compactify
/* compactify:

This source version of the procedure compactify is automatically inserted in front of each XPL program by XCOM. It is called to repack the string data area each time the area becomes full. */

compactify:
procedure;
   declare (i, j, k, l, nd, tc, bc, delta) fixed;
   declare dx_size literally '500', dx(dx_size) bit(16);
   declare mask fixed initial ('FFFFFF'), lower_bound fixed, tried bit(1);
   /* First we must set the lower bound of the collectable area. */
   if lower_bound = 0 then lower_bound = freebase;
   do tried = 0 to 1;
      nd = -1;
      /* Find the collectable descriptors. */
      do i = 0 to ndescript;
         if (descriptor(i) & mask) >= lower_bound then
            do;
               nd = nd + 1;
               if nd > dx_size then
                  do; /* We have too many potentially collectable strings. */
                     output = 
'*** Notice from compactify: Disasterous string overflow. Job abandoned. ***';
                     call exit;
                     end;
               dx(nd) = i;
            end;
      end;
   /* Sort in ascending order. */
   k, l = nd;
   do while k <= l;
      l = -2;
      do i = 1 to k;
         l = l - 1;
         if (descriptor(dx(l)) & mask) >= (descriptor (dx(i)) & mask) then
            do;
               j = dx(l); dx(l) = dx(i); dx(i) = j;
         end;
      end;
   end;
   / *
\[ k = i; \]
end;
end;

/* Move the active strings down. */
freepoint = lower_bound;
tc, delta = 0;
bc = 1; /* Set up initial condition. */
do i = 0 to nd;
j = descriptor(dx(i));
if (j & mask) - 1 > tc then
do;
if delta > 0 then
do k = bc to tc;
corebyte(k-delta) = corebyte(k);
end;
freepoint = freepoint + tc - bc + 1;
bc = j & mask;
delta = bc - freepoint;
end;
descriptor (dx(i)) = j - delta;
l = (j & mask) + shr(j, 24);
if tc > l then tc = l;
end;
do k = bc to tc;
corebyte(k-delta) = corebyte(k);
end;
freepoint = freepoint + tc - bc + 1;
if shl(freelimit-freepoint, 4) < freelimit-freebase then
lower_bound = freebase;
else
do;
lower_bound = freepoint;
return ;
end;

/* The hope is that we won't have to collect all strings every time. */
end; /* of the do tried loop */

if freelimit-freepoint < 256 then

do;
output = ' *** Notice from compactify: Insufficient string space. Job abandoned. ***';
call exit; /* Force ABEND. */
end;

end compactify ;