



中华人民共和国国家标准

GB/T 4092.11—1992

程序设计语言 COBOL 排 错 模 块

Programming language COBOL
Debug module

1992-08-04 发布

1993-05-01 实施

国 家 技 术 监 督 局 发 布

程序设计语言 COBOL
排错模块

GB/T 4092.11—1992

代替 GB 4092.11—83

Programming language COBOL
Debug module

1 排错模块引言

1.1 功能

排错模块提供了用户描述他的排错算法的方法，这些算法中包含一些在目标程序运行过程中数据项或过程被监测的条件。

决定监测什么以及在输出设备上显示什么信息由用户显式定义。COBOL 功能只是提供了有关信息的方便存取。

在标准 COBOL 的这一版本中视排错模块是过时成分，因为在标准 COBOL 的以后的修改版中要把它删掉。

1.2 级别特征

1 级排错提供了基本的排错功能，包括指明选择监测过程的能力。

1.3 语言概念

1.3.1 排错特征

支撑排错模块的 COBOL 语言特征是：

- a. 编译时的开关—WITH DEBUGGING MODE 子句。
- b. 目标时的开关。
- c. USE FOR DEBUGGING 语句。
- d. 特殊寄存器—DEBUG-ITEM。

1.3.2 特殊寄存器—DEBUG-ITEM

保留字 **DEBUG-ITEM** 是由支撑排错设施的实现者的代码自动产生的一个特殊寄存器的名字。每个程序只分配一个 **DEBUG-ITEM**。**DEBUG-ITEM** 中的下属数据项的名字亦为保留字。

1.3.3 编译时的开关

WITH DEBUGGING MODE 子句作为 **SOURCE-COMPUTER** 段的一部分写出。它用作写在程序中的排错语句的一个编译时开关。

当在程序中指出 **WITH DEBUGGING MODE** 子句时，所有的排错节按本标准本节中指出的那样编译。当未指出 **WITH DEBUGGING MODE** 子句时，所有的排错节作为注解行编译。

1.3.4 目标时的开关

目标时的开关动态地激活编译程序插入的排错代码。这个开关在程序中不可编址，它是在 COBOL 环境外受控制的。如果该开关合上，所有写在源程序中的排错语言有效。如果该开关关上，则描述的所有作用以及 **USE FOR DEBUGGING** 语句都被抑制。提供和取消这一功能并不需要源程序的重新编

译。

如果编译时在源程序中未指定 **WITH DEBUGGING MODE** 子句，则目标时的开关对目标程序的执行没有影响。

2 排错模块的环境部

2.1 WITH DEBUGGING MODE 子句

2.1.1 功能

WITH DEBUGGING MODE 子句指出所有排错节都要编译。如果未指出该子句，则所有排错节就如同是注解行一样编译。

2.1.2 一般格式

SOURCE-COMPUTER. [计算机名 [**WITH DEBUGGING MODE**].]

2.1.3 一般规则

(1) 如果在程序的配置节的 **SOURCE-COMPUTER** 段中指出了 **WITH DEBUGGING MODE** 子句，则编译所有的 **USE FOR DEBUGGING** 语句。

(2) 如果在一个程序的配置节的 **SOURCE-COMPUTER** 段中未指出 **WITH DEBUGGING MODE** 子句，则任何 **USE FOR DEBUGGING** 语句及所有相关的排错节作为注解行一样编译。

3 排错模块的过程部

3.1 一般描述

当 **COBOL** 源程序中出现排错模块的 **USE FOR DEBUGGING** 语句时，过程部中则包含申述过程。下面所示的为当出现 **USE FOR DEBUGGING** 语句时，过程部的一般格式。

PROCEDURE DIVISION.

DECLARATIVES.

{节名 **SECTION.**

USE FOR DEBUGGING 语句.

[段名.

[句子] ...] ...} ...

END DECLARATIVES.

{节名**SECTION.**

[段名.

[句子] ...] ...} ...

3.2 USE FOR DEBUGGING 语句

3.2.1 功能

USE FOR DEBUGGING 语句标识出将被相应的排错节监测的用户项。

3.2.2 一般格式

USE FOR DEBUGGING ON

过程名 1

ALL REFERENCES OF 标识符

过程名 1

过程名 1

ALL PROCEDURES

3.2.3 语法规则

(1) 排错节，如果指定的话，必须紧跟在 **DECLARATIVES** 首部后一起出现。

(2) 除在 **USE FOR DEBUGGING** 语句本身中外在排错节中不可有对其它非申述过程的任何引用。

(3) 出现在排错节组外的语句不可引用该排错节组中定义的过程名。

(4) 除 **USE FOR DEBUGGING** 语句本身外, 出现在某一给定排错节中的语句只可引用在不同的仅用 **PERFORM** 语句的 **USE** 过程中定义的过程名。

(5) 排错节中定义的过程名不可出现在 **USE FOR DEBUGGING** 语句中。

(6) 任一给定的过程名只可出现在一个 **USE FOR DEBUGGING** 语句中并且在该语句中也只能出现一次。

(7) **ALL PROCEDURES** 短语在一个程序中只可出现一次。

(8) 当指出了 **ALL PROCEDURES** 短语时, 在任何 **USE FOR DEBUGGING** 语句中都不可能指定过程名 1。

(9) 标识符 1 不可调用报表节中定义的数据项和计数器以外的任何数据项。

(10) 如果标识符 1 引用的数据项的数据描述款包含一 **OCCURS** 子句或者下属包含 **OCCURS** 子句的数据描述款, 则标识符 1 必须指定为不带通常所需要的下标或位标形式。

(11) 对专用寄存器 **DEBUG-ITEM** 的引用仅限于来自排错节的内部的引用。

3.2.4 一般规则

(1) 排错节的自动运行并不是由出现在排错节中的语句引起的。

(2) 当 **USE FOR DEBUGGING** 语句中指定了文卷名 1, 则该排错节在下列时刻执行:

- a. 在引用文卷名 1 的 **OPEN** 或 **CLOSE** 语句执行以后。
- b. 在 **READ** 语句执行以后 (并在其他指定的 **USE** 过程后), 但该 **READ** 语句并不是导致相关的 **AT END** 或 **INVALID KEY** 命令语句的执行。
- c. 在引用文卷名 1 的 **DELETE** 或 **START** 语句运行后。

(3) 当 **USE FOR DEBUGGING** 语句中指定了过程名 1 时, 则该排错节在下列时刻执行:

- a. 在命名过程的各次执行的紧前时刻。
- b. 引用过程名 1 的 **ALTER** 语句执行的紧后时刻。

(4) **ALL PROCEDURE** 短语使一般规则 3 中描述对程序中的每个过程名都有效, 除非这些出现在排错节内。

(5) 当指定 **ALL REFERENCES OF** 标识符 1 短语时, 则对显式引用标识符 1 的每个语句在下列每上时刻执行那个排错节:

- a. 对 **WRITE** 或 **REWRITE** 语句的情况, 为在那个 **WRITE** 或 **REWRITE** 语句执行的紧前并在此由于 **FROM** 短语存在而作的隐式的传送之后。
- b. 对带 **DEPENDING ON** 短语的 **GO TO** 语句的情况, 为控制转移的紧前并在控制转到的过程名相关的排错节执行之前。
- c. 对 **VARYING**、**AFTER** 或 **UNTIL** 短语引用标识符 1 的 **PERFORM** 语句的情况, 为在标识符 1 引用的数据项的内容每次初始化、修改或计算之紧后。
- d. 对其它 **COBOL** 语句的情况, 为在那个语句执行的紧后。

如果在不执行或未计算的短语中指定了标识符 1, 则相应的排错节不执行。

(6) 当指定的标识符 1 不带 **ALL REFERENCES OF** 短语, 则那个排错节在下列各个时刻执行

- a. 对显示引用标识符 1 的 **WRITE** 或 **REWRITE** 语句的情况, 则在 **WRITE** 或 **REWRITE** 语句执行的紧前, 并在由于 **FROM** 短语的存在引起的任何隐式传送之后。

b. 对 **PERFORM** 语句中 **VARYING**、**AFTER** 或 **UNTIL** 短语引用了标识符 1 的情况，则在标识符 1 引用的数据项的内容的各次初始化、修改或计算的紧后。

c. 在任何其它的显式引用并导致标识符 1 引用的数据项的内容变化的 **COBOL** 语句执行的紧后。

如果标识符 1 指定在未执行或未计算的语句中，则相应的排错节不执行。

(7) 作为单个语句的运行结果。对一个特定的运算分量，相应的排错节的执行不会多于一次，不管该运算分量在语句中使用了多少次。对导致引用的过程循环执行的 **PERFORM** 语句的情况，相应的排错节为每次循环而执行一次。

在一个命令语句中，对于排错的目的，一个命令动词的每次单个出现就标识为一个独立的语句。

(8) 当 **cd** 名 1 指定在 **USE FOR DEBUGGING** 语句中时，那个排错节在下列时刻执行：

a. 在引用 **cd** 名 1 的 **ENABLE**、**DISABLE** 和 **SEND** 语句执行之后。

b. 在引用 **cd** 名 1 的 **RECEIVE** 语句执行之后，该 **RECEIVE** 语句并不使得 **NO DATA** 命令语句执行。

c. 在引用 **cd** 名 1 的 **ACCEPT MESSAGE COUNT** 语句执行之后。

(9) 作为限定符对 **DEBUG-ITEM** 过程名 1 的引用，并不构成对上述一般规则描述的排错的那个项的引用。

(10) 与排错节的每次执行有关的专用寄存器 **DEBUG-ITEM**，提供了关于导致排错节执行的条件信息。**DEBUG-ITEM** 具有下列隐式描述：

01 **DEBUG-ITEM**.

02 **DEBUG-LINE** PICTURE IS X (6).

02 **FILLER** PICTURE IS X VALUE IS SPACE.

02 **DEBUG-NAME** PICTURE IS X (30).

02 **FILLER** PICTURE IS X VALUE IS SPACE.

02 **DEBUG-SUB-1** PICTURE IS S9999 SIGN IS LEADING SEPARATE CHARACTER.

02 **FILLER** PICTURE IS X VALUE IS SPACE.

02 **DEBUG-SUB-2** PICTURE IS S9999 SIGN IS LEADING SEPARATE CHARACTER.

02 **FILLER** PICTURE IS X VALUE IS SPACE.

02 **DEBUG-SUB-3** PICTURE IS S9999 SIGN IS LEADING SEPARATE CHARACTER.

02 **FILLER** PICTURE IS X VALUE IS SPACE.

02 **DEBUG-CONTENTS** PICTURE IS X (n).

(11) 在排错节的每次执行之前，**DEBUG-ITEM** 调用的数据项填以空格。然后对不属于 **DEBUG-ITEM** 的数据项的内容更新了，这种更新在控制传送到排错节的紧前按下列一般规则进行。在下列一般规则中未指出的数据项的内容为空格。

更新按照 **MOVE** 语句的规则完成，唯一的例外是到 **DEBUG-CONTENTS** 传送时，就好像是字符初等项到字符初等项的传送，并不进行数据从一种内部形式到另一种内部形式的转换。

(12) **DEBUG-LINE** 的内容是标识某一特定源语句的实现者定义的方法。

(13) **DEBUG-NAME** 包含导致执行排错节的名的前 30 个字符。

名的所有限制符在 **DEBUG-NAME** 中用字 'IN' 或 'OF' 隔开。如果有下标或位标的话，并不放入 **DEBUG-NAME**。

(14) 如果导致执行排错节的对某一数据项的引用有下标或位标，则各层的出现号必需分别存于 **DEBUG-SUB-1**、**DEBUG-SUB-2**、**DEBUG-SUB-3**。

(15) **DEBUG-CONTENTS** 是一足够大到容纳下列一般规则所需的数据的数据项。

(16) 如果程序中第一个非申述过程的第一执行导致执行排错节，则下列条件成立：

- a. **DEBUG-LINE** 标识那个过程的第一个语句。
- b. **DEBUG-NAME** 包含那个过程的名字。
- c. **DEBUG-CONTENTS** 包含 ‘**START PROGRAM**’。

(17) 如果 **ALTER** 语句中对过程名 1 的引用导致排错节的执行，则下列条件成立：

- a. **DEBUG-LINE** 标识引用过程名 1 的那条 **ALTER** 语句。
- b. **DEBUG-NAME** 包含过程名 1。
- c. **DEBUG-CONTENTS** 包含与 **ALTER** 语句的 **TO** 短语相关的可应用的过程名。

(18) 如果与 **GO TO** 语句执行相关的控制转移导致排错节的执行，则下列条件成立：

- a. **DEBUG-LINE** 标识其执行了把控制转到过程名 1 的 **GO TO** 语句。
- b. **DEBUG-NAME** 包含过程 1。

(19) 如果 **SORT** 或 **MERGE** 语句的 **INPUT** 或 **OUTPUT** 短语中对过程名 1 的引用导致排错节的执行，则下列条件成立：

- a. **DEBUG-LINE** 标识引用过程名 1 的 **SORT** 或 **MERGE** 语句。
- b. **DEBUG-NAME** 包含过程名 1。
- c. **DEBUG-CONTENTS** 包含：
 - 1) 如果对过程名 1 的引用在 **SORT** 语句的 **INPUT** 短语中，则包含 ‘**SORT INPUT**’。
 - 2) 如果对过程名 1 的引用在 **SORT** 语句的 **OUTPUT** 短语中，则包含 ‘**SORT OUTPUT**’。
 - 3) 如果对过程名 1 的引用在 **MERGE** 语句的 **OUTPUT** 短语中，则包含 ‘**MERGE OUTPUT**’。

(20) 如果从与 **PERFORM** 语句有关的控制机制的控制转移导致执行和过程名 1 有关的排错节的执行，则下列条件成立：

- a. **DEBUG-LINE** 标识引用过程名 1 的那个 **PERFORM** 语句。
- b. **DEBUG-NAME** 包含过程名 1。
- c. **DEBUG-CONTENTS** 包含 ‘**PERFORM LOOP**’。

(21) 如果过程名 1 是一个将被执行的 **USE** 过程，则下列条件成立：

- a. **DEBUG-LINE** 标识导致 **USE** 过程执行的那个语句。
- b. **DEBUG-NAME** 包含过程名 1。
- c. **DEBUG-CONTENTS** 包含 ‘**USE PROCEDURE**’。

(22) 如果控制从先前的顺序段隐式转到过程名 1 而导致排错节的执行，则下列条件成立：

- a. **DEBUG-LINE** 标识该先前语句。
- b. **DEBUG-NAME** 包含过程名 1。
- c. **DEBUG-CONTENTS** 包含 ‘**FALL THROUGH**’。

(23) 如果对文卷名 1 或 **cd** 名 1 的引用导致执行排错节，则：

- a. **DEBUG-LINE** 标识引用文卷名 1 或 **cd** 名 1 的源语句。
- b. **DEBUG-NAME** 包含文卷名 1 或 **cd** 名 1。
- c. 对 **READ**、**DEBUG-CONTENTS** 包含整个记录读入。
- d. 对文卷名 1 的所有其余引用，**DEBUG-CONTENTS** 包含空格。
- e. 对 **cd** 名 1 的任何引用，**DEBUG-CONTENTS** 包含与 **cd** 名 1 相关的区域的内容。

(24) 如果对标识符 1 的引用导致排错节执行，则：

- a. **DEBUG-LINE** 标识引用标识符 1 的源语句。
- b. **DEBUG-NAME** 包含标识符 1 的名字。

c. DEBUG-CONTENTS 包含标识符 1 引用的数据项在控制转到排错节时的内容 (见上述一般规则 5 和 6)

附加说明：

本标准由中华人民共和国机械电子工业部提出。

本标准由南京大学负责起草。

本标准主要起草人钱树人、王静英、冯惠、段祥。

本标准由 1983 年 12 月首次发布，1992 年 8 月第一次修订。