HONEYWELL

SOFTWARE BULLETIN

SERIES 200

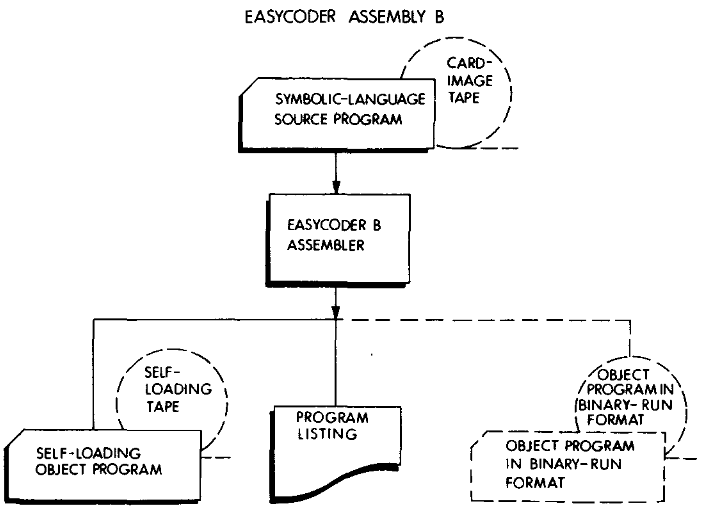
EASYCODER B ASSEMBLY SYSTEM

The Easycoder B Assembly System is a comprehensive package of assembly, operating, and utility programs which have been developed to simplify programming and to minimize operator tasks. Easycoder B is designed for card-oriented applications on a Series 200 computer having at least 8,192 characters of main memory. Magnetic tape drives can be used for increased speed, but they are not required, except by the Update program. Object programs assembled by the Easycoder Assembler B can be run on any Series 200 machine which offers sufficient equipment for the program.

The primary elements of the Easycoder B Assembly System are the Easycoder symbolic language and Easycoder Assembler B. Source programs written by the programmer in the symbolic language are translated by the assembler into machine-language programs. These object programs, on cards or tape, are in condensed, self-loading format. Optionally, object programs may be produced in a binary-run format for use with a card or tape loader. The assembler produces a printed listing of the user's program if desired.

The Easycoder B Assembly System also includes programs and routines which perform the following functions:

1. **Library Processing.** The Library Processor B program facilitates programming by allowing the programmer to utilize precoded program segments (macro routines) in his program. These macro routines consist of frequently used sequences of instructions in a generalized form. They are stored in a library file for easy reference. Library Processor B accepts macro calls which the programmer writes in his program, obtains the generalized macro routines called, specializes them for each program, and inserts them into an Easycoder symbolic program. Library Processor B can be used to specialize Honeywell-supplied library routines and/or generalized routines written by the user.
2. **Loading.** During assembly, the Card Loader B routine is generated as an integral part of each object program. This loading routine provides self-loading capabilities through a bootstrap area. A search and load routine generated by Update A enables the self-loading of object programs from magnetic tape. Card Loader-Monitor B is available for programs assembled in binary-run format. This loader provides programmed control of sequencing programs in a job.
3. **Updating.** During updating, a file of programs on a self-loading tape is created and maintained. The Update programs produce new files or update already existing files by adding new programs, copying existing programs, or deleting obsolete programs. Two Update programs are available. Update A generates a Search routine at the beginning of each tape. This routine locates and initiates loading of specified programs from the tape. Update B generates a Search routine at the beginning of each program on the file. Through the use of this routine, rewind time is diminished during searching. Update B has the ability to process programs in binary-run format as well as those in self-loading format. Thus, Update B provides compatibility with programs prepared by COBOL Compiler B and other Honeywell software packages. A directory listing is an optional output of both Update A and Update B.


EASYCODER ASSEMBLY B
SYMBOLIC-LANGUAGE SOURCE PROGRAM
CARD-IMAGE TAPE
EASYCODER B ASSEMBLER
SELF-LOADING TAPE
SELF-LOADING OBJECT PROGRAM
PROGRAM LISTING
OBJECT PROGRAM IN BINARY-RUN FORMAT
OBJECT PROGRAM IN BINARY-RUN FORMAT

EASYCODER SYMBOLIC LANGUAGE

The Easycoder symbolic language is versatile, powerful, and easy to use. It includes three types of symbolic statements:

1. **Data Formatting Statements.** These statements direct the assembler in setting up constants and reserved memory areas and in punctuating memory to indicate field boundaries.
2. **Assembly Control Statements.** Through the use of these statements, the programmer directs the assembler to perform a wide variety of functions related to the process of assembling an object program.
3. **Machine Instruction Statements.** These symbolic instructions are converted by the assembler into the machine-language commands of an object program.

(Continued on reverse side)

122.1105.000B.0-044
10666
Printed in U.S.A.

When ordering this publication please specify Title and Underscored portion of File Number.

Honeywell
ELECTRONIC DATA PROCESSING

DA statements are a useful feature of the Easycoder symbolic language. Through the use of DA statements, one or more contiguous areas in memory can be reserved. Within these areas, fields may be defined and punctuation generated. Automatic indexing of addresses can also be indicated. DA statements are particularly useful in areas such as double buffering.

Literals can also be used in the Easycoder language. Literals enable the programmer to write in the operands field of a symbolic statement the actual alphanumeric, binary, octal, or decimal *data* (as opposed to the *address* of the field containing the data) to be operated on by an instruction. Area-defining and address literals may be employed also. An area-defining literal defines and reserves a working area in memory without using a separate data formatting statement. An address literal enables the programmer to specify a symbolic address in the operands field of an instruction in such a way that the assembler will use the address as an operand.

In addition to the above capabilities, the Easycoder language provides both indexed and indirect addressing.

EQUIPMENT REQUIREMENTS

The Easycoder B Assembly System operates on a central processor with 8,192 characters of main memory, a card reader, and a card punch. Additional memory increments, which increase the space available for tag tables, can be added up to a maximum of 32,768 characters. Magnetic tape drives may be substituted for the card devices if desired. A printer is optional.

Update A and B require one tape drive, and Library Processor B requires the Advanced Programming Instructions.