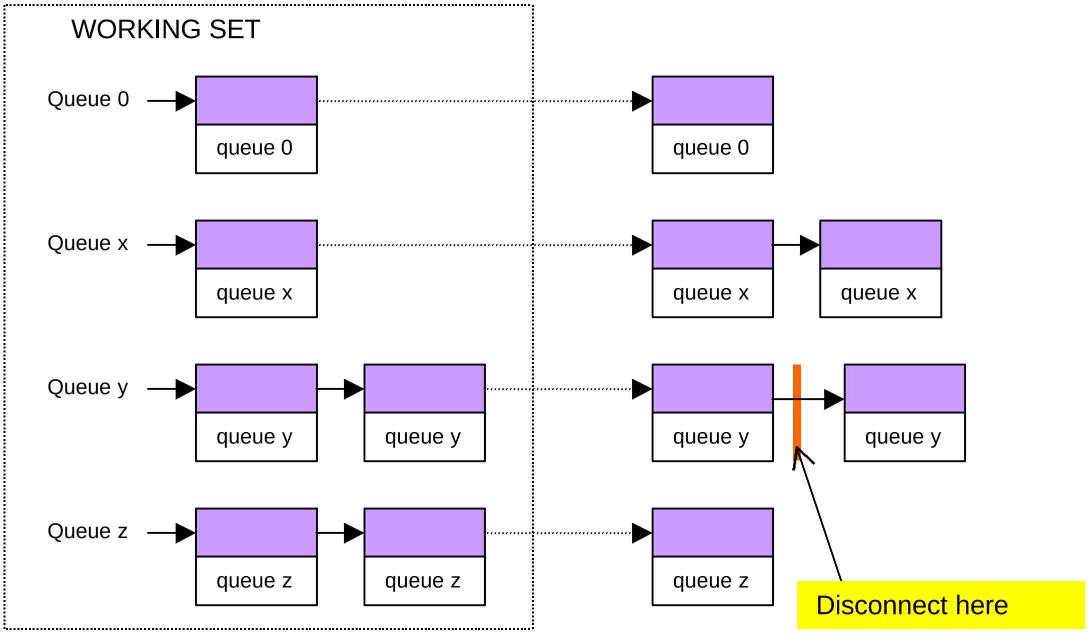


Figure xx3 - Example of parameter Encoding for *"TAGGED_ORB"*

3.2 **I2T** Disconnect command

In this case, the initiator has knowledge about ORBs associated with specified queue id. Thus the initiator may specify an ORB which belongs to a queue to be disconnected in a "disconnect" command format.



To specify it, the initiator shall enclose "TAGGED_ORB" parameter into the control information format described in clause 5.2 of IDT_r01 proposal. Also the initiator have to provide a registry to keep what ORBs had been processed in the queue.

The figure shown below gives an example of a control information format.

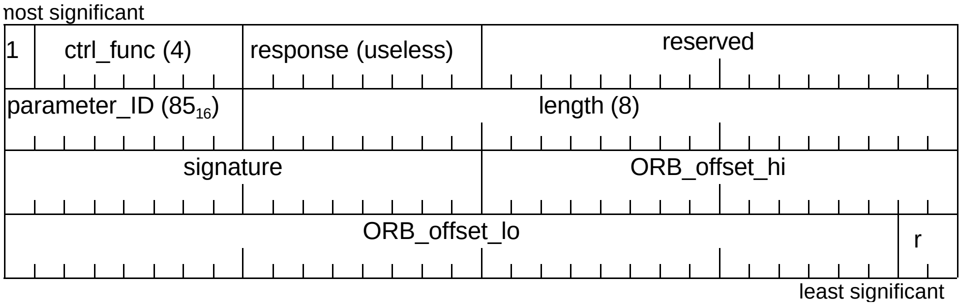


Figure 4 – example control information format

3.3 **T2I** Disconnect command

When a target invokes “disconnect” command to the initiator, the target does not expect whether a client program in the initiator side will place new request in the ORB list or not. However the target device want to abandon a connection between the initiator at this moment.

This connection is handled in the specified queue, fortunately the fetch agent of the target provides separate queues internally (this description would be an implementation issue). Thus, the target device simply has knowledge for the ORB which was most recently consumed associated in this queue. So, the target replies a status block *with an attention bet set to one*. (See IDT_r01 page 19)

Then the initiator places an empty transport flow control ORB with *queue id* set to zero and *direction* bit one. The target stores a control information including “TAGGED_ORB” parameter described in previous section in this note.

The initiator shall dispose rest of ORBs associated the specified queue from the ORB list.

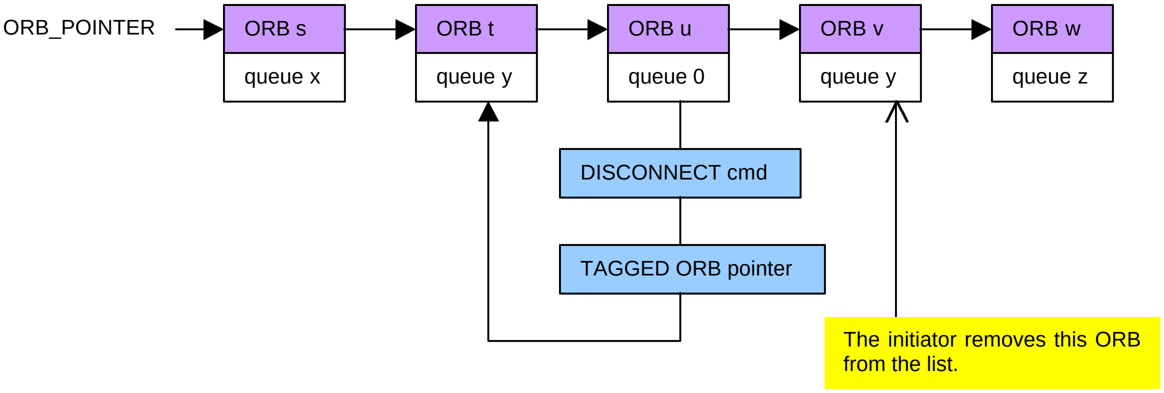


Figure 5 – an example of T2I disconnect

The figure shown above outlines behavior of the initiator. In this figure, the target specified queue y to be disconnected and the target showed that it had consumed “ORB t”.