new ecs facility

return authorization + block data transfer during call

will permit the return of at most 1 block of data & 2 group objects

1) format original call

The 2 words immediately preceding the IPLIST control the return authorization.

a) 1st word preceding IPLIST controls data return

<table>
<thead>
<tr>
<th>18</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>

up to C words can be returned starting at address A.
[If C=0, no words can be returned. Hence a second word refuses authorization.]

b) 2nd word preceding IPLIST controls object return

<table>
<thead>
<tr>
<th>18</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>

up to C objects can be returned starting at CList index A.
[as above, C=0 for no authorization]
2) **new versions of return**

IP 1 and IP2 will control data and object return, we will also retain old versions of return?

A) IP1 controls data return

```
  12  18
   C   A
```

Up to C words will be returned starting at address A

B) IP2 controls object return

```
  18  15
   C   A
```

Up to C objects will be returned starting at CH2, index A
Implementation of new forms of return & return

If both counts are zero, no action

If either count is nil, check for errors [c-list buffer and field
flush errors]

If errors, copy data and c-list entries to a buffer
perform swapback to calling subprogram
check if return control data, reduce counts to the minimum
at called and caller, much like for c-list flush & field flush
errors, if any, copy data and c-list entries to proper
location,
Now look for interrupt signals!

4) Variation on original call
change format-xj to #20

\[
\begin{array}{ccc}
\text{5} & \text{5} & \text{5} \\
\text{15} & \text{15} & \text{15} \\
\text{A} & \text{5} & \text{5} \\
\text{S} & \text{S} & \text{S} \\
\end{array}
\]

S will be skip field if no return [most use case now uses 0 or 1]
A address of a 2 word return control in form

\[
\begin{array}{cc}
A_i & C_i \\
C_i & A_i \\
A_i & C_i \\
\end{array}
\]

A_i, C_i address counts of data return authorization
A_i, A_i address counts of object return authorization
If A = 0, no authorization, if (A) = 0 read authorization
(1 + A) = 0 no object authorization
addition

The eis buffers can also be used to implement block data transfer during a call.

need "data block at length n"

1) new kind of parameter specification:

2) The IP list entry will be as word as follows:

<table>
<thead>
<tr>
<th>13</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>

If C=1, error
else transfer C words from A to the parameter area in called subprocess (via eis buffers, of course)

3) The next data word in parameter area of called subprocess will, of course, be in words beyond beginning of this block.

suggest parameter type be data search if length of to be transferred allocation is in called routine

the corresponding AP link word is in called routine and is set to a of the IP link entry