Identification

Corrigenda, BD.9.04
R. M. Graham, M. A. Padlipsky, J. M. Grochow

p.5   Insert the following at the bottom of the page
       (or at the top of p.6, if you prefer):

Since each entry for a condition handler in a particular
signals_n segment contains information pertaining to the
procedure which caused the entry to be established, the
user should be sure to call reversion before returning
from a procedure for any condition names he has called
condition for in the same procedure. Failure to do this
may cause chaos during future attempts to "signal" these
condition names.

Each "call" in a user's program that involves a ring
crossing will cause information to be added to the
<rtn_stk> and thus increment the invocation number.
As returns are made this information is removed and
the invocation number decremented. It is thus possible
that condition handlers will be left in various signals_n
segments with invocation numbers greater than the
current invocation number (or in some cases with an
invocation number equal to the present invocation number
but with references to stack frames that are no longer
active).

The procedure "signal_search" is invoked by signal
to search signals_n segments in other than the current
ring. Any signal vector entries that signal_search
encounters with an invocation number greater than the
invocation number current for the ring in which it is
searching will be removed by a special call to
reversion$ring with arguments condname (char(*)) and
rnum (char(2)). A message will also be put in the
user's error file to indicate the action taken. Note,
however, that this correction of user mistakes (reverting
condition handlers) only occurs when the mistake is
encountered in searching.

p.11   The description of <u>condition</u>, at the bottom of the page should read:

The calling sequence is

   call condition(condname, proc);

with declarations

   dcl condname char(*), proc entry;

where

<u>condname</u> is the name of a condition

<u>proc</u>   is the procedure which is to be invoked when this condition is signalled (this applies until and unless the procedure is either pushed down by a subsequent call to <u>condition</u> for <u>condname</u> or is popped off by a call to <u>reversion</u>)

p.12   Step 2. should be deleted.

In step 3., the reference for generate_ptr should read "(BY.13.02)"

p.13   In step b., the reference for link_change$make_ definition should read "(BY.13.03)"

p.15   The first paragraph of the description of <u>reversion</u> should be:

The calling sequence is

   call reversion (condname);

with declaration

   dcl condname char(*);

where <u>condname</u> has the same meaning as in <u>condition</u>.

p.16   In step 1., the sentence beginning "The argument
            procname" should be deleted.

p.16   The following should be inserted before the
            discussion of signal:

FIND_CONDITION

It is sometimes useful to be able to examine the contents
of a handler list. The find_condition routine is furnished
for this purpose; it operates in the protection ring it
is invoked from.

The calling sequence is

        call find_condition(condname, n, proc, flag);

with declarations

        dcl condname char(*), (n, flag) fixed bin (17), proc label;

where

        condname   is as in condition

        n   is the number of places down in the list to
                   examine (n=0 indicates the top of the list)

        proc   (returned by find_condition), is the handler
                   indicated in the nth entry in the list or the
                   last entry in the list if there are less than
                   n+1 entries

        flag   is set to zero if proc is indeed the nth entry
                   and is set to m, where m is the number of places
                   down the list the entry is, if proc is the
                   last entry rather than the nth. (If there is
                   no list of handlers for condname in the current
                   ring, flag is set to -1.)

p.16   The declarations for signal should be

        dcl condname char(*), rtn_flag fixed bin(17), ptr ptr;