

PL/I-80 Reference Manual

Appendix E: External Procedures

This appendix describes the use of the EXTERNAL attribute as applied to local procedure definitions, available in PL/I-80 version 1.3 or later. This is a non-standard feature, and must be avoided if upward compatibility with other Subset-G implementations is required.

It is often useful to group a set of separately compiled procedures into a single compilation, where the procedures reference the same global data. According to the Subset-G standard, each subroutine must be separately compiled, and the global data must be duplicated in each compilation. The individual modules are then combined using the linkage editor to produce the final object module. The EXTERNAL attribute can be applied to a procedure heading in PL/I-80 in order to make the procedure accessible outside the module. In order to be compatible with any future implementations of PL/I from Digital Research, only top-level procedures should be marked with the EXTERNAL attribute, and all globally-accessed data should be marked as STATIC. A compilation containing a group of EXTERNAL subroutines should consist only of subroutines, and no main program. The following program segment shows the use of the EXTERNAL attribute:

```
module:
  proc;
  dcl
    1 global_data static,
      2 a_field char(20) var init(''),
      2 b_field fixed      init(0),
      2 c_field float      init(0);
  set_a:
    _proc (c) external;
    dcl c char(20) var;
      a_field = c;
    end set_a;
  set_b:
    _proc (x) external;
    dcl x fixed;
      b_field = x;
    end set_b;
  set_c:
    _proc (y) external;
    dcl y float;
      c_field = y;
    end set_c;
  sum:
    _proc returns(float) external;
      return (b_field+c_field);
    end sum;
  display:
    _proc external;
      put skip list(a_field,b_field,c_field);
    end;
end module;
```

The program shown above defines five external procedures: set_a, set_b, set_c, sum, and display. These four procedures are accessed in the program shown below:

```
call_ext:
  proc options(main);
  dcl
    set_a entry (char(20) var),
    set_b entry (fixed),
    set_c entry (float),
    sum returns(float),
    display entry;
  call set_a('Johnson, J');
  call set_b(25);
  call set_c(5.50);
  put skip_list(sum());
  call display();
  end call_ext;
```

These two programs are separately compiled and linked together to form a single module. Note that, due to linkage editor format restrictions, long external names are truncated on the right and thus must be unique in the first six characters.