

UNIT 9

CALL Statement

CALL Statement

The **CALL** statement transfers control from one object program to another within the run unit

CALL { **identifier-1** } [**USING** [**BY CONTENT**/**BY REFERENCE**] **data-**
name-1 [**, data-name-2**] ...
 Literal-1]

[; **ON OVERFLOW** imperative-statement]

Figure 9-1 CALL Statement

Notes:

The **CALL** statement transfers control from one object program to another within the run unit.

The program containing the **CALL** statement is the calling program; the program identified in the **CALL** statement is the called subprogram. Called programs can contain **CALL** statements; however, a called program must not execute a **CALL** statement that directly or indirectly calls the calling program unless it has the **RECURSIVE** attribute.

CALL BY CONTENT/REFERENCE

- The **CALL....BY REFERENCE** technique allows the sub-program to access and process the data-items in the caller's storage.
- The **CALL....BY CONTENT** technique allows the sub-program to access and process a copy of the data-items from the caller's storage. The sub-program cannot change the original data values in the caller's storage.
- A single **CALL** statement may have both data passing techniques.

WORKING-STORAGE SECTION.

```
01 RECORD-A.  
   05 FIELD1 PIC .....  
   05 FILLER .....  
   .....
```

PROCEDURE DIVISION.

```
   .....  
   CALL "XXXXXXXX" USING BY REFERENCE RECORD-A  
                        BY CONTENT LENGTH OF RECORD-A
```

Figure 9-2 CALL BY CONTENT/REFERENCE

Notes:

BY REFERENCE Phrase

If the **BY REFERENCE** phrase is either specified or implied for a parameter, the corresponding data item in the calling program occupies the same storage area as the data item in the called program.

BY CONTENT Phrase

If the BY CONTENT phrase is specified or implied for a parameter, the called program cannot change the value of this parameter as referenced in the CALL statement's USING phrase, though the called program can change the value of the data item referenced by the corresponding data-name in the called program's Procedure Division header. Changes to the parameter in the called program do not affect the corresponding argument in the called program.

BY VALUE Phrase

The BY VALUE phrase applies to all arguments that follow until overridden by another BY REFERENCE or BY CONTENT phrase.

If the BY VALUE phrase is specified or implied for an argument, the value of the argument is passed, not a reference to the sending data item. The called program can modify the formal parameter corresponding to the BY VALUE argument, but any such changes do not affect the argument since the called program has access to a temporary copy of the sending data item.

These options are Compiler dependent.

LINKAGE SECTION

- **The LINKAGE SECTION of the DATA DIVISION describes data made available from another program**
- **Storage is NOT reserved**
- **The VALUE clause may not be specified (except for level-88 items)**

Figure 9-3 LINKAGE SECTION

Notes:

The Linkage Section describes data made available from another program or method. The Linkage Section may be composed of two entry types:

- Record-description-entry
- Data-item-description-entry

GLOBAL DATA

- The **GLOBAL** clause specifies that a data-name is available to every program contained within the program that declares it, as long as the contained program does not itself have a declaration for that name. All data-names subordinate to or condition-names or indexes associated with a global name are global names.
- The **GLOBAL** clause can be specified in the Working-Storage Section, the File Section, the Linkage section, and the Local-Storage Section, but only in data description entries whose level-number is 01 or FD.
- An item may have both **GLOBAL** and **EXTERNAL** clauses.
- **GLOBAL** only applies to Nested Programs.

Notes

GLOBAL in FILE SECTION.

The **GLOBAL** clause specifies that the file connector named by a file-name is a global name. A global file-name is available to the program that declares it and to every program that is contained directly or indirectly in that program.

A file name is global if the **GLOBAL** clause is specified in the file description entry for that file-name. A record – name is global if the **GLOBAL** clause is specified in the record description entry by which the record-name is declared or, in the cause of record description entries in the File Section, if the **GLOBAL** clause is specified in the file description entry for the file-name associated with the record description entry.

GLOBAL in Data Descriptions

The GLOBAL clause specifies that a data-name is available to every program contained within the program that declares it, as long as the contained program does not itself have a declaration for that name. All data-names subordinates to or condition-names or indexes associated with a global name are global names.

A data-name is global if the GLOBAL clause is specified either in the data description entry by which the data-name is declared or in another entry to which that data description entry is subordinate.

A statement in a program contained directly or indirectly within a program which describes a global name reference that name without describing it again.

Two programs in a run unit can reference common data in the following circumstances:

1. The data content of an external data record can be referenced from any program provided that program has described that data record.
2. If a program is contained within another program, both programs can refer to data possessing the global attribute either in the containing program or in any program that directly or indirectly contains the containing program.

Sample Program

```
ID DIVISION.  
PROGRAM-ID. MPGM.  
.....  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 MNTEC PIC X(32) GLOBAL.  
PROCEDURE DIVISION.  
    MOVE "BEGINNING CONTENTS" TO MNTEC  
    CALL "SUBPGM"  
    DISPLAY MNTEC.  
    STOP RUN.  
IDENTIFICATION DIVISION.  
PROGRAM-ID. SUBPGM.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
PROCEDURE DIVISION.  
    DISPLAY "ENTERING SUBPROGRAM".  
    DISPLAY MNTEC.  
    MOVE "ENDING CONTENTS" TO MNTEC.  
    DISPLAY "LEAVING SUBPGM".  
    EXIT PROGRAM.  
END PROGRAM SUBPGM.  
END PROGRAM MPGM.
```