

Published: 07/21/67  
 (Supersedes: BQ.6.02, 03/13/67)

### Identification

Calls to the Interprocess Communication Facility  
 Michael J. Spier

### Purpose

This section contains the list of all calls, to and from the Interprocess Communication Facility, which are functionally related to the Facility. Internal calls which remain "transparent" to the Facility's user are not included in this section.

Calls are grouped as follows:

Calls to the Event Channel Manager (ECM), ring 1

Calls to the Interprocess Group Event Channel Manager (IPGECM), ring 0

Calls to the Wait Coordinator (WC), ring 1, and

Calls to the Device signal Table Manager (DSTM) ring 0.

The header to each one of the above groups contains the number of the MSPM section in which that particular group of calls is described. Calls declared to be "internal" are calls within the Facility and unavailable to the user.

The list of calls is followed by the EPL declarations of the arguments, alphabetically sorted for easy reference.

CALLS TO THE EVENT CHANNEL MANAGER (Ring 1) (BQ.6.04)

### Calls by the user of the Facility:

a. Declaration calls (from all rings):

```
ecm$create_ev_chn(ev_chn,mode,signl_ring)
```

```
ecm$decl_ev_call_chn(ev_chn,proc_ptr,data_ptr,prior,  
level,sts)
```

```
ecm$decl_ev_wait_chn(ev_chn,sts)
```

```
ecm$give_access(ev_chn,acc_sw,acc_list)
```

```
ecm$delet_ev_chn(ev_chn,sts)
```

b. Event signalling (from rings 1-63 only):

ecm\$set\_event(rec\_prcs, ev\_chn, ev\_id, sts)

c. Declaration calls associated with the WC (from all rings):

ecm\$set\_call\_prior

ecm\$set\_wait\_prior

d. Calls associated with the quitting process (from ring 1):

ecm\$set\_wakeup\_sw(prcs\_id, sw)

ecm\$read\_wakeup\_sw(prcs\_id, sw)

Internal calls:

(All these calls originate in ring 1)

f=ecm\$read\_event(ev\_chn, ev\_ind, sts)  
(call made by the WC)

ecm\$get\_dev\_signal(dev\_sigl\_chn\_list)  
(call made by the WC)

CALLS TO THE INTERPROCESS GROUP EVENT CHANNEL MANAGER  
(IPGECM) (Ring 0) (BQ.6.05)

User call (from ring 0 only):

ipgecm\$set\_event(rec\_prcs, ev\_chn, ev\_id, sts)

Internal calls:

ipgecm\$set\_event(rec\_prcs, ev\_chn, ev\_id, sts)  
(This call is identical to the one above, but  
made internally by the ring 1 ecm\$set\_event)

ipgecm\$link\_dev\_chn(ev\_chn, dev\_inx)

ipgecm\$unlink\_dev\_chn(dev\_inx)  
(both calls are made by the ring 0 DSTM)

CALLS TO THE WAIT COORDINATOR (WC) (Ring 1) (BQ.6.06)

User call (from all rings):

f=wc\$test\_event(chn\_list,ev\_ind,sts)

wc\$wait(chn\_list,ev\_ind,sts)

WAIT COORDINATOR'S CALL TO USER (from ring 1 to all rings):

call [associated procedure] (data\_ptr,ev\_ind)  
(This is the call issued by the WC when an event  
has been signalled over an event-call channel)

CALLS TO THE DEVICE SIGNAL TABLE MANAGER (DSTM) (Ring 0)  
(BQ.6.07)

User calls:

a. Associated with Interprocess Communication (from ring 0):

dstm\$set\_dev\_signal(dev\_inx)

dstm\$set\_auth(dev\_inx,prcs\_id,ev\_chn)

dstm\$reset\_auth(dev\_inx)

b. Other:

f=dstm\$check\_auth(dev\_inx,prcs\_id)

dstm\$set\_route(dev\_inx,route)

dstm\$get\_route(dev\_inx,route)

Internal calls:

dstm\$read\_dev\_signal(dev\_inx,ev\_id,ev\_count)  
(call made by ecm\$get\_dev\_signal from ring 1)

ARGUMENT DECLARATION

```

dcl acc_list(n) character(50) /*Channel access list */;
dcl acc_sw bit(1) /*Access switch */;
dcl chn_list(n) bit(70) /*Event channel list */;
dcl data_ptr pointer /*Associated data pointer */;
dcl dev_inx fixed bin(17) /*Device index */;
dcl dev_sigl_chn_list
    fixed bin(17) /*Device signal channel list */;
dcl ev_chn bit(70) /*Event channel name */;
dcl ev_count fixed bin(17) /*Event count */;
dcl ev_id bit(70) /*Event identifier */;
dcl ev_ind(3) bit(70) /*Event indicator
    ev_ind(1)=ev_chn
    ev_ind(2)=ev_id
    ev_ind(3)=prcs_id */;
dcl f bit(1) /*'0'b=false, '1'b=true */;
dcl level fixed bin(17) /*Ass proc recursive call level*/;
dcl mode bit(1) /*'0'b=ev_count, '1'b=ev_queue */;
dcl prior fixed bin(17) /*Ev-call_channel lookup prior */;
dcl prcs_id bit(36) /*Process identifier */;
dcl proc_ptr pointer /*Associated procedure pointer */;
dcl rec_prcs bit(36) /*Receiving process identifier */;
dcl route bit(18) /*Dev physical route info */;
dcl sw bit(1) /*Receiving process' wakeup
    switch */;
dcl signl_ring fixed bin(17) /*Signalling ring number */;
dcl sts bit(36) /*Return status */;

```