



IBM PC/AT

硬件和XENIX系统资料汇编之十

COBOL 语言

孙 逢 举 译
韩 光 宇

北京科海总公司培训中心
中国科学院软件研究所

一九八七年四月

IBM PC / AT硬件和XENIX系统资料汇编

(之十)

COBOL 语



孙 逢 举

韩 光 宇

译

北京科海总公司培训中心

中国科学院软件研究所

一九八七年四月

目 录

第一章 简介	(1)
1.1 语法注释	(1)
1.2 Microsoft COBOL 符合 ANSI 标准	(2)
1.3 Microsoft COBOL所作的扩展	(3)
1.4 更多地了解COBOL	(3)
第二章 语言元素	(4)
2.1 源代码编写规则	(4)
2.2 字符集	(5)
2.3 标点	(5)
2.4 保留字	(6)
2.5 名字	(6)
2.5.1 命名规范	(6)
2.5.2 名字的限定	(7)
2.6 直接量	(7)
2.6.1 数值直接量	(8)
2.6.2 非数值直接量	(8)
2.6.3 象征常量	(8)
2.7 数据类型	(9)
2.8 语句	(9)
2.8.1 强制语句	(9)
2.8.2 条件语句	(9)
2.8.3 编译指示语句	(10)
2.9 算术语句	(10)
2.9.1 合成操作数	(10)
2.9.2 任选短语	(11)
2.10 算术表达式	(11)
第三章 COBOL程序结构	(13)
3.1 术语	(13)
3.2 层号及数据项	(14)
3.3 编译指示语句	(15)
第四章 标识部	(16)
4.1 标识部部头和一般格式	(16)
4.2 作者段	(16)
4.3 编译日期段	(17)

4.4	编写日期段.....	(17)
4.5	场所段.....	(17)
4.6	程序标识段.....	(18)
4.7	保密段.....	(18)
第五章	设备部	(19)
5.1	设备部部头及一般格式.....	(19)
5.2	配置节节头.....	(20)
5.2.1	目标计算机段.....	(20)
5.2.2	源计算机段.....	(21)
5.2.3	专用名段.....	(21)
5.3	输入/ 输出节节头.....	(23)
5.3.1	文件控制段.....	(24)
5.3.2	输入/ 输出控制段	(28)
第六章	数据部	(31)
6.1	数据部部头及一般格式.....	(31)
6.2	记录描述款	(33)
6.2.1	数据描述款和数据项	(34)
6.2.2	组合项	(35)
6.2.3	基本项	(35)
6.2.4	字母数字项和字母数字编辑项	(35)
6.2.5	数值项	(36)
6.2.6	数值编辑项	(38)
6.2.7	66层 (重命名) 项	(38)
6.2.8	77层 (独立数据描述) 项	(39)
6.2.9	88层 (条件名) 项	(39)
6.3	数据部限制	(40)
6.4	节	(41)
6.4.1	文件节和文件描述款	(42)
6.4.2	工作存贮节	(43)
6.4.3	连接节	(44)
6.4.4	屏幕节	(45)
6.5	子句	(48)
6.5.1	AUTO子句.....	(48)
6.5.2	BACKGROUND-COLOR子句.....	(48)
6.5.3	BELL子句	(49)
6.5.4	BLANK LINE子句.....	(49)
6.5.5	BLANK SCREEN子句	(49)
6.5.6	BLANK WHEN ZERO子句.....	(49)
6.5.7	BLINK子句	(50)

6.5.8	BLOCK子句	(50)
6.5.9	CODE-SET子句	(50)
6.5.10	COLUMN子句	(51)
6.5.11	DATA RECORD (S) 子句	(51)
6.5.12	FOREGROUND-COLOR子句	(52)
6.5.13	FROM/TO/USING子句	(52)
6.5.14	FULL 子句	(53)
6.5.15	HIGHLIGHT子句	(53)
6.5.16	JUSTIFIED子句	(53)
6.5.17	LABEL RECORD (S) 子句	(54)
6.5.18	LINAGE子句	(54)
6.5.19	LINE子句	(55)
6.5.20	OCCURS子句	(56)
6.5.21	PICTURE子句	(57)
6.5.22	RECORD子句	(61)
6.5.23	REDEFINES子句	(62)
6.5.24	RENAMES子句	(62)
6.5.25	REQUIRED子句	(63)
6.5.26	SECURE子句	(63)
6.5.27	SIGN子句	(64)
6.5.28	SYNCHRONIZED子句	(64)
6.5.29	TO子句	(65)
6.5.30	USAGE子句	(65)
6.5.31	USING子句	(66)
6.5.32	VALUE IS 子句	(66)
6.5.33	VALUE OF FILE-ID子句	(67)
第七章	过程部	(68)
7.1	过程部——部头和一般格式	(68)
7.2	算术语句	(69)
7.2.1	CORRESPONDING任选项	(70)
7.2.2	GIVING任选项	(71)
7.2.3	REMAINDER任选项	(71)
7.2.4	ROUNDED任选项	(71)
7.2.5	SIZE ERROR任选项	(71)
7.3	输入/ 输出错误处理	(72)
7.4	动态调试语句	(72)
7.5	MS-COBOL磁带语法	(73)
7.6	过程部语句	(73)
7.6.1	ACCEPT语句	(73)

7.6.2	ADD语句	(83)
7.6.3	ALTER语句.....	(84)
7.6.4	CALL语句.....	(84)
7.6.5	CHAIN语句	(84)
7.6.6	CLOSE语句	(85)
7.6.7	COMPUTE语句	(85)
7.6.8	COPY语句.....	(85)
7.6.9	DELETE语句	(85)
7.6.10	DISPLAY语句	(85)
7.6.11	DIVIDE语句	(87)
7.6.12	EXHIBIT语句	(88)
7.6.13	EXIT语句	(88)
7.6.14	EXIT PROGRAM语句	(89)
7.6.15	GO TO语句	(89)
7.6.16	IF语句.....	(90)
7.6.17	INSPECT语句	(94)
7.6.18	MERGE语句.....	(97)
7.6.19	MOVE语句	(97)
7.6.20	MULTIPLY语句.....	(98)
7.6.21	OPEN语句.....	(99)
7.6.22	PERFORM语句	(99)
7.6.23	READ语句.....	(102)
7.6.24	READY/RESET TRACE语句.....	(102)
7.6.25	RELEASE语句.....	(103)
7.6.26	RESET TRACE语句.....	(103)
7.6.27	RETURN语句	(103)
7.6.28	REWRITE语句	(103)
7.6.29	SEARCH语句	(103)
7.6.30	SET语句.....	(103)
7.6.31	SORT语句	(103)
7.6.32	START语句	(103)
7.6.33	STOP语句	(104)
7.6.34	STRING语句.....	(104)
7.6.35	SUBTRACT语句.....	(105)
7.6.36	UNLOCK语句	(106)
7.6.37	UNSTRING语句	(107)
7.6.38	USE语句.....	(108)
7.6.39	WRITE语句	(108)
第八章	程序间通信.....	(109)

8.1	CALL语句	(109)
8.1.1	USING短语	(109)
8.1.2	ON OVERFLOW短语	(110)
8.2	EXIT PROGRAM语句	(110)
8.3	CHAIN语句	(110)
8.4	CANCEL语句	(110)
8.5	带有USING/CHAINING短语的过程部首部	(111)
第九章	用索引方法管理表	(112)
9.1	索引名和索引数据项	(112)
9.2	下标	(112)
9.3	相对索引	(112)
9.4	SET 语句	(113)
9.5	SEARCH语句格式1	(113)
9.6	SEARCH语句格式2	(114)
第十章	顺序文件	(117)
10.1	顺序文件组织形式的定义	(117)
10.2	顺序文件输入输出的语法考虑	(117)
10.2.1	文件控制入口(环境部)	(117)
10.2.2	文件描述入口(数据部)	(118)
10.2.3	输入/输出控制段(环境部)	(118)
10.3	文件状态报告	(119)
10.4	过程部有关顺序文件的语句	(119)
10.4.1	CLOSE语句	(120)
10.4.2	OPEN语句	(120)
10.4.3	READ语句	(122)
10.4.4	REWRITE语句	(122)
10.4.5	WRITE语句	(123)
第十一章	索引文件	(125)
11.1	索引文件组织形式的定义	(125)
11.2	索引文件输入/输出的语法考虑	(125)
11.2.1	文件控制入口(环境部)	(125)
11.2.2	RECORD KEY子句	(126)
11.2.3	ALTERNATE RECORD KEY子句	(127)
11.2.4	文件描述入口(数据部)	(127)
11.2.5	输入/输出控制段(环境部)	(127)
11.3	文件状态报告	(128)
11.4	关于索引文件的过程部语句	(129)
11.4.1	CLOSE语句	(129)
11.4.2	DELETE语句	(130)

11.4.3	OPEN语句	(131)
11.4.4	READ语句	(132)
11.4.5	REWRITE语句	(133)
11.4.6	START语句.....	(133)
11.4.7	UNLOCK语句.....	(134)
11.4.8	WRITE语句.....	(134)
第十二章	相对文件	(136)
12.1	相对文件组织形式的定义	(136)
12.2	相对文件输入/ 输出的语法考虑	(136)
12.2.1	文件控制入口 (环境部)	(136)
12.2.2	RELATIVE KEY子句	(137)
12.2.3	文件描述入口 (数据部)	(137)
12.2.4	输入/ 输出控制入口 (环境部)	(137)
12.3	文件状态报告	(137)
12.4	对应于相对文件的过程部语句	(138)
12.4.1	CLOSE语句	(139)
12.4.2	DELETE语句	(139)
12.4.3	OPEN语句	(139)
12.4.4	READ语句	(140)
12.4.5	REWRITE语句	(141)
12.4.6	START语句.....	(141)
12.4.7	UNLOCK 语句	(142)
12.4.8	WRITE 语句	(142)
第十三章	分类/ 合并设置	(144)
13.1	语法考虑	(144)
13.1.1	文件控制入口	(144)
13.1.2	排序文件描述入口 (分类/ 合并)	(144)
13.1.3	输入/ 输出控制段	(144)
13.2	排序文件状态报告	(145)
13.3	SORT语句	(146)
13.4	MERGE语句	(146)
13.5	分类和合并顺序	(147)
13.5.1	输入过程和USING短语.....	(148)
13.5.2	输出过程和GIVING短语	(148)
13.6	限制	(148)
13.7	RELEASE语句	(149)
13.8	RETURN语句.....	(149)
13.9	例子	(150)
第十四章	声明区和USE语句.....	(159)

第十五章 分片功能.....	(161)
第十六章 COPY语句.....	(162)
第十七章 文件和记录上锁	(166)
17.1 文件上锁	(166)
17.2 记录上锁	(166)
17.3 语法考虑	(166)
17.3.1 文件控制入口 (SELECT子句)	(167)
17.3.2 OPEN、READ、START和UNLOCK语句	(168)
附录A 允许的MOVE操作数	(171)
附录B 嵌套条件 (IF) 语句	(172)
附录C 保留字.....	(174)
附录D ASCII字符集	(179)

第一章 简介

Microsoft_R COBOL在如下方面扩展了ANSI 1974 COBOL的能力:高级动词,为相对文件和索引文件处理构造的多用户文件共享,交互式 and 跟踪式调试用ACCEPT(接收)和DISPLAY(显示)语句实现的格式化屏幕管理。Microsoft COBOL的这些特点能为你提供必需的工具并为建立更好的商用程序。

怎样使用这个手册

第一章向你介绍Microsoft COBOL。

第二章和第三章讨论了MS_{TM}-COBOL语言,包括语言规范和概念,术语的定义,编码规则以及MS-COBOL编译器能够识别的语句种类。

第四章到第七章讨论了MS-COBOL程序的四个部(DIVISION),包括在这些部分中出现的一般语句和子句。每章的头一页给出了相应部分的一般格式,并按通常在MS-COBOL源程序的出现顺序安排。一般格式中的各个部分按字母表顺序讨论。

第八章到第十七章讨论了高级MS-COBOL功能。包括程序间通信、表处理、文件组织、排序/合并(SORT/MERGE)功能说明,USE语句,片,复制(COPY)语句,文件及记录上锁。

附录提供了允许的MOVE操作数,讨论了嵌套的条件(IF)语句,列出了所有COBOL保留字及ASCII字符。

1.1 语法规则

本手册中出现的语言元素的格式都将遵从下列规范:

大写 所有的大写字是Microsoft COBOL的保留字。设加下底线的保留字是任选项。它们的作用只是改善程序的可读性。Microsoft COBOL的保留字既可以大写形式也可以小写形式敲入。

大写 (加下底线)所有的加下底线的保留字都是关键字。除非包含它们的格式部分本身是任选项,否则这些字就是必需的。丢失或拼错关键字将产生错误。

小写 用小写字母组成的字代表用户必须用一个合法的条目来代替的项。

小写 (带后缀)如果在一个格式中一项出现了多次,那么必须加上一个字母或数字做后缀。后缀只是为了表达清楚而不改变那个项的意义(例: data-name-1, data-name-2)。

关系操作符(<, >, =)。字符小于(<),大于(>),等于(=),虽然它们没加下底线,但如果出现在一般格式中,也是必需的。

[] 任何语句或数据描述体中包在方括号中的部分都是任选的。

() 如果不会产生混淆的话,可以用小括号代替方括号描述任选项。

{ } 如果大括号把语句的几部分括起来,那么必须且只能选其中之一任选项。在可能引起混乱的情况下,也用大括号括起语句的部分并后随省略号来表示这个语句部

分的重复出现。

… 省略号 (…) 表示紧靠它们前面的单位将连续出现一或多次。一个单位可以是单独的小写字或由方括号或大括号括起来的一组字, 如果出现重复, 则单位将做为一个整体重复。

特殊字符 标点和特殊字符需要在一般格式中出现。标点将按2.3节说明的标点规则使用。

句号 句号在一般格式中作为分隔符出现。分号和逗号有时也可以用作分隔符。但它们是任选的。

分隔符后面的空格 作为分隔符, 所有的逗号, 分号、句号后面都要紧随一个空格。如果必要的话, 伴随一般格式的正文可以含有对格式使用的注解、限制及简化。

格式中所提到的子句 (如BLOCK子句) 或语句 (例PERFORM语句) 都将在正文的其余地方进行描述。

1.2 Microsoft COBOL符合ANSI标准

下面的表对Microsoft COBOL支持的模块与ANSI标准第二级做个比较, 给你一个Microsoft COBOL编译器的全部能力的了解。

Microsoft COBOL模块	实现级别
核心级	实现了全部
表处理	第二级特性
顺序输入/输出	
相对输入/输出	
索引输入/输出	
库	
程序间通信	实现了全部的第二级特性。动词CHAIN是语言准标的扩充。
排序/合并	实现了全部第二级特性, 扩充了SORT STATUS, 报告排序文件状态。
调试“报告一写者”	目前没实现, 但Microsoft COBOL包括了对ANST74标准COBOL扩展的跟踪式调试及交互式调试工具。
通讯	

1.3 Microsoft COBOL所作的扩展

Microsoft COBOL对ANSI 74标准COBOL做了如下的扩展:

1、提供了COMP-0, COMP-3及COMP-4等数据格式。COMP-3格式在每个字节中存放两位十进制数值数据。COMP-0和COMP-4则分别为2字节和4字节二进制整数。这三种数据类型都可以被用来减少数据的存贮需要的空间,减少数据文件的存贮空间以及加快一些运算的执行速度。

2、Microsoft COBOL能够定义屏幕属性,并可以以交互方式,在你的终端上显示这些属性及其它的屏幕定义。

这种能力是从Microsoft COBOL在数据部扩展的SCREEN SECTION及过程部扩展的ACCEPT及DISPLAY语句得到的。(见6.4.4节“屏幕节”和第七章“过程部”的ACCEPT语句的格式1,3,4及DISPLAY语句的格式3。)

3、小写字符被当做大写处理,除非它们是非数值直接量的一部分(用引号括起来)。

4、动态检错语句READY, RESET TRACE和EXHIBIT允许在程序的执行过程中显示过程名及数据项。

5、设备部中的SELECT子句支持带有RECORD KEY和ALTERNATE RECORD KEY子句的分开关键字任选项。见第11.2节“索引文件I/O的语法考虑”。

6、CHAIN语句和CHAINING短语扩大了程序间通信的范围,并且允许将程序装入内存并执行。

7、一个多任务文件共享机构支持多用户/多任务系统的文件处理。OPEN、READ、START、UNLOCK和SELECT子句中使用了新的语法。注意:UNLOCK子句是对标准语言的扩展。

8、对于排序文件,通过文件控制款(FILE-CONTROL)中的SORT STATUS子句实现了排序文件状态报告功能。

1.4 更多地了解COBOL

如果你刚接触COBOL程序设计不久,你大概想在使用本手册之前,更多地了解这个语言。下列文章都是专为初等程序员所写的教材:

1. 《COBOL程序设计:一个结构化方法》(COBOL Programming: A Structured Approach),作者Abel, Peter。Reston出版公司1980。
2. 《结构化COBOL程序设计简易指南》(A Simplified Guide to Structured COBOL Programming) 作者: McCracken, Daniel D。
3. 《学生用COBOL》(COBOL for Students), 作者Parkin, Andrew, Edward Arnold有限公司,1973。

《Microsoft COBOL》作者Seidel, Ken Dilithum印刷公司,1983。

4. 《结构化COBOL—原理及风格》(Structured COBOL—Fundamentals and Style) 作者Welburn, Tyler, Mayfield出版公司,1981。

第二章 语言元素

本章定义了本手册中使用的有关Microsoft COBOL程序的术语，也给出了Microsoft COBOL源程序编写规则及Microsoft COBOL能识别的命名规范。

2.1 源代码编写规则

源程序可以被写在标准程序纸上或终端上，下面给出的编写规则对两者都适用。尽管在个别终端上列号可能稍有不同，但其相应位置不变。

Microsoft COBOL编译器把源程序分为四个区域来解释：行号区（1—6列），续行区（第7列），A区（8—11列），B区（12—72列）。编译器忽略掉行号区以及72列以后各列的内容。

A区为下列源代码元素占用：部头、节头、段头、文件描述款标识（FD和SD），层号01和77。如果第7列的内容为星号（*）或斜杠（/），则A区也可以接收注解行。

B区为下列源代码元素占用：与FD或SD相关的文件名、SELECT子句、层号66、88及02到49和描述性子句及构成特定段描述款的数据描述。如果第7列含有连字符，B区也可以接收续行。

下面的图表说明了一行COBOL源程序中的A区、B区、续行区、忽略区及任选的行号区。

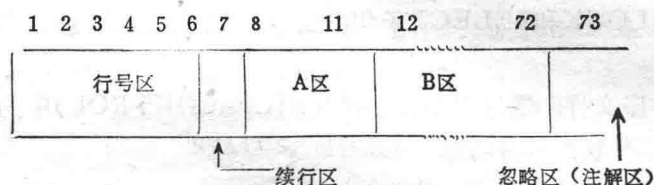


图 2.1 MS-COBOL源程序行格式

编写Microsoft COBOL程序要遵守下列规则：

1、代码的每一行可以在1至6列有一个六位数字的行号。这些行号必须从小到大排列。1至6列也可以为空格。

2、如果第7列是星号（*），这一行就被当作注解处理。将源程序列表时这一行将出现，其它情况下均被忽略。如果第7列是斜杠（/），这一行也被当做注解处理。但在对源程序列表时，将跳到新的一页开始显示这一行。

3、如果第7列是字符“D”，除非源计算机段中含有试调方式（WITH DEBUGGING MODE）子句，否则均被当做注解处理。见5.2.2节中关于源计算机段的内容。

4、如果第7列是连字符“-”，这一行将被当作上一行的续行处理。如果不是非数值直接量的续行的话，前一行的尾部空格及续行的头部空格均将被忽略掉。（见2.6.2节“非数值直接量”了解对非数值直接量续行的规定）。A区（8—11列）必须为空格。

5、部头、节头和段头的保留字必须从A区开始。过程名定义、层号01、77和FD及SD标识也要从A区开始。其它层号要从B区开始。

6、其它的语言元素都被限制在B区里。必须遵守语句的标点规则。

7、第72列以后的内容均被编译器忽略。

8、程序行中的制表字符要象《Microsoft COBOL编译器用户指南》中规定的那样展开。使用制表字符必须注意不能导致程序元素的非法位置。

2.2 字符集

Microsoft COBOL语言的字符集包括下列字符：

字母A到Z，a到z

空格

数字0到9。

特殊字符：

+ 加号 - 减号

* 星号 = 等号

> 关系符号 (大于) < 关系符号 (小于)

\$ 美元号 , 逗号

; 分号 . 句号或小数点

" 双引号 (左括号

) 右括号 ' 单引号

/ 斜杠

非数值直接量和注解所能使用的Microsoft COBOL字符集就是整个的计算机字符集。

2.3 标点

下列字符用作标点：

(左括号 ,) 右括号

, 逗号 . 句号 ; 分号

下面是编写源程序时使用标点的一般规则

1、当逗号、分号或句号被用作标点时，它们的前面可以没有空格，但后面必须有空格。

2、两个相连的字之间必须至少有一个空格。

3、关系符号的前后都必须有空格。

4、如果加号、减号、句号和逗号被用在PICTURE子句中，它们只须遵从数值编辑项的规则（见6.5.21节“PICTURE子句”中进一步的讨论）。

5、如果逗号被作为分隔符用在一个语句的连续的操作数之间，或两个下标之间，它的后面必须有一个空格（例：10, 30）。

6、分号或逗号可以用来分隔一系列的语句或子句。它们的后面必须有空格（例：SUBTRACT A FROM X, MOVE X TO Y）

2.4 保留字

保留字是在COBOL语言或Microsoft COBOL语言里有特定含义的字。它们在一般格式中以大写形式出现。它们可含有A到Z、a到z及数字0到9或连字符。最多可含30个字符。其中多数为谓词（例ADD、SUBTRACT、MOVE）或描述性短语（如：PICTURE、VALUE IS）。保留字不能被用作用户自定义字。

见附录C“保留字”中所有COBOL保留字的列表。

2.5 名字

任何不是Microsoft COBOL保留字的字都可用作用户自定义字，只要它符合下一节给出的命名规范。

2.5.1 命名规范

名字最长可达30个字符，而且必须只含字母A到Z、a到z及数字0到9和连字符。并且要满足：

- 1、除过程名以外的所有名字都必须至少含有一个字母或连字符。过程名则可以只由数字组成。

- 2、名字不能以连字符开头或结尾。一个名字可以含有多个连字符，它们也可以连在一起。

- 3、一个名字后面必须是空格或者是适当的标点。

- 4、如果一个用户名不唯一，就必须对它使用限定字。见2.5.2节“名字的限定”中对限定字的描述。

2.5.1.1 数据名

数据名是由用户给出的，用来标识程序中引用的数据项的字。

数据名在程序的数据部中定义。一个数据名总是对应于一个装有数据的区域，而不是对应于一个特定的值。因为在程序的执行过程中一个数据项可能取许多不同的值。

如果记录中的一些字符在程序执行过程中没有被引用，就不需要给它们一个数据名，而是用FILLER这个字来设置适当数量的空格。

数据名必须以字母开头。在每个数据描述款中，层号后面第一个字必须是数据名或关键字FILLER，并以下面的一般格式出现：

层号 { 数据名-1
FILLER }

见第三章“结构化COBOL程序设计”中关于层号的讨论。参看第六章“数据部”。

2.5.1.2 文件名

文件是一组数据记录，如打印的清单或磁盘的区域，它所包含的每个单独的记录都属于同种类型或应用。文件名在数据部的文件节中FD款定义，格式为：

FD 文件名

文件名的构成规则与数据名的相同。在设备部及过程部的OPEN、CLOSE和READ语句中能引用文件名。

2.5.1.3 条件名

条件名是分配给一个特定的值集的名字，或一个数据项能取到的一个完整值集中的值的范围的名字。条件名在数据部的88层款中定义。例：一个03层项“CLASS-NO”后面可能是从属的88层款：

88 VALID-NO VALUE IS '1'

88 INVALID-NO VALUE IS '2'

一个IF语句可以直接引用直接量 '1' 和 '2'，也可以引用条件名 VALID-NO和 INVALID-NO。

构成条件名的规则与数据名的相同。（见2.5.1.1节“数据名”）。条件名及其用法，在第六章“数据部”和第七章“过程部”中作了更全面的解释。

2.5.1.4 助记名

助记名（如PRINTER）指定了一个用户自定义名到设备名的联系。命名规则与数据名相同（见 2.5.1.1 节“数据名”）。在设备部的特殊名段中指定助记名。助记名可被 ACCEPT、DISPLAY语句及顺序文件的 WRITE语句引用。

2.5.1.5 过程名

过程名指定了由PERFORM或GOTO语句执行的或从程序其它部分转入的段或节。过程名在过程部中说明。过程名可以由纯碎的数字组成，否则，它必须遵从数据名的命名规则。（见2.5.1.1节“数据名”）。

2.5.1.6 索引名及索引数据项

索引名及索引数据项在用索引法进行表处理时使用。附加在OCCURE子句上的“INDEXED BY索引名”隐含地说明了索引名。索引数据项由USAGE IS INDEX子句定义。见第九章“表处理及索引法”中的深入讨论。

2.5.2 名字的限定

在数据名、条件名或段名（见第三章“COBOL程序结构”中关于段的描述）不唯一的情况下，引用这个名字的实例时要使用限定字。

如果有两个或更多的项叫YEAR，限定的引用：

YEAR OF HIRE

能够区分开HIRE或TERMINATION中的YEAR字段

限定字是跟在“OF”或“IN”后面的数据名或条件名。限定字必须指明包括引用的所有名字的多层组合。例如：HIRE必须是包含YEAR为从属组或以某种方式与YEAR有关系的数据名、条件名或库名（见第三章“COBOL程序结构”）。

段名可以被节名限定。COPY语句中的正文名可以被库名限定。

一个名字最多可有如下个限定字：段名与正文名只能有一个，其它项可有五个。

文件名和助记名必须唯一。

受限名可写在设备部、数据部和过程部中。如果对多次定义的段名的引用和定义在同一个节内，则不需使用限定符。

2.6 直接量

直接量就是常量。在程序中并没分配给它一个数据名，只对它的值进行引用，且其值保持不变。直接量可以是数值直接量、非数值直接量和象征常量。

2.6.1 数值直接量

数值直接量至少要含有一位数字但不得超过18位。数值直接量可由0到9的数字组成，前面的符号和小数点都是任选的。数值直接量只可以有一个符号字符和一个小数点。如果带有符号，那么它一定是数值直接量的最左字符如果无符号则认为是正的。

小数点可以出现在除去最右端以外的任何位置。如果一个数值直接量不含小数点则被认为是整数。

以下是数值直接量的例子：

72 +1011 3.14159 -6 -.333 0.5

在设备部的专用名 (SPECIAL NAMES) 段中包含 DECIMAL-POINT, IS COMMA款可以指定欧州记法 (句号与逗号互换)。在欧州记法中，数值直接量PI 将被写成3.14159。

2.6.2 非数值直接量

非数值直接量又叫“引号定界”直接量。它用双引号或单引号来定界，其中可含ASCII字符的任意组合。一般地说，如果用单引号做界符，则里面可以出现双引号，反之亦然。如果界符被用作直接量中的一个字符，则必须连续写两个。这种情况下，两个界符看成一个符号，例如：

"THE DATA-NAME" "VALID-NO" "IS ACCEPTED HERE"

将被解释成：

THE DATA-NAME "VALID-NO" IS ACCEPTED HERE

界符之间的空格都是直接量的一部分，并且在计算直接量长度的时候也把它们算进去。长度可以从1到120，界符不计算在内。

下面是非数值直接量的例子：

"ILLEGAL CONTROL CARD"

"CHARACTER-STRING"

"DO'S & DON'T'S"

非数值直接量可以从一行延续到下一行。

下面是续行规则：

- 1、续行的第7列要有连字符。
- 2、续行的A区为空格。
- 3、把界符放于B区，后面跟着延续的直接量。
- 4、上一行尾部的所有空格以及续行中从界符开始到续行结尾的空格都看成直接量的一部分。

2.6.3 象征常量

赋形常数是一种特殊类型的直接量。它代表了一个由Microsoft COBOL指定了保留数据名的值或字符。在程序编译的时候按需要提供这些值或字符。例如：象征常量SPACE将把它的整个域清为空格。LOW-VALUE表示机器的最小值。象征常量不需要用引号引起来。

在Microsoft COBOL中，以下的保留字是象征常数。这些字的复数形式编译器也能识别，并且与它们的单数形式等价。