

---

Honeywell

SOFTWARE RELEASE BULLETIN

---

SERIES 60 LEVEL 66

SOFTWARE RELEASE BULLETIN

FOR

CP-6 APL, RELEASE 1.0

PRODUCT NUMBER

SFL 6121

JUNE, 1979

© 1979, Honeywell Information Systems Inc.

CONTENTS

		Page
Section I	Introduction Supporting Documentation	1-1
Section II	General Information General Description	2-1 2-1
Section III	Installation Instructions	3-1
Section IV	Tips and Techniques	4-1
Section V	Known Errors	5-1

SECTION I

INTRODUCTION

This Software Release Bulletin (SRB) documents release 1.0 of the CP-6 APL processor.

This product operates on Control Program - Six (CP-6) software on Series 60 Level 66 hardware.

SUPPORTING DOCUMENTATION

The following document describes this product:

Order Number

Title

LADC 79-6

APL Language Reference Manual

## SECTION II

## GENERAL INFORMATION

GENERAL DESCRIPTION

APL is an interactive programming language designed for scientific and engineering applications. In addition to being a powerful language, APL is easy for the beginner to learn and use. APL has also found ready acceptance in many accounting application areas.

## Highlights:

- Operates under CP-6 as a shared processor
- Compatible superset of the IBM APLSV program language
- Batch or on-line operation
- Operation from CP-6-supported terminals without APL character set
- APLSV features
- File I/O capability
- Fast formatted output
- Compound statements
- Transparent I/O
- All normal CP-6 functions available at an APL terminal

SECTION III

INSTALLATION INSTRUCTIONS

CONTENTS OF DISTRIBUTION

This product is distributed in the form of a Run Unit and is included as part of the CP-6 Release Tape. The Run Unit is identified as:

File Name:           APL  
Account   :           :SYS

## SECTION IV

## TIPS AND TECHNIQUES

This section contains information to assist the site in the initial installation of this product.

1. The CP-6 APL Processor, in its original BETA test release form, is an absolutely plain run unit with well demarcated procedure and data sections, no special privileges and no overlays. Subsequent releases will modify this to add selected privileges and a single level overlay structure. Any standard procedures used to install processors in the :SYS account are applicable to APL.

2. The ZA6 module contains patch space as follows:

ZA6\$PATCH1	(procedure)	100 words
ZA6_PATCH0	(static )	120 words
ZA6_PATCH6	(constant )	20 words

Standard RUM patching mechanisms will be employed.

3. Since sealed workspaces are given access controls which specify APL as the access vehicle, it is inadvisable to install APL in accounts other than :SYS. Additionally, when processor privileges are added, they will be unavailable unless the run unit resides in :SYS.

SECTION V

KNOWN ERRORS

This section contains a description of the known errors in this release.

<u>ERROR</u>	<u>SEVERITY</u>	<u>IMPACT</u>
Response to asynchronous events is incomplete	2	Disconnect, over limits, error situations, ZAP and (to a lesser extent) breaks can result in a damaged workspace.
Shared Variables	2	Results in a "Nonce" error.
Processor not yet overlaid	2	Procedure requires 55K as opposed to planned 25K. Status data is 3K per user as opposed to planned 2K.
Slow Performance	2	Most operators are slower than CP-V counterpart even on 560.
Maximum memory feature authorization not yet available to the processor	3	Oversized workspace files cannot be copied into an already large workspace when user has a restricted memory allocation.

---

**Honeywell**

**SOFTWARE RELEASE BULLETIN**

---

SERIES 60    LEVEL 66

SOFTWARE RELEASE BULLETIN

FOR

CP-6 BASIC, RELEASE 1.0

PRODUCT NUMBER

SFL 6122

JUNE, 1979

© 1979, Honeywell Information Systems Inc.

## CONTENTS

		Page
Section I	Introduction Supporting Documentation	1-1
Section II	General Information General Description	2-1 2-1
Section III	Installation Instructions	3-1
Section IV	Tips and Techniques	4-1
Section V	Known Errors	5-1
Appendix A	Updates to Reference Manual	A-1

SECTION I

INTRODUCTION

This Software Release Bulletin (SRB) documents release 1.0 of the CP-6 BASIC processor.

This product operates on Control Program - Six (CP-6) software on Series 60 Level 66 hardware.

SUPPORTING DOCUMENTATION

The following document describes this product:

Order Number

Title

LADC 78-9

BASIC Language Reference Manual

## SECTION II

## GENERAL INFORMATION

GENERAL DESCRIPTION

BASIC is a programming language designed to be easy to teach, learn and use, allowing users with little or no programming experience to create, debug and execute programs via an on-line terminal. Such programs are usually small to medium sized applications of a computational nature. The CP-6 implementation is a unified processor consisting of a root or service module in continuous use, a compiler used during object program generation, and an execution support module used during program execution and utility operations. It is a compiler, not an interpretive processor, but differs from most compilers in that object programs are generated directly into users storage space and a separate 'load' process is bypassed. In this implementation the compilation and execution processes are in communication via the continuously resident root module.

## Highlights:

- Conforms to minimal ANS BASIC standard
- Conversational
- Immediate syntax checking of input
- Can be used in both on-line and batch modes
- Editing and compilation/execution modes of operation
- Direct execution capability
- Fast in-core compilation
- Efficient object code generation
- Minimum compiler core requirements
- Safe/fast compile option
- Extensive print editing and formatting capability
- Extensive intrinsic functions and matrix operations
- Computed GOTO capability (multiple-path branching)
- File input/output capability
- Chaining facility
- Variable program size
- Compilation and run-time diagnostics
- Restore statement for re-use of data
- Multiple line functions allowed
- No fixed limit to the number of constants
- Computation carried out in double precision or single precision at user option
- Chaining facility includes context preservation

SECTION III

INSTALLATION INSTRUCTIONS

CONTENTS OF DISTRIBUTION

This product is distributed in the form of a Run Unit and is included as part of the CP-6 Release Tape. The Run Unit is identified as:

File Name: BASIC

Account: :SYS

SECTION IV

TIPS AND TECHNIQUES

This section contains information to assist the site in the initial installation of this product.

1. Place run unit BASIC in :SYS to make it an unshared processor.
2. Patch space begins at the symbol 'O'  
One hundred words are provided.

## SECTION V

## KNOWN ERRORS

This section contains a description of the known errors in this release.

<u>ERROR</u>	<u>SEVERITY</u>	<u>IMPACT</u>
1. User Defined Function capability not checked out	1	This feature may not work properly
2. String arrays in double precision mode are not properly dynamically redimensioned	2	String arrays in single precision will work
3. DET function not checked out	2	
4. The statement "SHOW DIM" with the blank following the DIM gives an error	3	Without blank works
5. Multiple use of AND and OR does not compile properly	3	Change to simple logical statements which work properly
6. IF <relation statement> THEN assignment does not compile properly	3	Statement compiles properly if LET is used before the assignment
7. Compiler has not been checked out in shared mode	3	Compiler works in unshared mode

APPENDIX A

UPDATES TO REFERENCE MANUAL

The following pages represent updates to the CP-6 BASIC Language Reference Manual, Publication number LADC 78-9 dated December, 1978. They contain descriptions of several new features that were implemented since the manual was printed. These features are in the following categories:

- o New Edit Commands
- o New BASIC Statements
- o New BASIC Functions

## New Edit Commands

1. AUTO

The AUTO command causes automatic line numbering. If an error is made in typing a statement, the line number is repeated. Typing an unaccompanied carriage return terminates the operation. The syntax is:

```
AUT[O] [n][,m]
```

where

n is the starting line number.

m is the increment.

2. BATCH

The BATCH command enters jobs into the batch stream. If no command is specified, a RUNH command is inserted, following the command to load the specified file.

```
BAT[CH] [PW = password] [, (command)] fid
      [, "JOB CARD options" ] , ["resource card"]
```

3. BUILD

The BUILD command creates a file that contains embedded commands and direct statements. The NSAVE, NRESAVE, NSEAL, FETCH, CFETCH and SEQUENCE commands are designed to operate on build-mode type files. The BUILD prompt is a line number followed by a colon. BUILD mode programs can only be edited using the BUILD command. The user may type BUILD followed by the line number to be replaced. When BUILD prompts with the line number, the user should type the new line. Escape from BUILD mode is via a null line (carriage return). The syntax is:

```
BUI[LD] [n] [,m]
```

where

n is the starting line number.

m is the increment.

#### 4. Build Mode Commands

The following commands are for use in conjunction with BUILD mode.

##### a. NSAVE

The NSAVE command saves a file without line numbers. The syntax is:

```
NSAVE]   fid
```

##### b. NRESAVE

The NRESAVE command resaves a file without line numbers. The syntax is:

```
NRE[SAVE]   fid
```

##### c. FETCH

The FETCH command loads a file that does not have line numbers. The fetched file must have been build by EDIT or BASIC. The line numbers generated correspond to the key of the record read.

```
FETCH]   fid
```

##### d. CFETCH

The CFETCH command perform a clear and load of a non line-number file.

```
CFE[TCH]   fid
```

##### e. NSEAL

The NSEAL command seals and saves a non line-numbered file. The syntax is:

```
NSE[AL]   fid
```

f. SEQUENCE

The SEQUENCE command renumbers (with pseudo line numbers) a non line-numbered file (see BUILD). Although these files do not have line numbers in their 'saved' format, pseudo line numbers are appended when the program is 'fetched'. The syntax is:

```
SEQ[UENCE] [n][,m][,r]
```

where

n is the starting line number m is the increment

r is the range (see LIST)

5. CALC

The CALC command allows entry of non-line numbered statements. Expressions are evaluated and printed when they are typed. The calculator mode prompt is an asterisk. An unaccompanied carriage return terminates the operation. The syntax is:

```
CAL[C]
```

6. CATALOG

The CATALOG command list either the BASIC created files in a given account or all the files, regardless of type.

```
CAT[ALOG][ACCT = account] [,ALL]
```

where

account specifies the account to be listed

ALL specifies that all file types are to be listed

7. END

The END command returns the user to IBEX (see SYS).

8. FASTER/FASTEST

The FASTER or FASTEST commands cause a compile and execute with greater levels of code optimization than either RUN or FAST. The syntax is:

```
FASTER[H]
```

```
FASTEST[H]
```

9. GO

The GO command resumes execution after a break condition.  
The syntax is:

GO

Modified Edit Commands

RENUMBER

A range specification (same as LIST) has been added to  
RENUMBER. The syntax is:

REN[UMBER][n],[m],[r]

n is the starting line number.

m is the increment.

r is the range.

## New BASIC Statements

1. IBEX

The IBEX statement passes the input string to IBEX for execution as an IBEX command. The syntax is:

```
[line] IBEX str-expr
```

2. IF...THEN...ENDIF

A block structured capability has been added to IF...THEN. The syntax is:

```
IF   expr   THEN
    s1
    s2
    :
    sn
ENDIF
```

Note: IFs may not be nested.

3. IF...THEN...ELSE

A block structured else clause has been added to IF...THEN. The syntax is:

```
IF   expr   THEN
    s1
    s2
    :
    sn
ELSE
    s1
    s2
    :
    sn
ENDIF
```

Note: An ELSE clause may not contain an IF.

4. LINK

LINK allows transfer to any load module in the system. A command line can also be passed. The syntax is:

```
[line] LINK str-expr [, str - exp]
```

5. PROMPT

PROMPT will set the runtime prompt character to any string less than 32 characters long. The syntax is:

```
[line] PROMPT str-expr
```

6. WHEN

Two new forms of the WHEN statement have been added. The syntax is:

```
[line] WHEN MONERR GOTO- line | ERROR
[line] WHEN LIBERR GOTO line | ERROR
```

WHEN MONERR is similar to WHEN ERROR except that it applies to CP-6 monitor errors only.

WHEN LIBERR is similar to WHEN ERROR except that it applies to CP-6 common library errors only.

## Modified BASIC Statements

## 1. Successive involution.

Successive involutions are evaluated left to right as required by the ANS standard. Parenthesis are not required.

## 2. PRINT lists

A print list cannot start with a colon.

New BASIC Functions

1. DET

DET returns the determinant of a matrix. The syntax is:

DET (aname)

2. Error functions

Three new functions have been added to augment the WHEN statement.

ERT - returns the type of error:

- 1 = Monitor error.
- 2 = Common library error.
- 3 = BASIC runtime error.

LBE - returns the correct library error number.  
Prints zero if no error.

MNE - returns the correct monitor error number.  
Prints zero if no error.

Honeywell

SOFTWARE RELEASE BULLETIN

SERIES 60 LEVEL 66

SOFTWARE RELEASE BULLETIN

FOR

CP-6 FORTRAN, RELEASE 1.0

PRODUCT NUMBER

SFL 6120

JUNE, 1979

© 1979, Honeywell Information Systems Inc.

CONTENTS

		Page
Section I	Introduction Supporting Documentation	1-1
Section II	General Information General Description	2-1 2-1
Section III	Installation Instructions	3-1
Section IV	Tips and Techniques	4-1
Section V	Known Errors	5-1

SECTION I

INTRODUCTION

This Software Release Bulletin (SRB) documents release 1.0 of the CP-6 FORTRAN processor.

This product operates on Control Program - Six (CP-6) software on Series 60 Level 66 hardware.

SUPPORTING DOCUMENTATION

The following document describes this product:

Order Number

Title

LADC 79-9

FORTRAN Language Reference Manua

## SECTION II

## GENERAL INFORMATION

GENERAL DESCRIPTION

This language processor is a comprehensive algebraic programming language consisting of a compiler and an extensive library of subprograms. It is a superset of most available FORTRAN languages, containing many extended language features to facilitate program development and checkout. The compiler is designed to produce efficient object codes, thus reducing execution time and core requirements.

## Highlights:

Compatibility with American National Standard (ANS)  
-The language elements are compatible with the ANS FORTRAN 77 specification. Significant features include:

IF-THEN-ELSE statement

CHARACTER variables

Expanded I/O capabilities with OPEN, CLOSE, and INQUIRE statements.

## Features and Benefits:

Extended Language Features - Contains many language facilities, not found in other FORTRAN systems, which provide more flexibility and ease of use in program development. The extended features are:

Extended assignment statements  
N-dimensional arrays with generalized bounds  
Expressions in output lists  
Global variables  
Generalized DATA statement  
Generalized End-of-File processing  
Generalized REPEAT statement  
Automatic double precision  
Generalized subscripts  
Compound statements  
Global statement labels  
Direct I/O (define file)

INSTALLATION INSTRUCTIONS

CONTENTS OF DISTRIBUTION

This product is distributed in the form of a Run Unit and is included as part of the CP-6 Release Tape. The Run Unit is identified as:

File Name:       FORTRAN

Account:         :SYS

SECTION IV

TIPS AND TECHNIQUES

This section contains information to assist the site in the initial installation of this product.

The FORTRAN 77 compiler is a shared processor. It can be run as either a strung out (non overlaid) processor, or as an overlaid processor. The overlaid version of the processor is about 1/3 the size of the non overlaid version of the processor. However, if the FORTRAN 77 compiler is, typically, a heavily used processor on the CP-6 system, it would probably be more advantageous to use the strung-out version of the compiler instead of the overlaid version of the compiler (faster terminal response time-no overhead for fetching of compiler overlays).

## SECTION V

## KNOWN ERRORS

This section contains a description of the known errors in this release.

<u>ERROR</u>	<u>SEVERITY</u>	<u>IMPACT</u>
Checkout of the Code Generator has not been completed	1	This product will be completed and released on August 30, 1979

---

**Honeywell**

**SOFTWARE RELEASE BULLETIN**

---

SERIES 60    LEVEL 66

SOFTWARE RELEASE BULLETIN

FOR

CP-6 TEXT, RELEASE 1.0

PRODUCT NUMBER

SFP 6121

JUNE, 1979

© 1979, Honeywell Information Systems Inc.

CONTENTS

		Page
Section I	Introduction Supporting Documentation	1-1
Section II	General Information General Description	2-1 2-1
Section III	Installation Instructions	3-1
Section IV	Tips and Techniques	4-1
Section V	Known Errors	5-1

SECTION I

INTRODUCTION

This Software Release Bulletin (SRB) documents release 1.0 of the CP-6 TEXT processor.

This product operates on Control Program - Six (CP-6) software on Series 60 Level 66 hardware.

SUPPORTING DOCUMENTATION

The following document describes this product:

<u>Order Number</u>	<u>Title</u>
LADC 79-8	TEXT Reference Manual

SECTION II

GENERAL INFORMATION

GENERAL DESCRIPTION

TEXT is the text formatter of the CP-6 text processing system that provides the capability to create, edit, and print documents through typewriter-like terminals. The TEXT command language is simple to use and is oriented toward the nonprogrammer. Secretaries, technical writers, and others not familiar with computers can learn to use TEXT in a short period of time.

Highlights:

TEXT is a time and money saving aid in document preparation, which features:

- Automatic margin justification
- Automatic page numbering
- Rapid accurate proof preparation
- Reduced proofreading time
- Rapid reprint capability
- Rapid access and modification of data
- Password protection of documents as required

SECTION III

INSTALLATION INSTRUCTIONS

CONTENTS OF DISTRIBUTION

This product is distributed in the form of a Run Unit and is included as part of the CP-6 Release Tape. The Run Unit is identified as:

File Name:           TEXT  
Account   :           :SYS

Two data files are identified as:

File Name:           :ZTERRMSG  
Account   :           :SYS  
File Name:           :ZTH\$HELP  
Account   :           :SYS

The file :ZTERRMSG contains the TEXT error messages. The file :ZTH\$HELP contains the data for the HELP option of TEXT.

SECTION IV

TIPS AND TECHNIQUES

This section contains information to assist the site in the initial installation of this product.

TEXT is installed by booting the CP-6 release tape. No special installation actions are necessary.

For differences between CP-V TEXT and CP-6 TEXT see the CP-6 Conversion Guide Section 4.13.

For transfer of CP-V TEXT files see the CP-6 Conversion Guide Section 6.7.

To obtain a brief description of all the TEXT formatting commands, enter the following:

! TEXT X ON ME (HELP)

## SECTION V

## KNOWN ERRORS

This section contains a description of the known errors in this release.

- |                                                                                                                                                            |   |                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----------------------------------------------------------|
| 1. The output to the line printer is not yet fully supported.                                                                                              | 3 | The line printer can only be used for proofreading copies |
| 2. If an option with a parameter is specified e.g. (HELP=CK) an erroneous error message is printed.                                                        | 3 |                                                           |
| 3. The parameters A, E, and O on header blocks and footer blocks cannot be used in arbitrary sequence. Either A can be used <u>or</u> E and O can be used. | 3 | None. It will be mentioned in the manual                  |

SERIES 60 LEVEL 66  
SOFTWARE RELEASE BULLETIN

FOR

SORT/MERGE, RELEASE 1.0

PRODUCT NUMBER

JUNE, 1979

(c) 1979, Honeywell Information Systems, Inc.

## CONTENTS

Section I	Introduction Supporting Documentation	Page 1
Section II	General Information General Description	2
Section III	Installation Instructions	2
Section IV	Techniques for Using SORT/MERGE	3
Section V	Known Errors	4
Section VI	Shortfalls	4

## SECTION I

## INTRODUCTION

This Software Release Bulletin (SRB) documents release 1.0 of the CP-6 SORT/MERGE processor.

This product operates on Control Program - Six (CP-6) software on Series 60 Level 66 hardware.

SUPPORTING DOCUMENTATION

The following document describes this product:

Order Number

Title

Draft  
SORT/MERGE Reference Manual

## SECTION II

## GENERAL INFORMATION

The CP-6 Sort and Merge processors provide all of the functions planned for these products. Not all of these functions have been fully tested and some are limited pending future operating system developments.

Initial Release Limitations and Restrictions:

## Tape Processing

Although the Sort and Merge contain the logic to process tape files, this facility has not been thoroughly checked out.

## Label Processing

There are several restrictions on user label processing:

- 1) The structured definition of the label buffers for user label processing will become effective with Monitor version R. Until that time, the label buffer will contain only the array of labels and not the label count word.
- 2) User labels must be exactly 80 bytes long. There may be no more than four labels placed in the label buffer by the user labeling subroutine.
- 3) If the user labeling subroutine takes an alternate return, the Sort or Merge will come to an abnormal termination.

## SECTION III

## INSTALLATION INSTRUCTIONS

CONTENTS OF DISTRIBUTION

This product is distributed in the form of a Run Unit and is included as part of the CP-6 Release Tape. The Run Unit is identified as:

File Name:        SORT/MERGE

Account:            :SYS

SECTION IV

TECHNIQUES FOR USING SORT/MERGE

Creating Run-Units with Coexistent Sort:

User programs that are to co-exist with a Sort execution can be built into run-units in one of two ways:

- o Programs that can utilize the COBOL shared subroutine library may be linked as follows:

```
!LINK user-object-units,;  
  ZL_ELEMENT_OU,:LIBRARY,XUP$PARSE.:LIBRARY,;
```

```
  OVER user-run-unit (DCBSLOTS=34, SHAREDLIB=:SHARED_COBOL)
```

- o Programs that cannot utilize the COBOL shared subroutine library may be linked as follows:

```
!LINK user-object-units,;  
  ZL_ELEMENT_OU.:LIBRARY,ZL_SHARED_OU.:LIBRARY,;  
  XUP$PARSE.:LIBRARY,CSEQU.:LIBRARY;  
  OVER user-run-unit (DCBSLOTS=34)
```

Creating Run-units with Coexistent Merge:

User programs that are to co-exist with a Merge execution can be built into run-units in one of two ways:

- o Programs that can utilize the COBOL shared subroutine library may be linked as follows:

```
!LINK user-object-units,;  
  ZL_ELEMENT_OU.:LIBRARY, XUP$PARSE.:LIBRARY,;  
  OVER user-run-unit (DCBSLOTS=17, SHAREDLIB=:SHARED_COBOL)
```

- o Programs that cannot utilize the COBOL shared subroutine library may be linked as follows:

```
!LINK user-object-units,;  
  ZL_ELEMENT_OU.:LIBRARY,ZL_SHARED_OU.:LIBRARY,;  
  ZUP$PARSE.:LIBRARY,CSEQU.:LIBRARY;  
  OVER user-run-unit (DCBSLOTS=17)
```

SECTION V

KNOWN ERRORS

None at the present.

SECTION VI

SHORTFALLS

User label processing will not be available until the second beta site.

# Honeywell

## SOFTWARE RELEASE BULLETIN

SERIES 60 (LEVEL 66)/6000

COBOL-74

SOFTWARE RELEASE BULLETIN

June, 1979

Sec. I INTRODUCTION

Supporting Documentation  
Memory Requirements  
COBOL-74 Support of DM-IV

Sec. II GENERAL INFORMATION

General Description of COBOL  
COBOL-74 Interface with Systems Software  
Invoking the Compiler  
    Compiler Options  
    Linking the OU  
    Executing the RU  
    Run-time Facilities  
    File I/O  
    SORT/MERGE  
    RW  
    DEBUG  
    IPC (CALL/CANC) LIT only

Sec. III COBOL-74 EXTENSIONS

Multiple Entry Pt  
User Label Proc.  
Debug Schema  
Initialize  
Examine  
Block Clause  
Shorable User Object Code  
Read into VLR  
1 Character slew  
Data name up to 32 Ch

Sec. IV INSTALLATION PROCEDURE

---

Sec. V INVOKING THE COMPILER

Sec. VI COMPILER OPTIONS

Sec. VII KNOWN ERRORS

Sec. VIII SHORTFALLS FROM RELEASE 1.0 SPECIFICATIONS

Appendix A - Functionality Dependent on Language Interpretation

SECTION I

INTRODUCTION

This Software Release Bulletin (SRB) documents release CP-6.1 of the Series 60 (Level 66)/6000 COBOL (ANS COBOL X3.23-1974) compiler package. This compiler package is hereafter referred to as COBOL-74.

This release of COBOL-74 is supported by Series 60 (Level 66)/6000 hardware operating with CP-6 software.

SUPPORTING DOCUMENTATION

The following documents apply to this software release:

Order Number

Title

CP-6 COBOL-74 Reference Manual

MINIMUM MEMORY REQUIREMENTS

Systems on which COBOL-74 programs are to be compiled and executed must have at least 70 K words of memory.

COBOL-74 SUPPORT OF DM-IV DATA MANAGER

COBOL-74 is the host language compiler for the CP-6 Data Manager I-D-S/II. COBOL-74 is used to process I-D-S/II data when such data is in the COBOL-74 User Working Area (UWA). This working area exists only if I-D-S/II data is to be processed by the COBOL-74 program. Data is transferred in and out of the UWA by I-D-S/II routines.

SECTION II

GENERAL INFORMATION

GENERAL DESCRIPTION OF COBOL-74

COBOL is a programming language used throughout the world for programming business data processing applications. The COBOL language was developed by a group of computer users and manufacturers and the first documentation was distributed in April, 1960. Since then, COBOL has undergone many changes and extensions resulting from manufacturer experience with COBOL implementation and user experience with COBOL programming for Computers of many sizes and configurations. These changes and extensions are embodied in this version of the language termed ANS COBOL, X3,23-1974.

COBOL-74 is byte-oriented rather than word-oriented and all character data is internally processed as eight-bit characters rather than six-bit characters.

COBOL-74 INTERFACE WITH SYSTEM SOFTWARE

The COBOL-74 relationship with system software is as follows:

- o The batch operating system or Time Sharing control the various steps within the compilation activity.
- o Interface with source input MSSI.

- o Interface with job control language.
- o Interface with update lines M\$UI.
- o Places object code on object unit file M\$OU.
- o Object program interface with input/output system for logical and physical I/O.
- o Places diagnostics and hard copy information on system output M\$LO.
- o Interface with the Linker.
- o Interface with following intermediate files:
  - STAR1 Consecutive file
  - STAR2 Consecutive file
  - STAR3 Random file
  - 4STAR Consecutive file
  - 5STAR Consecutive file
  - GSTAR Consecutive file

#### COBOL-74 RUN-TIME FACILITIES

COBOL-74 object programs are executed in conjunction with major run-time support packages which provide the following COBOL-74 run-time facilities:

- o File Input/Output facility
- o SORT/MERGE facility.
- o Report Writer facility.
  
- o Debug facility.
- o Inter-Program Communication facility. (call "literal" only)

#### File Input/Output Facility

COBOL-74 uses CP-6 monitor file management services to accomplish the required file I/O for object program execution. The I/O functions provided by CP-6 are as follows:

- o Sequential I/O:

Provides the capability to access records of a file in established sequence. The sequence is established when the records are written to the file. Sequential I/O also provides for the specification of rerun points and the sharing of memory areas among files.

- o Relative I/O:

Provides the capability to access records of a mass storage file in either a random or sequential manner. Each record in a relative file is uniquely identified by an integer value greater than zero which specifies the record's ordinal position in the file.

- o Indexed I/O:

Provides the capability to access records of a mass storage file in either a random or sequential manner. Each record in an indexed file is uniquely identified by the value of a key within that record.

### SORT/MERGE Facility

The SORT/MERGE facility provides the capability to order one or more files of records, or to combine two or more identically ordered files of records, according to a set of user specified keys contained within each record. Optionally, a user may apply some special processing to each of the individual records by input or output procedures. This special processing may be applied before and/or after the records are ordered by the sort or after the records have been combined by the merge.

### Report Writer Facility

The Report Writer facility provides for producing formatted reports by specifying the physical appearance of the report rather than requiring specification of the detailed procedures necessary to produce the report.

A hierarchy of levels is used to define the logical organization of a report. Each report is divided into report groups, which in turn are divided into sequences of items. This hierarchical structure permits explicit reference to a report group with implicit reference to other levels in the hierarchy. A report group contains one or more items to be presented on one or more lines.

The Report Writer facility emphasizes organization, format, and contents of an output report. Although a report can be produced using the standard COBOL language, the Report Writer language characteristics provide a more concise method for report structuring and report production. Much of the Procedure Division coding which would normally be supplied by the user is instead supplied by the Report Writer Control System (RWCS). Thus, the user is relieved of writing procedures for moving data, constructing print lines, counting lines on a page, numbering pages, producing heading and footing lines, recognizing the end of logical data subdivisions, updating sum counters, etc. All of these operations are accomplished by the RWCS from source language statements that appear primarily in the Report Section of the Data Division of the source program.

### Debug Facility

The Debug facility assists in error detection by:

- o Monitoring transfers of control to user selected procedures during program execution.
- o Monitoring values of user selected data items during program execution.

The user supplied statements required to accomplish such monitoring are included in the source program and can be compiled or not depending on the presence or absence of the WITH DEBUGGING MODE clause in the source program. After the user statements have been compiled into the program, they can be executed or ignored at object program execution according to the setting of a run-time switch. The decisions concerning what to monitor and what information to display on the output device are explicitly at the discretion of the user. The main purpose of the COBOL Debug facility is to provide convenient access to such information.

### Inter-Program Communication Facility

Inter-Program Communication is a facility through which a program can communicate with one or more other programs. The communication is provided by the following:

- o The ability to transfer control from one program to another program within a run unit.
- o The ability for both programs to have access to the same data items.

SECTION III  
COBOL 74 EXTENSIONS

- o Initialize
- o Examine
- o Block Clause
- o Multiple Entry Point
- o Debug Schema
- o COBOL-74 CP-6 Options
- o Shareable User Object Code
- o CALL with Literal Name

MULTIPLE ENTRY POINTS

The user may specify an ENTRY phrase in the Procedure Division header of the Inter-Program Communication facility to allow multiple entry points to be defined in a COBOL-74 program. The syntax construct is as follows:

```
PROCEDURE DIVISION      [ USING data-name-1 [, data-name-2] ... ]  
      [ ENTRY section-name-1 [, section-name-2] ... ] .
```

The first six characters of the section-name specified in an ENTRY phrase must be unique in any given run unit of which the referenced section is a member.

If both the USING phrase and the ENTRY phrase are present in the Procedure Division header, the rules for calling a section-name within an ENTRY phrase are identical to those specified for calling a program-name.

THE CALL 'LITERAL' RESTRICTION

- o The CALL "literal" format must be used only to invoke subprograms.

Refer to Section XIV of the COBOL-74 User's Guide for additional information concerning inter-program communication.

SECTION IV  
INSTALLATION PROCEDURES

This software release is supported by Series 60 (Level 66) 6000 hardware operating with CP-6 software. Systems on which COBOL-74 programs are to be compiled and executed must have at least 70K words of memory.

CONTENTS OF DISTRIBUTION

The following magnetic tape comprises the physical element of this software release:

Software Library Number

Description

CP-6 Processor tape

CP-6 PROCESSOR TAPE

The CP-6 processor tape will contain the following files for COBOL:

<u>File Number</u>	<u>Description</u>
#1	COBOL-74 compiler in system loadable format
#2	COBOL-74 run-time subroutine object library
#3	COBOL/SORT shared library

## SECTION V

### INVOKING THE COMPILER

The compiler is invoked by using the standard invocation line. For example:

```
!COBOL fid_SI(,fid_UI) (over)
                    ( on ) (fid_OU,) (fid_LØ) (Options)
                    (onto)
```

where: fid\_SI designates a source input file,  
fid\_UI designates an update file,  
fid\_OU designates an output file,  
fid\_LO designates a listing file.

If an update file is specified, it must not be null.

If an output file is not specified, the output unit will be written to the \*G file.

If a listing file is not specified, the listing will be returned to the terminal if entered from time sharing or if the job is entered as a batch job, the listing will be sent to the on-line printer.

For further information concerning the standard invocation line reference the Programmer's Reference Manual.

## SECTION VI

## COBOL-74 COMPILER OPTIONS

The following list represents the CP-6 COBOL-74 compiler options. The (N) indicates that the option is not to be applied. (M) indicates the minimal set of the option. The underline indicates those options that are implied if no options have been indicated on the !COBOL JCL. If any options are specified, then all options that are desired must be explicitly declared. (Reference CP-6 Processor Standards and Conventions for detailed information on the CP-6 compiler options.)

<u>OPTION</u>	<u>DESCRIPTION</u>
(N) SO	new source file including updates
(N) <u>OU</u>	object unit is desired
(N) <u>LS</u>	source listing is desired
(N) LO	symbolic object listing desired
**	
(M) PM (AP)	procedure map list
(N)	
(M) DM (AP)	data map list
(N)	
(N) WA (RN)	list the warning messages
(M) XR (EF)	create a cross-reference listing
(N)	
SR (CH) (LIST)	library file search list
(N) UI	update the source
*	
(M) <u>SC</u> (HEMA)	create a debug schema
(N)	
(N) UR (EF)	list unreferenced data items

The following list represents the COBOL-74 CP-6 compiler options that are local to the COBOL-74 compiler.

<u>OPTION</u>	<u>DESCRIPTION</u>
COPY	copy statements are included in the source
DUMP	compiler dump if aborted
IDS	create an IDS sub-schema List
CLST	create a copy list
RESEQ	resequence the source in columns 1-6
REPLACE	a replace clause is present
DEBUG	the source program contains debug statements
SUB	the source program being compiled is a subprogram not a main program

If PMAP is declared, all Entry Labels, procedure labels, section labels, and individual source statements should have a corresponding relative memory location listed. If MPMAP is declared, unlabeled source statements should not have a corresponding location listed.

If DMAP is declared, all Data Division entries should have a corresponding relative memory location listed. If MDMAP is declared only level 01 or 77 data items will have the corresponding relative location listed.

If SCHEMA is declared, all schema items are included in the debug schema. If MSHEMA is declared, referenced schema items only will be included in the debug schema.

In the case where no options are indicated (i.e., !COBOL) the compiler should assume LS, OU, MSHEMA, NWARN. NWARN is always assumed unless specifically turned on (WARN).

If OU and NSHEMA are specified with DM, or PM as other options, the compiler will over-ride NSHEMA with SCHEMA. If OU and NSHEMA are specified with LO, MDM, or MPM as other options, the compiler will over-ride NSHEMA with MSHEMA.

SECTION VII

KNOWN ERRORS

1. If the user does not specify an OU file as the COBOL invocation line, the compiler is to default to the \*G file. At the current time if an OU file is not specified the compiler will abort. This error should not greatly impact the user since an OU file is normally specified. (Low severity problem)
2. If a COBOL program contains DISPLAY statements, and if the run unit of the program is executed from a remote terminal, the output from the DISPLAY statements will always be displayed on the remote terminal. This problem could impact users who want to direct this output to some other device or file. (Moderate severity level.)
3. The SO option causes the COBOL Compiler to produce binary output rather than the CP-6 format. This error will not allow users to create new source files that have been updated by UI files or COPY files. (High severity problem.)
4. Data names greater than 27 characters in length are not processed correctly by the compiler. This problem should not place a burden on the user. (Low severity problem.)
5. SWITCH STATUS - Several tests fail due to switch settings. The problem may be in the incorrect setting of switches or in the interpretation of the switches. (Low severity problem.)
6. Three tests fail due to the incorrect sign on a single character value. (Moderate severity problem.)
7. The compiler accounting is incorrect -- it always lists the figure of 240K of core as the space allocated to the processor. (Low severity problem.)
8. Call with "Literal Name" does not work for a test program. (Moderate severity problem.)

SECTION VII

SHORTFALLS FROM SPECIFICATIONS

- I. To be rectified or implemented between now and General Release 1.0.
  - A. Keyed file processing.
  - B. An M\$DO facility for outputting diagnostic messages.
  - C. Although use of the rerun, same area and multi-file clauses are inappropriate for CP-6, the same area clause should not produce an execution time error.
  - D. RESEQ option availability
  - E. User label processing to satisfy Sort and conversion considerations.
  - F. UNGMAP invocation by the compiler

APPENDIX A

FUNCTIONALITY DEPENDENT ON LANGUAGE INTERPRETATION

Two language items have been submitted to the American National Standard COBOL Committee (X3J4) and the Federal COBOL Interpretation Committee (FCIC) for resolution. The results of these interpretations may affect object code in future software releases. The two items are:

1. UNSTRING Statement

UNSTRING identifier DELIMITED BY ALL { identifier  
literal }

DELIMITER IN identifier-5.

The Standard is not clear whether a single occurrence of the delimiter is moved to the 'DELIMITER IN' field, or all contiguous occurrences of the delimiter are moved to that field. Currently, all contiguous occurrences of the delimiter are moved to the 'DELIMITER IN' field.

However, since the CODASYL Journal of Development (JOD) was recently changed, the interpretation may specify that only one occurrence of the delimiter can be moved to the 'DELIMITER IN' field.

In the example:

UNSTRING identifier-1 DELIMITED BY ALL  
"AB" INTO identifier-4 DELIMITER IN  
identifier-5, identifier-7.

Given: ID-1 PIC X(10) VALUE "12ABABABCD".  
ID-4 PIC X(8) VALUE SPACE.  
ID-5 PIC X(8) VALUE SPACE.  
ID-7 PIC X(8) VALUE SPACE.

Currently, the results would be:

ID-4 = 12~~bbbbbb~~  
ID-5 = ABABAB~~bb~~  
ID-7 = CD~~bbbbbb~~

Possible results after interpretation:

ID-4 = 12~~bbbbbb~~  
ID-5 = AB~~bbbbbb~~  
ID-7 = CD

Recommendation:

If use of the  
...ALL { identifier } ... DELIMITER IN ...  
          { literal }

phrase of the UNSTRING statement is required, test the contents of the identifier that receives the delimiter for both a single occurrence and multiple occurrences of that delimiter.

2. COPY Statement

The Standard is not clear whether or not the COPY statement is functional when it is used in a paragraph within the Identification Division (within a comment-entry). Currently, the compiler will attempt to process the COPY statement whenever it is encountered, including occurrences within a comment-entry.

An interpretation being studied by the X3J4 and FCIC committees suggests that if the COBOL word COPY does not immediately follow the paragraph header where a comment-entry may occur, the word COPY is part of the comment-entry and is therefore not functional. If COPY is the first word following the paragraph header within the Identification Division and it is syntactically correct, it is a functional COPY statement.

NOTE: The CODASYL COBOL language committee has deleted comment-entries from the JOD; however, this does not affect the current version of the Standard (1974).

Recommendation:

If use of the COPY statement is required within a paragraph of the Identification Division, ensure that the word COPY is the first word in the paragraph. (This suggestion applies only to paragraphs within the Identification Division.)

## S R B OUTLINE

### I. INTRODUCTION

- A. Purpose
- B. CP-6 Features
- C. Supporting Publications

### II. HARDWARE CONSIDERATIONS

- A. Central System
- B. Front End Processor

### III. RELEASE CONTENT

- A. PO Tape
- B. Processor tapes
- C. Known Problem Areas

### IV. INSTALLING CP-6

- A. Hardware preparation
- B. Booting
- C. Beginning Production

Ed

S R B

Section 1

Introduction

This Software Release Bulletin (SRB) describes the A00 release of Honeywell Control Program-Six (CP-6) for Honeywell Series 60 Level 66 mainframes. Included are summary descriptions of significant features, the hardware requirements, a detailed list of release contents with known problems designated, and a guide to installation of CP-6 on the (as-the?) target hardware.

CP-6 is a comprehensive, multi-use operating system designed for Honeywell Series 60 Level 66 mainframes configured with Honeywell minicomputers. Three operational modes of access are provided:

- o Time sharing
- o Batch processing
- o Remote processing

CP-6 supports these access modes with balanced service and no inherent emphasis on any single access mode. Programs do not require alteration to run in any particular access mode. These modes are designed to operate concurrently. Several programs utilizing different modes can be simultaneously resident in memory. The system design allows the user to select only the mode or modes required for a given task. CP-6 performs equally well whether a single mode or multiple modes are used. CP-6 functional elements are essentially the same for all programs, regardless of the access mode.

CP-6 time sharing provides a highly productive environment suited to online program development and debugging. Features include:

- o Up to 500 interactive time sharing terminals may be connected to the system.
- o Rapid access and response from CP-6 creates an atmosphere in which each time sharing user appears to have the entire system dedicated to his tasks.
- o Highly interactive response that is practically independent of system load.
- o Access to all types of peripheral devices.
- o Support of a wide variety of terminals.

- o Dynamic timing algorithms that can be set for each individual terminal.
- o Terminal access without translation, providing transparent control for special purpose devices.

CP-6 batch processing provides maximum utilization of system resources by minimizing conflicts in resource use. Features include:

- o Up to 500 batch jobs can be processed concurrently.
- o Batch jobs can be submitted from on-line terminals or remote work stations.
- o Batch jobs are channeled into the stream best able to handle the individual requirements of the job, consistent with throughput and resource constraints determined by installation management.
- o System recovery preserves the batch queue.

CP-6 remote processing provides flexible communication between CP-6 and a variety of remote terminals using synchronou<sup>s</sup> protocols. Features include:

- o Remote terminals and associated devices can be organized into work stations.
- o Work stations can range from a simple card reader/line printer combination to a complete large-scale computer system with an assortment of peripheral devices.
- o Communication with any supported device at one or several remote sites.
- o Any time sharing terminal can be used as the operator's console for a CP-6 work station
- o A CP-6 system can act as a central site to several remote terminals and as a remote terminal to other computers, simultaneously.
- o Dynamic modification of workstation definition during system operation.

#### CP-6 FEATURES — *entire*

CP-6 equals and exceeds the industry standards for performance, convenience, and cost-efficiency.

#### EASE OF USE:

- o A simple, yet comprehensive execution control language that is common to all access modes.
- o An extensive HELP facility that provides information about the system and its processors.
- o System default conventions that minimize the need for execution control commands.

- o Terminal personality that includes type-ahead, echoplex, and a variety of escape and control key-ins, providing an unparalleled interactive interface.
- o Quick terminal response.
- o Program and data file compatibility in all modes of access.
- o A comprehensive remote batch system that allows entry of jobs from a variety of terminals.
- o System I/O design allows the writing of handlers that permit access to special devices.
- o The installation can supply command processors to tailor system use to specific applications.

#### OPTIMIZED FILE MANAGEMENT:

- o A single central file management system.
- o Files are compatible across operating modes and language processors.
- o File access is device independent.
- o Graduated levels of file access security.
- o Comprehensive file backup system.
- o File integrity assured by system recovery.
- o Self-contained "sets" of disk packs provide removable public file segments.

#### AN EFFICIENT MONITOR:

- o Event-driven, priority-adjustable scheduler.
- o Full utilization of hardware addressing and security features.
- o Shared re-entrant programs and system processors.
- o A comprehensive, easily-accessed set of system services.
- o High I/O performance via tree-structure file indexes with several forms of I/O caches and program disassociated buffering.
- o Multiprogramming and multiprocessing.

#### MINIMAL OPERATIONS COST:

- o A small staff requirement for installation and system support.
- o Hardware is easy to maintain..
- o System recovery does not require operator intervention and automatically determines the appropriate level of recovery.
- o System can be run without an operator in attendance.
- o Online hardware diagnostics are available at time sharing terminals at both local and remote sites.

- o On-line remote access software debugging and patching facilities are available.
- o Full system use accounting.

#### SYSTEM HIGHLIGHTS:

- o An integrated performance monitor that measures system performance simultaneously with normal operation.
- o Majority of operating system and processing code is written in a high-level structured language (PL-6).
- o A modern, extensive data base management system that is interfaced with COBOL, APL, FORTRAN, PL-6, IDP, and assembly language.
- o Communication with other operation systems through ANS labeled tape and the HASP and 2780 communications protocols.
- o A sophisticated debugger that can be run in either the interactive or batch environment, and which possesses a comprehensive set of functions suitable for debugging FORTRAN, PL-6, COBOL, GMAP-6 and other language programs.
- o Common calling sequences generated by all languages, allowing programs written in several languages to be loaded and run together.
- o Up to 256 K word of procedure and data with up to 128 K <sup>words</sup> cards of data additional.
- o Superior hardware assures a secure environment.
- o More concurrent batch streams can be defined with priority, class, and dependent job scheduling.
- o A common command language controls on-line and batch jobs.
- o Any standard remote or local terminal can be defined as an operator console.
- o APL, BASIC, COBOL, FORTRAN, RPG-II and TEXT are improved to a level exceeding current commercial state-of-the-art.
- o Hierarchical budget accounting for control of system charges and usage.
- o Remote communications concentrators provide fast local response, error control over long lines, and economical use of lines via full-duplex protocols.
- o Comprehensive user documentation.
- o Additional file access methods, and the ANS labels and blocking methods are supported.

\*\*\*\*\*

## Supporting Publications

The following table identifies the reference manuals which describe the CP-6 Operating System. All of these manuals are new issue.

<u>Order No.</u>	<u>Title</u>	<u>LADC ID No.</u>
TBD	CP-6 Concepts and Facilities	78-1
TBD	CP-6 Programmer Reference Manual	79-10
TBD	CP-6 Monitor Services Reference Manual	79-3
TBD	CP-6 System Support Reference Manual	79-11
TBD	CP-6 Operators' Manual	79-12
TBD	CP-6 DELTA Reference Manual	79-9

The following table identifies reference manuals included with the initial release package; ordinarily, these manuals and the processors they describe are ordered separately.

<u>Order No.</u>	<u>Title</u>	<u>LADC ID No.</u>
TBD	CP-6 GMAP-6 Reference Manual	79-13
TBD	CP-6 FORTRAN 77 Language Reference Manual	79-7
TBD	CP-6 COBOL-74 Language Reference Manual	79-2
TBD	CP-6 SORT/MERGE Reference Manual	78-10
TBD	CP-6 I-D-S/II Programmer Reference Manual For COBOL-74 Users	79-4
TBD	CP-6 I-D-S/II Data Base Administrator Reference Manual	79-5
TBD	CP-6 Interactive Database Processor (IDP) Reference Manual	79-14
TBD	CP-6 TEXT Reference Manual	79-8
TBD	CP-6 BASIC Language Reference Manual	78-9
TBD	CP-6 APL Language Reference Manual	79-6
TBD	CP-6 Report Program Generator II (RPG II) Language Reference Manual and CP-6 Supplement	79-15
TBD	CP-6 PL-6 Language Reference Manual	78-5

## SECTION 3

## Release      Content

The CP-6 release package consists of this document, the supporting documentation, and two types of magnetic tapes; the first type, known as the PO tape, is a bootable copy of the CP-6 operating system, complete with the files and service processors required to make CP-6 operational; the second type, known as the processor tape(s), consists of the remaining service processors, various unsupported tools found useful, and language processors ordered by the customer.

The PO tape consists of the following:

- o AARDVARK, the bootstrap and mini-J10 system used to load CP-6 into memory.
- o Firmware for the tape, disk, and unit record subsystems. Firmware tapes are also distributed separately by Honeywell Field Engineering Division(FED).
- o A core image of the basic CP-6 monitor.
- o XDELTA, the system debugger and patch processor.
- o TIGR, the Table Initializing and Generation Routine. TIGR constructs tables for the actual hardware running CP-6 and other system tables related to installation defined parameters, e.g., total number of users allowed on the system.
- o GHOST 1, system initialization ghost. GHOST 1 reads the system ghost<sup>s</sup>, the service processors, and various control files from the PO tape.

The following system ghosts are present on the PO tape:

- o ELF            ( Error Logging Facility)    Logs system-detected errors on permanent storage.
- o FEPANLZ      Debugs and down-line loads the front end processors.
- o INSYM        Processes job control card decks and terminal-initiated batch jobs.
- o KEYIN        Processes input and output to the operator's console.
- o OUTSYM       Processes files destined for spooled devices, e.g., line printers and card punches
- o PRESCAN     Processes all batch JCL prior to submission to the multi-batch-scheduler.

- o SLUG (Startup and Logon User <sup>Shost</sup>) Starts up all batch, time sharing and ghost users on CP-6; log<sup>s</sup> on remote operator's consoles; multi-batch schedules.
- o TEMPLAT Creates the operator interface used by KEYIN.
- o RCVRZ<sup>2</sup> In the event of a general system crash or single-user abort, generates a dump file of pertinent data for later analysis.
- o PIG (Pack Initialization Ghost) Controls the mounting and access to the file management pack sets.
- o FROG (Front - end Operations Ghost) Controls flow of certain messages between the front-ends and the <sup>host</sup> last system.

The following service processors are on the PO tape:

- o CONTROL Displays and sets values for various system parameters which affect the performance and behavior of CP-6.
- o DELTA The general purpose program debugger. Performs patching of service processors.
- o EFT (Efficient File Transfer) The file maintenance processor.
- o IBEX (Interactive and Batch EXecutive) The command program for CP-6. It controls the flow of execution for all users of the system.
- o LOGON Initializes each user created by SLUG to default parameters established by SUPER.
- o PCL General purpose file transfer utility.
- o SPIDER Permits dynamic installation, deletion, and replacement of shared processors.
- o SUPER Defines valid logon ids and establishes maximum and default limits for users logging <sup>on</sup> in those accounts.
- o VOLINIT Initializes the headers on disk packs used by PIG and performs surface checks.

The following control files are present on the PO tape:

- o %FEP The down-line load file used by FEPANLZ to initialize the front end processors (FEP).

- o SFEPCTL Contains the FEP configuration data.
- o :OCKEY Contains the information defining the operator interface of KEYIN.
- o OC~~KEY~~~~S~~ Source file used to create the :OCKEY file.
- o :OCMSG Contains the information defining the format of output messages generated by KEYIN.
- o OC~~MSG~~~~S~~ Source file used to create the :OCMSG file
- o CONTROL~~S~~ Contains a set of CONTROL commands defining operational parameters and batch partitions.
- o :ERRMSG Contains the text of all error messages generated by the CP-6 operating system and its service processors.

\*\*\*\*\*

## KNOWN PROBLEM AREAS

This section of the SRB documents the currently known problem areas of CP-6; a problem is defined as a feature planned for general release which does not work in the present integration (QUAIL). The problems are broken down into three categories:

1. Planned for general release, not in QUAIL.
2. Planned for AΦΦ general release but deferred to later general releases.
3. FΦΦ CP-V features not in the AΦΦ CP-6 release.

The QUAIL problem list is in two sections, general capabilities and service processors.

## GENERAL CAPABILITIES

### Job Management

- o Full authorization checks for batch jobs <sup>and</sup> is not available.
- o Batch interface with system console does not have full complement of messages.
- o <sup>Lesson</sup> Fagan password scrambling is not performed.

### Accounting

- o Job-step accounting is not available.
- o Forms accounting is not available.
- o Job history file is not complete.
- o Accounting statistics during recovery are not generated.

### Resource Management

- o Allocation by attribute is not complete.
- o Online resource allocation is not available.
- o Authorization check is not performed.

### Communications

- o FEP information for STATS display is not available.
- o FEPCON, the front-end control processor, is not available.
- o 2780/3780 and binary (transparent) HASP support is not available.
- o Connection of CP-6 as a slave HASP workstation is not available.

- o Formatted printout of FEP dumps by FEPANLZ is not available.
- o L6XDELTA is not available.
- o A<sup>u</sup> synchronous features not available:
  - AUTOSAVE on line disconnect.
  - Autoband<sup>u</sup> detection of any but 300 or 1200 band<sup>u</sup> terminals.
  - Paper tape support.
  - Logical pagination.
  - ESC-W sequence.
  - Read time-out.

### Operat<sup>or</sup> Interface

#### File Management

#### Recovery

#### Online Diagnostics

- o POLTS does not support CP-6 unit record devices.
- o COLTS is not available.
- o MOLTS is not available.
- o PACT is not available.
- o Independent distribution of FED diagnostic and<sup>f</sup> firmware tapes is not available.
- o SYSCON is not available.

#### Error Lagging

- o CPU startup/shutdown records are not logged.
- o System identification records are not logged.
- o Timestamp records are not logged.
- o Diagnostic user is not logged.
- o Channel mailbox contents are not logged.
- o Resource exhausted entries are not logged.
- o MPC errors and statistics are not logged.
- o Single-bit correction errors are not logged.
- o Extended status on I/O errors is not logged.

I/O System RMA

- o Retry channel errors on different path is not available.
- o Retry on MPC errors is not available.
- o Thresholding of errors is not available.

SERVICE PROCESSORS

- o [ IBEX features not available on incomplete:
  - DISPLAY command options
  - HELP command is not available.
  - ACCEPT command is not available.
  - TERMINAL command options.
  - RELEASE command is not available.
  - RESOURCE command for online users is not available.
  - CHECK command is not available.
  - STATUS command is not available.
  - LIMIT command problem: <sup>TP indented</sup> If user requests a service limit greater than he is authorized for, a garbled error message is generated.
  - SET command problems:
    1. [ If more than one ACCESS option is specified, only the last is used.
    2. [ If multiple serial numbers are specified as part of the fid, only the last is used.
  - Interactive problems:
    1. [ If a command passed by MYC results in an error other than syntax, IBEX will look <sup>P</sup> on the error.
    2. [ Starting a program under DELTA while in control-Y mode causes IBEX to abort.
  - Calling sequence problems:
    1. [ If source field is not specified but object or listout fields are, the implicit DCB's are erroneously assigned.
    2. [ Lower case E or EDIT does not make it to the processor.

- GOTO problem:

If the GOTO command is issued from a terminal, the next command is ignored.

- BATCH command problem:

If the !JOB card in the file to be batched is continued, the command will not work.

- SAVE command problem:

Files for IBEX use are not protected from a !SAVE OVER file command.

o [ LINK features incomplete or not available:

- ~~Library~~ searching for unresolved references

- Temporary run units.

- Certain privileges for the target run unit.

- Assignment of DCBs to the related command line fields.

o [ PCL features incomplete or not available:

- Certain MODIFY command options.

- Tabs handling is not correct.

-C ME INTO IDSFIL does not work.

o [ RATES features incomplete or not available:

- No known problems.

o [ EDIT features incomplete or not available:

- No known problems

o [ CONTROL features incomplete or not available:

- No known problems.

o [ SYSCON features incomplete or not available.

*SYSCON is not available in QUAIL*  
- Mapping a virtual resource name into device, IOM number, and channel number.

o [ SUPER features incomplete or not available:

- Synchronous device TERMINAL definition

- Various PROFILE options

- STATION definition

- WSN definition

- DEVICE definition

- FORMS definition

- PIG interface for file management account authorization.
- LIST does not display all items which may be defined.
- o DEF features incomplete or not available:
  - DEF is not ~~included with~~ available in *QUAIL*
  - o VOLINIT
    - No known problems.
- o DELTA features incomplete or not available.:
  - Expression evaluation is very primitive
  - COBOL and FORTRAN specific input system
  - COBOL and FORTRAN array displays
  - Structure display with elements broken out
  - Floating point and decimal input
  - Data break points
  - Update and extension of attachments
  - Display of variable length items
  - ON CALLS tracing
  - Trace transfer with shared libraries
  - Display auto variables not in current frame
- o - RUM interface problems:
  1. ] DISPLAY does not work
  2. ] LET does not work
  3. ] DCB's, start address, and privileges can not be modified.
  - ANLZ interface problems:
    1. ] No results on M \$ ALIB return
    2. ] DISPLAY does not work
    3. ] LET does not work

\*\*\*\*\*

Deferred items originally planned for first release

- Terminal Coupling

CP-V features not in first release

- Transaction Processing
- Real Time
- Online echo of operator console messages pertaining to tapes.

CP-V features dropped from CP-6

- DCT-2000 remote batch terminal support
- 2741 support
- VISTA support
- All half-duplex modems and terminals