

# Creation of XRef Data

XRef data is written for Assembler, COBOL, or PL/I programs. The following general rules apply when writing XRef data for these types of programs:

- The programs are declared as programs of type P.
- One XRef record is written for each program with the member name as the entry into the program.
- XRef data is written for the file and each field in the file that are accessed in the program using an EXEC ADABAS GENERATE or a COPY statement.

## What Types of Code Use are Documented in XRef Data

XRef data is written for the following types of code use:

- CALL and ENTRY statements. For COBOL and PL/I, the Preprocessor scans the source code for these statements.  
For Assembler (BAL, 370-Assembler), preprocessor statements have to be included in the source code to identify the call of external programs and functions.
- Use of copy code and include code.
- EXEC CICS statements: All Assembler, COBOL and PL/I programs are scanned for several types of EXEC CICS statements.

The different types are described in the sections below.

### Documenting CALL Statements and Entry Points

COBOL or PL/I programs are scanned for CALL statements and entry points.

The following rules apply:

- The name of the entry or external program can be up to 8 characters long.
- In COBOL a maximum amount of 50 entries and 300 called modules are allowed.
- A COBOL entry point is defined by an ENTRY statement.
- A PL/I entry point is defined as the label before a PROC, PROCEDURE or ENTRY statement.
- Equalization of module/entry names in PL/I will be performed. If an external program and an entry are found with both the same name, the existence of an internal procedure will be presumed.
- Assembler programs will be scanned for the macros CALL and CM\$CALL. The names following these statements will be written as the external member names. The entry points and called external programs may also be specified in separate preprocessor statements.

### Documenting the Use of Copy Code

All Assembler, COBOL and PL/I programs are scanned for copy code.

- Assembler and COBOL programs are scanned for the COPY statement; the first eight characters of the word following the COPY statement (in the same line) are stored as copy code name.
- PL/I programs are scanned for the INCLUDE statement. All copy/include code names within an INCLUDE statement are stored as XRef data.

## Documenting EXEC CICS Statements

All Assembler, COBOL and PL/I programs are scanned for a variety of EXEC CICS statements. The following types of EXEC CICS statements are found and documented:

- EXEC CICS LINK PROGRAM ('literal')
- EXEC CICS XCTL PROGRAM ('literal')
- EXEC CICS LOAD PROGRAM ('literal')
- EXEC CICS SEND MAP ('literal')
- EXEC CICS RECEIVE MAP ('literal')
- EXEC CICS HANDLE ABEND PROGRAM ('literal')

The literal will be written into XRef data as the external member name. If the literal is longer than eight characters, or the brackets contain a variable without apostrophes or quotes, \*DYNAMIC will be written.