

ORACLE FUSION MIDDLEWARE

Oracle B2B Technical Note

Technical Note #004

PL/SQL Enqueue / Dequeue scripts for B2B queues

Oracle B2B has multiple options to enable communications to the middleware software. These internal Delivery Channels are: Oracle AQ, JMS Queues, File and FTP.

These scripts enable the user to enqueue and de queue messaged to the internal delivery channels. In a typical end-to-end Business Process Integration these script enable the unit testing of Oracle B2B.

Contents

Sample PL/SQL for enqueue / dequeue B2B queues 2

Script: msgid_b2b.sql 2

Script: enq_b2b.sql 2

Script: deq_b2b.sql 3

Sample PL/SQL for enqueue / dequeue B2B queues

(IP_IN_QUEUE/IP_OUT_QUEUE)

SQL Script	Description
msgid_b2b.sql	Create db sequence to generate unique msg id for enqueue
Enq_b2b.sql	Enqueue a message into b2b queue. User can use either varchar2 or clob as the message payload.
Deq_b2b.sql	Dequeue a message from b2b queue.

Script: msgid_b2b.sql

```
CREATE SEQUENCE message_seq INCREMENT BY 1 START WITH 1000;
```

Script: enq_b2b.sql

```
DECLARE
  enqueue_options          dbms_aq.enqueue_options_t;
  message_properties       dbms_aq.message_properties_t;
  msg_handle               RAW(16);
  ipmsg                    IP_MESSAGE_TYPE;
  /* xml_payload           varchar2(30000); */
  xml_clob                 clob;
  msg_id                   number(10);
  subscribers              dbms_aq.aq$_recipient_list_t;

BEGIN
  select message_seq.nextval into msg_id from dual;
  /* xml_payload              := '<?xml version="1.0" encoding="UTF-8"?>'; */
  xml_clob                  := '<?xml version="1.0" encoding="UTF-8"?>';

  subscribers(1)           := SYS.AQ$_AGENT('b2buser', null, null);
  message_properties.RECIPIENT_LIST := subscribers;

  ipmsg                     := IP_MESSAGE_TYPE
    (msg_id, null, 'Acme', 'GlobalChips', 'Process_850', '850', '4010', 1, xml_clob, null);

  dbms_aq.enqueue(queue_name => 'IP_IN_QUEUE',
    enqueue_options => enqueue_options,
    message_properties => message_properties,
    payload => ipmsg,
    msgid => msg_handle);
  commit;
END;
/
show errors;
```

Script: deq_b2b.sql

```
DECLARE
  dequeue_options          dbms_aq.dequeue_options_t;
  message_properties       dbms_aq.message_properties_t;
  message_handle           RAW(16);
  ipmsg                    IP_MESSAGE_TYPE;
BEGIN
  dequeue_options.consumer_name := 'b2buser';
  dequeue_options.dequeue_mode  := DBMS_AQ.REMOVE;
  dequeue_options.navigation    := DBMS_AQ.NEXT_MESSAGE;
  dequeue_options.wait          := DBMS_AQ.FOREVER;

  DBMS_AQ.DEQUEUE(
    queue_name              => 'IP_IN_QUEUE',
    dequeue_options         => dequeue_options,
    message_properties      => message_properties,
    payload                 => ipmsg,
    msgid                   => message_handle);

  DBMS_OUTPUT.PUT_LINE('Message ID : ' || ipmsg.msg_id);
COMMIT;
END;
/
show errors;
```