



中华人民共和国国家标准

GB/T 4092.13—1992

程序设计语言 COBOL 通信模块

Programming language COBOL
Communication 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 通信模块

代替 GB 4092.13—1983

Programming language COBOL Communication module

1 引言

1.1 功能

通信模块提供存取、处理和建立消息或部分消息的功能。它提供用通信设备的消息控制系统(MCS)进行通信的能力。

1.2 级别特征

1级通信对通信描述款提供局部功能。在过程部中,1级通信对 **RECEIVE** 和 **SEND** 语句提供局部功能而对 **ACCEPT MESSAGE COUNT** 语句提供完整功能。

2级通信对通信描述款提供完整功能。在过程部中,2级通信为 **ACCEPT MESSAGE COUNT**、**DISABLE**、**ENABLE**、**PURGE**、**RECEIVE** 和 **SEND** 语句提供完整功能。

2 通信模块的数据部

2.1 通信节

通信节位于源程序的数据部。通信节描述源程序中作为消息控制系统(MCS)和程序之间接口的数据项。这个 MCS 接口区由通信描述款定义。通信描述款后跟以零个、一个或多个记录描述款。

下面给出通信节的一般格式:

COMMUNICATION SECTION.

[通信描述款

[记录描述款]...]

2.1.1 通信描述款

在 COBOL 程序中,通信描述款(CD 款)代表通信节中最高一级的组织。通信节首后跟以通信描述款,该描述款是由层指示符(CD)、cd 名以及一串独立子句组成。描述款本身是由句号终止的。

对输入通信描述款,这些子句指明输入的队列、消息日期和时间、符号源、正文长度、状态和末端键以及消息计数。对输出通信描述款,这些子句指明目的地计数、正文长度、状态键、错误键和符号目的地。对输入输出通信描述款,这些子句指明消息日期和时间、符号终端、正文长度、状态和末端键。

2.1.2 记录描述结构

与通信描述款相关联的记录域可以隐含地通过用户指定紧跟在通信描述款之后书写的各种记录描述款来重定义。

记录描述款由一组描述特定记录之特性的数据描述款组成。每个数据描述款由层号后随数据名或 **FILLER** 子句(若指明),当需要时跟以一串独立子句组成。记录描述可以有层次结构,因此使用某一款的子句可以颇为不同,这依赖于其后是否跟以从属款。记录描述结构和记录描述款中允许的元素的结构

在层的概念和数据描述款中解释。数据描述款中特定子句的可用性依赖于实现支撑的核心模块的级。

2.2 通信描述款

2.2.1 功能

通信描述款指明消息控制系统(MCS)和 COBOL 程序间的接口区。

2.2.2 一般格式

格式 1:

CD cd 名 1

```
FOR INPUT
[[SYMBOLIC QUEUE IS 数据名 1]
[SYMBOLIC SUB-QUEUE-1 IS 数据名 2]
[SYMBOLIC SUB-QUEUE-2 IS 数据名 3]
[SYMBOLIC SUB-QUEUE-3 IS 数据名 4]
[MESSAGE DATE IS 数据名 5]
[MESSAGE TIME IS 数据名 6]
[SYMBOLIC SOURCE IS 数据名 7]
[TEXT LENGTH IS 数据名 8]
[END KEY IS 数据名 9]
[STATUS KEY IS 数据名 10]
[MESSAGE COUNT IS 数据名 11]]
[数据名 1,数据名 2,数据名 3,数据名 4,
数据名 5,数据名 6,数据名 7,数据名 8,
数据名 9,数据名 10,数据名 11]
```

格式 2:

CD cd 名 1FOR OUTPUT

```
[DESTINATION COUNT IS 数据名 1]
[TEXT LENGTH IS 数据名 2]
[STATUS KEY IS 数据名 3]
DESTINATION TABLE OCCURS 整数 1 TIMES
INDEXED BY {位标名 1}...
[ERROR KEY IS 数据名 4]
[SYMBOLIC DESTINATION IS 数据名 5]
```

格式 3:

CD cd 名 1

```
FOR OUTPUT
[[ MESSAGE DATE IS 数据名 1]
[MESSAGE TIME IS 数据名 2]
[SYMBOLIC TERMINAL IS 数据名 3]
[TEXT LENGTH IS 数据名 4]
[END KEY IS 数据名 5]
[STATUS KEY IS 数据名 6]]
[数据名 1,数据名 2,数据名 3,
数据名 4,数据名 5,数据名 6]
```

2.2.3 语法规则

所有格式:

(1) **CD** 款必须而且仅能在通信节中出现。

格式 1 和 3:

(2) 在单个程序中, **INITIAL** 子句仅能在一个 **CD** 描述款中指明。 **INITIAL** 子句不能用于过程前或 **USING** 短语的程序中。

(3) 这些任选子句可以按任意次序书写。

(4) 若在指定的接口区中无任选子句, 那么 01 层数据描述款必须跟着该 **CD** 描述款。任何一个任选可以跟着 01 层数据描述款。

格式 1:

(5) 跟着输入 **CD** 描述款的记录描述款隐含地重定义了由输入 **CD** 描述款建立的记录区, 而且必须描述为确是 87 个标准数据格式字符的记录。这个记录允许多次重定义; 但只有第一次重定义可以包含 **VALUE** 子句。然而, **MCS** 总是根据一般规则 2 中定义的数据描述来引用这个记录的 (见 GB/T 4092.2 中 5.15 **VALUE** 子句)。

(6) 数据名 1、数据名 2、...、数据名 11 在 **CD** 描述款中必须唯一。在这个序列中, 任一数据名可以用保留字 **FILLER** 来替换。

格式 2:

(7) 任选子句的书写顺序是无关紧要的。

(8) 若 **CD** 描述款中没有指明任选子句, 层号为 01 的数据描述款必须跟着 **CD** 描述款。

(9) 从属于输出 **CD** 描述款的记录描述款, 隐含地重新定义了由输出 **CD** 描述款建立的记录区。这个记录的多次重定义是允许的, 但仅第一次重定义可以包含 **VALUE** 子句。然而 **MCS** 总是按在一般规则 16 中定义的数据描述来引用这个记录的 (见核心模块 5.15 **VALUE** 子句)。

(10) 数据名 1、数据名 2、...、数据名 5 在 **CD** 描述款中必须是唯一的。

(11) 将假定有一个错误键和符号目的地域。此种情况, 当引用这些数据项时, 带下标是不允许的。

(12) 若指明有 **DESTINATION TABLE OCCURS** 子句, 数据名 4 和数据名 5 仅可以带下标来引用。

(13) 在 1 级中, 由数据名 1 引用的数据项的值必须是 1。

格式 3:

(14) 跟着输入输出 **CD** 描述款的记录描述款隐含地重定义了由输入输出 **CD** 款建立的记录区, 而且必须描述为确是 33 个标准数据格式字符的记录。这个记录允许多次重定义; 但只有第一次重定义可以包含 **VALUE** 子句。 **MCS** 总是按一般规则 24 中定义的数据描述来引用这个记录 (见 GB/T 4092.2 中 5.15 **VALUE** 子句)。

(15) 数据名 1、数据名 2、...、数据名 6 在 **CD** 款中必须是唯一的。这个序列中, 任一数据名可以用保留字 **FILLER** 来替换。

2.2.4 一般规则

格式 1:

(1) 输入 **CD** 描述款的信息构成消息控制系统 (**MCS**) 和消息处理程序之间的通信。这个信息不可以从终端作为消息的一部分来到的。

(2) 对于每个输入 **CD** 描述款, 分配一个有 87 个相邻的字符位置的记录域。这个记录域对 **MCS** 定义如下:

a. **SYMBOLIC QUEUE** 子句定义数据名 1 为占有该记录中的第 1-12 位的 12 个字符的字符型初等数据项。

b. **SYMBOLIC SUB-QUEUE-1** 子句定义数据名 2 为占有该记录中的第 13-24 位的 12 个字符的字符型初等数据项。

c. **SYMBOLIC SUB-QUEUE-2** 子句定义数据名 3 为占有该记录中的第 25-36 位的 12 个字符的字符型初等数据项。

d. **SYMBOLIC SUB-QUEUE-3** 子句定义数据名 4 为占有该记录中的第 37-48 位的 12 个字符的字符型初等数据项。

e. **MESSAGE DATE** 子句定义数据名 5 为一个数据项的名,它隐含的描述为占据该记录的第 49-54 字符位的无正负号的 6 位数字的整数。

f. **MESSAGE TIME** 子句定义数据名 6 为一个数据项的名,它隐含的描述为占据该记录的第 55-62 字符位的无正负号的 8 位数字的整数。

g. **SYMBOLIC SOURCE** 子句定义数据名 7 为占据该记录的第 63-74 位的 12 个字符的字符型初等数据项的名。

h. **TEXT LENGTH** 子句定义数据名 8 为一个初等数据项的名,隐含的描述为包括该记录中第 75-78 位的无正负号的 4 位数字的整数。

i. **END KEY** 子句定义数据名 9 为占据该记录中的第 79 位的一个字符的字符型初等数据项的名。

j. **STATUS KEY** 子句定义数据名 10 为占据该记录中的第 80-81 位的两个字符的字符型初等数据项的名。

k. **MESSAGE COUNT** 子句定义数据名 11 为初等数据项的名,它隐含地描述为占据该记录的第 82-87 字符位的无正负号的 6 位数字的整数。

任何任选产生一个记录,它的隐含描述等价于:

隐含描述		注 解	
01	数据名 0.		
02	数据名 1	PICTURE X(12).	符号队列
02	数据名 2	PICTURE X(12).	符号子队列
02	数据名 3	PICTURE X(12).	
02	数据名 4	PICTURE X(12).	
02	数据名 5	PICTURE 9(6).	消息日期
02	数据名 6	PICTURE 9(8).	消息日期
02	数据名 7	PICTURE X(12).	符号源
02	数据名 8	PICTURE 9(4).	正文长度
02	数据名 9	PICTURE X.	末端键
02	数据名 10	PICUTRE XX.	状态键

02 数据名 11 PICTURE 9(6). 消息计数

注意在“注解”下的信息只是为了阐明而不是数据描述的部分。

(3) 由数据名 2、数据名 3 和数据名 4 引用的数据项的内容必须包含空格。

(4) 由数据名 1 引用的数据项分别包含队列符号名。所有符号名必须遵守系统名的形成规则,并且必须预先对 MCS 定义。

(5) RECEIVE 语句导致从 CD 中的描述款所指定的队列中串行回送“下一个”消息。

在 RECEIVE 语句的执行期间,输入 CD 域必须包含,在数据名 1 的内容中,符号队列的名。数据名 2、数据名 3 和数据名 4 规定的数字项的内容包含空格。

当指明队列结构的给定层时,那么也必须指明所有队列结构的较高层。若有小于所有队列层次的层,那么由 MCS 在输入 CD 指定的队列和/或子队列中决定存取下一个消息或者部分消息。

RECEIVE 语句执行以后,由数据名 1 到数据名 4 引用的数据项的内容将包含队列结构的所有层的符号名。

(6) 每当 MCS 调度一个程序处理消息时,那个程序建立一个运行单位并且具有这种动作性的队列结构的符号名作为可应用的放进与 INITIAL 子句相关的 CD 描述款的数据名 1 到数据名 4 所引用的数据项中。在所有其它情况中,与 INITIAL 子句相关联的 CD 描述款的数据名 1 到数据名 4 引用的数据项的内容,初始化为空格。

符号名插入或初始化成空格是在过程部第一个语句执行之前完成的。

执行指明的数据名 1 到数据名 4 所引用的数据项的同一内容的下一个 RECEIVE 语句回送使程序进行调度的实在消息。仅在此时修改 CD 描述款的其余部分。

(7) 若 MCS 提示调度一个缺少 INITIAL 子句的程序,其结果将是无定义的。

(8) RECEIVE 语句执行期间,在数据名 5 引用的数据项中,MCS 提供它认为该消息完成时的日期,格式为‘YYMMDD’(年、月、日)。由数据名 5 引用的数据项内容仅能由 MCS 作为 RECEIVE 语句的执行部分来修改。

(9) RECEIVE 语句执行期间。在数据名 6 引用的数据项中,MCS 提供它认为消息完成时的时间,格式为‘HHMMSSSTT’(小时、分、秒、百分之一秒)。由数据名 6 引用的数据项内容仅能由 MCS 作为 RECEIVE 语句的执行部分来修改。

(10) 在 RECETRVE 语句的执行期间。由数据名 7 引用的数据项内,MCS 提供传输消息源通信终端的符号名。这个符号名必须遵循系统名的形成规则。然而,若 MCS 不知道通信终端的符号名,那么由数据名 7 引用的数据项内容将包含空格。

(11) MSC 借助于数据名 8 引用的数据项内容来指出执行 RECEIVE 语句得到的填满的字符位置数。

(12) 由数据名 9 引用的数据项内容仅能由 MCS 作为执行 RECEIVE 语句的部分结果来置值,置值根据下列规则进行:

a. 当指明 RECEIVE MESSAGE 短语时,那么:

1) 若检测到消息组末端,那么由数据名 9 引用的数据项的内容置为 3。

2) 若检测到消息末端,那么由数据名 9 引用的数据项的内容置为 2。

3) 若不足一个消息被传输,那么由数据名 9 引用的数据项的内容置为 0。

b. 当指明 RECEIVE SEGMENT 短语时,那么:

1) 若检测到消息组末端,那么由数据名 9 引用的数据项的内容置为 3。

2) 若检测到消息末端,那么由数据名 9 引用的数据项的内容置为 2。

3) 若检测到消息段末端,那么由数据名 9 引用的数据项的内容置为 1。

4) 若不足一个消息段被传输,那么由数据名 9 引用的数据项的内容置为 0。

c. 当上述条件同时满足一个以上,则以列表次序中首先满足的规则来决定数据名 9 引用的数据项的内容。

(13) 由数据名 10 引用的数据项内容指出先前执行过的 **RECEIVE**、**ACCEPT**、**MESSAGE**、**COUNT** 语句的状态条件。

由数据名 10 引用的数据项内容和状态条件本身的实际联系在下面的图 1(通信状态键条件)中定义。

(14) 数据名 11 引用的数据项的内容指出消息的数目,这些消息存在于队列中。**MCS** 把数据名 11 引用的数据项的内容只能作为 **ACCEPT MESSAGE COUNT** 语句执行的部分进行修改。

格式 2:

(15) 输出 **CD** 描述款信息的实质是不发送到终端,但它构成有关消息处理的程序和 **MCS** 之间的通讯。

(16) 在 1 级中,对每个输出 **CD** 分配 23 个相邻字符位置的记录域。

在 2 级中,对于每个输出 **CD** 描述款,根据下列公式: $(10+15 \times \text{整数 } 1)$ 分配具有相邻字符位置的记录域。

这个记录域的隐含描述是:

a. **DESTINATION COUNT** 子句定义数据名 1 为一个数据项的名,它隐含描述为占据记录中第 1-4 字符位的无正负号的整数。

b. **TEXT LENGTH** 子句定义数据名 2 为初等数据项的名,它隐含描述为记录中占据第 5-8 字符位的无正负号的 4 位数字的整数。

c. **STATUS KEY** 子句定义数据名 3 是一个占据记录的第 9-10 位的 2 个字符的字符型初等数据项。

d. 第 11 到 23 位的字符位置形成如下描述的表项:

1) **ERROR KEY** 子句定义数据名 4 为 1 个字符的字符型初等数据项名。

2) **SYMBOLIC DESTINATION** 子句定义数据名 5 为 12 个字符的字符型初等数据项名。

使用上面子句产生一个记录,它的隐含描述等价于:

隐含描述				注解
01	数据名 0.			
02	数据名 1	PICTURE	9(4).	目的地计数
02	数据名 2	PICTURE	9(4).	正文长度
02	数据名 3	PICTURE	XX.	状态键
02	数据名			目的地表
03	数据名 4	PICTURE	X.	错误键
03	数据名 5	PICTURE	X(12).	符号目的地

注意:在“注解”下的信息是为了阐明而不是数据描述的部分。

(17) **SEND** 语句的执行期间,由数据名 1 引用的数据项的内容对 **MCS** 指出符号目的地的数,这些符号目的地使用数据名 5 引用的区域。

MCS 找到由数据名 5 引用的区域的第一次出现的第一个符号目的地名,由数据名 5 引用的区域的第二次出现中的第二个符号目的地名,...,直到包含出现由数据名 1 的内容中的由数据名 5 引用的区域为止。

在执行 **SEND** 语句期间,若由数据名 1 引用的数据项的值在 1 之外,那么产生一个错误条件,对任何目的地不采取动作,且终止执行

(19) 作为 **SEND** 语句执行的一部分, **MCS** 把数据名 2 引用的数据项内容解释为用户对数据项的最左字符位置个数的指示, 该数据项是由传送该数据的相应的 **SEND** 语句中的标识符所引用的(见 3. 6**SEND** 语句)。

(21) 由数据名 3 引用的数据项的内容指出先前执行的 **SEND**、**PURGE**、**ENABLE OUTPUT** 或 **DISABLE OUTPUT** 语句的状态条件。

(22) 在 `SEND [REDACTED]` 语句的执行期间,若 MCS 判定有一错误,则数据名 3 引用的数据项的内容及数据名 4 的 `NEED_REFRESH` 内容,直到包括数据名 1 内容指明的出现均被修改。

格式 3:

(24) 对于每个输入输出 CD 描述款,分配一个有 33 个相邻字符位置的记录域。这个记录域对 MCS 定义如下:

b. MESSAGE TIME 子句定义数据名 2 为一个数据项的名。它隐含的描述为占据该记录的第 7~14 字符位置的无正负号且为 8 位数字的整数。

d. TEXT LENGTH 子句定义数据名 4 为一个初等数据项的名。它隐含的描述为占据该记录的第 27-30 字符位置的无正负号的 4 位数字的整数。

e. **END KEY** 子句定义数据名 **5** 为占据该记录的第 **31** 位的一个字符的字符型初等数据项的名。

[illegible]

任何任选产生一个记录,它的隐含描述等价于:

注解

02 数据名 1 PICTURE 9(6).

消息日期

02	数据名 2	PICTURE 9(8).	消息时间
02	数据名 3	PICTURE X(12).	符号终端
02	数据名 4	PICTURE 9(4).	正文长度
02	数据名 5	PICTURE X.	末端键
02	数据名 6	PICTURE XX.	状态键

注意：在“注解”下的信息是为了阐明而不是数据描述的部分。

(25) 当 MCS 调度一个程序处理消息时，引用具有 INITIAL 子句的输入输出 CD 的第一个 RECEIVE 语句回送使程序进行调度的实际消息。

(26) 数据名 1 具有格式 ‘YYMMDD’ (年、月、日)。它的内容表示 MCS 认为该消息完成时的日期。数据名 1 引用的数据项的内容仅由 MCS 作为 RECEIVE 语句的执行部分来修改。

(27) 数据名 2 有格式 ‘HHMMSSTT’ (时、分、秒、百分之一秒)。它的内容表示 MCS 认为该消息完成时的时间。

数据名 2 引用的数据项的内容仅由 MCS 作为 RECEIVE 语句的执行部分来修改。

(28) 每当 MCS 调度一个程序处理消息时，那个程序建立一个运行单位并且这程序调用的消息源的通信终端的符号名作为可应用的放进与 INITIAL 子句相关的输入输出 CD 描述款的数据名 3 引用的数据项中。这个符号名必须遵循系统名的形成规则。

在所有其它情况，与 INITIAL 子句相关的输入输出 CD 描述款的数据名 3 引用的数据项的内容初始化为空格。

符号名插入或初始化为空格是在过程部第一个语句执行之前完成的。

(29) 若 MCS 提示调度一个缺少 INITIAL 子句的程序，其结果将是无定义的。

(30) 当对输入输出 CD 指定了 INITIAL 子句而且 MCS 调度该程序时，数据名 3 引用的数据项的内容不能被程序改变。若内容被改变了，则执行任何引用 cd 名 1 的语句是不成功的，而且当应用时数据名 6 引用的数据项被指示成不认识的源或目的地 (见表 1 通信状态键条件)。

(31) 对输入-输出 CD 当程序未被 MCS 调度到时，程序必须在第一个引用 cd 名 1 的语句执行之前在数据名 3 里规定源或目的地的符号名。

在第一个引用 cd 名 1 的语句执行以后，数据名 3 引用的数据项的内容不可被程序改变。若该内容被改变了，则执行任何引用 cd 名 1 的语句是不成功的，而且当应用时，数据名 6 引用的数据项被标记为不认识的源或目的地 (见表 1 通信状态键条件)。

(32) MCS 借助于数据名 4 引用的数据项内容来指出执行 RECEIVE 语句得到的填满的字符位置数 (见 3.5 RECEIVE 语句)。

作为 SEND 语句执行的一部分，MCS 把数据名 4 引用的数据项的内容解释为用户对数据项的最左字符位置数的指示，该数据项是由传送该数据的相应的 SEND 标识符所引用的 (见 3.6 SEND 语句)。

(33) 数据名 5 引用的数据项内容仅能由 MCS 作为执行 RECEIVE 语句的部分结果来置值，按下列规则置值：

a. 当有 RECEIVE MESSAGE 短语时：

- 1) 若检测到消息组末端，那么由数据名 5 引用的数据项的内容置为 3。
- 2) 若检测到消息末端，那么由数据名 5 引用的数据项的内容置为 2。
- 3) 若不是一个消息被传送时，那么由数据名 5 引用的数据项的内容置为 0。

b. 当指明 **RECEIVE SEGMENT** 短语时：

- 1) 若检测到消息组末端,数据名 5 引用的数据项的内容置为 3。
- 2) 若检测到消息末端,那么由数据名 5 引用的数据项的内容置为 2。
- 3) 若检测到消息段末端,那么由数据名 5 引用的数据项的内容置为 1。
- 4) 若不足一个消息段被传输,那么由数据名 5 引用的数据项内容置为 0。

c. 当几个条件同时满足时,则以列表次序中首先满足的规则来决定数据名 5 引用的数据项的内容。

(34) 由数据名 6 引用的数据项内容指出先前执行过的 **DISABLE INPUT/OUTPUT** **RECEIVE** 或 **SEND** 语句的状态条件。

由数据名 6 引用的数据项内容和状态条件本身之间的实际联系在下面的表 1(通信状态键条件)中定义。

2.2.5 通信状态键条件

在所示的各语句完成时,下面的图指出格式 1 中数据名 10、格式 2 中数据名 3 和格式 3 中数据名 6 引用的数据项的可能的内容。在语句行的行上的“X”指明该行所表明的状态键码对该语句是可能的

表 1 通信状态键条件

注 ①	注 ②	注 ③	注 ④	注 ⑤	注 ⑥	注 ⑦	注 ⑧	注 ⑨	注 ⑩	注 ⑪	状态 键值	
X	X	X	X	X	X	X	X	X	X	X	00	未发现错误,动作完成
		X	X								10	断开一个或多个目的地,动作完成(见错误键码)
	X										10	断开目的地,不做动作
					X	X	X	X	X	X	15	符号源,或一个或多个队列或目的地已经断开/接通(见错误键码)
	X	X	X				X			X	20	一个或多个目的地未知,对已知的目的地动作完成(见错误键码)
X				X	X			X			20	一个或多个队列或子队列未知,不做动作
X						X			X		21	符号源未知,不做动作
		X	X				X			X	30	目的地计数非法,不做动作
					X	X	X	X	X	X	40	口令非法,不发生接通/断开动作
	X	X									50	正文长度超过标识符 1 的长度
	X	X									60	请求发送的部分消息其正文长度为零或缺了标识符 1,不做动作
		X									65	超过输出队列的容量(见错误键码)

注：①RECEIVE ②SEND input-output-ud ③SEND output-cd ④PURGE ⑤ACCEPT MESSAGE COUNT
⑥ENABLE INPUT ⑦ENABLE INPUT/I-O TERMINAL ⑧ENABLE OUTPUT
⑨DISABLE INPUT ⑩DISABLE INPUT/I-O TERMINAL ⑪DISABLE OUTPUT

续表 1

注 ①	注 ②	注 ③	注 ④	注 ⑤	注 ⑥	注 ⑦	注 ⑧	注 ⑨	注 ⑩	注 ⑪	状态 键值	
			×								70	一个或多个目的地没有与其相关联的部分消息,对其它目的地动作完成
		×	×		×		×	×		×	80	出现了至少两个状态键条件 10,15 和 20 的联合,(见错误键码)
											9X	实现者定义的状态

注: ①RECEIVE ②SEND input-output-ud ③SEND output-cd ④PURGE ⑤ACCEPT MESSAGE COUNT
 ⑥ENABLE INPUT ⑦ENABLE INPUT/I-O TERMINAL ⑧ENABLE OUTPUT
 ⑨DISABLE INPUT ⑩DISABLE INPUT/I-O TERMINAL ⑪DISABLE OUTPUT

2.2.6 错误键值

在所示的各语句完成时,下面的表 2 指出格式 2 中数据名 4 引用的数据项的可能的内容。在语句列的行上的‘×’指明该行所表明的有关错误键值对该语句是可能的。符号 2 指示在 1 级中不可用的 2 级元素。

表 2 错误键值

注 ①	注 ②	注 ③	注 ④	错误 键值	
×	×	×	×	0	无错误
×	×	×	×	1	符号目的地未知
×	×			2	符号目的地断开
	×			4	无与引用的符号目的地有关的部分消息 ²
		×	×	5	符号目的地已经接通/断开 ²
×				6	超出输出队列容量
				7-9	为进一步使用而保留
				A-Z	实现者定义的条件

注: ①SEND ②PUREC² ③ENABLE OUTPUT² ④DISABLE OUTPUT²

3 通信模块的过程部

3.1 ACCEPT MESSAGE COUNT 语句

3.1.1 功能

ACCEPT MESSAGE COUNT 语句使得队列中的完整消息的个数是可用的。

3.1.2 一般格式

ACCEPT cd 名 1 MESSAGE COUNT**3.1.3 语法规则**

(1) **cd 名 1** 必须引用一个输入 **CD** 描述款。

3.1.4 一般规则

(1) **ACCEPT MESSAGE COUNT** 语句使 **cd 名 1** 指明的消息计数数据项被修改,以指出在队列结构中存在的完整消息的个数,该队列结构由 **cd 名 1** 引用域的数据名 1(**SYMBOLIC QUEUE**)

和由 **SYMBOLIC SUB-QUEUE-3** 指定的数据项的内容来指定。

(2) 当 **ACCEPT MESSAGE COUNT** 语句执行时,由通信描述款指明的域的内容必须至少包含要测试的符号队列名。测试该条件导致数据项名 10(**STATUS KEY**)引用的数据项的内容和与通信描述款有关的域中的数据名 11(**MESSAGE COUNT**)内容作相应地修改(见 2.2 通信描述款)。

3.2 DISABLE 语句**3.2.1 功能**

DISABLE 语句通知 **MCS** 对于输出禁止指定的输出队列和目的地之间进行数据传输,或对于输入禁止指定的源和输入队列之间进行数据传送,或对于输入输出禁止程序和指定的一个源或目的地之间进行数据传输。在标准 **COBOL** 的这一版本中视 **WITH KEY** 短语是一个过时成分,因为在标准 **COBOL** 的以后的修改版中要把它删掉。

3.2.2 一般格式

$$\text{DISABLE} \left\{ \begin{array}{l} \text{INPUT} \quad [\text{TERMINAL}] \\ \text{I-O} \quad \text{TERMINAL} \\ \text{OUTPUT} \end{array} \right\} \text{cd 名 1} \left[\text{WITH KEY} \left\{ \begin{array}{l} \text{标识符 1} \\ \text{字值 1} \end{array} \right\} \right]$$
3.2.3 语法规则

(1) 当有 **INPUT** 短语时,**cd 名 1** 必须引用一个输入 **CD** 描述款。

(2) 当有 **I-O TERMINAL** 短语时,**cd 名 1** 必须引用一个输入输出 **CD** 描述款。

(3) 当有 **OUTPUT** 短语时,**cd 名 1** 必须引用一个输出 **CD** 描述款。

(4) 由标识符 1 引用的数据项的内容或字值 1 必须定义为字符型。

3.2.4 一般规则

(1) **DISABLE** 语句使得 **MCS** 和指定的源或目的地之间实现逻辑断开。当这种逻辑断开已经存在,或者已由这个程序之外的某些其他手段处理,那么在这个程序中 **DISABLE** 语句是不需要。当指定已经断开的源或目的地的 **DISABLE** 语句被执行时,除了状态键数据项中的值指示这个条件外,不做其它工作。**DISABLE** 语句不影响 **COBOL** 程序和 **MCS** 之间的数据传输的逻辑路径。

(2) **MCS** 保证执行 **DISABLE** 语句导致在源或目的地在刚开始不活动时的逻辑断开。执行 **DISABLE** 语句使得在传进或传出终端期间永不会停止传输该消息的余下部分。

(3) 当指明不带任选字 **TERMINAL** 的 **INPUT** 短语,那么 **cd 名 1** 引用域的数据名 1(**SYMBOLIC QUEUE**)到数据名 4(**SYMBOLIC SUB-QUEUE-3**)的内容所指明的有关队列和子队列和所有可能相联的源之间的逻辑路径为不活动的。

(4) 当指明带有任选字 **TERMINAL** 的 **INPUT** 短语时,源和所有有关的队列和子队列之间的逻辑路径为不活动的,这个源由数据名 7(**SYMBOLIC SOURCE**)所引用的数据项的内容定义。

(5) 当指明 **I-O TERMINAL** 短语时,源和程序间的逻辑路径为不活动的。这个源由数据名 1(**SYMBOLIC TERMINAL**)所引用的数据项的内容定义。

(6) 当指明 **OUTPUT** 短语时,由数据名 5 每次出现的内容规定的全部目的地和直到包含由 **cd 名 1** 引用域的数据名 1 的内容所规定的出现之间的逻辑路径都是不活动的。

(7) 字值 1 或由标识符 1 引用的数据项内容需和建进系统的口令相匹配。仅当字值 1 或标识符

引用的数据项内容和系统口令相匹配时,DISABLE 语句才有效。若字值 1 或标识符 1 引用的数据项内容和系统口令不匹配,那么,cd 名 1 引用域内的 STATUS KEY 项的值要修改。

MCS 必须处理包含从 1 个字符到 10 字符的口令。

3.3 ENABLE 语句

3.3.1 功能

ENABLE 语句通知 MCS 允许输出时在指定的输出队列和目的地之间或者输入时在指定的源和输入队列之间进行数据传输,或者在输入输出时在程序和指定的一个源或目的地之间进行数据传输。在标准 COBOL 的这一版本中视 WITH KEY 短语是一个过时成分,因为在标准 COBOL 的以后的修改版中要把它删掉。

3.3.2 一般格式

$$\text{ENABLE} \left\{ \begin{array}{l} \text{INPUT} \\ \text{I-O} \\ \text{OUTPUT} \end{array} \right\} \left\{ \begin{array}{l} [\text{TERMINAL}] \\ \text{TERMINAL} \end{array} \right\} \text{cd 名 1} \left[\text{WITH KEY} \left\{ \begin{array}{l} \text{标识符 1} \\ \text{字值 1} \end{array} \right\} \right]$$

3.3.3 语法规则

- (1) 当指明 INPUT 短语时,cd 名 1 必须引用一个输入 CD 描述款。
- (2) 当指明 I-O TERMINAL 短语时,cd 名 1 必须引用一个输入-输出 CD 描述款。
- (3) 当指明 OUTPUT 短语时,cd 名 1 必须引用一个输出 CD 描述款。
- (4) 字值 1 或者由标识符 1 引用的数据项内容必须定义成字符型的。

3.3.4 一般规则

(1) ENABLE 语句在所指定的源或目的地和 MCS 之间建立逻辑联系。当这种逻辑联系已经存在,或者已由该程序以外的某些其他手段处理,那么在该程序中的 ENABLE 语句是不需要的。当指定已经联系的源或目的地的 ENABLE 语句被执行时,除了状态键数据项中的值指示这个条件外,不做其它工作。COBOL 程序和 MCS 之间数据传输的逻辑路径不受 ENABLE 语句影响。

(2) 当指明不带任选字 TERMINAL 的 INPUT 短语时,那么 cd 名 1 引用域内的数据名 3(SYMBOLIC QUEUE)到数据名 4(SYMBOLIC SUB-QUEUE-3)的内容指明有关队列和子队列和所有有关的源之间的逻辑路径是活动的。

(3) 当指明带有任选字 TERMINAL 的 INPUT 短语时,源和所有有关的队列和子队列之间的逻辑路径是活动的。这个源由数据名 7(SYMBOLIC SOURCE)所引用的数据项的内容定义。

(4) 当指明 I-O TERMINAL 短语时,源和程序间的逻辑路径是活动的。这个源由数据名 8(SYMBOLIC TERMINAL)所引用的数据项的内容定义。

(5) 当指明 OUTPUT 短语时,对于由数据名 5 每次出现内容所规定的全部目的地直至并包括由 cd 名 1 引用域的数据名 1 所规定的出现之间的逻辑路径都是活动的。

(6) 字值 1 或标识符 1 引用的数据项的内容需和建进系统的口令相匹配。仅当字值 1 或标识符 1 引用的数据项的内容和系统口令相匹配时,ENABLE 语句才有效。若字值 1 或标识符 1 引用的数据项的内容和系统口令不匹配,那么修改由 cd 名 1 引用的域内的 STATUS KEY 项的值。

MCS 必须能处理从 1 个字符到 10 个字符的口令。

3.4 PURGE 语句

3.4.1 功能

PURGE 语句从消息控制系统(MCS)清除由一个或多个 **SEND** 语句释放了的部分消息。

3.4.2 一般格式

PURGE cd 名 1

3.4.3 语法规则

(1) cd 名 1 必须引用一个输出 **CD** 或输入输出 **CD** 描述款。

3.4.4 一般规则

(1) 执行 **PURGE** 语句使 **MCS** 清除由 cd 名 1 引用的 **CD** 中指出的待传送到目的地的部分消息。

(2) 任何已经与 **EGI** 或 **EMI** 关联的消息不受执行 **PURGE** 语句的影响。

(3) cd 名 1 引用域的状态键数据项的内容与错误键数据项的内容(若可用)由 **MCS** 修改(见 2.2 通信描述款)。

3.5 RECEIVE 语句

3.5.1 功能

RECEIVE 语句使得消息 [MESSAGE] 和有关那个数据的信息是可用的。

3.5.2 一般格式

RECEIVE cd 名 1 { **MESSAGE** } INTO 标识符 1
[**NO DATA** 命令语句 1]
[**WITH DATA** 命令语句 2]
[**END-RECEIVE**]

3.5.3 语法规则

(1) cd 名 1 必须引用一个输入 **CD** [SYMBOLIC QUEUE] 描述款。

3.5.4 一般规则

(1) 若 cd 名 1 引用一个输入 **CD**, 则 cd 名 1 引用域内的数据名 1 (**SYMBOLIC QUEUE**) [SYMBOLIC QUEUE] 指定的数据项的内容来表示包含这个消息的队列结构(见 2.2 通信描述款)。

(2) 若 cd 名 1 引用一个输入输出 **CD**, 则 cd 名 1 引用域的数据名 3 (**SYMBOLIC TERMINAL**) 指定的数据项的内容表示消息源(见 2.2 通信描述款)。

(3) 消息 [MESSAGE] 采用不用空格填补的左对齐方式传送到标识符 1 引用域内的接受字符的位置。

(4) 在 **RECEIVE** 语句的执行期间, 当 **MCS** 使标识符 1 引用的数据项中的数据成为可用时, 若指出 **NO DATA** 短语, 则忽略 **NO DATA** 短语并把控制转移到 **RECEIVE** 语句的结束处, 或若指出 **WITH DATA** 短语, 则控制转移到命令语句 2。当控制转移到命令语句 2 时, 就按照命令语句 2 中指出的各语句的规则继续执行。若过程转移或显式引起控制转移的条件语句被执行, 则控制就按照那个语句的规则进行转移; 否则, 在命令语句 2 的执行完成之后, 控制就转向 **RECEIVE** 语句的结束处。

(5) 在 **RECEIVE** 语句的执行期间, 若 **MCS** 不能使标识符 1 引用的数据项中的数据成为可用的, 则会出现下列三个动作之一。造成数据不可用的条件由实现者定义。

a. 若 **RECEIVE** 语句指明 **NO DATA** 短语, 则用动作完成指示来终止 **RECEIVE** 操作并把控制转向命令语句 1。随后按照命令语句 1 中指出的各语句的规则继续执行。若过程转移或显式引起控制转移的条件语句被执行, 则控制按照那个语句的规则进行转移; 否则, 在命令语句 1 执行完成以后, 控制就转向 **RECEIVE** 语句的结束处, 并且忽略 **WITH DATA** 短语(若已指明)。

b. 若 **RECEIVE** 语句没指明 **NO DATA** 短语,则目标程序的执行就挂起,直到标识符 1 引用的数据项中的数据成为可用时为止。

c. 若一个或多个队列 **MESSAGE** 对 **MCS** 是未知的,则存储适合的状态键码,然后就象数据已经可用时那样执行控制的转移(见 2.2.5 通信状态键条件)。

(6) **cd** 名 1 标识的数据项在 **RECEIVE** 语句的每次执行中由 **MCS** 进行相应地修改(见 2.2 通信描述款)。

(7) **RECEIVE** 语句的一次执行从不回送给标识符 1 引用的数据项多于一个消息(当用 **MESSAGE** 短语时)或者一个消息段(当用 **SEGMENT** 短语时)。然而,即使 **RECEIVE** 语句指明 **SEGMENT** 短语,整个消息在对 **MCS** 可用之前, **MCS** 不把任何部分消息送给目标程序。

(8) **MESSAGE** 消息段末端指示符是不考虑的,在数据传输中应用下面的规则:

a. 若消息和标识符 1 引用域有相同的长度,那么消息存入标识符 1 引用的域中。

b. 若消息的长度小于标识符 1 引用域,按照由标识符 1 引用域的最左字符对位而且未被消息占有的字符位置的内容保持不变。

c. 若消息的长度大于标识符 1 引用域,则消息用消息的最左字符开始从左到右填满标识符 1 引用域。在 1 级中,消息的余下部分是无定义的。

在 2 级中,消息的余下部分能传送到下一个 **RECEIVE** 语句中引用同一个队列,子队列... 的由标识符 1 引用的域中。应用规则 8a、8b 和 8c,消息的余下部分作为新的消息处理。

d. 若把组末端指示符与 **RECEIVE** 语句存取的正文相联系,那么消息末端指示符的存在是隐含的。

(9) 当使用 **SEGMENT** 短语时,应用下列各规则:

a. 若消息段和标识符 1 引用域有相同长度,那么该消息段存入由标识符 1 引用的域。

b. 若消息段的长度小于由标识符 1 引用域,按照由标识符 1 引用域的最左字符对位。而且未被消息段字符占有的内容保持不变。

c. 若消息段的长度大于标识符 1 引用域,该消息段用消息段的最左字符开始从左到右填满由标识符 1 引用域。消息段的余下部分能传送到下一个调用同一个队列,子队列... 的 **RECEIVE** 语句的标识符 1 引用域中。应用规则 9a、9b 和 9c,消息段的剩余部分作为新的消息段处理。

d. 若把消息末端指示符或组末端指示符与 **RECEIVE** 语句存取的正文相联系,则消息段末端指示符的存在是隐含的。

(10) 每当 **RECEIVE** 语句执行一次就回送部分消息一次,仅当那个运行单位中的 **RECEIVE** 语句的下一执行才回送消息段的余下部分。

(11) **END-RECEIVE** 短语限定 **RECEIVE** 语句的作用域(见 GB/T 4092.1 中 6.6.4.3 语句的作用域)。

3.6 SEND 语句

3.6.1 功能

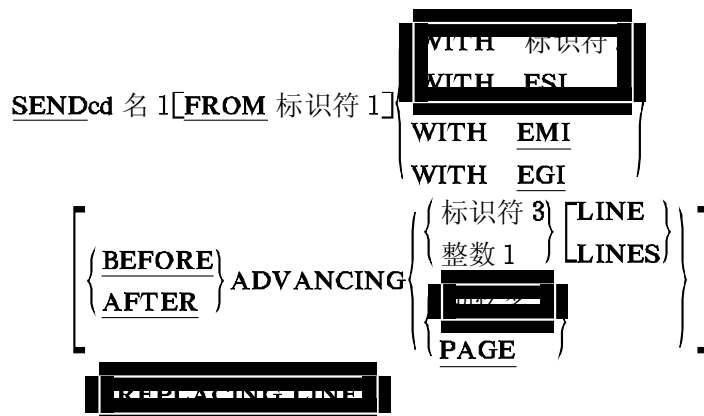
SEND 语句释放消息 **MESSAGE** 到一个 **QUEUE** 由 **MCS** 维护的输出队列中。

3.6.2 一般格式

格式 1:

SEND **cd** 名 1 **FROM** 标识符

格式 2:



3.6.3 语法规则

(1) **cd** 名 1 必须引用一个输出 **CD** 或输入输出 **CD** 描述款。

(3) 标识符 3 必须引用一个整数数据项。

(4) 当使用助忆名短语时,该名由实现者指定的特征来标识。助忆名定义在环境部的 **SPECIALNAMES** 段。

(5) 整数 1 或标识符 3 引用的数据项的值可以是 0。

3.6.4 一般规则

所有格式:

(1) 当接收用的通信设备(打印机、屏幕显示器、卡片穿孔机等)有固定的行长度:

a. 每一个消息从物理行的最左字符位置开始。

b. 小于物理行长度的消息释放时右边填满空格。

c. 消息的超出字符并不截断。字符将紧凑到等于该物理行的长度,然后输出到该输出设备。对超出字符将在下一行上继续处理。

(2) 当接收用的通信设备(纸带穿孔、另一台计算机等)能处理变长消息,那么每一个消息

始自该通信设备的下一个可用字符位置。

(3) 作为 **SEND** 语句执行的一部分,**MCS** 把 **cd** 名 1 引用的域中的正文长度数据项内容翻译成由标识符 1 引用的已传输的数据项的最左字符位置数的用户指示(见 2.2 通信描述款)。

若 **cd** 名 1 引用的域中的正文长度数据项内容是 0,那么由标识符 1 引用的数据项的字符不进行传输。

若 **cd** 名 1 引用的域中的正文长度数据项的内容超出 0 到标识符 1 引用的数据项的长度,由 **cd** 名 1 引用的域中的状态键数据项的值指出其错误,且数据不传输(见 2.2.5 通信状态键条件)。

(4) 作为 **SEND** 语句执行的一部分,**MCS** 修改 **cd** 名 1 引用域中的状态键数据项内容(见 2.2 通信描述款)。

(5) 在标识符 1 引用的数据项的内容里,特定控制字符的功能是未定义的。

(6) 格式 2 表示的 **SEND** 语句的一次执行从不释放给 **MCS** 多于一个消息或一个消息段,

或者由 **EMI** 或 **EGI** 指示符来指出。

然而,在整个消息释放给 MCS 之前,MCS 不传送任何部分消息段到通信设备去。

(7) 在该运行单位的执行期间,不用 EMI 或 EGI 来终止的或还未被 PURGE 语句清除的部分消息的安排是无定义的。然而,该消息对 MCS 来说逻辑上并不存在,因此不能发送到目的地,

(8) 每当 SEND 语句执行一次就把部分消息释放给 MCS 一次,仅在同一个运行单位中的 SEND 语句的下一次执行时才能导致释放该消息的余下部分。

格式 2:

(9) 由标识符 2 引用的数据项的内容指出标识符 1(若指明)引用的数据项的内容是与消息段末端指示符、消息末端指示符、消息组末端指示符或无指示符(隐含部分消息或部分消息段)相关联。若没有指明标识符 1,只有该指示符被送往 MCS。

若由标识符 2 引用的数据项的内容是	则由标识符 1 引用的数据项的内容有关于	含 义
0	无指示符	部分消息或部分消息段
1	消息段末端指示符(SEI)	当前消息段末端
2	消息末端指示符(EMI)	当前消息末端
3	消息组末端指示符(EGI)	当前消息组末端

与 1、2、3 不同的任何字符解释成 0。

若标识符 2 引用的数据项内容不是 1、2 或 3,以及标识符 1 未指明,那么由 cd 名 1 引用的域中的状态键数据项值指出错误,且没有数据被传送。

(10) WITH EGI 短语对 MCS 指出消息组是完整的。

WITH EMI 短语对 MCS 指出消息是完整的。

MCS 将识别这些指示并建立维护消息组、消息控制的必要部分。

(11) 末端指示符的层次是 EGI,EMI 和 EMI。EGI 不先于 EMI。

(12) ADVANCING 短语允许在应用纵向定位的通信设备上对每个消息进行纵向定位的控制。若纵向定位不能在该设备上应用的话,MCS 对指明的或隐含的纵向定位不加考虑。

(13) 若指明有标识符 2 且标识符 2 引用的数据项内容是 0,那么 MCS 对 ADVANCING 短语和 REPLACING 短语(若指明)不加考虑。

(14) 在应用纵向定位的设备上且 ADVANCING 短语未加指明的话,那么由实现者来提供自动推进,其动作如同用户指明 AFTER ADVANCING 1 LINE 一样。

(15) 若隐式或显式地指明有 ADVANCING 短语并且应用于纵向定位的话,那么采用下列规则:

a. 若指明有标识符 3 或整数 1 的话,传输到通信设备的字符将在推进等于标识符 3 引用的数据项的值或整数 1 的行数以后重新进行纵向定位。

b. 若标识符 3 引用的数据项的值是负的,那么结果是无定义的。

c. 若使用 BEFORE 短语,在纵向定位以前根据上面的一般规则 15a 在通信设备上表示消息。

d. 若使用 BEFORE 短语,在纵向定位以前根据上面的一般规则 15a 在通信设备上表示消息。

e. 若使用 AFTER 短语,在纵向定位以后根据上面的一般规则 15a 在通信设备上表示消息。

息

f. 若指明 **PAGE**, 在该设备定位于下页以前或以后(依赖于使用的短语)在该设备上表示传输到通信设备上的字符。若指明 **PAGE** 但对指定设备而言页并无意义, 那么推进由实现者来提供, 其作用如同用户指明 **BEFORE** 或 **AFTER**(依赖于使用的短语) **ADVANCING 1 LINE** 一样。

(16) 当接收的通信设备是可在其上的同一位置呈现两个或多个字符的字符图像设备时, 而且该设备对那种行允许两种选择: (1) 或者在已经显示了字符的那个位置上叠加第二或后继字符; (2) 或者每个字符出现在先前的字符位置上。则:

a. 若指明 **REPLACING** 短语, 那么由 **SEND** 语句传输的字符替换先前已被传输到同一行上的全部字符从该行的最左字符位置开始。

b. 那么由 **SEND** 语句传输的字符叠加在先前可能已被传输到同一行的字符上, 从该行的最左字符位置开始。

(17) 当接收的通信设备不支撑字符的替换时, 由 **SEND** 语句传输的字符叠加在先前可能已被传输到同一行的字符上, 从该行的最左字符位置开始。

(18) 当接收的通信设备不支撑在同一位置叠加两个或多个字符时, 不考虑是否指明 **REPLACING** 短语, 由 **SEND** 语句传输的字符替换先前可能已被传输到同一行上的全部字符, 从该行的最左字符位置开始。

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。