TO: C. Clingen
          F. J. Corbató
          R. Freiburghouse
          J. Gintell
          N. Morris
          R. Roach
          J. Saltzer ✓
          M. Weaver
          S. Webber
          B. Wolman

FROM: V. Voydock, R. Feiertag

DATE: September 3, 1971

SUBJECT: Special Condition and Cleanup Procedures

Enclosed is a copy of a document distributed this August. We have
received no comments critical of the plan it describes. If you have any
last minute reservations please get in touch with us as soon as possible.
If we have received no critical comments by Friday, September 10, we
will begin implementation of the plan. For compatibility the condition
mechanism will continue to recognize the current methods of establishing
these special handlers for at least a year. At this time either the
compiler can be changed to diagnose the obsolete usage at compile time
or the condition establishing subroutines can be changed to diagnose
it at run time.

TO: C. T. Clingen   J. H. Saltzer
             F. J. Corbató   M. L. Weaver
             R. Freiburghouse   S. H. Webber
             J. W. Gintell   B. L. Wolman
             N. I. Morris

FROM: R. J. Feiertag, V. L. Voydock

DATE: August 19, 1971

SUBJECT: Special Condition and Cleanup Procedures

There are two things currently called conditions that really are
not conditions. The first of these, the cleanup mechanism, is really a
means of specifying an action to be taken when a particular activation
of a procedure is to be aborted by a non-local transfer. The cleanup
mechanism, conceptually, has nothing to do with conditions. The second
of these, the unclaimed signal mechanism, is a means of specifying a
default handler for when no specific handler for a raised condition has
been established. Unclaimed signal is not, in itself, a condition, it
is only an indicator of a default action to the condition mechanism.

A new feature similar to unclaimed signals has been proposed. This
new mechanism could indicate a local default handler for the case where
no specific handler has been established for the raised condition in a
particular activation (stack frame) of a procedure. This new feature
has been known as the "matches_all", "all", or "any" condition .

This new local default mechanism would fit into the condition
signalling mechanism as follows. When a condition is raised a search is
made for a handler. Each stack frame in the stack is searched for a
handler for the condition. If no handler is found for the condition in
a particular stack frame the local default is invoked if one exists for
that frame. If not the search is continued in the next frame. If
no specific handler or local default handler is found in the current

stack the most recently established global default handler (unclaimed signal hander) is invoked. If no global default handler has been established, the crawl out mechanism is invoked.

In order to make clear that there is a difference between the above mentioned mechanisms and true conditions, the manner in which they are established should be distinct. In other words establishing a global and local default handler as well as establishing a cleanup procedure should be performed by separate routines, not the normal handler establishing routine, condition_. This has the secondary benefit of not necessitating the reservation of the special names "cleanup" and "unclaimed_signal".

By separating out these special mechanisms we will also be better able to take advantage of their special properties thereby enabling us to make the entire condition mechanism more efficient.

In order to implement the above proposals the following calls would be implemented:

```
call establish_cleanup_procedure_ (proc);
call set_local_default_handler_ (handler);
call set_global_default_handler_ (handler);
```

where proc and handler are entries. Calls would also be provided for reverting the handlers. For compatibility purposes the condition mechanism would continue to recognize the current methods of establishing these special handlers for at least a year. At this time either the compiler can be changed to diagnose the obsolete usage at compile time or the condition establishing subroutines can be changed to diagnose it at run time.