

TO: MSPM Distribution
FROM: D. L. Stone
SUBJECT: BF.20.03
DATE: 01/30/68

This revision mirrors the updating of BF.20.01 and adds the declaration of the data bases:

cat_cst

CCT

DCT

Status_segment

A new argument has been added to the assign call.

(Supersedes: Published: 01/30/68
 BF.20.03, 12/01/67,
 BF.20.03, 05/05/67,
 BF.20.03, 02/08/67,
 BF.20.03, 12/06/66)

Identification

A Summary of GIM Calls and Data Bases
 S. D. Dunten, T. P. Skinner, D. L. Stone

Purpose

This section provides a summary of all external calls to the GIOC Interface Module (GIM). EPL declarations for the parameters are given, as well as for Data Bases referenced.

The GIM Interface

All the calls specified below are accessible from ring 0 only. To call the GIM from other rings, the "hcs_entry name" mechanism must be used.

Call `giminit$assign(name,devx,event,type,rcode);`

```

dcl   name char (32),          /* symbolic channel name */
      devx fixed bin(17),     /* device index returned
                               by the GIM for this
                               channel */
      event bit(70),          /* passed on to the DSTM
                               for device signalling */
      type char(*),           /* passed on to IØ
                               assignment module */
      rcode fixed bin(17);    /* error return, 0=ok */

```

Call `giminit$fsassign(name,devx,rcode);`

Call `giminit$list_size(devx,listsize,rcode);`

```

dcl   devx fixed bin(17),     /* device index from assign */
      listsize fixed bin(12), /* size of list to be
                               allocated */
      rcode fixed bin(17),    /* error return */

```

Call `gim$list_change(devx,dcwp,datap,listx,count,rcode);`

```

dc1  devx fixed bin(17),
      (dcwp,
       datap) ptr,
      (listx,
       count)fixed bin(17);
      /* pointer to "dcw_array" */
      /* pointer to "data_array" */
      /* index within DCW_list,
       (starting with 1) of first
       DCW to be changed */
      /* number of elements in
       "dcw_array" which are to
       be used */

```

where the data referred to by "dcwp" and "datap" is accessed as:

```

dc1  dcw_array(count)bit(72)based(dcwp),
      /* a list of real DCW's */
      1 data_array(count)based(datap),
      2 p ptr,
      /* pointer to user workspace
       if nth DCW is data DCW */
      2 rw bit(2);
      /* "10"b = readDDCW
       "01"b = writeDDCW */

```

(Note: for privileged users, datap should be null).

Call `gim$list_connect(devx,ciw,listx,rcode);`

```

dc1  devx fixed bin(17),
      ciw fixed bin(18),
      listx fixed bin(12),
      rcode fixed bin(17);
      /* the CIW is treated as an
       18 bit string which is the
       right half of a CIW */

```

Call `gim$get_cur_status(devx,listx,dcwt,rcode);`

```

dc1  devx fixed bin(17),
      listx fixed bin(12),
      dcwt fixed bin(12),
      rcode fixed bin(17);
      /* current index of dcw being
       processed */
      /* current dcw tally */

```

Call `gim$get_status(devx,status_array_ptr,array_size,outsize,waiting,rcode);`

```

dc1  devx fixed bin(17),
      status_array_ptr ptr,
      (array_size,
       outsize)fixed bin(17),
      waiting fixed bin(17),
      rcode fixed bin(17);
      /* pointer to "status_array"*/
      /* maximum number of status
       elements to be returned */
      /* actual number returned */
      /* count of status words
       waiting but not returned
       in this call */

```

```

dc1  1 status_array(array_size)based(status_array_ptr),
    2 status,
    3 type bit(4)          /* bits 0-3 of (modB)
                           status word */
    3 int_sig bit(2),     /* bits 4-5 */
    3 adapter bit(12),   /* bits 18-29 */
    2 time bit(52),      /* time at which stored
                           interrupt was processed
                           by handler */
    2 listx fixed bin(12), /* which DCW caused
                           status store? */
    2 dcwt fixed bin(12); /* dcw tally at interrupt,
                           if applicable */

```

Call giminit\$unassign(devx,rcode);

```

dc1  devx fixed bin(17),
    rcode fixed bin(17);

```

GIM DATA BASES

The Channel Assignment and Channel Status Table

This segment is the major system-wide GIM data base. It is wired down and contains all per-user channel information and per-gioc information.

```

dc1  1 cat_cst based(p),
    2 chan (0;260),      /* per-device-index */
                           /* information accessed */
                           /* by the "devx" */
                           /* presented in the GIM */
                           /* calls */
    3 cctno bit (18),   /* segment number of the */
                           /* CCT for this user */
                           /* -only accessed by one */
                           /* process */
    3 dcw_rel_add bit (18), /* offset of dcw list */
                           /* within dcw segment */
                           /* Zero is interpreted */
                           /* as dcw-list not */
                           /* yet allocated unless */
                           /* "priv" is on */
    3 dcw_list_len bit (12), /* size of dcw list in */
                           /* dcw's */
                           /**/
    3 giocno bit (2),   /* which gioc is being */
                           /* used for this channel */
                           /**/

```

```
3 conno bit (2),          /* which connect */
                          /* channel are we using */
                          /**/
3 priv bit (1),          /* on if user can */
                          /* supply absolute */
                          /* address in data DCW's */
3 dir_chan bit (1),     /* on if direct channel */
                          /**/
3 channo bit (12),      /* physical channel */
                          /* number */
                          /**/
3 status_lost bit (1),  /* ON if status lost */
                          /**/
3 pad1 bit (5),         /* guess again */
                          /**/
2 gioc (2),             /* per-gioc information */
                          /**/
3 mbx_base ptr,         /* pointer to mailbox */
                          /* area */
                          /**/
3 portno bit (3),       /* which port is this */
                          /* gioc on */
                          /**/
3 stat_base bit (3),    /* offset for use in */
                          /* avoiding bad status */
                          /* channels */
                          /*-GIM uses only one */
                          /* status channel */
                          /* normally */
3 connect (0:2),        /* connect channel info */
                          /**/
4 lock bit (36),        /* uses locker */
4 ccw bit (36),         /* ccw to be connected */
                          /* is physically here */
4 cpw bit (72),         /* fixed pointers to the */
                          /* ccw's above */
                          /**/
3 status (0:1),         /* status channel info */
                          /**/
4 lock bit (36),        /* standard lock */
4 scwa bit (36),        /* status channel word */
                          /* A-refill */
4 oldest fixed bin (17), /* index to oldest */
                          /* status */
4 basep ptr,           /* ptr to top of status */
                          /* queue */
4 intp fixed bin (17), /* index to last */
                          /* interrupt processed */
4 thresh fixed bin (17), /* min. amount of free */
                          /* stat. storage needed */
4 endindex fixed bin(17), /* size in status words */
```

```

3 devx (8:130),      /* back ptr to devx entry
                    in "chan" */
4 id bit (12);      /* */

```

The Channel Copy Table

This table is allocated as a separate segment for each user of the assign call and contains information about the DCW lists for each user.

```

dcl 1 cct based(p),
    2 copy_dcw fixed bin (12),      /* index of the last */
                                    /* dcw for which copying */
                                    /* has been done*/
    2 copy_word fixed bin (12),     /* index of the last */
                                    /* word for the "copy_dcw" */
                                    /* which has been moved */
                                    /* to the users area. */
    2 nreads fixed bin (12),        /* number of read */
                                    /* DCW's in list */
    2 nalloc fixed bin (12),        /* number of buffers */
                                    /* allocated */
    2 addr_list (4096),              /* allocated to fit; */
                                    /* information about */
                                    /* data for dcw's */
    3 wksp ptr,                      /* users data area */
    3 length allocated               /* size of wired down */
    fixed bin (18);                 /* buffer allocated */

```

The Device Configuration Table

This per-system table describes all of the channels attached to all of the GIOC's which are administered by the GIM.

```

dcl 1 dct based(p),                 /* device configuration
                                    table */
    2 ndev fixed bin(17),            /* number of devices */
    2 desc(300 /* dev_nam_max */ ), /* start of device
                                    description */
    3 dev_nam char(32),              /* device name */
    3 phys_nam char(32),             /* name of physical
                                    channel and GIOC */
    3 devx bit(17),                  /* device index */
    3 giocno fixed bin(17),          /* GIOC number of this
                                    device */
    3 phychn fixed bin(17),          /* half the LPW channel
                                    number of this device */
    3 conno fixed bin(17),           /* symbolic connect
                                    channel number */
    3 direct_chan bit(1);            /* ON if direct channel */

```

The Status Segment

All hardware stored status is kept in a single wired down segment. Per GIOC, two status queues are kept - one for channel 0 and one for user status; associated with each status queue is a time queue administered by the interrupt handler. The base of the various status queues is determined from the `cat_cst.gioc(N).status(M).basep` entries in the `cat_cst`. Given the base of a queue, the queue can be accessed by:

```

dc1  1 status_q based (basep),
      2 status (0:endindex /*from cat_cst*/)bit(36),
      /*ModA GIOC*/
      2 pad bit(72),
      2 time (0:endindex),
      3 t bit(52),
      3 pad bit(2);

```

Buffer Areas

In addition to the above tables, one wired down contiguous segment is used to store all `dcw_lists` and data buffers associated with `dcw's`. Each buffer is individually allocated within the segment as an area.