

## 目录

### 第一章 介绍

#### 1.1 使用数据描述规范（DDS）建立文件

##### 1.1.1 填写 DDS 格式

##### 1.1.2 输入源语句

##### 1.1.3 建立文件

#### 1.2 语法规则

##### 1.2.1 键字和参数值

##### 1.2.2 DDS 命名规则

#### 1.3 语法编码举例

### 第二章 物理文件和逻辑文件

#### 2.1 定义物理文件

#### 2.2 定义逻辑文件

##### 2.2.1 单格式和多格式逻辑文件

##### 2.2.2 连接逻辑文件

##### 2.2.3 在逻辑文件指定记录格式

#### 2.3 位置项（1 到 44 列）

##### 2.3.1 顺序号（1 到 5 列）

##### 2.3.2 格式类型（第 6 列）

##### 2.3.3 注释（第 7 列）

##### 2.3.4 条件（第 8 到 16 列）

##### 2.3.5 名字或规范的类型（第 17 列）

##### 2.3.6 保留（第 18 列）

##### 2.3.7 名字（第 19 到 28 列）

##### 2.3.8 引用（第 29 列）

##### 2.3.9 长度（第 30 到 34 列）

##### 2.3.10 数据类型（第 35 列）

##### 2.3.11 小数位（36 到 37 列）

##### 2.3.12 用法（第 38 列）

##### 2.3.13 位置（第 39 到 44 列）

#### 2.4 键字项（第 45 到 80 列）

##### 2.4.1 ABSVAL（绝对值）

##### 2.4.2 ALIAS（替换名）

##### 2.4.3 ALL（全部）——仅用于逻辑文件

##### 2.4.4 ALTSEQ（交替分配顺序）

##### 2.4.5 ALWNULL（允许空值）——仅用于物理文件

##### 2.4.6 CCSIP（编码字符集标识）——仅用于物理文件

##### 2.4.7 CHECK（检查）

##### 2.4.8 CHKMSGID（检查信息标识）

##### 2.4.9 CMP（比较）

##### 2.4.10 COLHDG（栏目标题）

##### 2.4.11 COMP（比较）

##### 2.4.12 CONCAT（连接）

##### 2.4.13 DATFMT（日期格式）

- 2.4.14 DATSEP (日期分隔符)
- 2.4.15 DESCEND (降序)
- 2.4.16 DFT (缺省值) ——仅用于物理文件
- 2.4.17 DIGIT (数字)
- 2.4.18 DYNSLT (动态选择) ——仅用于逻辑文件
- 2.4.19 EDTCDE (编辑码) 和 EDTWRD (编辑字)
- 2.4.20 FCFO (先修改先出)
- 2.4.21 FIFO (先进先出)
- 2.4.22 FLTPCN (浮点精度)
- 2.4.23 FORMAT (格式)
- 2.4.24 JDFTVAL (连接的缺省值) ——仅用于连接逻辑文件
- 2.4.25 JOURNALSEQ (连结的重复顺序) ——仅用于连接逻辑文件
- 2.4.26 JFILE (被连接的文件) ——仅用于连接逻辑文件
- 2.4.27 JFLD (被连接的字段) ——仅用于连接逻辑文件
- 2.4.28 JOIN (连接) ——仅用于连接逻辑文件
- 2.4.29 JREF (连接引用) ——仅用于连接逻辑文件
- 2.4.30 LIFO (后进先出)
- 2.4.31 NOALTSEQ (无交替顺序)
- 2.4.32 PFILE (物理文件) ——仅用于逻辑文件
- 2.4.33 RANGE (范围)
- 2.4.34 REF (引用) ——仅用于物理文件
- 2.4.35 REFACCPATH (引用访问路径) ——仅用于逻辑文件
- 2.4.36 REFFLD (被引用字段) ——仅用于物理文件
- 2.4.37 REFSHIFT (引用换档)
- 2.4.38 RENAME (重命名) ——仅用于逻辑文件
- 2.4.39 SIGNED (符号)
- 2.4.40 SST (子串) ——仅用于逻辑文件
- 2.4.41 TEXT (正文)
- 2.4.42 TIMFMT (时间格式)
- 2.4.43 TIMSEP (时间分隔符)
- 2.4.44 TRNTBL (转换表) ——仅用于逻辑文件
- 2.4.45 UNIQUE (唯一)
- 2.4.46 UNSIGNED (无符号)
- 2.4.47 VALUES (值)
- 2.4.48 VARLEN (变长字段)
- 2.4.49 ZONE (零)

### 第三章 显示文件

- 3.1 定义显示文件
- 3.2 位置项 (1 到 44 列)
  - 3.2.1 顺序号 (1 到 5 列)
  - 3.2.2 格式类型 (第 6 列)
  - 3.2.3 注释 (第 7 列)
  - 3.2.4 条件 (7 到 16 列)
  - 3.2.5 多键字字段条件

- 3.2.6 名字或规范的类型（第 17 列）
- 3.2.7 保留（第 18 列）
- 3.2.8 名字（19 到 28 列）
- 3.2.9 引用（第 29 列）
- 3.2.10 长度（30 到 34 列）
- 3.2.11 数据类型/键盘换档（第 35 列）
- 3.2.12 小数位（36 到 37 列）
- 3.2.13 用法（第 38 列）
- 3.2.14 位置（39 到 44 列）
- 3.3 键字项（45 到 80 列）
  - 3.3.1 ALARM（报警）
  - 3.3.2 ALIAS（替换名）
  - 3.3.3 ALTHELP<sup>A</sup>（替换帮助键）
  - 3.3.4 ALTNAME（替换记录名）
  - 3.3.5 ALTPAGEDWN/ALTPAGEUP（替换的翻下页/翻上页键）
  - 3.3.6 ALWGPH（允许图形）
  - 3.3.7 ALWROL（允许滚动）
  - 3.3.8 ASSUME（假定）
  - 3.3.9 AUTO（自动）
  - 3.3.10 BLANKS（空白）
  - 3.3.11 BLINK（闪烁）
  - 3.3.12 BLKFOLD（空白折叠）
  - 3.3.13 CAnn（命令注意）
  - 3.3.14 CFnn（命令功能）
  - 3.3.15 CHANGE（修改）
  - 3.3.16 CHCACCEL（选项加速说明）
  - 3.3.17 CHCAVAIL（可用的选项颜色或显示属性）
  - 3.3.18 CHCCTL（选项控制）
  - 3.3.19 CHCSLT（被选择的选项颜色/显示属性）
  - 3.3.20 CHCUNAVAIL（不可用选项的颜色/显示属性）
  - 3.3.21 CHECK（检验）
  - 3.3.22 CHGINPDFT（改变输入缺省值）
  - 3.3.23 CHKMSGID（检验信息标识）
  - 3.3.24 CHOICE（选项）
  - 3.3.25 CHRID（字符标识）
  - 3.3.26 CLEAR（清除）
  - 3.3.27 CLRL（清除行）
  - 3.3.28 CMP（比较）
  - 3.3.29 CNTFLD（连续输入字段）
  - 3.3.30 COLOR（颜色）
  - 3.3.31 COMP（比较）
  - 3.3.32 CSRINPONLY（光标移至仅输入位置）
  - 3.3.33 CSRLOC（光标位置）
  - 3.3.34 DATE（日期）

- 3.3.35 DFT (缺省)
- 3.3.36 DFTVAL (缺省值)
- 3.3.37 DLTCHK (删除检验)
- 3.3.38 DLTEDT (删除编辑)
- 3.3.39 DSPATTR (显示属性)
- 3.3.40 DSPMOD (显示方式)
- 3.3.41 DSPRL (由右至左显示)
- 3.3.42 DSPSIZ (显示尺寸)
- 3.3.43 DUP (重复)
- 3.3.44 EDTCDE (编辑码)
- 3.3.45 DETMSK (编辑屏蔽)
- 3.3.46 EDTWRD (编辑字)
- 3.3.47 ENTFLDATR (进入字段属性)
- 3.3.48 ERASE (清除)
- 3.3.49 ERASEINP (清除输入)
- 3.3.50 ERRMSG (错误信息) 和 ERRMSGID (错误信息标识)
- 3.3.51 ERRSFL (错误子文件)
- 3.3.52 FLDCSRPRG (光标前进字段)
- 3.3.53 FLTFIXDEC (浮点到定点十进制)
- 3.3.54 FLTPCN (浮点精度)
- 3.3.55 FRCDTA (强制数据)
- 3.3.56 GETRETAIN (保留)
- 3.3.57 HELP<sup>A</sup> (帮助)
- 3.3.58 HLPARA (帮助区)
- 3.3.59 HLPBDY (帮助边界)
- 3.3.60 HLPCLR (帮助清除)
- 3.3.61 HLPCMDKEY (帮助命令键)
- 3.3.62 HLPDOC (帮助文件)
- 3.3.63 HLPEXCLD (禁止帮助)
- 3.3.64 HLPFULL (全屏帮助)
- 3.3.65 HLPIID (帮助标识)
- 3.3.66 HLPPNLGRP (帮助面板组)
- 3.3.67 HLPRCD (帮助记录)
- 3.3.68 HLPRTN (帮助返回)
- 3.3.69 HLPSCIDX (帮助查询索引)
- 3.3.70 HLPSEQ (帮助顺序)
- 3.3.71 HLPSHELF (帮助书架)
- 3.3.72 HLPTITLE<sup>A</sup> (帮助标题)
- 3.3.73 HTML (超级文本标识语言)
- 3.3.74 HOME (home 键)
- 3.3.75 INDARA (指示器区)
- 3.3.76 INDTXT (指示器正文)
- 3.3.77 INVITE (申请)
- 3.3.78 INZINP (初始输入)

- 3.3.79 INZRCD (初始记录)
- 3.3.80 KEEP (保持)
- 3.3.81 LOCK (加锁)
- 3.3.82 LOGINP (日志输入)
- 3.3.83 LOGOUT (日志输出)
- 3.3.84 LOWER (小写)
- 3.3.85 MDTOFF (关闭修改数据标记)
- 3.3.86 MLTCHCFLD (多选项选择字段)
- 3.3.87 MNUBAR (菜单项)
- 3.3.88 MNUBARCHC (菜单条选项)
- 3.3.89 MNUBARDSP (菜单条显示)
- 3.3.90 MNUBARSEP (菜单条分隔符)
- 3.3.91 MNUBARSW (菜单条开关键)
- 3.3.92 MNUCNL (菜单取消键)
- 3.3.93 MUUBTN (鼠标键)
- 3.3.94 MSGALARM (信息报警)
- 3.3.95 MSGCON (信息常量)
- 3.3.96 MSGID (信息标识)
- 3.3.97 MSGLOC (信息位置)
- 3.3.98 NOCCSID (无编码字符集标识)
- 3.3.99 OPENPRT (打开打印机文件)
- 3.3.100 OVERLAY (复盖)
- 3.3.101 OVRATTR (复盖属性)
- 3.3.102 OVRDTA (复盖数据)
- 3.3.103 PAGEDOWN/PAGEUP (下页/上页键)
- 3.3.104 PASSRCD (传送记录)
- 3.3.105 PRINT<sup>A</sup> (打印)
- 3.3.106 PROTECT (保护)
- 3.3.107 PSHBTNCHC (按钮字段选项)
- 3.3.108 PSHBTNFLD (按钮字段)
- 3.3.109 PULLDOWN (下拉菜单)
- 3.3.110 PUTOVR (放置明显复盖)
- 3.3.111 PUTRETAIN (设置保留)
- 3.3.112 RANGE (范围)
- 3.3.113 REF (引用)
- 3.3.114 REFFLD (引用字段)
- 3.3.115 RETKEY (保持功能键) 和 RETCMDKEY (保持命令键)
- 3.3.116 RETLCKSTS (保持加锁状态)
- 3.3.117 RMVWDW (取消窗口)
- 3.3.118 RULLUP/RULLDOWN (上翻/下翻)
- 3.3.119 RULLUP/RULLDOWN (例子)
- 3.3.120 RTNCSRLOC (返回光标位置)
- 3.3.121 RTNDTA (返回数据)
- 3.3.122 SETOF (设置断开)

- 3.3.123 SETOFF (设置断开)
- 3.3.124 SFL<sup>A</sup> (子文件)
- 3.3.125 SFLCHCCTL (子文件选项控制)
- 3.3.126 SFLCLR (子文件清除)
- 3.3.127 SFLCSRPRG (子文件光标处理)
- 3.3.128 SFLCSRRRN (子文件光标相对记录号)
- 3.3.129 SFLCTL (子文件控制)
- 3.3.130 SFLDLT (子文件删除)
- 3.3.131 SFLDROP (子文件撤消)
- 3.3.132 SFLDSP (子文件显示)
- 3.3.133 SFLDSPCTL (子文件显示控制)
- 3.3.134 SFLEND (子文件结束)
- 3.3.135 SFLENTER (子文件 enter 键)
- 3.3.136 SFLFOLD (子文件折叠)
- 3.3.137 SFLINZ (子文件初始化)
- 3.3.138 SFLLIN (子文件行)
- 3.3.139 SFLMLTCHC (子文件多选项选择列表)
- 3.3.140 SFLMODE (子文件方式)
- 3.3.141 SFLMSG (子文件信息) 和 SFLMSGID (子文件信息标识)
- 3.3.142 SFLMSGKEY (子文件信息键)
- 3.3.143 SFLMSGRCD (子文件信息记录)
- 3.3.144 SFLNXTCHG (子文件的下一个修改)
- 3.3.145 SFLPAG<sup>A</sup> (子文件页)
- 3.3.146 SFLPGMQ (子文件程序信息队列)
- 3.3.147 SFLRCDNBR (子文件记录号)
- 3.3.148 SFLRNA (子文件非活动记录)
- 3.3.149 SFLROLYAL (子文件翻卷值)
- 3.3.150 SFLRTNSEL (子文件返回选择的选项)
- 3.3.151 SFLSCROLL (子文件上翻)
- 3.3.152 SFLSIZ (子文件大小)
- 3.3.153 SFLSNGCHC (子文件单选项选择列表)
- 3.3.154 SLNO (开始行号)
- 3.3.155 SNGCHCFLD (单项选项选择字段)
- 3.3.156 SYSNAME (系统名)
- 3.3.157 TEXT (正文)
- 3.3.158 TIME (时间)
- 3.3.159 UNLOCK (解锁)
- 3.3.160 USER (用户)
- 3.3.161 USRDFN (用户定义)
- 3.3.162 USRDSPMGT (用户显示管理)
- 3.3.163 USRRSTDSP (用户重存显示)
- 3.3.164 VALNUM (有效数字)
- 3.3.165 VALUES (值)
- 3.3.166 VLDCMDKEY (有效命令键)

3.3.167 WDWBORDER (窗口边界)

3.3.168 WDWTITLE (窗口标题)

3.3.169 WINDOW (窗口)

3.3.170 WRDWRAP (字重叠)

## 第四章 打印文件

4.1 定义一个打印文件

4.2 在打印设备描述中有 AFP(\*YES)的键字

4.3 位置项 (1 到 44 列)

4.3.1 顺序号 (1 到 5 列)

4.3.2 格式类型 (第 6 列)

4.3.3 注释 (第 7 列)

4.3.4 条件 (7 到 16 列)

4.3.5 名字或规范的类型 (第 17 列)

4.3.6 保留 (第 18 列)

4.3.7 名字 (19 到 28 列)

4.3.8 引用 (第 29 列)

4.3.9 长度 (30 到 34 列)

4.3.10 数据类型 (第 35 列)

4.3.11 小数位数 (36 到 37 列)

4.3.12 用法 (38 列)

4.3.13 位置 (39 到 44 列)

4.4 键字项 (45 到 80 列)

4.4.1 ALIAS (替换名)

4.4.2 BARCODE (条型码)

4.4.3 BLKFOLD (空白折叠)

4.4.4 BOX (方框)

4.4.5 CDEFNT (编码字型名)

4.4.6 CHRID (字符标识)

4.4.7 CHRSIZ (字符尺寸)

4.4.8 COLOR (颜色)

4.4.9 CPI (每英寸字符数)

4.4.10 CVTDTA (转换数据)

4.4.11 DATE (日期)

4.4.12 DFNCHR (定义字符)

4.4.13 DFT (缺省)

4.4.14 DLTEDT (删除编辑)

4.4.15 DRAWER (纸盘)

4.4.16 DTASTMCMD (数据流命令)

4.4.17 EDTCDE (编辑码)

4.4.18 EDTWRD (编辑字)

4.4.19 ENDPAGE (结束页)

4.4.20 FLT FIXPEC (浮点到定点十进制)

4.4.21 FLTPCN (浮点精度)

4.4.22 FNTCHRSET (字形字符集)

- 4.4.23 FONT (字形)
- 4.4.24 GDF (图形数据文件)
- 4.4.25 HIGHLIGHT (高亮度)
- 4.4.26 INDARA (指示器区)
- 4.4.27 INDTXT (指示器正文)
- 4.4.28 INVMMAP (激活中间映象)
- 4.4.29 LINE (线)
- 4.4.30 LPI (每英寸行数)
- 4.4.31 MSGCON (信息内容)
- 4.4.32 OVERLAY (复盖)
- 4.4.33 PAGNBR (页号)
- 4.4.34 PAGRTT (页旋转)
- 4.4.35 PAGSEG (页段)
- 4.4.36 POSITION (位置)
- 4.4.37 PRTQLTY (打印特性)
- 4.4.38 REF (引用)
- 4.4.39 REFFLD (引用字段)
- 4.4.40 SKIPA (打印后跳行)
- 4.4.41 SKIPB (打印前跳行)
- 4.4.42 SPACEA (打印后空行)
- 4.4.43 SPACEB (打印前空行)
- 4.4.44 TEXT (正文)
- 4.4.45 TIME (时间)
- 4.4.46 TRNSPY (透明度)
- 4.4.47 TXTRTT (正文旋转)
- 4.4.48 UNDERLINE (下划线)

## 第五章 系统内部通迅功能文件

- 5.1 定义一个系统内部通讯功能文件
- 5.2 位置项 (1 到 44 列)
  - 5.2.1 顺序号 (1 到 5 列)
  - 5.2.2 格式类型 (第 6 列)
  - 5.2.3 注释 (第 7 列)
  - 5.2.4 条件 (7 到 16 列)
  - 5.2.5 名字或规范类型 (第 17 列)
  - 5.2.6 保留 (第 18 列)
  - 5.2.7 名字 (19 到 28 列)
  - 5.2.8 引用 (29 列)
  - 5.2.9 长度 (30 到 34 列)
  - 5.2.10 数据类型 (35 列)
  - 5.2.11 小数位数 (36 到 37 列)
  - 5.2.12 用法 (38 列)
  - 5.2.13 位置 (39 到 44 列)
- 5.3 键字项 (45 到 80 列)
  - 5.3.1 ALIAS (替换名)

- 5.3.2 ALWWRT (允许写)
- 5.3.3 CANCEL (取消)
- 5.3.4 CNLINVITE (取消请求)
- 5.3.5 CONFIRM (确认)
- 5.3.6 CTLDTA (控制数据)
- 5.3.7 DETACH (脱离)
- 5.3.8 DFREVOKE (延迟激活)
- 5.3.9 ENDGRP (分组结束)
- 5.3.10 EOS (会话结束)
- 5.3.11 EVOKE (激活)
- 5.3.12 FAIL (失败)
- 5.3.13 FLTPCN (浮点精度)
- 5.3.14 FMH (功能管理首标)
- 5.3.15 FMTNAME (格式名)
- 5.3.16 FRCDTA (强制数据)
- 5.3.17 INDARA (指示器区)
- 5.3.18 INDTXT (指示器正文)
- 5.3.19 INVITE (请求)
- 5.3.20 NEGRSP (负应答)
- 5.3.21 PRPCMT (准备落实)
- 5.3.22 RCVCANCEL (接收取消)
- 5.3.23 RCVCONFIRM (接收确认)
- 5.3.24 RCVCTLDTA (接收控制数据)
- 5.3.25 RCVDETACH (接收脱离)
- 5.3.26 RCVENDGRP (接收结束分组)
- 5.3.27 RCVFAIL (接收失败)
- 5.3.28 RCVFMH (接收功能管理标题)
- 5.3.29 RCVNEGRSP (接收负应答)
- 5.3.30 RCVROLLB (接收返回应答指示器)
- 5.3.31 RCVTKCMT (接收使用落实应答指示器)
- 5.3.32 RCVTRNRND (接收转向)
- 5.3.33 RECID (记录标识)
- 5.3.34 REF (引用)
- 5.3.35 REFFLD (引用字段)
- 5.3.36 RQSWRT (申请写)
- 5.3.37 RSPCONFIRM (应答确认)
- 5.3.38 SECURITY (安全)
- 5.3.39 SUBDEV (子设备)
- 5.3.40 SYNLVL (同步级)
- 5.3.41 TEXT (正文)
- 5.3.42 TIMER (计时器)
- 5.3.43 TNSSYNLVL (传输同步级)
- 5.3.44 VARBUFMGT (可变缓冲区管理)
- 5.3.45 VARLEN (变长)

修改一览表:

显示文件新增加的键字:

显示文件增加了下列键字:

CSRINONLY (光标移至仅输入位置)  
HLPSHELF (帮助书架)  
MOUBTN (鼠标按键)  
PSHBTNCHC (按钮开关字段选择)  
PSHBTNFLD (按钮开关字段)  
SFLCHCCTL (子文件选择控制)  
SFLMLTCHC (子文件多项选择表)  
SFLRTNSEL (子文件返回选择)  
SFLSCROLL (子文件上卷)  
SFLSNGCHC (子文件单项选择表)  
VALNUM (有效数字)  
WDWTITLE (窗口标题)  
WRDWWRAP (字卷)

这些键字在第三章“显示文件”按字母顺序来讲述。

加到 DBCS 设备中一些新的显示文件键字 (具有调用日语 DDS 的能力):

下面键字加到显示文件中:

GRDATR (坐标属性)  
GRDBOX (坐标框)  
GRDCLR (坐标清除)  
GRDLIN (坐标行)  
GRDRCD (坐标记录)

这些键字在附录 E 中以字母顺序来讲述。

以下新的键字加到系统内部通讯功能文件中:

PRPCMT (准备落实)  
RCVROLLB (接收返回响应指示器)  
RCVTKCMT (接收取落实响应指示器)  
TNSSYNLVL (转换同步级)

这些键字在第五章“系统内部通讯功能文件”中以字母顺序讲述。

下面键字加到 DBCS 文件中:

CNTFLD (续行项字段)

这个键字在附录 E 中讲述。

以下是显示文件中修改过的键字:

EDTWRD (编辑字)  
MLTCHCFLD (多项选择字段)  
MNUBAR (菜单项)  
PULLDOWN (下拉菜单)  
RTNCSRLOC (返回光标位置)

**SFLEND** (子文件结束)  
**SFLRCDNBR** (子文件记录号)  
**SFLSIZ** (子文件大小)  
**SNGCHCFLD** (单项选择字段)  
**WINDOW** (窗口)

这些键字在第三章“显示文件”中以字母顺序讲述。

以下是打印文件修改过的键字：

**GDF** (图形数据文件)  
**OVERLAY** (复盖)  
**PAGSEG** (页段)

这些键字在第四章“打印文件”中以字母顺序讲述。

系统内部通讯功能文件修改过的键字：

**SYNLVL** (同步级)

这个键字在第五章“系统内部通讯功能文件”中讲述。

需在打印设备描述中有 AFP(\*YES)的键字：

对于 V3R1，这种先进功能的打印系统 (AFP) 是做为 OS/400 中称作打印服务设备/400 (PSF/400) 的独立部分定购的。为了能够打印特殊值的某些键字所规定的值，需要 PSF/400。这些键字在 4.2 中讲述。

# 第一章 介 绍

说明数据属性的传统方法是在应用程序中指定它们自己的数据属性（诸如：记录和字段的名字与长度）。但在 AS/400 系统上用一个更方便更有效的替换方法，数据描述规范（DDS）可以在处理数据的应用程序外部来描述数据的属性。DDS 可以用来描述下列类型的文件：

- 物理文件（DDS 是可选的）
- 逻辑文件（DDS 是必须的）
- 显示文件（DDS 是可选的）
- 打印机文件（DDS 是可选的）
- ICF 文件（DDS 是必须的）

## 1.1 使用数据描述规范（DDS）建立文件

要使用 DDS 建立文件，遵循以下步骤：

1. 填写数据描述规范（DDS）的表格

2. 在源文件中进入源语句，这个源文件可以是 AS/400 数据库的一部分，（例如：在一个 IBM 提供的 QGRL 库中的 QDOSSRC 的一个源物理文件中）或在一张软磁盘上。

3. 使用相应的建立文件命令建立这个文件。

注：通过屏幕设计辅助工具（SDA）实用程序，可使用提供给显示文件的功能就可以建立和测试显示文件，而不用写 DDS 语句，详见 SDA 用户指南。

### 1.1.1 填写 DDS 格式

图 1-1 给出缩小了的简单数据描述规范表格。

DDS 表格的左边（1~44 列）是称为位置项的固定格式项。位置项用来定义像字段的名字，长度等记录格式和字段的最常用的属性。关于最重要的位置项的简要说明，参照图下面的 1~7。更详细的解释看第二章到第五章的位置项部分。

DDS 表格的右边（第 45~80 列）是 DDS 键字。DDS 键字定义文件、记录格式和字段的不太常用的和变化较多的属性。它们遵循控制语言句法规则。关于键字项的简要说明见图下面的 8 项。更详细的解释，请看在第二章到第五章的键字项部分。

Pic1

### 图 1-1 位置项和关键字项的格式

- 1.顺序号和格式类型在 DDS 中是可选的。格式类型指出源文件是一个 DDS 源语句。这一项对所有的文件类型都是有效的。
  - 2.在第 7 列如果是“\*”表示这行是一个注释行。这一项对所有类型的文件都有效。当在第 7 列给出 A(and)或 O(or)或一个空白时，8~16 列可以提供这行或紧接这行的某些条件。对于物理或逻辑文件条件是无效的。
  - 3.名字或规范的类型（第 17 列）指明名字项（19~28 列）或者规范的类型：
- | 名字项 | 说明       | 文件的类型        |
|-----|----------|--------------|
| R   | 一个记录格式名  | 所有的          |
| 空白  | 一个字段名    | 所有的          |
| K   | 一个关键字字段  | 仅用于物理文件和逻辑文件 |
| S   | 一个选择字段的名 | 仅用于逻辑文件      |
| O   | 一个省略字段的名 | 仅用于逻辑文件      |
| J   | 一个连接文件规范 | 仅用于连接逻辑文件    |
| H   | 一个帮助规范   | 仅用于显示文件      |
- 4.在第 29 列有 R 指出了这个名字字段（在 19~28 列）引用了在另外一个地方定义的某一字段的属性。逻辑文件不用这列。
  - 5.长度、数据类型的小数位指定了在一个记录格式内命名字段的属性。这对所有的文件都有效。
  - 6.用法（USAGE）指明了这个字段是作为输入、输出，输入/输出，既不输入也不输出，复盖信息或是程序到系统的字段。关于字段的使用，每一类型的文件皆有其自己的限制。
  - 7.位置指定了一个字段在显示屏上或打印机页上的位置。只对显示和打印机文件使用。
  - 8.通过关键字的使用，对不同类型文件的不同层次指定某些功能，如下所示：

关键字的层次	适用的文件类型
FILE (文件)	所有的文件类型
RECODE (记录)	所有的文件类型
FIELD (字段)	所有的文件类型
JOIN (逻辑)	仅用于连接逻辑文件
KEY FIELD (关键字字段)	仅用于物理文件和逻辑文件
SELECT OR OMIT FIELD (选择或省略字段)	仅用于逻辑文件
HELP (帮助)	仅用于显示文件

对于显示和打印文件，在单引号中的常数做为显示或打印的缺省值。

#### 1.1.2 输入源语句

在填完这个表格之后，把源语句输入到源文件中。可以交互或以批处理的方式输入这些源语句。

使用 SEU 交互式的输入源语句。详细信息见 SEU 用户指南。在命令行进入 STRSEU 命令来调用 SEU。

使用批处理的方式输入源语句是用软盘，使用下面的方法之一：

- 在磁盘上输入含有 DDS 源码的控制语言命令的语句，然后用启动磁盘读命令（STRDKTRDR）启动一个假脱机读动作。
- 在磁盘上仅输入源语句，然后使用复制文件（CPYF）命令把产生的数据文件复制到一个源物理文件中。
- 在磁盘上仅输入源语句并键入一个建立文件的命令。命令中的 SRCFILE 参数写数据

文件名，在 SRCMBR 参数上写\*FILE。

注：这个方法不能建立一个源物理文件。

### 1.1.3 建立文件

通过生成文件命令建立文件。可以根据要建立的文件类型使用相应的命令。文件类型和命令是：

文件类型	命令
物理文件	CRTPF
逻辑文件	CRTLE
显示文件	CRTDSPF
打印机文件	CRTPRTF
ICF 文件	CRTICFF

发出一个生成文件命令时，从源文件中检索 DDS 并检查有效性，如下图所示建立一个文件。仅在 DDS 中的错误不大于建立文件命令的 GENLVL 参数中指定的严重程度时，才能建立文件。当建立文件时，可以使用 GENLVL 参数来控制可允许的错误严重性。根据在 OPTION 和 FLAG 参数上指定的选项，还可以产生 DDS（或编译）源清单，DDS 清单包括数据描述和错误信息。

可以用 FLAG 参数规定要打印的 DDS 信息的级别。例如，可以不打印字复盖的警告信息。

PIC2

## 1.2 语法规则

这一部分介绍 DDS 键字和名字的一般信息。

### 1.2.1 键字和参数值

DDS 键字和参数值的语法与 CL（控制语言）语法相似。DDS 的语法规则是：

除单引号中的字符值和在双引号中的扩展名外，所有的 DDS 项必须大写。

DDS 键字及其相关项要在同一行或其续行上。

至少用一个空白分隔多个键字。键字的参数值必须用括号括起来。左括号必须紧跟在键字之后。例如：

KEYWORD(VALUE)

这个规则与控制语言略有不同。用控制语言时，参数值可以是以位置确定的。DDS 的语法要求指定键字，除非指定 DFT 键字和常数参数值。

使用至少一个空格来分隔同一键字的多个参数值。如：

KEYWORD(VALUEA VALUEB)

参数表达式用括号括起来。参数表达式是指一个特定的值紧跟着参数的值。这个特定值

以左括号开头，后面写\*号，参数值放在右括号之前。至少用一个空格分开特定值与参数值。参数表达式表示一个参数值，必须至少用一个空格同其它参数值分开。例如：

KEYWORD(VALUEA (\*特定值 VALUEB) VALUEC)

使用单引号括起字符值。数字值不用单引号。在语法中字符值可以出现在两个地方：

——作为某些键字的参数值。如 TEXT (对所有类型的文件) 和 COLHDG (数据库文件)，需要字符串作为正文描述。其它的键字，象 CAnn 和 CFnn，使用字符串作为应答指示器的正文描述。

——做显示和打印文件常数字段的缺省值 (用或不用 DFT 键字)。在显示文件中，也可以对命名字段指定一个字符常数。既使没指定 DFT 键字，只指定一个字符常数就隐含 DFT 键字。

字符串本身有一个单引号，要写两个单引号，这样在输出时才有一个单引号出现。如：

KEYWORD('customer" s name)

打印时为 Customer's name

当一个键字和它的参数值在一行上放不下时可以用加号 (+) 或减号 (-) 作为续行字符。这个符号必须是功能字段 (45 - 80 列) 的最后一个非空字符。一个语句最多续 5000 个字符位。

——一个减 (-) 号的意义是在下一行的第 45 位开始续行 (功能字段的第一位，即第 45 列)

——一个加 (+) 号的意义在下一行上的功能字段中的第一个非空白 (第一个有意义的) 字符开始续行

如果在参数值中指定了一个续行字符，那么在这个参数值中的续行符之前的空白都属于这个参数值。

对一个要到下一行上指定条件的键字，用 (+) 号作为这一行的最后一个非空字符。这一点在条件中包含了 n 个可选的指示器并应用于几个键字时是有帮助的。

OS/400 继续一个 DDS 语句直到出现下面之一的条件为止：

——一个记录格式名 (第 17 列出现 R)

——一个字段的规范 (字段名或位置)

——对于物理或逻辑文件，一个键字字段规范 (第 17 列出现 K)

——对于逻辑文件，一个选择或省略的规范 (第 17 列出现 S 或 O)

——对于连接逻辑文件，一个连接规范 (第 17 列出现 J)

——对于显示文件，一个帮助规范 (第 17 列出现 H)

——对于设备文件，一个可选指示器或一个键字，字段或字段位置的条件名

——DDS 语句的最大长度 (5000 字符)，这里包含语句中第一行的位置项 (第 1 列到第 44 列)，这样键字用的最大长为 4956 位。

键字描述使用下面这些符号说明键字的语法：

( ) 包括在其中的值是必须的

[ ] 包含在其中的值是可选的

[...] 需要时指定附加值

{ } 上面的值为缺省值 (见 REFFLD)

| 指定左边的值或者是右边的值，二者选一 (可引用可选值)

### 1.2.2 DDS 命名规则

用于 DDS 的命名规则如下：

限定名：

——用斜线来分开限定名各部分，不允许嵌入空格。例如：

KEYWORD(library/file)

——对于大多数和有限定名参数的键字，可以用\*LIBL 或\*CURLIB 做库名。如果不指定库名，就用\*LIBL。但不能用库名\*USR LIBL。这同 CL 语言的规则是不同的，CL 语言允许\*USR LIBL。

——目标名最多 10 个字符。如果用引号括住名字，那么在引号之间最多允许 8 个字符。这个规则与 CL 语言是不同的。CL 语言允许在引用符号内出现 10 个字符。有关内容请参考 CL 指南。

记录和文件名：

——名字必须少于或等于 10 个字符。

——名字必须以字母字符 (A 到 Z, @, \$和#) 开头。其后的字符可以是字母数字 (A 到 Z, 0 到 9, @, \$, #和\_)。没有嵌入的空格。

——在系统内部通讯功能 (ICF) 文件中，记录名不能以\$\$开头。

——限定字段名和限定名相似。例如：

KEYWORD(记录名/字段名)

各种高级语言在名字上有特定长度和限制值。请看相应的高级语言指南中有关部分。

ALIAS (替换字段) 名字：

——替换字段名的长度是 1 到 30 个字符。第一个字符必须是 A 到 Z。接下来字符 A 到 Z, 0 到 9，或者下划线。

——因为 DDS 不执行语言规定的句法检查，所以替换字段名字要遵循高级主语言的命名规则。当名字在程序中出现的时候，高级语言编译程序检查名字的句法。

信息标识：

——信息标识必须为 7 个字符长，前 3 个字符是信息前缀。

——信息前缀的第一个字符必是 A 到 Z, @, \$和#，接下来的字符必须是 A 到 Z, @, \$, #, \_ 和 0 到 9。

——后面 4 个字符必须是十六进制的值 (0 到 9, A 到 F)。

标号、文本和文件夹名：

——一个联机帮助信息的标号名必须 1 到 10 个字符长，而且必须由大写的 A 到 Z, @, #或\$开头。标号名不能包括逗号、单引号或空格。

——文本名（和一个简单的文件夹名）必是 1 到 8 个字符，后面可以跟着一个圆点和 1~3 个字符称为扩展符，常用字符是 A 到 Z, 0 到 9, @, #, \$和\_。

——如果一个文件夹名是连接起来的，每个文件夹之间要用斜线 (/) 分开。文件夹名的总长度不能超过 63 个字符。

——在 DDS 中一个文本、一个文件夹名或联机帮助信息标号名可以用引号括起来。当名字包括一个(或()、一个引号时，必须用引号括起，如果名字中有引号，要用两个引号编码。如果文件夹名是连接起来的，如果有括号，必须括起整个连接的名。

### 1.3 语义编码举例

图 1-2 到图 1-7 是语义编码的例子。除 HLPARA, JFILE, JFLD 和 PFILE 之外，在这些例子中所显示的其它键字都不是实际的键字。只是简单的指出应在什么地方写这些键字。

### PIC 3

图 1-2 物理文件的语法

- 1.注释（可选的）：注释可出现在 DDS 中的任意一行。它们是由在第 7 列上的“\*”来标识的。
- 2.文件层（可选的）：文件层出现在记录格式名（在 00040 行上的 RECORD）之前。
- 3.记录层（在物理文件中仅允许出现一个）：在第 17 列上给出 R 指明 RECORD 是一个记录格式名。记录层延展到第一个命名字段结束。
- 4.字段层（要求至少有一个字段名，除非在记录格式中规定 FORMAT 键字）：对于物理文件中的字段至少要指定一个名字和长度。其它的属性可以明确的指定或通过缺省指定。
- 5.键字段层（可选的）：在第 17 列指定 K，指出这个字段是键字字段（索引字段），对每个键字字段都要指定 K，在字段层之后通过重复某些字段名（这里是 FIELDA），指出是键字段层。

### PICTURE 4

图 1-3 简单逻辑文件的语法

- 1.注释（可选的）：注释可以出现在 DDS 的任意一行。它们是由在第 7 列上的一个“\*”来标识的。
- 2.文件层（可选的）：文件层关键字出现在记录格式名之前（在 00040 行上的 RECORD1）。
- 3.记录层（要求至少有一个）：在第 17 列的 R 指出 RECORD1 是记录格式名。在简单或多格式的逻辑文件中，对每一个记录格式都要求有 PFILE 键字。记录层直到第一个字段名出现时结束。
- 4.字段层：对于逻辑文件不要求一定指出字段名与字段属性。详细信息见第三章“逻辑文件”。
- 5.键字段层（可选的）：在第 17 列上指定 K，表示这个字段是键字字段，对每个键字段都要指定 K。在字段层规范之后通过重复一个或多个字段名（象 FIELDA 一样）来指定键字段层。
- 6.选择和省略层（可选的）：在第 17 列上指明 S 表示这个字段（象 FIELDDB）做为选择字段，在 17 列上的 O 指明这个字段是一个省略字段。选择和省略层在键字段层之后。  
注：要构成一个多格式的逻辑文件，在文件中要重复 3~6 项来包括更多的记录格式，或在 PFILE 键字上指定多个文件。

PICTURE 5

图 1-4 连接逻辑文件的语法

- 1.注释（可选）：注释可以出现在 DDS 中的任意一行。它们是由在第 7 列上给出“\*”来标识的。
- 2.文件层（可选）：文件层关键字出现在记录格式名（在 00040 行上的 REDORD1）之前。
- 3.记录层（只能是一个）：在第 17 列给出 R 指明 RECORD1 作为一个记录格式名。在连接逻辑文件中，其记录格式要有 JFILE 键字。记录层在遇到第一个连接规范时结束。
- 4.连接层：在连接层至少指定一个连接规范。第 17 列的 J 指出一个连接规范的开始。  
每一个连接规范至少要有一个 JFLD 键字。如果在文件中有多个连接规范，那么在一个逻辑文件中的每一个规范都必须有一个 JOIN 键字。一个连接规范在遇到下一个连接规范或一个字段名时结束。
- 5.字段层：对于连接逻辑文件要求最少有一个用法不是 N 的字段名。
- 6.键字段层（可选）：在第 17 列上指定 K，表示这个字段是键字字段，对每个键字段都要指定 K，在字段层规范之后通过重复一个或多个字段名（象 FIELDA 一样）来指定

键字段层。

7.选择和省略层（可选）：在第 17 上给 S 指明 FIELDB 做为一个选择字段，第 17 列上给出 O 指明字段作为省略字段，选择和省略层是跟在键字段层之后的。

PICTURE 6

图 1-5 显示文件的语法

- 1.注释（可选）：注释可以出现在 DDS 中的任意一行。它是在第 7 列上给出星号 “\*” 来标识的。
- 2.文件层（可选）：文件层键字出现在第一个记录格式名（在 00040 行的 RECORDA）之前。
- 3.记录层（要求最少有一个）：在第 17 列给出 R 指明 RECORDA 作为记录格式名。记录层在遇到一个字段名或第一个帮助规范时结束。
- 4.帮助层（可选）：第 17 列上给出 H 指明帮助规范的开始，一个帮助规范直到下一个出现在 17 列上的 H 或第一个字段时结束。每一个帮助规范最少要指明一个 HLPARA 键字和一个 HLPRCD 或 HLPDOC 键字。
- 5.字段层（可选）：在显示设备和程序之间传递的显示文件字段必须命名。并且必须指定一个长度。其它的属性可以通过显式的或缺省的指定。常数字段（没命名）要求有一个位置和键字，象在第四章显示文件中描述的 DATE, DFT, TIME 和 MSGCON 键字就是这样。第 17~38 列不能给常量字段用。

注：在显示文件中要指定一个新的记录格式可重复 3~5。

PICTURE 7

图 1-6 打印文件语法

1. 注释（可选）：注释可以出现在 DDS 中的任意一行。它们是通过在第 7 列上给出星号“\*”来标识的。
2. 文件层（可选）：文件层关键字出现在第一个记录格式名（在 00040 行上的 RECORDA）之前。
3. 记录层（要求最少有一个）：在第 17 列上给出 R 指明 RECORDA 作为一个记录格式名。记录层遇到第一个指定的字段时结束。
4. 字段层（在文件中的每一个记录格式中要求最少有一个字段，无论它是否命名）：从程序传递到打印机的打印文件字段必须是一个命名字段并且必须指定长度。其它的属性可以显式的或缺省的指定。常数仅要求位置和关键字。象在第五章“打印文件”中描述的 DATE, DFT, PAGNBR, TIME 和 MSGCON 键字就是这样。

注：在打印文件中可以通过重复 3 和 4 项来指定一个新的记录格式。

#### PICTURE 8

图 1-7 ICF 文件的语法

1. 注释（可选）：注释可以出现在 DDS 中的任意一行。它们是由在第 7 列上给出星号“\*”来标识的。
2. 文件层（可选）：文件层关键字出现在第一个记录格式名（在 00040 行上的 RECORDA）之前。
3. 记录层（要求最少一个）：第 17 列给 R 指明 RECORDA 是一个记录格式名。记录层直到第一个指定字段时结束。
4. 字段层（可选）：ICF 文件的字段至少要有一个名字（象 FIELDA）和长度。其它的属性可以显式的或由缺省的来指定。

注：在 ICF 文件中可以通过重复 3 和 4 项来指定一个新的记录格式。

## 第二章 物理文件和逻辑文件

这一章讲述有关物理文件和逻辑文件的下述信息：

- 定义
- 位置项
- 键字项

在 2.3 位置项（1-44 列）给出填写 DDS 表格 1-44 列的例子和规则。

在 2.4 键字项（45-80 列）给出规定 DDS 键字的例子和规则。

键字是以字母顺序描述的。

关于物理文件和逻辑文件位置项和键字选择的其它信息，见 DB2/400 数据库指南。

### 2.1 定义物理文件

一个物理文件只有一个记录格式。可以用以下两种方法来指定记录格式：

- 定义一个新的记录格式：对于一个新的记录格式要指定字段和键字段的规范
- 共享一个已存在的记录格式：用 FORMAT 键字来指定 OS/400 使用预先定义在一个物理的或逻辑文件中的记录格式。当使用 FORMAT 键字时，即使已有的记录格式已指定键字字段层规范，也必须再一次指定它（如果希望使用键字顺序访问路径）。

要定义一个物理文件，按下面的顺序指定这些项：

- 1.文件层项（可选）
- 2.记录层项
- 3.字段层项
- 4.键字段层项（可选）

注：文件名是通过建立物理文件（CRTPF）命令指定的而不是在 DDS 中指定的。

可以在第一章的“介绍”中找到文件层、记录层和键字层规范的解释。

可以在第一章的“语法规则”中找到关于指定 DDS 键字的语法规则。

一个记录格式中字段的最大数量是 8000。如果记录格式的某些字段是日期、时间、时间标记、变长或允许空值字段，那么实际上字段的最大数少于 8000。字段的最大个数取决于存在记录格式中字段及连接字段的个数。如果不包括变长字段，那么一个记录格式中字节数最多不超过 32766 个。如果包括变长字段，那么最多不超过 32740。图 2-17 给出了确定记录格式总长度的规则。

### 2.2 定义逻辑文件

在一个应用程序读操作时逻辑文件决定如何选择和定义数据记录。逻辑文件可以是单格式的、多格式的和连接逻辑文件。单格式的逻辑文件只有一个记录格式并且在 PFILE 键字上指定一个文件。多格式逻辑文件既可以含有多个记录格式亦可在 PFILE 键字上指定多个文件。连接逻辑文件有一个记录格式并有 JFILE 键字，可以最多指出 32 个文件。

### 2.2.1 单格式和多格式的逻辑文件

对单格式和多格式逻辑文件必须在记录层指定 PFILE 键字，在多格式逻辑文件中，在 PFILE 键字中指定的物理文件中有公共字段的记录格式才可以使用。

按下面的顺序指定这些项来定义一个单格式的或多格式的逻辑文件：

- 1.文件层项（可选）
- 2.记录层项
- 3.字段层项（可选）
- 4.键字段层项（可选）
- 5.选择和省略字段层项（可选）

文件中的每个记录格式要重复 2-5 项。

图 2-1 给出多格式逻辑文件编码的例子。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* LOGICAL FILE EXAMPLE
00020A* INVENTORY FORMAT
00030A      R INVFMT          PFILE(INVENTORY)
00040A      K ITEM
00050A*
00060A* ORDER FORMAT
00070A      R ORDFMT          PFILE(ORDER)
00080A              TEXT(' ORDER ANALYSIS')
00090A      ITEM
00100A      ORDER      10
00110A      SUPPLY      +2
00120A      SHPDAT          CONCAT(SHPMO SHPDA SHPYR)
00130A      QTY      5P      RENAME(QTYDUE)
00140A      K ITEM
00150A      K SHPYR
00160A      K SHPMO
00170A      K SHPDA
00180A      O QTYDUE          CMP(LT 1)
00190A*
00200A* ACCOUNTING FORMAT
00210A      R ACTFMT          PFILE(ACCOUNTS)
00220A              FORMAT(ACCOUNTL)
00230A      K ITEM
A
```

图 2-1 多格式逻辑文件的编码的例子

### 2.2.2 连接逻辑文件

连接逻辑文件是把多个物理文件中的数据字段连接到一个记录格式中。对连接逻辑文件必须在记录层指定 JFILE 键字。

要定义一个连接逻辑文件按下面的顺序指定这些项：

- 1.文件层项（可选）
- 2.记录层项
- 3.连接层项
- 4.字段层项（可选）
- 5.键字段层项（可选）
- 6.选择和省略层项（可选）

在一个连接逻辑文件中仅允许有一个记录格式，所以这些项只可以指定一次。下图给出一个逻辑文件编码的例子。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A* 从两个物理文件把字段合并到一个记录格式  
00020A      R RECORD1          JFILE(PF1 PF2)  
00030A      J                  JOIN(PF1 PF2)  
00040A          JFLD(NAME NAME)  
00050A      NAME             JREF(1)  
00060A      ADDR  
00070A      PHONE  
A
```

图 2-2 连接逻辑文件编码的例子

可以在第一章“介绍”中找到文件、记录、连接、字段、键字段和选择/省略字段层的解释。

可以在第一章的“语法规则”中找到指定 DDS 键字的语法规则。

关于使用单格式、多格式和连接逻辑文件的其它信息，参看 DB2/400 数据库程序设计指南。

### 2.2.3 在逻辑文件中指定记录格式

如果在一个逻辑文件中指定了多个记录格式，那么必须对每个记录格式指定 PFILE 键字。

有三种方法指定记录格式中的字段：

- 指定记录格式名和 PFILE 键字
- 指定记录格式名，PFILE 或 JFILE 键字，并且至少一个字段
- 指定记录格式名，PFILE 键字和 FORMAT 键字

图 2-1 给出了指定字段的这三种方法。

对于这三种方法，可以规定如下的访问路径：

- 不指定键字段（到达顺序访问路径）：除非指定了 DYNSLT 键字外，不能指定选择/

省略字段。对逻辑文件仅可以用在 PFILE 中的物理文件的一个记录格式。

- 指定一个或多个键字段（键字顺序访问路径）：如果在逻辑文件中指定了多个记录格式，那么每个记录格式至少要指定一个键字段。可以对这个文件中的一个记录格式指定选择/省略字段。
- 指定 REFACCPATH 键字（键字顺序访问路径），将另外的物理文件或逻辑文件的访问路径信息复制到你正在定义的文件中。

记录格式中的最大字段个数不超过 8000。如果记录格式中的某些字段是日期、时间、时间标志、变长或允许空值，那么实际上字段的最大长度小于 8000。实际上字段的最大个数依赖于存在于记录格式中的字段和联接字段的个数。如果变长字段不包括在内，则一个记录格式的最大字节数是 32766，否则是 32740。图 2-17 给出了确定记录格式总长度的规则。

### 2.3 位置项 (1-44 列)

这一节描述如何为逻辑文件和物理文件指定 DDS 表格的前 44 列。DDS 表格的剩余编码见键字项 (45-80 列)。

图 2-3 是物理文件的位置项举例。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* PHYSICAL FILE CODING EXAMPLE
00020A                               REF(INVENCTL/INVENTORY)
00030A                               UNIQUE
00040A      R ORDFMT                TEXT('Format for Purchase Orders')
00050A      ORDNBR     7 0          COLHDG('Order' 'Number')
00060A      ITMNBR     R 10
00070A      SUPNBR     R +2          REFFLD(SUPID SUPLIB/SUPMST)
00080A      QTYORD     5B
00090A      K ORDNBR
00100A      K ITMNBR               ABSVAL
A
```

图 2-3 物理文件编码例子

在图 2-1 和图 2-2 给出了多格式和连接逻辑文件的位置项。

#### 2.3.1 顺序号 (1-5 列)

使用这几列为 DDS 表格的每一行指定一个顺序号。顺序号是可选的并且仅用来编排文件。

#### 2.3.2 格式类型 (第 6 列)

在这列键入一个 A 以指明这是一个 DDS 表格，格式类型是可选的并且仅有编排文件的意义。

### 2.3.3 注释（第 7 列）

在第 7 列键入一个星号 “\*”，指明这行是一个注释行。8-80 列用于注释正文。一个空行（在 7-80 列没有字符）也做为注释行处理。注释可以出现在 DDS 的任意一行，并且仅保留在源文件中。它们是被打印在源文件清单上的，但不在扩展的源文件清单上打印。

### 2.3.4 条件（8-16 列）

物理或逻辑文件不用这几列。但可以用它们做注释的正文。

### 2.3.5 名字或规范的类型（第 17 列）

在这键入一个值用以指定名字的类型，对于逻辑文件用于指定规范的类型。如果指定了一个名字的类型，在 19-28 列指定名字。

对物理文件有效的项如下：

项	意义
R	记录格式
空格	字段名
K	键字段名

注：仅指定一个 R 是因为物理文件只能有一个记录格式。

对逻辑文件有效的项如下：

项	意义
R	记录格式
J	连接规范
空白	字段名或选择/省略的 AND 条件
K	键字段名
S	选择字段名
O	省略字段名

关于名字类型的其它信息，见 2.3.7 中“名字（9-28）”，关于连接规范的其它信息，见 2.4.28 中“JOIN”。

### 2.3.6 保留（第 18 列）

任何文件类型都不用这一位。除用它写注释正文外，保留这个位置为空白。

### 2.3.7 名字（19-28 列）

使用这几列指定下面的名字：

- 物理文件或逻辑文件的记录格式名
- 为每一个记录格式指定构成它的那些字段名（除非指定了 FORMAT 或在记录层指定了 PFILE 键字）
- 为每一个记录格式指定那些用来作索引的字段名

- 对于逻辑文件，指定那些用来做选择/省略规范的字段名

注：文件名是通过建立物理文件（CRTPF）命令来指定的而不是在 DDS 中指定的。

当要用 DDS 中指定记录或字段名时，参考 1.2 的语法规则中所述规则。

名字必须在第 19 列开始。

除了指定字段名或选择/省略的 AND 条件之外，必须在第 17 指定名字的类型。

图 2-3 描述了物理文件中如何编码名字

图 2-1 和图 2-3 描述如何在逻辑文件中编码名字。

#### 2.3.7.1 记录格式名

当在 17 列上指定 R 时，19-28 列指定的名字是一个记录格式名。

物理文件：物理文件仅允许有一个记录格式名。只能用下面两种方法之一指定记录格式名：

- 在物理文件中指定有字段名的新记录格式名，记录格式名可以做为建立物理文件（CRTPF）命令中的文件名。但如果名字不唯一将出现一个警告信息。这是因为一些高级语言不允许文件名和记录格式名相同。RPGIII就是这样一个高级语言。对系统而言记录格式名和字段名不必是唯一的，在不同的文件中可以出现相同的名字。
- 使用另一个已预先定义物理文件中的记录格式名，这时必须指定 FORMAT 键字，可以不指定字段名和属性。见本章的“格式（FORMAT）”中关于 FORMAT 键字的解释。
- 单格式或多格式逻辑文件：可以指定多个记录格式名，但在这个文件中每个名字必须是唯一的。对于这个解释见相应的高级语言手册。

用下面三种方法之一指定记录格式名：

- 用在 PFILE 键字上指定的第一个物理文件的记录格式名作为这个记录格式名，如果没有指定 FORMAT 键字并且没在这个记录格式中指明各个字段的名字，那么要求使用这个方法。
- 用逻辑文件中指定了字段名的一个新的记录格式名，每一个字段必须规定名字。没有命名的物理文件字段不能是逻辑文件记录格式中的一部分。做为 RENAME 和 CONCAT 键字参数的物理文件字段是逻辑文件记录格式的一部分。除了在别处指定之外，做为 SST 键字参数的物理文件字段不是逻辑文件记录格式的一部分。
- 用在物理和逻辑文件中预先定义了的记录格式名，不用指定字段名和属性，但是必须指定 FORMAT 键字。对于如何指定 FORMAT 键字的描述见本章的“FORMAT”。

记录格式名可以与在建立文件命令中指定的文件名字相同。然而，如果名字不唯一，将送出一个警告信息。象 RPGIII那样的一些高级语言处理程序不允许记录格式名与文件名相同。

使用记录格式名和 PFILE 键字来指定与记录格式相关的物理文件。一个记录格式可以在 PFILE 键字上指定多个物理文件。如果没有定义字段并且没有指定 FORMAT 键字，那么在 PFILE 键字中指定第一个文件的格式名用来作为所有物理文件的格式名（这个格式被用来作属性的引用、属性以及名字的检验）。

连接逻辑文件：仅可以指定一个记录格式名。指定一个记录格式名作为逻辑文件中指定了字段的一个新的记录格式名。对于连接逻辑文件，在记录格式中的每一个字段必在 19-28 位中给出名字。只有在这个记录格式之外的其地方指定了这些字段名，物理文件的字段作为有 RENAME、CONCAT 和 SST 键字的参数时，这些字段才能是逻辑文件记录格式的一部分。

JFILE 键字必须在记录层指定。它指定记录格式连接哪些物理文件。

### 2.3.7.2 字段名

当第 17 列是空白时，19-28 列指定的名字是一个字段名。如果指定了 FORMAT 键字，则不能指定字段名。

物理文件要求命名每个字段，并且在一个记录格式中这些名字必须是唯一的。这些字段以它们在 DDS 中指定的相同顺序出现在物理缓冲区中。

如果描述一个单格式的或多格式的逻辑文件，可以使用这个逻辑文件所依赖的物理文件中的已存在的记录格式，而不用指定字段名。

如果不使用已经存在的物理文件中的记录格式，那么必须在逻辑文件中命名每个字段。在单格式或多格式逻辑文件中，在记录格式中的每个字段名必须是唯一的并且必须与在物理文件记录格式中的一个字段相对应。字段名的顺序是使用逻辑文件的程序中字段出现的顺序。

在逻辑文件记录格式中出现的字段名通常是与在物理文件记录格式中相应的字段名相同。如果名字不同，那么这两个名字必须用 RENAME 键字使其相等。逻辑文件记录格式中的一个字段可以来自物理文件的两个或更多字段的连结。（见 CONCAT（连结）键字）。SST 键字可以用物理文件字段的子串作为逻辑文件的一个字段。

注：在逻辑文件中字段名顺序是重要的，如果在逻辑文件中一个记录格式中多次使用相同的物理文件字段，（使用 RENAME 或 CONCAT），那么字段在逻辑文件中顺序是数据传送到物理文件的顺序。那样，在逻辑文件中最后一次指定的字段值作为在物理记录中的值。

### 2.3.7.3 键字段名

当第 17 列给出 K 时，19-28 列指定的名字是一个键字段名。它必须是物理文件的记录格式中字段名中的一个。这个字段的内容做为从数据库中检索记录的顺序。指定键字段是可选的。如果没有指定键字段，缺省的顺序是到达顺序（即这个记录输入到数据库时的顺序）。见数据库指南中关于访问路径的信息和它们在保存/重存操作时的影响。

使用这个键字段（可选地，选择/省略字段）在逻辑文件成员的记录格式中定义一个键字顺序访问路径。逻辑文件成员包括在建立逻辑文件（CRTLF）或增加逻辑文件成员（ADDLMF）命令中参数 DTAMBRS 指定的物理文件成员。

通过指定顺序处理的键字，可以改变记录从文件中读出的顺序。这些顺序处理的键字是 ALTSEQ、NOALTSEQ、SIGNED、UNSIGNED、ABSVAL、ZONE、DIGIT、DESCEND、FIFO 和 LIFO。关于更多的信息参照这些键字的讨论。

当没有对键字段指定顺序时，它的缺省顺序是升序的。字符字段、十六进制字段、日期字段、时间字段和时间标记字段的缺省值是 UNSIGNED 属性。数字字段的缺省值是 SIGNED

属性，对区位十进制字段（35 位指定了 S），下面的情况是例外：

- 在文件层指定了 ALTSEQ，这个文件中的所有区位十进制键字段的缺省值为 UNSIGNED。
- 对一个区位十进制键字段指定了 DIGIT 或 ZONE 时，这个字段的缺省是 UNSIGNED。

如果对一个逻辑文件指定多个记录格式或 PFILE 键字指定了多个物理文件，那么必须在这个逻辑文件的所有记录格式至少指定一个键字段。

一个键访问路径可以有多个键字段，称为复合键。在复合键中要按重要性（主要到次要）的顺序指定键字段的名字，一个键字段的名字占一行。

图 2-4 给出了一个两个记录格式的多格式逻辑文件，其中一个使用了复合键。在这个例子中，RECORD1 有一个单一的键字段 FIELD1，RECORD2 有一个复合键，包含了 FIELD4 和 FIELD5。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1          PFILE (PF1)
00020A      FIELD1
00030A      FIELD2
00040A      FIELD3
00050A      K FIELD1
00060A*
00070A      R RECORD2          PFILE (PF2)
00080A      FIELD4
00090A      FIELD5
00100A      K FIELD4
00110A      K FIELD5
A
```

图 2-4 有两个记录格式的多格式逻辑文件

如果没有指定键字段，定义的逻辑文件用到达顺序访问路径。

组成一个键的字段个数限制于 120 个以内。总长度不超过 2000 字节。（如果指定 FCFO 键字，总长度不超过 1995 字节）。键的总长度包括了每个键字段的长度。如果键字段允许空值，那么每个字段加一个字节放空值。OS/400 使用这个额外的字节确认此键是否包含空值。如果键字段是变长的，那么每个变长的键字段加两个字节。OS/400 使用这两个额外字节来存放字段分配的长度。

在一个逻辑文件中指定多个记录格式时，则需要对第一个\*NONE 键字位置附加一个字节。对每个附加键字段位置也需一个附加字节。OS/400 使用这些附加字节，以使记录区别于重复键值的物理文件。

例如：假定一个键由字段名 FIELD A、FIELD B 和 FIELD C（以这个顺序）构成的，如图 2-5 那样给出 DDS：

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* SAMPLE COMPOSITE KEY (PHYSICAL FILE)
00020A      R RECORD
00030A      FIELD A      3 0
00040A      FIELD B      3 0
00050A      FIELD C      3 0
00060A      FIELD D      3 0
00070A      K FIELD A
00080A      K FIELD B
00090A      K FIELD C
A

```

图 2-5 复合键

注：00070 到 00090 行组成复合键。

记录是按下面的顺序排序的：

- 根据 FIELD A 的内容来排序
- 如果多个记录在 FIELD A 中具有相同值，OS/400 按 FIELD B 中的值来排序
- 如果某些记录在 FIELD A 和 FIELD B 中的值都相同，则按 FIELD C 的值排序。

考虑下面文件：

Record	FIELD A	FIELD B	FIELD C
1	333	99	67
2	444	10	45
3	222	34	23
4	222	12	01
5	222	23	45
6	111	06	89
7	222	23	67

假定所有的字段是升序的，以下面的顺序检索记录：

Record	FIELD A	FIELD B	FIELD C
6	111	06	89
4	222	12	01
5	222	23	45
7	222	23	67
3	222	34	23
1	333	99	67
2	444	10	45

- 由于记录 3、4、5 和 7 在 FIELD A 字段中有相同的内容，所以 FIELDB 成为决定字段。
- 在这四个记录中，5 和 7 在 FIELDB 中的值是相同的，所以 FIELDC 成为一个决定字段。
- 如果在 FIELDC 中也含有重复的值，那么这些记录是按着先进先出（FIFO）或后进先出（LIFO）的顺序来检索。要保证这个顺序，要指定 FIFO 或 LIFO 键字。指定 UNIQUE 键字可以防止重复键值的出现。

关于一个含有负（-）内容的键字段的例子，见 2.4.39。

当在建立物理文件（CRTPF）命令或建立源物理文件（CRTSRCPF）命令中使用了 FILETYPE(\*SRC) 时，键字段有特殊的限制。关于在源文件中键字段规范的信息见 DB2/400 数据库程序设计。

对于逻辑文件，下列规则用于键字段：

- 对于单格式和多格式逻辑文件，同定义的字段相匹配的键字段名使用下面的检索顺序：
  1. 在 DDS19-28 列上指定的字段。
  2. 在 CONCAT 或 RENAME 键字上作为参数指定的字段。
- 如果多次的指定了一个字段名，使用第一次出现的。
- CONCAT 或 RENAME 键字中的字段名与 19-28 列中的相关字段不能同时作为键字段。
- SST 键字上的参数名，除非在另一个地方用逻辑文件格式定义了之外，不能作为键字段。
- 对于连接逻辑文件，指定的键字段名必须是主文件（在 JFILE 键字上指定的第一个物理文件）中的一个字段，并且必须在字段层的 19-28 列中指定。

注：如果指定了一个字段作为 CONCAT、RENAME 或 SST 键字的参数值，而没在连接逻辑文件的 19-28 列指定这个字段，就不能指定这个字段作键字段。

如果正在连接字符和十六进制数字字段，不能指定数字字段作为键字段。如果连接区位十进制和任一其它类型的数字字段，就不能指定其它数据类型的字段作为键字段。

图 2-6 给出哪个连结字段可用/不可用来作为键字段。

00010A	R RECORD1	PFILE(PF1)
00020A	FLD1	
00030A	FLD2	
00040A	Z	CONCAT(ZFLD PFLD)
00050A	A	CONCAT(AFLD NFLD)
00060A	K ZFLD	
00070A	K AFLD	
	A	

图 2-6 正确的不正确的连接字段

在物理文件 PF1 中, ZFLD 是区位十进制而 PFLD 是压缩十进制。因此 Z 是区位十进制, PFLD 不能作键字段。ZFLD 和 Z 都可以用来作为键字段, 但不能在同一个记录格式中。

在物理文件 PF1 中, AFLD 是一个字符字段而 NFLD 是一个数字段。因此 A 是字符型, 并且 NFLD 不能被作键字段。AFLD 和 A 可作键字段, 但不能在同一记录格式中。

访问路径关键字: 可以指定一个或多个访问路径关键字对 OS/400 建立和使用键值的方法产生影响。这些访问路径的关键字是:

文件层	键字段层
ALTSEQ	DESCEND
FCFO	DIGIT
FIFO	SIGNED
LIFO	UNSIGNED
REFACCPTH	ZONE
UNIQUE	

在复合键中的不同键字段可能有不同的访问路径关键字。

有多个记录格式的逻辑文件: 在一个逻辑文件中指定了多个记录格式时, 必须为每一个记录格式至少指定一个键字段。在每个关键字中不要求相同的键字段数, 在一个记录格式中指定的键字段必须与在同一逻辑文件中其它记录格式中的相应的键字段有相同属性和访问路径关键字。一个变长的键字段后面不允许跟一个定长的键字段, 即使字段的类型和长度是相同的。

每个记录格式要求有一个键, 这样逻辑文件成员对每个记录格式的有序记录都有单一的访问路径。当记录从逻辑文件依据的物理文件的不同成员中返回时, 它们是根据逻辑文件成员访问路径的关键字值来合并的。

当逻辑文件成员的记录被排序时, OS/400 为每个记录用键字段值的连接建立一个键值。然后用这个键值建立程序使用的访问路径。关于由 OS/400 允许的 I/O 操作的信息见数据库程序设计。

复合键中的每一个键字段有一个键位置。第一个键字段是在位置 1, 第二个键字段在位置 2, 以此类推。在对逻辑文件进行 I/O 操作时, OS/400 比较从数据库读出或写入的记录键值。当建立多记录格式 (有或没有不同键字段) 的逻辑文件时, OS/400 检查键位置属性。要使键位置属性检验成功, 在同一键位置上的不同记录格式的键字段必段具有相同的数据类型, 长度, 小数点位置和在键字段层指定的访问路径关键字。这能保证在 I/O 操作期间产生有意义的记录排序。

用做关键字的浮点字段必须有相同的数据类型和精度, 但不要求有相同的长度和小数点位置。

在图 2-7 中, FIELD1、FLD1 和 F1 必须具有相同的属性, 而且 FIELD2、FLD2、F2 也必须具有相同的属性; FIELD1、FLD1、F1 是在键位置 1 中, FIELD2、FLD2、F2 是在键位置 2 中。一个记录格式可以比另一个记录格式有更多的键字段, 这些附加字段不需要做键位置属性检验。FLD3 就是一个这样的字段。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1                  PFILE (PF1)
00020A      FIELD1       5  0
00030A      FIELD2       10
00040A      FIELD3       10
00050A      K FIELD1
00060A      K FIELD2                  DESCEND
00070A*
00080A      R RECORD2                  PFILE (PF2)
00090A      FLD1        5  0
00100A      FLD2        10
00110A      FLD3        20
00120A      K FLD1
00130A      K FLD2                  DESCEND
00140A      K FLD3
A*
A      R RECORD3                  PFILE (PF3)
A      F1        5  0
A      F2        10
A      F3        30
A      K F1
A      K F2                  DESCEND
A
A

```

图 2-7 键字段属性检查

关于在逻辑文件有多个记录格式的键字段的例子见图 2-1。在图 2-1 中，每一个键中都有 ITEM 字段。对记录格式 INVFMT 和 ACTFMT，指定 ITEM 为唯一键字段。记录格式 ORDFMT 指定了复合键。在这个复合键中包含了 ITEM、SHPYR、SHPMO 和 SHPDA。用在键字中的每一个字段必须也在字段层说明。因此，ITEM 必须存在于物理文件 INVENTORY 的记录格式中以便可以把它复制到逻辑文件 INVFMT 中。同样，ITEM 必须存在于逻辑文件 ACCOUNTL 的记录格式中以便它能够被复制到逻辑文件 ACTFMT 中。ITEM 还必须存在于物理文件 ACCOUNTS 中。

在键字段使用\*NONE：下面两个条件在具有相同键位置的键字段不能比较时发生。这两个条件是：

- 这两个键字段不具有相同的属性（数据类型、长度、小数位或在字段层的访问路径关键字）。
- 这两个键字段具有相同的属性，但是你不想合并他们并一起排序。

要避免两个键字段之间不想发生的比较，在其中一个指定\*NONE 并且把替代键字段移到下一个键位置。OS/400 比较在\*NONE 之前和之后的键位置值，而按逻辑文件 DDS 中指

定的记录格式的顺序检索有效的记录。

你可以两次或更多次的在下一行指定\*NONE 来将一个键字段移位到与你的应用程序相关的键字段属性比较的键位置上。

图 2-8 显示了\*NONE 作为一个键字段。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A       R RECORD1          PFILE(PF1)  
A       FIELD1      6A  
A       FIELD2      4A  
A       FIELD3     10A  
A       K FIELD1  
A       K FIELD2  
A       K FIELD3  
A  
A       R RECORD2          PFILE(PF2)  
A       FLD1      3A  
A       FLD2      4A  
A       FLD3     12A  
A       K *NONE  
A       K FLD2  
A  
A       R RECORD3          PFILE(PF3)  
A       F1      6A  
A       F2      4A  
A       F3     10A  
A       K F1  
A       K F2  
A       K F3  
A
```

图 2-8 规定\*NONE 做键字段

在图 2-8 中，FIELD2、FLD2 和 F2 的属性必须一致。由于对第二个记录的第一个键字段指定了\*NONE，FIELD1 和 F1（第一记录的第一键字段和第三记录的第一键字段）必须有一致的属性。FIELD3 和 F3 也必须有一致的属性，在第二个记录格式中没有相对应的字段。

图 2-9 到图 2-12 解释了如何使用键字段。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A       R CLSHST          PFILE(CLSHSTP)  
00020A       K EMPNBR    1  
00030A       K CLSDTE   2  
00040A*
```

00050A	R JOBHST	PFILE(JOBHSTP)
00060A	K EMPNBR	1
00070A	K JOBDTE	2
	A	

图 2-9 规定键字段 (例 1)

记录格式	键位值 1	键位置 2
CLSHST	EMPNBR	CLSDTE
JOBHST	EMPNBR	JOBDTE

例 1: 在图 2-9 中, 一个逻辑文件通过两个不同的记录格式访问两个物理文件: CLSHST (课程日记) 和 JOBHST (作业日记) 的记录。在这个逻辑文件中, 来自两个物理文件的记录合并在一起并且可以通过在键位置 1 指定 EMPNBR 排序。

所有的 EMPNBR 键值相同的记录都属于同一雇员, 为了把一个给定雇员的所有记录合并且排序到各自的课程日记和作业日记中, 要对两个记录格式在键位置 2 指定 CLSDTE (课程日期) 和 JOBDTE (作业日期), 如图 2-9 所示。

假定作业分配日期和课程日期 (月/年) 是分配和课程的开始日期, 3 个学生的记录按下面的顺序检索出来:

EMPNBR	CLSDTE	JOBDTE	Description
1005	3/79		Completed class
1005	4/79		Left to begin new job
1005		4/79	Completed job
1005	6/79		Completed class
1006		1/79	Completed job
1006		2/79	Completed job
1006	3/79		Completed class
1006	5/79		Transferred to new location
1007		1/79	Completed job
1007		4/79	Completed job
1007		7/79	Completed job
1007	8/79		Left because of

上面报表提供每个学生连续的历史记录

例 2: 在图 2-10 中, 另一个逻辑文件使用例 1 所示的相同二个物理文件, 但逻辑文件的第 2 个记录格式在键位置 2 中规定了\*NONE。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A R CLSHST PFILE(CLSHSTP)
00020A K EMPNBR 1

00030A	K CLSDTE	2
00040A*		
00050A*		
00060A	R JOBHST	PFILE(JOBHSTP)
00070A	K EMPNBR	1
00080A	K *NONE	2
00090A	K JOBDTE	3
A		

图 2-10 规定键字段 (例 2)

记录格式	键位置 1	键位置 2	键位置 3
CLSHST	EMPNBR	CLSPTE	*NONE
JOBHST	EMPNBR	*NINE	JOBDTE

象图 2-9 一样，两个物理文件的所有记录首先是按雇员号 (EMPNBR)，合并到一起并排序的。但每个学生的记录首先按 CLSDTE 排序，然后再按作业 JOBDTE 来排序。在图 2-9 中使用的一组记录象下面那样检索出：

EMPNBR	CLSDTE	JOBDTE	Description
1005	3/79		Completed class
1005	4/79		Left to begin new job
1005	6/79		Completed class
1005		4/79	Completed job
1006	3/79		Completed class
1006	5/79		Transferred to new location
		1/79	Completed job
1006		2/79	Completed job
1007	8/79		Left because of illness
		1/79	Completed job
1007		4/79	Completed job
1007		7/79	Completed job

当几个相邻的记录格式在同一键位置 有\*NONE 时，在它们之前和之后相关的记录格式形成一组，这个功能象一个独立的记录格式在排序一样。在\*NONE 之后指定的键字段起到在组内格式的合并和排序记录的作用。

例 3：在图 2-11，考虑一个基于五个物理文件的雇员逻辑文件。

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

00010A	R EMPMST	PFILE (EMPMSTP)
00020A	K EMPNBR	1
00030A*		
00040A	R CLSREG	PFILE (CLSREGP)
00050A	K EMPNBR	1
00060A	K CLSDTE	2
00070A*		
00080A	R CLSHST	PFILE (CLSHSTP)
00090A	K EMPNBR	1
00100A	K CLSDTE	2
00110A*		
00120A	R JOBHST	PFILE (JOBHSTP)
00130A	K EMPNBR	1
00140A	K *NONE	2
00150A	K JOBDTE	3
00160A*		
00170A	R ACTHST	PFILE (ACTHSTP)
00180A	K EMPNBR	1
00190A	K *NONE	2
00200A	K ACTDTE	3 A

图 2-11 规定键字段（例 3）

记录格式	键位置 1	键位置 2	键位置 3
EMPMST	EMPNBR	*NONE	*NONE
CLSREG	EMPNBR	CLSDTE	*NONE
CLSHST	EMPNBR	CLSDTE	*NONE
JOBHST	EMPNBR	*NONE	JOBDTE
ACTHST	EMPNBR	*NONE	ACTDTE

这些记录是按下面那样来合并和排序:

- 1.所有的记录按雇员号合并且排序。
- 2.对于给定的雇员号，记录是按下面的情况来排序的：
  - a.主记录（EMPMST 格式）
  - b.CLSREG 和 CLSHST 格式的记录，按 CLSDTE（键位置 2）的值合并和排序。
  - c.JOBHST 和 ACTHST 格式的记录，按 JOBDTE 和 ACTDTE（键位置 3）合并和排序。

在键定义时指定\*NONE 可以实现如下的排序:

- \*NONE 和字段名、CLSDTE 出现在相邻格式 CLSHST 和 JOBHST 的第二个键位置上，这样有效地分开了在前边的键位置（位置 1）之后的两个格式。上面分开格式的记录是以在 EMPNBR 的值下分开的记录来合并和排序的。
- 在格式 EMPMST 的第二键位上隐含的\*NONE 强制做相似的分开。

- 在键位置 2 有 \*NONE，使 JOBHST 和 ACTHST 格式形成一组并且仅在合并和排序这两个格式记录时才比较 JOBDTE 和 ACTDTE 的值。

由前面键字定义的记录顺序完全依赖于格式指定的顺序。例如，如果在 CLSHST 之前指定了 JOBHST，那么键位置 2 将读到：

\*NONE、CLSDTE、\*NONE、CLSDTE、\*NONE

这里，CLSRG 中的 USDTE 的值将不用在 CSHST 中的 CLSDTE 值排序，JOBDTE 也将不用 ACTDTE 排序。

**例 4：**在图 2-12 中，假定一个雇员有重复的课程，要用相同的 EMPNBR 和 CLSDTE 值来排序这两个记录，在记录格式 CSHST 中指定第 3 个键字段 DATE。但由于 JOBDTE 和 ACTDTE 出现在其它格式的下一个键位置上，所以 DATE 不能出现在下一个可用的键位置（位置 3）上。如果在这个位置上指定 DATE，那 DATE 的属性将与 CSHST 和 JOBHST 的属性做比较，这个键的定义就不能用。

要得到需要的排序，在 DATA 之前指定 \*NONE，代替键字 DATE 到键位置 4 上。

像图 2-12 中那样，DATE 字段可以出现在位置 4 上。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R EMPMST          PFILE (EMPMSTP)
00020A      K EMPNBR
00030A*
00040A      R CLSRG          PFILE (CLSRGP)
00050A      K EMPNBR
00060A      K CLSDTE
00070A*
00080A      R CSHST          PFILE (CSHSTP)
00090A      K EMPNBR
00100A      K CLSDTE
00110A      K *NONE   1
00120A      K DATE    1
00130A*
00140A      R JOBHST          PFILE (JOBHSTP)
00150A      K EMPNBR
00160A      K *NONE
00170A      K JOBDTE
00180A*
00190A      R ACTHST          PFILE (ACTHSTP)
00200A      K EMPNBR
00210A      K *NONE
00220A      K ACTDTE
A
```

图 2-12 规定键字段（例 4）

Record Format	Key Positions			
	1	2	3	4
EMPMST	EMPNBR	*NONE	*NONE	*NONE
CLSREG	EMPNBR	CLSDTE	*NONE	*NONE
CLSHST	EMPNBR	CLSDTE	*NONE	DATE
JOBHST	EMPNBR	*NONE	JOBDTE	*NONE
ACTHST	EMPNBR	*NONE	ACTDTE	*NONE

在键位置 4 指定 DATE 能够使来自物理文件 CLSHST 中带有一致值的 EMPNBR 和 CLSDTE 记录根据 DATE 的值来合并和排序。详细信息请看数据库程序设计。

注：在前面例子中，键字中实际值是为了保证顺序，但在多格式逻辑文件中，如规定了 \*NONE，重复键值的结果是不可预测的。

#### 2.3.7.4 选择/省略字段名

使用选择/省略字段告诉 OS/400，当应用程序使用这个记录格式检索记录时如何选择或省略记录。受影响的只有那些在这个记录格式中用 PFILE 或 JFILE 键字指定的物理文件的记录。

下面的规则适用于逻辑文件中的选择/省略字段：

- 仅当对这个文件也指定了键字段或 DYNSLT 键字时，才可以指定选择/省略字段。
- 当应用需要非键字段时，也可以指定 \*NONE 做为键字段来满足其对一个键字段的要求。
- 对于单格式的和多格式逻辑文件，OS/400 使用下面的检索顺序用已定义的字段来匹配选择/省略字段：

— 在 DDS19-28 位指定的字段

— 作为 CONCAT 或 RENAME 键字参数指定的字段

如果多次指定一个字段名，那么使用第一次出现的。

在 CONCAT 或 RENAME 键字上的字段名和在 19-28 列上出现的相关联的字段名，不能都指定作为选择/省略字段。

在 SST 键字上的参数名作为选择/省略字段是无效的，除非它在这个逻辑文件记录格式中的其它地方已被定义过。

对于连接逻辑文件，指定的选择/省略字段名必须在字段层的 19-28 列上指定。

当使用选择/省略字段时，在 17 列指定 S 或 O，隐含了选择/省略语句是“或”的关系。系统处理“或”关系的选择、省略语句是互相独立的。也就是如果遇到选择或省略的情况，记录被选择或被省略。如果条件不适合，系统执行下一条语句。

在第 17 列指定一个空格，隐含了选择/省略语句是“与”关系。必须用组合关系来选择或省略记录。见图 2-13 和 2-14。在 19-28 列上指定一个字段名，其内容在处理时根据对这个字段指定选择/省略键字来决定这个记录或选择或省略。选择/省略键字是 COMP、RANGE 和 VALUES。最后一个选择/省略规定可以用 ALL 键字来做，但是不允出现字段名。

字段必须在物理文件和逻辑文件记录格式中都出现。一个记录格式中，选择/省略语句必须在所有的字段和键字段层项之后。对同一记录格式可以同时指定选择/省略语句。它们提供下面的信息：

- 如果对同一个记录同时指定了选择和省略语句，那么指定的顺序是很重要的。选择/省略语句是按着它们被指定的顺序来处理的；如果一个记录满足了一个语句，那么这个记录象指定的那样被选择或省略，并且对这个记录保持这个选择/省略语句不再处理。见图 2-15。
- 如果同时指定了选择和省略语句，那么可以规定不满足条件的记录如何处理。详细信息见本章后面的“ALL”。
- 如果没指定 ALL 键字，那么对于不满足选择/省略值的记录的处理做最后一个语句指定的类型的相反动作。即：不符合选择的值的记录被省略，不符合省略的值的记录被选择。

在一个简单的逻辑文件中指定的选择/省略语句的个数是有限的。如果指定了许多选择/省略字段就不能生成文件，可在规范中按下面给出的内容来修改，这样来减少文件的开销：

- 减少在这个文件中记录格式的数量。
- 减少在 PFILE 或 JFILE 键字上指定的物理文件的数量。
- 减少在选择/省略规范中所使用的字段（单一出现的）的数量。

不能用浮点字段来作为选择/省略字段。

可用到达顺序处理有选择/省略处理的访问路径。例：CPYF 可以指定 FROMRCD(1)，高级语言可以不用键字处理，在这种情况下，就象指定了 DYNSLT 键字一样来处理。

图 2-13 显示了如何使用“或”条件的选择语句指定选择/省略字段。

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A            R RECORD1                                PFILE (PARTS)
00020A            PNO
00030A            DSC
00040A            UPR
00050A            QOH
00060A            K PNO
00070A            S UPR                                    COMP (GT 5.00)
00080A            QOH                                    COMP (LT 10)
00090A            O                                        ALL

A

图 2-13 规定选择/省略字段（例 1）

上图中，仅当记录满足两个选择条件时，才选择它：第一个语句用来选择那些 UPR 字段的值大于 5.00 的记录，第二个语句用来选择那些 QOH 字段小于 10 的记录。对于字段 QOH 来说，在第 17 列没有 S，因此，这两个选择语句是“与”的关系。因此程序读出的记录两

个指定的条件一定都为真。

图 2-14 显示了如何用一个省略语句与两个“与”关系的选择语句“或”在一起来指定选择/省略字段。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1          PFILE(PARTS)  
00020A      PNO  
00030A      DSC  
00040A      UPR  
00050A      QOH  
00060A      K PNO  
00070A      O DSC          COMP(EQ 'HAMMER')  
00080A      S UPR          COMP(GT 5.00)  
00090A      QOH          COMP(LT 10)  
00100A      O          ALL  
A
```

图 2-14 规定选择/省略字段（例 2）

在上图中，如果一个记录通过了下面的检验，才能提供给程序：

- DSC 字段的内容不等于 HAMMER
- UPR 字段大于 5.00 且 QOH 字段小于 10

图 2-15 显示了规定同样的选择/省略逻辑的几种方法。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      S ST          COMP(EQ 'NY')  
00020A      REP          COMP(EQ 'JSMITH') 1  
00030A      YEAR          COMP(LT 78)  
00040A      O          ALL  
A  
00050A      O YEAR          COMP(GE 78)  
00060A      S ST          COMP(EQ 'NY') 2  
00070A      REP          COMP(EQ 'JSMITH')  
00080A      O          ALL  
A  
00090A      O REP          COMP(NE 'JSMITH')  
00100A      O ST          COMP(NE 'NY') 3  
00110A      S YEAR          COMP(LT 78)  
  
00120A      O          ALL  
A
```

图 2-15 规定选择/省略字段（例 3）

在上图中,要选择在 1978 年以前在纽约洲的名字叫 JSIMITH 的推销代理人的所有记录。可以有下面的三种方法来编码选择/省略字段:

- 1.在记录被选择/省略之前,所有的记录必须与选择字段 ST、RFP 和 YEAR 进行比较。
- 2.第一次进行比较后省略所有的 1978 年之后的记录,然后,只将 1978 年以前的记录与 ST 和 REP 进行比较,这样,仅有两个选择的字段需要做比较。显然这个方法比方法 1 效率更高。
- 3.前两次比较省略所有在 Newyork 洲的不是 JSIMITH 的记录。然后只将剩下的记录与 YEAR 比较。这个方法比方法 1 或 2 的效率都高。

### 2.3.8 引用 (第 29 列)

对于逻辑文件,第 29 列为空格。对于所有指定字段,所有逻辑文件都自动提供引用能力。逻辑文件的没有明确指定的属性都可以由物理文件记录格式的相应字段提供。

对物理文件,这列给出 R 表示要引用一个预先定义的命名字段(称为被引用字段)的属性,且必须指定 REF 或 REFFLD 键字。被引用字段可在预先定义的物理文件中也可以在预先建立的数据库文件中,被引用字段的属性是字段的长度、数据类型和小数位置,以及 ALIAS、COLHDG、DATFMT、DATSEP、FLTPCN、REFSHIFT、TEXT、TIMFMT、TIMSEP、VARLEN、编辑和有效检查键字的属性。

如果这位没有指定 R,那么必须规定字段属性。

注:如果一个被引用字段的 DATFMT 键字为\*ISO、\*EUR、\*USA 或\*JIS,则不能引用 DATFMT 键字属性。

在文件层和记录层的第 29 列必须是空白。

如果引用同一文件中的字段,那么字段不能同名,如果同名那么要指定被引用字段的文件名作为 REF 或 REFFLD 键字的参数。如果不同名,则需用 REFFLD 键字指定被引用字段的名字。详细内容请见键字 REF 和 REFFLD 的解释及附录 A 的有关内容。

要替换被引用字段的一些属性,可以对正在定义的字段指定那些属性。另外:

- 如果对这个字段指定了 EDTCDE(编辑码)或 EDTWRD(编辑字),则不要从被引用字段中复制。
- 如果在这个字段上指定了 CHECK(AB、ME、MF、M10、M10F、M11、M11F、VN 或 VNE)CHKMSGID、COMP、RANGE 或 VALUES,也不要从被引用字段复制有效检验规范。
- 如果对字段指定了数据类型、长度、小数位,则即不复制被引用字段的编辑也不复制有效检验键字。

注:在一个物理文件建立之后,被引用的文件可以被删除或修改而不影响物理文件中字段的描述。如要把引用文件做的修改包括进去,则先删除后重建这个物理文件。

### 2.3.9 长度 (30-34 列)

对物理文件,用这些位置为每个命名的字段指定字段长度(除非从被引用字段复制它

们)。对于数字类型的字段指定数位数，对一个字符类型的字段指定字符数。

对逻辑文件，使用这些位置指定逻辑文件长度。仅当逻辑文件要改变引用的物理文件中相应字段长度或欲使其无效时才指定。如果这几列是空白，那么逻辑文件定义的字段与引用的物理文件中的相应字段的长度是相同的。如果物理文件的字段是变长的并有空位，那么引用它的逻辑文件中字段也是变长的。如果指定一个长度，在逻辑文件中的字段是定长的，除非也指定 VARLEN 键字。另外，SST 键字也可用来控制逻辑文件字段的长度，可以规定一个字段是另外字段的某一部分字符串。详情请见 SST 键字的说明。

如果指定长度，那么必须右对齐，前置 0 是可选的。

图 2-16 显示了物理文件字段长度的正确和错误的编码。

....+....1....+....2....+....3....+....4....+....5
00010A           FIELD1           7
A
00020A           FIELD2           7
A
00030A           FIELD3        R    +7
A

图 2-16 物理文件字段长度的编码

注：FIELD1 的长度规定是不正确的，FIELD2 和 FIELD3 是正确的。

有效的长度规范是：

数据类型	有效长度
字符型	1-32766 个字符
十六进制	1-32766 个字节
二进制	1-9 位数字
区位十进制	1-31 位数字
压缩十进制	1-31 位数字
浮点单精度	1-9 位数字
浮点双精度	1-17 位数字
日期	6, 8 或 10 个字符
时间	8 个字符
时间标记	26 个字符

数据类型为 L (日期)，T (时间) 或 Z (时间标记) 的字段长度是由系统决定的。不要在 30-34 列规定其字段长度。

日期和时间字段长度包括分隔符。

时间标记有下列固定格式：

YYYY-MM-DD-hh.mm.ss.uuuuuu

单精度数据最多为 9 位数，而双精度是 17 位数。OS/400 程序支持的浮点精度是：单精度 7 位，双精度 15 位。

在一个记录中所有字段所占有的字节总数一定不能超过 32766 字节，关于记录格式中决定总长度的有关规则见图 2-17。在实际存贮中占用的字节数是由系统按照下面的情况决定的：

数据类型	占有的存贮字节数
字符	字符的个数
十六进制	字节的个数
二进制 1-4 位	2 个字节
5-9 位	4 个字节
区位十进制	数字的个数
压缩十进制	(数字的个数/2) +1 (取整)
单精度浮点	4 个字节
双精度浮点	8 个字节
日期	没有 DATFMT 为 10 个字符，若有，为 6, 8, 10 个字符
时间	8 个字符
时间标记	26 个字符

注：系统执行算术操作时，使用压缩十进制数据比用区位十进制的效率高。

图 2-17 描述了确定总长度的规则。

情 况	动 作
记录格式中包括变长字段？	1. 在总长度中加上 24 个字节。 2. 对每个变量字段加 2 个字节长度。
记录格式中有允许空值的字段？	字段的总数除以 8，取整为下一个最高整字节， 加上格式长度。

图 2-17 确定总长度的规则

为了替换物理文件的被引用字段或逻辑文件的字段长度，指定一个新值或修改长度，要增长度，指定+n，n 是增量，要减少长度，指定-n，n 是减量。例如+4，表示逻辑文件中的这个数字字段比在被引用的物理文件中的这个字段长 4 位数字。图 2-3 例子给出如何使物理文件字段长度。图 2-1 给出如何修改和替换逻辑文件字段长度。

如果物理文件记录格式中的字段是有小数位的二进制数据，引用它的逻辑文件中就不能替换其长度。如果是用引用的物理文件记录格式中某些字段连结而成的字段，那么，在逻辑文件中就不能规定长度，物理文件字段长度总和是由系统计算的。

如果在 30-34 列指定了一个值，程序接收这个长度，但在物理文件中相应的字段的长度

不变。这可能会引起数据转换错误。当试图向文件加成员或打开成员时，OS/400 可能发送一个文件映象错误信息。在下面的情况下，OS/400 也可送一个映象的错误信息：

- 从一个减少了物理文件中字段长度的逻辑文件中读时；
- 当向一个增加了物理文件字段长度的逻辑文件写时。

例如：如果物理文件字段定义为 4 个字符长而逻辑文件字段比其少 2 个字符，那么程序读不到物理文件中的 ABCD 值，而仅能读到 AB，在这种情况下，程序的写总是可以成功的。对于字符字段数据左对齐，用空白填满物理文件剩余的字段。对于数字字段，数据右对齐，用 0 填满剩余物理文件字段。

30-34 列仅对字段层有效。在键字段，选择/省略字段，连接层、记录层和文件层这几位必须保留空白。

注：高级语言可以限制字段长。应该遵守由高级语言使用的长度限制。

#### 2.3.10 数据类型（第 35 列）

对于物理文件来说，使用这一列来指定在数据库中字段的数据类型。

在逻辑文件中指定数据类型只是为了改变或替换逻辑文件引用的物理文件中相应字段的类型。如果这一位为空白，那么定义的字段有一个与逻辑文件引用的物理文件中相应的字段相同的属性。

有效的数据类型项如下：

项	含义
P	压缩十进制
S	区位十进制
B	二进制
F	浮点数
A	字符
H	十六进制
L	日期
T	时间
Z	时间标记

图 2-3 和图 2-1 显示了数据类型是如何编码的。

对于物理文件，如果没指定数据类型或没从被引用字段复制数据类型，OS/400 分配下面的缺省值：

- A（字符）：如果小数位（36 和 37 位）是空白时；
- P（压缩十进制）：如果小数位（36 和 37 位）有一个范围在 0-31 间的数时。

注：

1. 在 37 列给出 0，表示这是一个压缩十进制，区位十进制或二进制的整数字段。
2. 在 35 列给出 F，表示这是一个单精度浮点数的字段。使用 FLTPCN 键字来指定一个双精度字段或改变一个已经指定的浮点字段的精度。
3. 在第 35 列指定一个 H（十六进制），表示一个字段的内容是不允许被系统中断的。在多数情况下，十六进制字段以字符字段处理，除非一个十六进制字段的内容不能解释为任何

字符集或编码页。

下表给出在物理文件和逻辑文件之间有效的数据转换类型：

物理文件 数据类型	逻辑文件数据类型								
	字符	十六进制	区位十进制	压缩十进制	二进制	浮点	日期	时间	时间标记
字符	有效	有效	看注 1	无效	无效	无效	无效	无效	无效
十六进制	有效	有效	看注 1	无效	无效	无效	无效	无效	无效
区位十进制	看注 1	看注 1	有效	有效	看注 2	有效	无效	无效	无效
压缩十进制	无效	无效	有效	有效	看注 2	有效	无效	无效	无效
二进制	无效	无效	看注 2	看注 2	看注 3	看注 2	无效	无效	无效
浮点	无效	无效	有效	有效	看注 2	有效	无效	无效	无效
日期	无效	无效	看注 4	无效	无效	无效	有效	无效	无效
时间	无效	无效	看注 4	无效	无效	无效	无效	有效	无效
时间标记	无效	无效	无效	无效	无效	无效	看注 5	看注 5	有效

注：

- 1.只有当字符（字节）个数等于数字个数并且字符（或十六进制）字段不是变长字段时才有效。
- 2.二进制字段没有小数部分时才有效。
- 3.两个字段有相同的小数位时才有效。
- 4.系统自动生成字段长度，所以不要在 30-34 列给出长度，长度不包括分隔符。
- 5.字段是仅输入时有效。

#### 2.3.10.1 数字型数据间的转换

物理文件记录格式中任何一种数字型转换都是允许的。例如：物理文件中二进制字段在逻辑文件转换成区位十进制字段。

#### 2.3.10.2 在区位十进制与字符或十六进制之间的转换

在提供的字段长度相同时，可以把区位十进制字段转换成字符型或十六进制字段，反之也可。程序中字段的类型是逻辑文件中指定数据类型。在 I/O 操作时，如果传递的数据仅含有数字字符（0-9）不产生错误。但程序不能做试图把不是 0-9 的数字字符从字符或十六进制字段传送到区位十进制字段中的 I/O 操作。OS/400 送一个信息并不能完成这个 I/O 操作。

例如：假定这是物理文件中的一个区位十进制字段。如果面向程序指定字符型（A），那么当这个字段通过逻辑文件返回到物理文件中时，必须确保这个字段中仅含有数字字符（0-9）。

另一个例子是，假定在物理文件中的是一个字符字段。如果在逻辑文件指定这个字段为区位十进制字段并且作为一个键字段，那么除非在这个物理文件中的所有记录都仅含有数字字符（0-9），否则不能生成这个逻辑文件。

### 2.3.10.3 从浮点数据转换为压缩十进制、区位十进制或二进制。

如果把浮点数字段（在物理文件中）转换为一个压缩十进制、区位十进制或二进制（在逻辑文件中），则必须明确确定其长度和小数点位置。当将浮点数据转换成定点数时，要确保指定的长度和小数位数的值足够大来放下这些数据，物理文件长度和小数位只是一个表示值而不是指出这个数值的大小。

### 2.3.10.4 连结字段时数据类型的转换

如果定义来自相关物理文件字段的连结字段，由 CONCAT 键字指定，不能指定数据类型。OS/400 程序基于被连结字段的数据类型来分配，一般规则如下：

- 如果连结包括一个或多个十六进制（H）字段，结果数据类型是十六进制（H）。
- 如果连结包括一个或多个字符（A）字段，但没有十六进制字段，则结果数据类型是字符型（A）。
- 如果连结仅包括数字（S，P，B）字段，结果数据类型是区位十进制（S）。

### 2.3.10.5 取子串时数据类型的转换

如果定义的字段是一个来自逻辑文件或相关物理文件字段的子串，（通过 SST 键字指定），必须指定其数据为字符型，原字段必须是字符型、十六进制或区位十进制（A，H 或 S）。

## 2.3.11 小数位（36 和 37 列）

对物理文件用这两个位置来指定在一个压缩十进制，区位十进制，二进制或浮点字段中小数点位置。用一个 0-31 的二进制数规定小数点右边的位数，（这个数必须小于字段长度）。在图 2-3 给出如何编码小数位字段。

在系统中实际存贮的数据不带有小数点。小数点是隐含的。例如：1.23 存贮为 123，如果没有指定编辑方式，在显示或打印文件中出现 123。

要使被引用字段（第 29 位 R）的这位无效，即可以指定一个新的值也可以在这个位置上修改。要增加位数，指定+n，n 为增量。要减少位数指定-n，n 为减量。例如：一个+4 的项指出，这个字段比被引用字段的小数点右边的数多 4 位。如果小数点的位数超出了允许最大数，那么送出一个错误信息。

对于逻辑文件，仅在它要改变或使引用的物理文件中相应字段的小数位无效时才用这二列。这两列为空白，那么定义的字段与它依据的物理文件的相应字段具有相同的小数位置。

要改变或使在一个压缩十进制或区位十进制字段内的小数点位置无效，指定 0-31 之中的数以指明小数点右边的小数位的数目。这个数必须不大于字段长度。当在物理文件中相应的字段是一个带有小数的二进制（数据类型 B）时，不能改变或使原指定的小数位无效。当逻辑文件的字段是二进制而在物理文件中相应的字段不是二进制时（在逻辑文件中 35 位指定 B），二进制字段的小数位必须是 0。

可以通过在这个位置上指定一个新的值或指出增、减的量来使这个字段的原定义的位数无效。要增加位数，指定+n，那里 n 为增量；要减少位数，指定-n，n 为减量。例如：一个

+4 指明，这个字段比被引用的字段在小数点右边多 4 位。

如果在 36 和 37 列上有一个值，并且程序通过这个逻辑文件字段对物理文件字段写或检索，OS/400 按小数点对齐这个数据。根据情况不同，可引起小数值的舍入，或引起一个数据转换错误。在下面的情况下，要舍入小数值：

- 当从一个比物理文件中指定的小数位少的逻辑文件中读时；
- 当往一个比物理文件中的小数位多的逻辑文件写时。

例如：如果物理文件中指定了一个 4 位长带 2 位小数的字段，而逻辑文件减少这个小数位为 0 位小数，那么在物理文件中为 0.20 的一个值在逻辑文件中变成了 0，而在物理文件中的一个 2.52 的值在逻辑文件中变成了 2。

当舍去小数值时，这个字段的左边是用 0 来填充的。

在下面的情况下将发生数据转换错：

- 当往一个比在物理文件指定了的小数位少的逻辑文件写时；
- 当从一个比在物理文件指定了的小数位多的逻辑文件中读时。

由于有过多的数字将被传送到小数点左边的可用空间，所以将产生一个数据转换错。例如：象上面的例子那样物理文件中定义了一个有两位小数的 4 位长的字段，并且在逻辑文件中减少了这个小数位使其为 0，那么对逻辑文件写的一个 3322 的值是不能装入到物理文件中的，这是因为在物理文件中小数点的左面仅允许出现 2 位数字。

要避免数据转换错，用与小数位相同的增量或减量来增加或减少逻辑文件的长度(30-34 列)。

如果对字段指定 CONCAT 键字，那么不能指定小数位。在一个连结字段中不能包括物理文件中有小数位的字段。

注：高级语言可能有小数位长度和值的限制，请遵循这些规定。

### 2.3.12 用途 (第 38 列)

用这项指定字段是仅输入的，输入/输出的，或不可输入/不可输出的。

对于物理文件，可指定下列项：

项	意义
空白	缺省值为 B (允许输入和输出)
B	允许输入和输出

由于缺省值即为 B，所以无需指定这一项。

在 38 列上的项是不能由 REF 和 REFFLD 键字引用的。因此当显示文件引用物理文件中的一个 38 列上为 B 的字段是没有影响的。

对于逻辑文件，可指定下列项：

空白 (缺省值)：

如果 38 列是空白，发生下面情况：

- 对于简单和多格式逻辑文件 (在记录层指定了 PFILE)，这个字段是可输入/输出的字段 (B)。
- 对于连接逻辑文件 (在记录层指定 JFILE 键字)，这个字段是只输入的 (I)。

B (既输入又输出)：

如果 38 列是 B，那么这个字段是一个既输入又输出字段。也就是说，程序既可以从此字段中读数据亦可以向这个字段写数据。对于连接逻辑文件二者都可的字段是无效的，这是因为连接逻辑文件是一个只读文件。

I (只输入)：

如果 38 列是 I，这是一个仅输入的字段。也就是说，程序可以从这个字段读数据，但不能改变它的内容。作为只输入字段的一个典型的情况是键字字段 (为了降低对访问路径的维护开销)，用户可以看但不能改的敏感字段 (如：在职员记录中的工资)，以及这些字段指定 SST 或 TRNTBL 键字。

如果程序对指定了只输入字段的记录格式执行一个修改操作，那么只输入字段不被修改，并且也不送出信息。如果程序对一个指定了只输入的字段的记录格式执行一个输出操作，那么只输入字段采用缺省值 (见 2.4.16DEF 键字)。

在物理文件中只输入字段是无效的。

N (二者都不是)：

如果 38 列是 N，则这是一个既不输入也不输出的字段并且只对连接逻辑文件有效。这样的字段可以用来作为连接逻辑文件的一个连接字段，但是程序不能用这个字段。

当在物理文件中连接字段的属性不匹配时，使用这个字段。在这种情况下，必须重定义一个或两个连接字段，但可能不想在记录格式中包含这些重定义的字段 (也就是说，不想让应用程序看到这些重定义的字段)，因此，为这些重定义的字段编码 N，则它们就不出现在这个记录格式中。

在 38 列上有 N 的字段不出现在由程序使用的缓冲区中。但用显示文件字段描述命令 (DSPFFD) 时将显示这个字段的描述。

既不输入亦不输出字段不能作为选择/省略字段或键字段。

38 列上的项不能由 REF 或 REFFLD 键字引用。因此显示文件引用逻辑文件中的在 38 位上为 I 或 B 的字段时，是无作用的。

### 2.3.13 位置 (39-44 列)

这几位物理文件或逻辑文件不用，除非用它做为注释行的正文外，要为空格。

### 2.4 键字项 (45-80 列)

这一节给出对物理文件和逻辑文件有效的键字项。它们是在 45-80 位上键入的（功能）。关于指定这些键字的通用规则见 1.2。

下面这些键字对于物理文件和逻辑文件都是有效的。（除括号中有注释）

ABSVAL	FIFO
ALIAS	FLTPCN
ALL (仅用于逻辑文件)	FORMAT
ALTSEQ	LIFO
ALWNULL (仅用于物理文件)	NOALTSEQ
CCSID (仅用于物理文件)	RANGE
CHECK	REF (仅用于物理文件)
CHKMSGID	REFFLD (仅用于物理文件)
CMP	REFSHIFT
COLHDG	RENAME (仅用于逻辑文件)
COMP	SIGNED
CONCAT (仅用于逻辑文件)	SST (仅用于逻辑文件)
DATFMT	TEXT
DATSEP	TIMFMT
DESCEND	TIMSEP
DFT (仅用于物理文件)	TRNTBL (仅用于逻辑文件)
DIGIT	UNIQUE
DYNSLT (仅用于逻辑文件)	UNSIGNED
EDTCDE	VALUES
EDTWRD	VARLEN
FCFO	ZONE

下面这些键字仅对单格式和多格式逻辑文件有效：

PFILE            REFACCPFH

下面键字仅对连接逻辑文件有效：

JPFTVAL        JFILE        JOIN  
JPUPSEQ        JFLD        JREF

当用 DDS 来描述一个源文件时（通常使用 CRTSRCDF 命令而不是 DDS），或当用源物理文件作逻辑文件时，不能使用下面的键字：



ABSVAL	NOALTSEQ
ALTSEQ	SIGNED
DESCEND	UNIQUE
FCFO	VARLEN
FIFO	ZONE
LIFO	

#### 2.4.1 ABSVAL (绝对值)

使用这个键字段层关键字指出当 OS/400 用一个数字段值做关键字排序时省略这个字段的符号。

这个关键字没有参数。

下面给出了用区位十进制作键字段的 6 个记录。

记录	数字键字段 (区位十进制)	十六进制表示
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

如果没指定任何顺序关键字或 ALTSEQ 关键字，键字段的缺省顺序是 SIGNED (有符号)，在这种情况下，这些记录是以下面的顺序来排序：

记录	键字段数字 (区位十进制)	十六进制表示
3	98-	F9D8
2	00	F0F0
5	20	F2F0
4	97	F9F7
1	98	F9F8
6	99	F9F9

如果指定了 ABSVAL 关键字，则负数字段用绝对值排序，可以得到下面的排序结果：

记录	键字段数字 (区位十进制)	十六进制表示
2	00	F0D0
5	20	F2F0
4	97	F9F7
1	98	F9F8
3	98-	F9D8
6	99	F9F9

**ABSVAL** 对字符型、日期、时间、时间表和十六进制数据字段无效。它不能与 **DIGIT**、**SIGNED**、**UNSIGNED** 或 **ZONE** 键字一起使用。

**ABSVAL**（键字段层键字）导致省略 **ALTSEQ**（文件层键字）。如果对一个键字段指定了 **ABSVAL**，那么即使在文件层指定了 **ALTSEQ**，对这个键字段仍然是 **NOALTSEQ** 起作用。无论是否指定了 **NOALTSEQ** 键字，都是如此。

图 2-18 给出了如何来指定 **ABSVAL** 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      ORDAMT      5 0  
00020A      K ORDAMT          ABSVAL  
A
```

图 2-18 规定 **ABSVAL** 键字

#### 2.4.2 ALIAS（替换名）

使用这个字段层键字为一个字段指定替换名。在程序编译时，替换名传给程序代替 DDS 中的字段名。是否使用 **ALIAS** 名，由高级语言编译程序决定的。关于 **ALIAS** 支持的语言的信息请参照相应的高级语言参考手册。

这个键字的格式是：

**ALIAS**（替换名）

对于 **ALIAS** 命名转换参考 1.2 中语法规则。

替换名必须不同于其它所有的替换名，也不同于记录格式中 DDS 字段。如果发现重名，将在字段名或替换名上出现一个错误信息。

替换名不能在 DDS 内部或任一其它的 OS/400 功能中使用（例如：作为键字段名，作为一个 REFFLD 键字中指定的字段名或是作为一个在复制文件（CPYF）命令中使用的字段名）。

当引用一个带有 **ALIAS** 键字的字段时，除非明确的指定引用字段的 **ALIAS** 之外，被引用的字段的 **ALIAS** 键字也将被复制。

图 2-19 显示了如何指定 **ALIAS** 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      FIELDA      25A      ALIAS (CUSTOMERNAME)  
A
```

图 2-19 规定 **ALIAS** 键字

在上图中 **FIELDA** 的替换名是 **CUSTOMERNAME**。

#### 2.4.3 ALL（全部）——仅用于逻辑文件

使用这个选择/省略层键字来指定在对逻辑文件所有选择/省略处理之后的动作。在 17 位上带有 S 的 ALL 指示 OS/400 选择所有没有满足其它选择/省略规则的记录。在 17 位上指定 0，指示 OS/400 省略所有没有满足其它选择/省略规则的记录。如果指定 ALL 必须是在其它的选择/省略键字之后，在 ALL 键字上不能指定字段名。

这个键字没有参数。

如果没指定 ALL 键字，缺省的动作是采取为这个文件所做的最后一个选择/省略规范的反面，即：如果最后的规范是选择，那么缺省为省略所有的，而如果最后一个规范是省略，则缺省为选择所有的。

图 2-20 显示如何指定 ALL 键字。

00010A	S ACT	COMP(EQ 3000)
00020A	S ACT	COMP(GT 3100)
00030A	O AMT	COMP(LT 0)
00040A	O	ALL
A		

图 2-20 规定 ALL 键字

#### 2.4.4 ALTSEQ (交替分配顺序)

如果文件是键字访问的一个键，那么用这个文件层键字来指示 OS/400，当检索一个文件成员的排序记录时，使用一个交替分配顺序表。

这个键字的格式：

ALTSEQ ([库名]/表名)

交替分配顺序表的名字是一个必须有的参数，库名是可选的。如果不指定库名，OS/400 使用文件生成时用的库列表 (\*LIBL)。

在下列条件下 ALTSEQ 键字是无效的：

- 当在建立物理文件 (CRTPF) 或建立逻辑文件 (CRTLTF) 命令中指定了 FILETYPE(\*SRC) 时。关于 FILETYPE(\*SRC) 参数值的信息见数据库程序设计。
- 当键字段为压缩十进制、二进制或浮点数据类型时。
- 当键字段规定 ABSVAL 或 SIGNED 时。对这些字段，假定是 NOALTSEQ 且无需指定，可以对不要求交替顺序的组合键中的任意字段指定 NOALTSEQ。
- 在建立物理文件 (CRTPF) 或建立逻辑文件 (CRTLTF) 的 SRTSEQ 参数中指定了不是 \*SRC 的值。
- ALTSEQ 键字不能与 REFACCPTH 键字一起使用。

对交替分配顺序表，要有使用的权限。交替分配顺序表是使用建立表命令 (CRTTBL) 生成的。

ALTSEQ 使区位键字段做无符号的缺省顺序。可以对某个键字指定 SIGNED 键字来使该缺省无效。

图 2-21 给出了如何在逻辑文件中指定一个 ALTSEQ 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                                ALTSEQ(TABLELIB/TABLE1)  
00020A      R RECORD1                  PFILE (PF1)  
00030A      :  
00040A      :  
00050A      :  
00060A      NAME          20  
00070A      :  
00080A      :  
00090A      K NAME  
A
```

图 2-21 规定 ALTSEQ 键字

格式为 RECORD1 的记录是按照键字段 NAME 根据替换对照顺序表来排序的 (TALBELIB 库中的 TABLE1)。

#### 2.4.5 ALWNULL (允许空值) ——只用于物理文件

使用这个字段层关键字来定义这个字段允许空值。

这个关键字没有参数。

当指定 ALWNULL 键字，能在 30-34 列指定的最大长度是 32765 字节。(如果字段是变长的为 32739 字节)。

对于物理文件，如果指定 DATFMT 的值为 \*JOB、\*MDY、\*DMY、\*YMD 或 \*JUL 并且字段允许空值，必须为这个字段用 DFT 键字指定一个有效日期。

图 2-22 给出如何指定 ALWNULL 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1  
00020A      FIELD1      75A      ALWNULL  
00030A      FIELD2      100A  
00040A      FIELD3      L      ALWNULL  
00050A  
00060A  
A
```

图 2-22 规定 ALWNULL 键字

FIELD1 被定义为允许空值。FIELD1 的缺省值是空值。FIELD2 不允许空值。FIELD2 的缺省值是空格。

#### 2.4.6 CCSID (编码字符集标识)

使用这个物理文件的文件层或字段层键字，逻辑文件的字段层键字为字符字段，指定一个编码字符集标识。

键字格式如下：

CCSID (值 [字段—显示—长度])

值最大可为 5 位数，它定义一个编码表标识、字符集标识、编码页标识和其它有关信息，它为字段中的数据唯一指定了可使用的编码图形字符表示。

对于逻辑文件的字段，如果使用 CCSID 键字，下列条件必须为真：

如果 CCSID 使用 UCS-2 级 1 编码方案，那么字段的数据类型必须是 G，而相应的物理文件字段的类型必须是 A 或 G。如果 CCSID 不用 UCS-2 级 1 编码方案，字段的数据类型必须是 A、O 或 G，相应物理文件字段的类型必须是 G 且 CCSID 必须用 UCS-2 级 1 编码方案。

“字段—显示—长度”参数是可选的。仅当这个字段被显示文件引用时才用它。这个参数仅在用 UCS-2 级 1 编码方案时才有效。它允许用户根据存储在 UCS-2 级 1 中的数据类型来控制字段的尺寸，详细内容请参考附录 H。

当在物理文件的文件层中指定时，除非字段层有 CCSID 键字指定，否则这个 CCSID 键字将提供给文件中的每一个字符字段。如果在字段层有 CCSID 规定用 UCS-2 编码方案，字段的类型必须是 G。

如果在文件层没指定 CCSID 键字，且并非所有字符字段都有 CCSID 键字指定，那么文件建立时，字段被分配为作业缺省的 CCSID。

对 AS/400 系统有效的 CCSID，请见国际语言手册。图 2-23 给出如何指定物理文件的 CCSID 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                                CCSID(285)  
00020A      R RECORD1  
00030A      FIELD1      75G      CCSID(13488)  
00040A      FIELD2      150A  
00050A      FIELD3      20A  
A
```

图 2-23 规定物理文件的键字

分配给 FIELD1 的 CCSID 值是 13488，分配给 FIELD2 和 FIELD3 的 CCSID 值是 285。

图 2-24 给出在相应的逻辑文件中如何规定 CCSID 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00000A  
00010A      R RECORD1  
00020A      FIELD1      75A      CCSID(37)
```

00030A	FIELD2	150G	CCSID(13488 80)
00040A	FIELD3	20A	
	A		

图 2-24 规定逻辑文件的 CCSID 键字

分配给逻辑文件的 FIELD1 的 SBCSCCSI 值为 37。由于物理文件字段有 UCS-2 数据，所以在物理文件和逻辑文件的 FIELD1 字段间发生转换，逻辑文件的 FIELD2 的 UCS-2 CCSID 的值为 13488，与 FILED1 字段同样原因，FIELD2 也有转换发生。FIELD3 没规定 CCSID。

#### 2.4.7 CHECK (检查)

用这个字段层键字来对在显示文件中使用此字段时进行有效性检验。

这个键字的格式是：

CHECK (编辑检验码[...])

CHECK 对定义它的物理和逻辑文件没有影响。当在显示文件中定义一个可输入字段时，通过在 29 列指定 R 和 REF 或 REFFTD 键字引用现在定义的字段。在建立显示文件时，OS/400 从物理文件把这个字段的 CHECK 键字和其它的属性复制到显示文件中的字段中，也可在显示文件中指定有效检验键字来替换原字段的有效性检验键字。详细信息见 3.2.9 中“引用”（第 29 列）。

在物理文件和逻辑文件中指定这个键字的规则与对一个显示文件的引用规则是相似的，但对物理文件和逻辑文件仅允许下面的编码：

编码	意义
AB	允许空白
ME	强制性输入
MF	强制性填充
M10	IBM 模 10 自检验算法
M10F	IBM 模 10 自检验算法
M11	IBM 模 11 自检验算法
M11F	IBM 模 11 自检验算法
VN	有效日期名
VNE	有效日期名扩展

对浮点字段（第 35 列上是 F）上不能指定 CHECK(AB)、CHECK(VN)、CHECK(VNE)、CHECK(M10)、CHECK(M11)、CHECK(M10F)或 CHECK(M11F)键字。

详细信息见 3.3.21 中 CHECK (检验) 的例子。

#### 2.4.8 CHKMSGID (检查信息标识)

使用这个字段层键字来定义一个与有效检验键字有关的错误信息。如果没有指定 CHKMSGID 键字，使用系统提供的信息。如果对某字段指定了 CHKMSGID 键字而在建

立的显示文件中引用的话，有效的检验信息和 CHKMSGID 键字被复制到显示文件中。如果在从屏幕输入时发现了有效检验错误，那么在 CHKMSGID 键字中规定的错误信息在信息行上显示出来。

CHKMSGID 对物理文件或逻辑文件没有影响。它的格式如下：

CHKMSGID (信息标识 [库/] 信息文件[信息数据字段])

如果指定了信息数据字段参数，那么这个字段必须在物理文件或逻辑文件中定义，但如果有 CHKMSGID 键字的字段在建立显示文件时被引用，信息数据字段则必须在显示文件中定义。(用与 CHKMSGID 键字中字段的相同格式定义)。

只在有 VALVES、RANGE、CMP、COMP、CHECK(M10)、CHECK(M11)、CHECK(VN) 或 CHECK(VNE) 键字的字段才允许出现 CHKMSGID。

详细信息及例子见 3.3.23CHKMSGID。

#### 2.4.9 CMP (比较)

这个键字与 COMP 键字等效。

这个键字的格式是：

CMP (关系操作符值)

一般都用 COMP 键字。详细解释见 2.4.11 中 COMP (比较) 的描述。

#### 2.4.10 COLHDG (栏目标题)

使用这个字段层键字指定由正文管理、查询实用程序、数据文件实用程序 (DFU) 和屏幕设计辅助工具 (SDA) 使用的这个字段的栏目标题。

格式如下：

COLHDG ('TF-1' [ '行-2' [ '行-3' ]])

最多允许 3 行，每行允许 20 个字符。每一行必须用单引号括起来，在栏目标题中如果要用单引号，那么需使用双引号 ("")。必须使用多于一个的空白来分隔每行。

对于物理文件，如果没指定 COLHDG 并不从被引用字段复制，那么使用字段名。如果在逻辑文件中没指定 COLHDG，除非这个字段是个连接字段（缺省值为该字段名）之外，使用相应物理文件字段的栏目标题。

如果指定了 COLHDG 而没指定 TEXT，则栏目标题的 50 位信息用来作为正文。例如 COLHDG ('Order' 'Date') 与 TEXT ('Order' Date') 是等效的。

图 2-25 显示了如何在物理文件中指定 COLHDG 键字。

00150A	ORDDAT	5 0	COLHDG('Order' 'Date')
00160A	NAME	20	COLHDG('Customer''s Name')
00170A	CITY	20	COLHDG('Customer' 'City' 'Field')

A

图 2-25 规定 COLHDG 键字

因为 Order date 是数字字段（图中的 NNNNN），那么对 ORDDAT 必须指定小数位或数据类型。

下面显示给出了当运行正文管理、查询、DFU 或 SDA 时栏目标题出现的形式。

---

---

Customer		
Order		City
Date	Customer's Name	Field
NNNN	XXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXX

---

#### 2.4.11 COMP 比较

使用这个字段层关键字来指定在显示文件引用这个字段时进行有效性检验。对于逻辑文件，也可以在选择/省略字段层指定这个关键字。COMP 与 CMP 是等效的。

它的格式是：

COMP (关系操作符值)

在选择/省略层，格式如下：

COMP (关系操作符 字段名)

有效的关系操作符是：

关系操作符	意义	关系操作符	意义
EQ	等于	GT	大于
NE	不等于	NG	不大于
LT	小于	LE	小于等于
NL	不小于	GE	大于等于

既可以在字段层亦可以在选择/省略层指定值参数。在选择/省略层仅可指定字段名参数。

##### 2.4.11.1 在字段层指定 COMP

在字段层，COMP 键字对描述它的逻辑文件和物理文件是没有影响的。但在显示文件描述一个可输入字段时，可以通过在 29 列指定 R 和指定 REF 或 REFFLD 键字引用这个字段。在建立显示文件期间，OS/400 把逻辑文件的字段及 COMP 键字和其它的字段属性复制到这个显示文件的字段中。可以对显示文件中的字段指定有效性检验键字来使对逻辑文件字段中原指定的键字无效（同其它有效检验键字和 CHKMSGLD 键字一样）。详细信息见 3.2.9 的引用（第 29 列）。

对字段层的 COMP 键字不能指定字段名来作为参数值。

对于字段层的 COMP 键字不能指定\*NULL 作为参数值。

不能对浮点类型字段（第 35 列为 F）或十六进制字段（第 35 列为 H）指定 COMP 键

字。也不要在日期、时间、时间标识字段（第 35 列为 L、T 或 Z）指定 COMP 键字。

在逻辑文件中或在物理文件中指定这个关键字的规则与在显示文件中相同。关于如何指定这个关键字的信息见 3.3.31 中 COMP（比较）。

定义一个数字字段，当一个工作站的用户键入数据时，OS/400 根据这个文件中的小数位数来对齐键入的字符。当这个字段传给程序时，其前后的空白都是用 0 来填充。如果没键入小数点，OS/400 认为小数点在被键入字符的最右边。例如：对于一个 5 位长带有二位小数的 1.2 解释为 001.20，而 100 认为是 100.00。

#### 2.4.11.2 在选择/省略层指定 COMP

在选择/省略层可以对 COMP 指定字段名或\*NULL 作为参数。

如果指定一个值，下面的规则是合适的：

- 如果定义字符字段，指定字符常数或十六进制的字符串。
- 指定字符串要用单引号（见图 2-26）
- 用 X 开头，后跟单引号中的数字 0—9 与字母 A—F 表示一个十六进制字符串。十六进制数字的个数必须是指定字段长度的 2 倍（见图 2-27）
- 如果定义一个数字串，指定数字 0—9，不用单引号，象图 2-27。
- 如果定义一个日期字段，那么使用在 DATFMT 键字中同样的格式，使用在 DATSEP 键字中同样的分隔符。例如，如果 DATFMT(\*MDY)，DATSEP 为 ‘/’，那么 COMP(EQ ‘12/15/91’ )是缺省值。
- 如果定义时间字段，使用在 TIMFMT 键字中同样的格式，使用在 TIMSEP 键字中同样的分隔符。例如：如果 TIMFMT(\*ISO)，那么 COMP(EQ ‘11.00.00’ )是缺省值。  
ISO 缺省的分隔符是一个隔点 (.)。

如果定义时间标识字段，必须按以下格式指定缺省值：

COMP(EQ ‘YYYY—MM—DD—HH.MM.SS.UUUUUU’ )。

如果指定了\*NULL，关系比较符必须为 EQ 或 NE。

当程序把一个输入操作送到定义的记录格式时，COMP 从这个逻辑文件依据的物理文件中选择或不要记录。OS/400 程序根据用选择/省略字段的值测试的结果来选择或不用记录，这个可以是你规定的字段名或空值。

图 2-26 给出了如何为字符串和数字串指定 COMP 键字。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8								
00010A	R RECORD			PFILE(PF1)				
00020A								
00030A	FIELD A	1 0	COMP(NE 0)	1				
C0040A	FIELD B	1	COMP(NE 'A')	1				
00050A	FIELD C							
00060A	FIELD D							
00070A	FIELD E							
00080A	K FIELD B							
00090A	S FIELD C		COMP(EQ FIELD D)	2				

00100A	S FIELD A	COMP(NE 0)	2
00110A	S FIELDE	COMP(NE *NULL)	2
00120A	O FIELDB	COMP(GE 'A')	2
	A		

图 2-26 规定 COMP 键字 (例 1)

1. 对 FIELD A 和 FIELDB 指定 COMP 作为引用它们的显示文件做有效性检验键字。
2. 对 FIELD A、FIELDB、FIELD C 和 FIELDE 指定 COMP 做为选择/省略键字。根据下面的比较结果通过逻辑文件记录格式，从物理文件 PF1 中检索记录：

- FIELD C：当 FIELD C 等于 FIELD D 时选择记录。
- FIELD A：当 FIELD A 不等于 0 时，选择那些不满足 FIELD C 检测的记录。
- FIELDE：当 FIELDE 非空值时，不要那些满足 FIELD A 测试条件的记录。

图 2-27 给出用十六进制字符串指定 COMP 键字。

00010A	R RCD1	PFILE(PF1)
00020A	CODEA	
00030A	FLD1	
00040A	FLD2	
00050A	K FLD1	
00060A	S CODEA	COMP(EQ X'51')
	A	

图 2-27 规定 COMP 键字 (例 2)

COMP 被指定做为 CODEA 的选择/省略键字 (一个字节的字段)，仅当 CODEA 的值是十六进制 51 时，通过这个文件记录格式从物理文件 PF1 中检索记录。

#### 2.4.12 CONCAT (连结) —— 仅用于逻辑文件

当要将来自物理文件记录格式中的两个或两个以上的字段连接成定义的逻辑文件记录格式中的一个字段时，使用这个字段层键字。连结字段的名字必须在 19-28 位上给出。

这个键字的格式是：

CONCAT (字段 1 字段 2...)

按连结顺序依次指定物理文件的字段名，中间用空白隔开。

如果在逻辑文件的记录格式中多次指定了同一个物理字段 (也就是说，用 RENAME 和 CONCAT)，那么在逻辑文件中指定这个字段的顺序是在修改或插入操作传送到物理字段的数据顺序。这样，在物理字段中最后出现的值是放入到这个物理记录的值，并且这个值被用来建立物理字段的所有键字。同一个字段以前的值都被省略。

如果想用一个由 CONCAT 键字定义的字段或一个在 CONCAT 键字中做参数的字段作为

一个键字段，见 2.3.7.3 的内容。

在连结字段不能包括有小数位的字段，但可以包括小数位为 0 的字段，在这个情况下它是作为整型数字来处理的。

OS/400 程序分配连接字段的长度为所有被连接字段的总长度（数字和字符）。

OS/400 程序根据连接的字段情况来确定字段的定长或变长属性。通常规则是：

- 定长字段和变长字段的连接结果为变长字段。
- 除了对规定 CONCAT 键字的字段同时指定 VARLEN 键字外，定长字段与定长字段的连接结果还是定长字段。

注：如果连接的结果字段是变长字段或允许空值，那么 CONCAT 字段必为仅输入（第 38 列为 I），假如逻辑文件中包括连接字段，那么它依据的物理文件字段不能有空值。

OS/400 程序根据被连接字段的数据类型来分配数据类型，通常的规则是：

- 如果连接包括一个或多个十六进制 (H) 字段，那么结果数据类型是十六进制 (H)。
- 如果连接包括一个或多个字符 (A) 字段，但没有十六进制字段，那么结果数据类型是字符 (A)。
- 如果连接仅包括数字 (S,P,B) 字段，那么结果数据类型是区位十进制 (S)。

当连结数字字段时，在连结中最右边字段的符号被用作连结后字段符号，其它字段符号被省略，但在连结后字段中它们仍存在。因此，如果一个字段是负值，但不是最后一个字段，你必须采用相应的动作来清除被嵌入的符号（例如：将连结字段转换成压缩十进制型）。

连结字段的最大长度根据连接字段的数据类型和长度不同是可变的。如果是区位十进制 (S)，总长度不得超过 31 字节。如果是字符 (A) 或十六进制 (H)，总长度不得超过 32766 字节。如果连接字段是变长字段，总长度不得超过 32740 字节（如果允许空值则为 32739 字节）。

在连结字段中，不能包括浮点、日期、时间、时间标记字段。

在连接逻辑文件中，被连结的字段必须来自同一个物理文件。在 CONCAT 键字中指定的第一个字段指明使用哪个物理文件。因此，第一个字段在连接逻辑文件依据的物理文件中必须是唯一的，而且必须用 JREF 键字来指定使用哪个物理文件。

图 2-28 到图 2-31 给出如何来指定 CONCAT 键字。

在物理文件中的 MTH、DAY 和 YEAR 字段被连结成逻辑文件中的 DATE 字段。象图 2-28 所示。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
00010A	R RECORD1	PFILE (PF1)
00020A	DATE	CONCAT (MTH DAY YEAR)
A		

图 2-28 规定 CONCAT 键字（例 1）

图 2-29 中，如果程序是将 DATE 的 01 03 81 改为 02 05 81，由于最后指定的字段是 MTH (值为 01)，DAY (值为 03) 和 YEAR (值为 81)，所以放在物理记录中的值是没修改。但，

如果 MTH、DAY 和 YEAR 变成了一个新值，那么在物理记录中 DATE 的值也被改变。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD2          PFILE(PF1)  
00020A      DATE              CONCAT(MTH DAY YEAR)  
00030A      MTH  
00040A      DAY  
00050A      YEAR  
  
A
```

图 2-29 规定 CONCAT 键字（例 2）

图 2-30 中，来自物理文件的那些字段在逻辑文件中被连结成多个字段。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD3          PFILE(PF1)  
00020A      DATE              CONCAT(MTH DAY YEAR)  
00030A      CMPDAT            CONCAT(DAY MTH YEAR)  
  
A
```

图 2-30 规定 CONCAT 键字（例 3）

在图 2-31，如果来自 PF1 的字段是：

FIXED1 是定长字段

FIXED2 是定长字段

VARLEN1 是变长字段

结果字段是：

FIELD1 是变长字段

FIELD2 是定长字段

FIELD3 是变长字段

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD4          PFILE(PF1)  
00020A      FIELD1            CONCAT(FIXED1 VARLEN1)  
00030A      FIELD2            CONCAT(FIXED1 FIXED2)  
00040A      FIELD3            CONCAT(FIXED1 FIXED2)  
00050A      VARLEN  
  
A
```

图 2-31 规定 CONCAT 键字（例 4）

#### 2.4.13 DATFMT (日期格式)

使用这个字段层键字来指定日期字段格式。这个键字仅对日期字段（数据类型 L）或由物理文件字段是日期类型形成的逻辑文件区位字段（数据类型 S）有效。

格式如下：

#### DATFMT (日期格式)

日期格式参数为日期指定格式。下表描述了各种日期格式和它们的缺省分隔符。

格式的名字	日期格式参数	日期格式和分隔符	字段长度	例 子
Job Default	*JOB			
Month/Day/Year	*MDY	mm/dd/yy	8	06/21/90
Day/Month/Year	*DMY	dd/mm/yy	8	21/06/90
Year/Month/Day	*YMD	yy/mm/dd	8	90/06/21
Julian	*JUL	yy/ddd	6	90/172
International Standards Organization	*ISO	yyyy-mm-dd	10	1990-06-21
IBM USA Standard	*USA	mm/dd/yyyy	10	06/21/1990
IBM European Standard	*EUR	dd. mm. yyyy	10	21. 06. 1990
Japanese Industrial Standard Christian Era	*JIS	yyyy-mm-dd	10	1990-06-21

如果指定\*JOB，缺省是作业属性。

对于物理文件，如果没指定 DATFMT 键字，缺省为\*ISO。

对于逻辑文件，如果没指定 DATFMT 键字，缺省是来自物理文件的日期格式。

对于逻辑文件和物理文件，如果没指定 DFT 键字，缺省值是当前日期。

如果指定了\*ISO、\*USA、\*EUR 或\*JIS 值，那么就不能指定 DATSEP 键字。这些格式有一个固定的分隔符。

DATFMT 键字替换日期字段的作业属性。它不改变系统缺省值。

对于物理文件，如果你指定了\*JOB、\*MDY\*、\*DMY、\*YMD 或\*JUL 值并且允许空值，必须为这个字段用 DFT 键字指定一个有效日期。

图 2-32 给出如何指定 DATFMT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A
00020A      R RECORD
00030A      DATFLD1      L      DATFMT (*JUL)
00040A      DATFLD2      L      DATFMT (*EUR)
A
```

图 2-32 规定 DATFMT 键字

如果当前日期是 1990 年 6 月 21 日，系统当前日期格式是 MDY，分隔符是 /，DATFLD1 中为 90/172（指 1990 年第 172 天），DATFLD2 中为 21.06.1990。

#### 2.4.14 DATSEP (日期分隔符)

使用这个字段层键字为一个日期字段指定分隔符。这个键字仅对日期字段(数据类型 L)有效。

格式如下：

DATSEP(\*JOB | ‘日期分隔符’ )

日期分隔符参数指定出现在年、月、日之间的分隔符号。可选值是斜线 (/)、下划线 (\_)、隔点 (.)、逗号 (,) 或空格 ( )。参数必须括在单引号里。

如果指定 \*JOB，缺省值是作业属性。

对于物理文件，如果不指定 DATSEP 键字，缺省值是作业属性。

对于逻辑文件，如果不指定 DATSEP 键字，缺省值是来自物理文件的日期分隔符。如果没有为物理文件字段指定 DATSE[P] 键字，也没有 DATFMT 规定 (\*ISO、\*USA\*、EUR 或 \*JIS)，对于 DATSEP，缺省值是作业属性。

如果在 DATFMT 键字上指定了 \*ISO、\*USA、\*EUR 或 \*JIS 日期格式值，就不能指定 DATSEP 键字。这些格式是有固定的日期分隔符。

DATSEP 键字替换作业属性。它不改变系统缺省值。

图 2-33 给出如何指定 DATSEP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A  
00020A      R RECORD1  
00030A      DATFLD2      L      DATFMT(*DMY) DATSEP(' -')  
00040A      DATFLD4      L      DATSEP(' ')  
A
```

图 2-33 规定 DATSEP 键字

如果当前日期是 1990 年 6 月 21 日，系统当前日期格式是 MDY，并且系统日期分隔符是 ‘/’，DATFLD2 为 21-06-90，DATFLD4 为 06 21 90。

#### 2.4.15 DESCEND (降序)

使用这个键字段层键字来指定字符、十六进制或数字键字段的值是按降序来检索的。缺省值是升序的。用 DESCEND 键字做数据存贮的例子，见 2.4.39 的“SIGNED (符号)”中的例子。

此键字没有参数。

图 2-34 给出如何为逻辑文件指定 DESCEND 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      K ITEM
00020A      K BALDUE          DESCEND
A

```

图 2-34 规定 DESCEND 键字

#### 2.4.16 DFT (缺省值) ——仅用于物理文件

使用这个字段层的键来指定一个字段的缺省值。

这个键字的格式是：

DFT ('值' | 数字值 | X '十六进制值' | \*NULL)

如果没有这个键字，字符和十六进制字段缺省值是空格，而数字字段缺省值是 0。如果为字段指定 ALWNULL 键字，那么字符、十六进制和数字字段的缺省值是空值。

指定值时要按下列规则做：

- 如果定义为字符字段，那么指定一个字符常数、十六进制值或\*NULL，字符串用单引号括起来。如果字段是变长 (VARLEN)，那么串长度必须小于等于分配长度。
- 在 X 这后跟随数字 0-9 与字母 A-F，且用单引号括起来。在单引号内的十六进制的位数必须是这个字段分配长度的两倍。如果字段是变长 (VARLEN)，那么在单引号中十六进制的位数必须是分配长度的两倍。
- 如果字段定义为十六进制字段，指定字符常数，十六进制值或\*NULL。

注：如指定一个字符常数，它的十六进制表示值是缺省值。

- 用单引号括起指定的字符串。如果字段为变长 (VARLEN)，那么串长度必须小于等于分配的长度。
- 如果定义一个数字型的字段，指定一个数字值(数字 0-9，无须使用单引号)或\*NULL。对于 36 和 37 列不是 0 的字段，在 DDS 中的相应位置指定数字常数的一个小数点字符。
- 如果指定\*NULL，那么必须也指定 ALWNULL 键字。
- 如果没有指定任何值 (DFT (' '))，那么零长度的串为缺省值，并且仅当字段是变长时有效。(也必须指定 VRALEN 键字)。
- 如果定义了一个日期字段，用与 DATFMT 指定的同样格式指定一个有效日期，并且用 DATSEP 指定的同样分隔符。例如，在 DATFMT 指定了\*MDY 并且 DATSEP 指定了分隔符为 '/' 的情况下，DFT ('12/15/91') 是缺省值。
- 如果没有指定 DFT 键字，缺省值是当前日期。
- 如果指定一个时间字段，用与 TIMFMT 中同样的格式来指定一个有效时间，并且用在 TIMSEP 中同样的分隔符。例如：如果为 TIMFMT 指定\*ISO，DFT ('11.00.00') 是有效值。对于\*ISO 来说，缺省分隔符是分隔点 (.)。
- 如果没有指定 DFT 键字，缺省值是当前时间。
- 如果指定一个时间标记字段，就必须按如下格式指定缺省值：  
DFT ('YYYY-MM-DD-HH.MM.SS.UUUUUU')。

- 如果没指定 DFT 键字，缺省值是当前时间。

在下面的情况下，字段被赋予规定的值：

- 当程序向一个以物理文件为依据的逻辑文件做输出操作并且在逻辑文件中的记录格式给这个字段命名时。
- 当用初始化物理文件成员 (INZPFM) 命令时。
- 当使用带有 FMTOPT(\*MAP) 的复制文件 (CPYF) 命令，而且 To—文件的某一个字段不在 From—文件中时。

当程序对一个连接逻辑文件作输入操作并且下面的所有情况都为真时提供给程序指定的值：

- 对连接逻辑文件指定了 JDFTVAL 键字。
- 文件被定义为连接逻辑文件中的次文件。
- 当发生输入操作并且次文件的连接没产生记录时，这个关键字不对物理文件的输入操作产生影响。

图 2-35 显示了如何指定 DFT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1
00020A      CHARFLD1    20A      DFT(' Sample field')
00030A      CHARFLD2    5A       DFT(X' D985955185')
00040A      HEXFLD1     3H       DFT(' ABC')
00050A      HEXFLD2     3H       DFT(X' C1C2C3')
00060A      NUMFLD1     5S 0    DFT(99999)
00070A      NUMFLD2     5S 2    DFT(999. 99)
00080A      NUMFLD3     5S 2    DFT(999)
00090A      NUMFLD4     5S 2    DFT(*NULL)
00100A          ALWNULL
00110A      NUMFLD5     5S 2    DFT(999. 99)
00120A          ALWNULL
00130A      DATFLD1     L       DATFMT(*MDY) DATSEP(' -')
00140A          DFT(' 12-31-91')
00150A      TIMFLD1     T       DFT(' 11. 15. 00')
A
```

图 2-35 规定 DFT 键字

CHARFLD1 的缺省值是 ‘Sample field’。CHARFLD2 的缺省值是十六进制 D985955185。HEXFLD1 有缺省值是 C1C2C3 (字符常量的十六进制表示法)。HEXFLD2 的缺省值是 C1C2C3。NUMFLD1 的缺省值是 99999 (由于这个字段没有小数位所以不需要小数点)。NUMFLD2 的缺省值是 999.99。NUMFLD3 的缺省值是 999 (如果不需要指定小数值则不需要小数点)。NUMFLD4 是缺省值是空值 (如果 DFT(\*NULL)，那么需要 ALWNULL 键字)。NUMFLD5 的缺省值是 999.99；这个字段也允许空值。DATFLD1 的缺省值是 12-31-91。

TIMFLD1 的缺省值是 11.15.00 (\*ISO格式)。

#### 2.4.17 DIGIT (数字)

用这个键字段层关键字来指定当用这个字段做索引字段时仅使用它每个字节的数字部分(最右边的4个二进制位), 其高位部分填零。

这个关键字没有参数。

这个关键字是对整个键字段提供的(不只是这个字段中的一部分)。它仅对字符型、十六进制和压缩十进制型有效。

不能与 ABSVAL、SIGNED、ZONE 关键字一起使用这个关键字。

如果对一个键字段指定了 DIGIT, 这个字段的值是作为一个无符号的二进制数据处理的, 对于区位十进制字段它是一个缺省值。

图 2-36 显示如何指定 DIGIT 关键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00040A          K ORDTYP           DIGIT  
A
```

图 2-36 规定 DIGIT 关键字

如果 ORDTYP 是一个3字节的字段, 那么三个不同记录的字段值可以是下边这样的。

值	十六进制	用作关键字的数字
C4J	C3F4D1	341
CMA	D3D4C1	341
3D1	F3C4F1	341

#### 2.4.18 DYNSLT (动态选择)

使用这个文件层关键字来指明对文件的选择/省略检验是在处理时做的。这个关键字指定动态选择/省略而不是访问路径的选择/省略。

这个关键字没有参数。

当程序对一个有 DYNSLT 关键字的逻辑文件做输入操作时, 系统检测相关物理文件中的所有记录看其是否满足选择/省略的值, 只有满足的记录才提供给程序, 对每一个记录的检测可能导致较慢的 I/O 性能, 但可能比维护一个文件的访问路径更为有效, 有时这对只读文件往往是特别合理的, 特别是当依据的物理文件频频修改时, 就更为有效。使用动态选择/省略对带选择频率比较高的文件更有效。

关键字顺序访问文件, 访问路径是在生成文件时建立的, 并且是根据在生成逻辑文件(CRTLF)或修改逻辑文件(CHGLF)命令中的 MAINT 参数的规定来维护这个文件。DYNSLT 关键字不影响关键字顺序文件的访问路径维护。

对于所有规定 DYNSLT 关键字的单格式逻辑文件, 不需要为索引字段规定选择/省略字段。但对所有规定 DYNSLT 关键字的多格式逻辑文件, 不必须至少规定一个索引字段, 可以为这

个字段指定\*NONE。

当要用选择或省略字段而且下面的任一为真时，必须使用 DYNSLT 键字：

- 逻辑文件使用到达顺序（没指定索引字段），见图 2-37。
- 逻辑文件是一个规定了 JDFTVAL 键字的连接逻辑文件。
- 逻辑文件是一个连接逻辑文件且选择/省略字段来自多个物理文件，并且下面之一为真：

—选择/省略字段在同一选择或省略语句中见图 2-39。

—选择/省略字段选择和省略混合的语句中，见图 2-40。

—选择/省略字段在被“或”在一起的选择语句上。

—选择/省略字段在被“与”在一起的省略语句上。

不能与 REFACCPTH 键字一起指定 DYNSLT 键字。

对于连接逻辑文件，选择/省略字段可以出现在任意一个在 JFILE 键上指定的物理文件中。在连接逻辑文件中使用 JFILE 键字来指出字段的来源并避免二意性。

下面例子给出如何指定 DYNSLT 键字。

图 2-37 显示如何在到达顺序中指定动态选择

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                      DYNSLT  
00020A      R RECORD1          PFILE(PF1)  
00030A      FLD1  
00040A      FLD2  
00050A      S FLD1           COMP(GT 2)
```

图 2-37 规定 DYNSLT 键字（例 1）

由于没有索引字段，所以要用 DYNSLT 键字。

逻辑文件以到达顺序为程序提供记录。假定物理文件 PF1 有如下记录：

FLD1	FLD2
1	aaaa
2	dddd
3	jjjj
4	bbbb

当程序做一个输入操作时，系统根据选择/省略值检测前两个记录，但没有一个符合，不能传给程序。程序仅可以得到后两个记录：

FLD1	FLD2
3	jjjj
4	bbbb

图 2-38 给出如何用键字顺序访问路径指定动态选择。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               DYNSLT  
00020A      R RECORD1                PFILE(PF1)  
00030A      FLD1  
00040A      FLD2  
00050A      K FLD1  
00060A      S FLD2                COMP(GT 'bbbb')  
A
```

图 2-38 规定 DYNSLT 键字（例 2）

在图 2-38 中，DYNSLT 键字不是必需要的。逻辑文件以键字顺序为程序提供记录。假定物理文件有下面这些记录：

FLD1	FLD2
1	aaaa
2	dddd
3	jjjj
4	bbbb

当程序请求一个记录时，系统根据选择/省略值检测记录的 FLD2 值。程序可以看到下面的记录：

FLD1	FLD2
2	dddd
3	jjjj

图 2-39 给出如何指定来自两个物理文件，有选择/省略规定的连接逻辑文件。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               DYNSLT  
00020A      R RECORD1                JFILE(PF1 PF2)  
00030A      J                   JFLD(FLD1 FLD3)  
00040A      FLD1                 JREF(PF1)  
00050A      FLD2                 JREF(PF1)  
00060A      FLD3                 JREF(PF2)  
00070A      FLD4                 JREF(PF2)  
00080A      S FLD1               COMP(GT FLD4)  
A
```

图 2-39 规定 DYNSLT 键字（例 3）

FLD1 和 FLD2 来自主文件 (PF1)，而 FLD3 和 FLD4 来自次文件 (PF2)。选择规定来自主文件的 FLD1 与来自次文件的 FLD4 作比较。因此，需要 DYNSLT 键字。图 2-40 给出如何指定来自多个物理文件的有选择和省略字段的连接逻辑文件。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8	
00010A	DYNSLT
00020A R JREC	JFILE(PF1 PF2)
00030A J	JOIN(PF1 PF2)
00040A	JFLD(FLD1 FLD2)
00050A FLD1	JREF(PF1)
00060A FLD2	JREF(PF1)
00070A FLD3	JREF(PF2)
00080A K FLD1	
00090A S FLD1	COMP(GT 0)
00100A O FLD3	COMP(GT 4)
A	

图 2-40 规定 DYNSLT 键字 (例 4)

FLD1 和 FLD3 来自不同的物理文件，并且有混合的选择和省略语句。因此要用 DYNSLT 键字。

#### 2.4.19 EDTCDE (编辑码) 和 EDTWRD (编辑字)

用这个字段层键字来编辑显示或打印文件。引用字段的 EDTCDE 和 EDTWRD 键字时物理文件或逻辑文件没有影响。

EDTCDE 键字格式是：

EDTCDE (编辑码[\*！浮动货币符号])

EDTWRD 键字格式是：

EDTWRD ('编辑字')

当定义一个显示文件输入字段时，引用字段要在 29 列指定 R 并使用 REF 或 FEFFLD 键字来引用，在生成显示文件时，OS/400 程序从物理文件或逻辑文件字段复制 EDTCDE 和 EDTWRD 及其它属性到显示文件字段中。可用显示和打印文件中的新编辑键字来替换原来指定的编辑，通过指定 DLTEDT 键字来清除所有的编辑属性。在打印或显示文件中删除所有字段编辑要指定 DLTEDT 键字。详细信息见 3.2.9 “引用” (第 29 列)。

不能对浮点字段 (35 列为 F) 和十六进制字段 (35 列为 H) 指定 EDTCDE 或 EDTWRD 键字。不能对日期、时间或时间标识字段 (35 列 L、T 或 Z) 指定 EDTCDE 或 EDTWRD 键字。

在物理文件或逻辑文件中指定这些键字的规则与显示文件是相同的。详细信息见 3.3.44 及 3.3.46。

图 2-41 给出如何在物理文件中指定 EDTCDE 和 EDTWRD 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECORD
A
A          PRICE      5  2      EDTCDE(J)
A
A          SALES     7  2      EDTCDE(K $)
A
A          SALARY     8  2      EDTCDE(1 *)
A
A          BALANCE    7  2      EDTWRD(' $    0.  &CR')
A
A          DATE       6  0      EDTCDE(Y)
A
```

图 2-41 规定 EDTCDE 键字

PRICE、SALES、SALARY 和 DATE 字段指定了编辑属性。当由显示或打印文件引用这些字段时，没有指定新的编辑，引用这些字段的应用程序使用这些字段的标准编辑属性。

#### 2.4.20 FCFO（先修改先输出）

使用这个文件层键字指定如果是来自同一个物理文件或逻辑文件的记录有重复键值时，那么先检索先被修改的键值的记录，即以先修改先输出的顺序检索。

这个键字没有参数。

先修改先输出（FCFO）不允许与 FIFO、LIFO、UNIQUE 或 REFACCPTH 键字一起用。

如果不指定 FCFO、LIFO、FIFO 或 UNIQUE，具有重复键值的记录是以 FIFO（先入先出）、FIFO（后入先出）或 FCFO（先修改先出）顺序检索的，但哪个顺序并非一定。

对于 FCFO 键字，是依记录键字值改变的先后排序。对于 FIFO 和 LIFO，记录是由相关记录号是排序的。

在规定 FCFO 键字的文件中至少要有一个键字段。当在生成物理文件（CRTPF）或生成逻辑文件（CRTLTF）命令中指定 FILETYPE(\*SRC) 时，FCFO 键字是无效的。

图 2-42 给出如何在物理文件中指定 FCFO 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               FCFO
00020A          R CUSREC           TEXT(' CUSTOMER RECORD' )
00030A          CUSNAMEF        10A
00040A          CUSNAMEM        1A
00050A          CUSNAMEL        10A
```

```
00060A      K CUSNAMEL  
A
```

图 2-42 规定 FCOF 键字

#### 2.4.21 FIFO (先进先出)

使用这个文件层键字来指定对来自同一物理文件或逻辑文件成员中有重复键值的记录则按先进先出的顺序来检索它们。

这个键字没有参数。

FIFO 不允许与 LIFO、UNIQUE 或 REFACCPTH 一同使用。

如果没有指定 FCFO、LIFO、FIFO 和 UNIQUE 键字，那么检索有重复键值记录的顺序既可能是先进先出的，也可能是后进先出或先修改先出，但无法确定这些记录检索的顺序。

有 FIFO 键字的文件中至少要指定一个键字字段。在生成物理文件 (CRTPF) 或生成逻辑文件 (CRTLTF) 命令上指定了 FILETYPE(\*SRC) 时，FIFO 键字是无效的。

图 2-43 给出了如何为一个物理文件指定 FIFO 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A          FIFO  
00020A      R CUSREC           TEXT (' CUSTOMER RECORD ')  
00030A      CUSNAMEF        10A  
00040A      CUSNAMEM        1A  
00050A      CUSNAMEL        10A  
00060A      K CUSNAMEL  
A
```

图 2-43 规定 FIFO 键字

#### 2.4.22 FLTPCN (浮点精度)

使用这个字段层键字来指定一个浮点字段的精度。

这个键字格式如下：

FLTPCN (\*SINGLE | \*DOUBLE)

SINGLE 是单精度，DOUBLE 是双精度，这个键字仅对浮点字段（数据类型为 F）有效。

如果没指定 FLTPCN 键字，那么浮点字段的缺省值是单精度。单精度字段最多有 9 位数字，双精度字段有 17 位数字，如果指定的单精度字段长度大于 9 位或双精度字段长度大于 17 位，系统将送出一个错误信息。

图 2-44 给出如何指定 FLTPCN 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00090A      FIELDA        17F 4      FLTPCN(*DOUBLE)  
A
```

图 2-44 规定 FLTPCN 键字

FIELDA 是一个双精度浮点字段。

#### 2.4.23 FORMAT (格式)

用这个记录层键字来指定记录格式共享一个预先定义的记录格式的字段。定义的记录格式名必须与预先定义的记录格式名相同。

这个键字格式是：

FORMAT ([库名/]数据文件名)

数据文件名是必须的，它是预先定义的物理或逻辑文件的名字。

库名是可选的，如果没指定库名，那么用在生成文件时使用的库列表 (\*LIBL)。

如果指定了 FORMAT 键字，则不能对这个记录格式指定字段规范。如果需要可以指定键字段和选择/省略规范。(它们与以前定义的记录格式可以相同也可以不同)。

在连接逻辑文件中，不允许 FORMAT 键字同时也不能指定连接逻辑文件名作为 FORMAT 键字的参数。

如果使用的数据库文件的记录格式被删除，那么只要某些文件还使用这个记录格式它就仍然保持存在。例如：在 FILE2 中的 RECORD 使用 FORMAT 键字来共享在 FILE1 中的 RECORD 规范。两个文件都生成了，如果 FILE1 被删除，然后用不同的 DDS 对其进行重建，RECORD 仍然存在于 FILE2 中，它可以由用 FORMAT 键字的其它文件引用。

在这个键字中不能使用 DDM 文件。

图 2-45 显示了如何为一个逻辑文件指定 FORMAT 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD          PFILE(FILE2)  
00020A          FORMAT(FILE1)  
A
```

图 2-45 规定 FORMAT 键字

这个逻辑文件的记录格式与预生成的文件 FILE1 中的记录格式相同。记录格式名 (RECORD) 必须与在 FILE1 中的记录格式名相同。

#### 2.4.24 JDFTVAL (连接的缺省值) ——仅用于逻辑文件

在连接逻辑文件中使用这个文件层键字，这样当连接次文件中没有相应记录时，系统为这个字段提供一个缺省值。JDFTVAL 仅对连接逻辑文件有效。

这个键字没有参数。

字符型、十六进制型字段，系统的缺省值是空白，数字字段，缺省值为 0。在物理文件中的字段可以指定 DFT 键字来修改字段的缺省值。(见 2.4.16)。

如果指定了 JDFTVAL，程序检索在次文件中没有相应值的记录。如果没有指定

DFTVAL，则跳过在次文件中没有相应记录的主文件记录。

如果连接 3 个或更多个文件，并且对被用作连接的那些字段指定了 JDFTVAL 键字，则使用次文件中没有字段的缺省值。记录是根据缺省值来选择和省略的，即如果这个字段作为一个连接字段来连接其它次文件，那么来自其它次文件的记录根据这个缺省值送给程序。

图 2-46 显示了如何指定 JDFTVAL 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               JDFTVAL  
00020A      R RECORD1                JFILE(PF1 PF2)  
00030A      J                      JOIN(PF1 PF2)  
00040A                           JFLD(NAME NAME)  
00050A      NAME                  JREF(1)  
00060A      ADDR  
00070A      BAL  
A
```

图 2-46 规定 JDFTAVL 键字

PF1 是主文件，PF2 是次文件。假定 PF1 和 PF2 有如下的记录：

PF1		PF2	
NAME	ADDR	NAME	BAL
Anne	120 1st St.	Anne	5.00
Doug	40 Pillsbury	Doug	6.50
Mark	2 Lakeside Dr.	Sue	2.00
Sue	120 Broadway		

在连接逻辑文件中指定了 JDFTVAL，程序读到下面的记录（按出现顺序显示）。

NAME	ADDR	BAL
Anne	120 1st St.	5.00
Doug	40 Pillsbury	6.50
Mark	2 Lakeside Dr.	0.00
Sue	120 Broadway	2.00

在连接逻辑文件中没有规定 JDFTVAL，程序仅可以读到 3 个记录（对于 Mark 没有记录）。在这个例子中，如果指定了 JREF(2)代替 JREF(1)，那么送给程序中的记录是与上面不同，而是象下面那样：

NAME	ADDR	BAL
Anne	120 1st St.	5.00
Doug	40 Pillsbury	6.50

	2 Lakeside Dr.	0.00
Sue	120 Broadway	2.00

#### 2.4.25 JDUPSEQ (连接的重复顺序) ——仅用于连接逻辑文件

使用这个连接字段的键字来指定当程序读一个连接逻辑文件时, 为有重复连接字段的记录给出顺序。

这个键字的格式是:

JDUPSEQ (排序字段名 [\*DESCEND])

这个键字不影响那些唯一记录的排序顺序。如果没指定这个键字, 系统对那些有重复连接字段的记录顺序是没保证的。

如果在一个连结规范中指定了多个 JDUPSEQ 键字, 那么 JDUPSEQ 键字的顺序决定重复记录的出现顺序。这与指定一个附加的键字段是相似的, 以它来决定带有重复键值记录的顺序。

这个键字仅对连接逻辑文件有效。

在一个连接逻辑文件规范中, JFLD 键字和 JDUPSEQ 键字中指定的字段的总长度不得超过 120 个字节。

排序字段名必须是: (1)存在于连接规范的(to)文件中, (2)不在连接规范的 JFLD 键字上指定为 (TO) 字段, 这个排序字段名可以是一个 CONCAT 字段或是一个 SST 字段。对于这个连接逻辑文件, 排序字段名不是必须在记录格式中规定的。

另外, 可以指定\*DESCEND 来修改重复记录的顺序。没有\*DESCEND, 按下面的缺省顺序提供重复记录:

- 对数字字段按有符号升序。
- 对字符字段按升序。

图 2-47 和 2-48 给出如何指定 JDUPSEQ 字段。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JREC                  JFILE(PF1 PF2)
00020A      J                      JOIN(PF1 PF2)
00030A      JFLD(NAME1 NAME2)
00040A      JDUPSEQ(PHONE)
00050A      NAME1
00060A      ADDR
00070A      PHONE
```

图 2-47 规定 JDUPSEQ 键字 (例 1)

假定 PF1 和 PF2 中有如下记录:

PF1		PF2	
NAME1	ADDR	NAME2	TELEPHONE

Anne	120 1st St.	Anne	555-1111
Doug	40 Pillsbury	Anne	555-6666
Mark	2 Lakeside Dr.	Anne	555-2222
		Doug	555-5555

在 PF2 中对 Anne 有 3 个记录，显示了 3 个电话号，由于指定了 JDUPSEQ，记录是象下面这样返回到程序的：

NAME	ADDR	TELEPHONE
Anne	120 1st St.	555-1111
Anne	120 1st St.	555-2222
Anne	120 1st St.	555-6666
Doug	40 Pillsbury	555-5555

JDUPSEQ 仅当有重复记录存在时才有影响。

图 2-48 中的逻辑文件与图 2-47 依据相同的物理文件，在 PF2 中对 Anne 有 3 个记录，显示了 3 个电话号码。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JREC          JFILE(PF1 PF2)
00020A      J              JOIN(PF1 PF2)
00030A          JFLD(NAME1 NAME2)
00040A          JDUPSEQ(PHONE *DESCEND)
00050A      NAME1
00060A      ADDR
00070A      PHONE
A

```

图 2-48 规定 JDUPSEQ 键字（例 2）

当指定了 JDUPSEQ(\*DESCEND)时，记录是象下面那样返回的：

NAME1	ADDR	TELEPHONE
Anne	120 1st St.	555-6666
Anne	120 1st St.	555-2222
Anne	120 1st St.	555-1111
Doug	40 Pillsbury	555-5555

在这个表中 Anne 的电话号码是降序的。

#### 2.4.26 JFILE（被连接的文件）——仅用于连接逻辑文件

使用这个记录层键字来指明连接逻辑文件访问数据所依据的物理文件。

这个关键字格式是：

JFILE ([库名/]物理文件名 [.....32])

这个关键字除了指明文件作为一个连接逻辑文件外与 PFILE 键字是相似的。JFILE 键字不允许与 PFILE 键字同时使用。

在连接逻辑文件中的记录层必须有 JFILE 键字。在一个 JFILE 键字中最少要有两个物理文件名，可以多次的指定同一个文件名。

第一个文件称为主文件，连接将从这个文件开始，所有其它的文件称为次文件，且最多可有 31 个（在 JFILE 键字上总共可有 32 个文件）。

在远程系统上建立逻辑文件时，在 JFILE 键字上允许给出分布式数据管理（DDM）文件。详细的信息见 DDM 用户指南。

下面适用于在 JFILE 键字上指定物理文件:

- 如果几个物理文件的记录个数不同，那么在 JFILE 键字中靠左侧指定较少记录的物理文件。主文件的记录数应该少于或等于次文件的记录数。这样，当读文件时可以改善性能。
  - 在连接规范中指定的主文件和次文件必须有特定的顺序。这个顺序取决于在 JFILE 键字中指定的顺序。见图 2-55。
  - JOIN 和 JREF 可以使用由 JFILE 键字中确定的相关文件号。在 JFILE 键字中的第一个文件的相关文件号是 1，第 2 个文件的相关文件号是 2，直到 32。如果使用相关文件号来代替 JOIN 和 JREF 键字上的文件名，那么在 JFILE 键字上文件的顺序能够影响在 JOIN 和 JREF 键上指定的路径。

注：如果物理文件的名字不唯一，那么必须指定相关文件号。

图 2-49 和图 2-50 给出如何指定 JFILE 键字。

图 2-49 规定 JFILE 键字 (例 1)

在连接逻辑文件中，PF1 是主文件，PF2 次文件。

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A R JREC JFILE (MYLIBA/PHYSICAL1 +  
00020A MYLIBB/PHYSICAL2 MYLIBC/PHYSICAL3)  
00030A J JOIN(1 2)  
00040A JFLD(FIELD1 FIELD2)  
00050A J JOIN(1 3)  
00060A JFLD(FIELD1 FIELD2)

图 2-50 规定 JFILE 键字 (例 2)

在连接逻辑文件中，MYLIBA 库中的 PHYSICAL1 是主文件，在 MYLIBB 中的文件 PHYSICAL2 和在 MYLIBC 库中的文件 PHYSICAL3 是次文件。

#### 2.4.27 JFLD (被连接的字段) ——仅用于连接逻辑文件

使用这个连接层关键字来指明一个逻辑文件中，用来连接物理文件的 From 字段和 To 字段。这些字段都是作为连接字段来引用的。

这个关键字的格式是：

JFLD (From—字段名 To—字段名)

连接字段必须与作为连接规范的 JOIN 关键字上指明的物理文件中的字段名一致。在 JFLD 关键字上指定的名必须与在物理文件中的名相同，除非在连接逻辑文件中对其进行了重命名。如果没指定 JOIN 关键字，则使用 JFILE 关键字。

这个关键字仅对连接逻辑文件有效。

对于每个连接规范至少要有一个 JFLD 关键字，一个连接规范是在 17 列的 J 指明的。由于在一个连接逻辑文件中至少要有一个连接规范，所以对每一个连接逻辑文件至少要有一个 JFLD 关键字。

这些字段不必指定作为连接逻辑文件记录格式中的一个字段。

为了在连接物理文件时指定另外的连接字段，要指定多个 JFLD 关键字。

在 JFLD 关键字上指定的字段名必须或者是在连接记录格式中字段层指定了的名字，或者是在 JFILE 关键字上指定的一个物理文件中的字段名之一。

OS/400 使用下面的检索顺序来匹配使用的连接字段：

1. 在连接逻辑文件的字段层，19-28 列上指定的字段。

注：那些指定了 CONCAT、RENAME 或 SST 关键字的字段可作为连接字段。CONCAT、RENAME 或 SST 关键字中做参数指定的字段不能作为连接字段。

2. 在 JOIN 关键字中指定的物理文件中的字段。

指定连接字段的规则如下：

- from 字段必须能在 JOIN 关键字上指定的 from 文件中找到。
- To 字段必须能在 JOIN 关键字上指定的 to 文件中找到。
- 连接字段不必在连接记录格式中定义。
- from 和 to 字段必须具有相同的属性（长度、数据类型和小数位），但不必有相同的名字。当在物理文件中连接字段具有不同的定义时，必须重定义其中的一个或两个都重定义。如果重定义这些字段，则可能发生数据转换错。见 2.3.9“长度”及 2.3.10、2.3.12 的说明。

注：字符字段无须有同样长度，短的连接字段用空格填充使它与长的连接字段长度相同。

在简单连接规范中，JFLD 关键字中指定的 to 字段和在 JDUPSEQ 关键字中指定的字段，总长度最多为 120 个字节。

图 2-51 和图 2-52 给出如何指定 JFLD 关键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JREC          JFILE(PF1 PF2)
00020A      J              JOIN(PF1 PF2)
00030A          JFLD(NAME1 NAME2)
A

```

图 2-51 规定 JFLD 键字（例 1）

在连接逻辑文件中，JFLD 键字指定了在物理文件 PF1 中的 NAME1 与物理文件 PF2 中的 NAME2 连接。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R JREC          JFILE(PF1 PF2)
00020A      J              JOIN(PF1 PF2)
00030A          JFLD(NAME1 NAME2)
00040A          JFLD(ADDR1 ADDR2)
A

```

图 2-52 规定 JFLD 键字（例 2）

在连接逻辑文件中，JFLD 键字指定物理文件 PF1 中的 NAME1 和 ADDR1 连接物理文件 PF2 中的 NAME2 和 ADDR2。

#### 2.4.28 JOIN（连接）——仅用于连接逻辑文件

使用这个连接层键字来指明哪一对文件由连接规范来做连接。

这个键字的格式是：

JOIN (from 文件 to 文件)

这个键字仅对连接逻辑文件有效。

可以使用文件名或相关文件号来指明连接哪些文件。如果在 JFILE 键字中多次指定了同一个文件，那么必须指定相关文件号。

如果指定文件名，那么必须选择在 JFILE 键字上只指定了一次的文件。在每个 JFILE 键字上，from 文件必须在 to 文件之前出现。

如果指定文件号，那么要与在 JFILE 键字上指定的文件相对应。下面是有效值：

文件	有效值
from 文件号	1-31
to 文件号	2-32

from 文件号必须总是小于 to 文件号。

对 from 文件和 to 文件顺序的特别规则，详细信息见 2.4.28.1 中的图 2-55。

在一个连接逻辑文件中，每个次文件仅作 to 文件一次。

#### 2.4.28.1 连接规范

要描述一个连接规范应做下面的事：

- 在紧跟记录层之后的 17 列上指定 J (在 19-28 列的第一个字段名之前)。17 列的 J 指出一个连接规范的开始。
- 指定 JOIN 键字。当在 JFILE 键字中仅指定两个文件时，JOIN 键字是可选的，在 JFILE 键字上指定了多于两个文件时，对于每一个次文件要求有一个 JOIN 键字。
- 对每一个连接规范至少要指定一次 JFLD 键字。
- 有另外一个 17 列上的 J 或 19-28 位上出现字段名，标志这个连接规范结束。

对在 JFILE 键字上指定的每个次文件必须有一个连接规范，因此，在一个连接逻辑文件中要求至少有一个连接的规范。

在一个连接规范内，只能指定一次 JOIN 键字。

图 2-53，图 2-54，图 2-55 给出如何指定 JOIN 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1          JFILE(PFA PFB PFC)  
00020A      J                  JOIN(PFA PFB)  
00030A          JFLD(NAME1 NAME2)  
00040A      J                  JOIN(PFA PFC)  
00050A          JFLD(NAME1 NAME3)  
00060A      NAME1  
A
```

图 2-53 规定 JOIN 键字 (例 1)

图 2-53 中 PFA 与 PFB 和 PFC 连接在一起。

图 2-54 给出如何使用相关文件号指定 JOIN。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1          JFILE(PFA PFB PFC)  
00020A      J                  JOIN(1 2)  
00030A          JFLD(NAME1 NAME2)  
00040A      J                  JOIN(1 3)  
00050A          JFLD(NAME1 NAME3)  
00060A      NAME1  
A
```

图 2-54 规定 JOIN 键字 (例 2)

图 2-54 与图 2-53 是等效的。PFA 是在 JFILE 键字上指定的第一个物理文件，其相关文件号是 1。PFB 和 PFC 分别是在 JFILE 键字上指定的第二和第三个文件，它们的相关号分别是 2 和 3。

图 2-55 给出了物理文件连接的顺序。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R J3          JFILE(VENDORS PARTS PARTWARE +  
00020A                  WAREHOUSE  
                          1  
00030A      J          JOIN(1   2)    2  
00040A                  JFLD(VNBR VNUM)  
00050A      J          JOIN(2   3)    3  
00060A                  JFLD(PNBR PNBR)  
00070A      J          JOIN(3   4)    3  
00080A                  JFLD(WNBR WNBR)  
00090A      VNAME  
00100A      VAD1  
00110A      VAD2  
00120A      PNBR          JREF(2)  
00130A      WNBR          JREF(4)  
00140A      BIN  
00150A      QOH  
A
```

图 2-55 规定 JOIN 键字 (例 3)

图中连接逻辑文件是以 4 个物理文件为基础的。其中在 **JFILE** 键字上指定的第一个文件 **VENDDRS** 是主文件，并且其相关文件号是 1，而 **PARTS**、**PARTWARE** 和 **WAREHOUSE** 是次文件，它们的相关文件号分别用 2、3 和 4。

注：在 **JOIN** 键字上指定号的方式为：

1. 在第一个 **JOIN** 键字上的第一个参数（第一个 From 文件）必须是主文件。
2. 在 **JOIN** 键字上的第二个参数（To 文件）必须与在 **JFILE** 键字上次文件的顺序相同。

如果用文件名而不用相关文件号，那么要象下面那样指定顺序：

```
J      JOIN(VENDORS PARTS)  
J      JOIN(PARTS PARTWARE)  
J      JOIN(PARTWARE WAREHOUSE)
```

3. 在每一个 **JOIN** 键字上 from 和 to 文件必须是按升序来指定的。

注：一个文件可以多次的被指定作为 from 文件。

例如：上面 **JOIN** 键字的参数还可以象下面这样来指定：

```
J      JOIN(1 2)  
J      JOIN(2 3)  
J      JOIN(2 4)
```

但一个文件只可一次被指定作为 to 文件。

#### 2.4.29 JREF (连接引用) ——仅用于连接逻辑文件

在连接逻辑文件中, 如果使用的字段名在多个物理文件中存在, 用这个字段层关键字指明在哪个物理文件有你要使用的字段。

这个关键字的格式是:

JREF (文件名 | 相关文件号)

既可以指定文件名, 亦可以指定相关文件号, 如果在 JFILE 键字上两次使用了同一个物理文件名, 那么必须指定相关文件号。相关文件号对应于在 JFILE 键字上指定的物理文件名。例如: 指定 JREF(1) 将一个字段与在 JFILE 上指定的第一个物理文件联系在一起, 指定 JREF(2) 则是将一个字段与在 JFILE 键字上指定的第二个物理文件联系在一志。见图 2-57。

这个关键字仅对连接逻辑文件有效。

连接逻辑文件是依据两个或更多 (最多 32 个) 的物理文件。在连接逻辑文件记录格式中指定的字段名必须是唯一的对应这个逻辑文件依据的物理文件的某一个字段。例如: 如果连接逻辑文件引用两个物理文件, 且每个物理文件都有一个名为 NAME 的字段, 那么必须规定 JREF 键字指出字段来自哪一个物理文件。

当字段名在 JFILE 键字中给出的物理文件中是唯一的, JREF 键字是可选的。例如一个连接逻辑文件是由两个物理文件连接而成, 且仅有一个物理文件中有字段名 NAME1, 那么就无需指定 JREF 键字。

如果一个连接逻辑文件是仅用一个物理文件连接而成的 (在 JFILE 键字中同名文件出现两次), 那么必须为每一个字段指定 JREF 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R JOINREC          JFILE(PFA PFB PFC)  
00020A      :  
00030A      :  
00040A      :  
00050A      NAME            JREF (PFB)  
A
```

图 2-56 规定 JREF 键字 (例 1)

在图 2-56 中, NAME 出现在 PFA 和 PFB 两个物理文件中, 指定 JREF(PFB) 表明用 PFB 中的这个字段。

图 2-57 给出如何用相关文件号的 JREF

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R JOINREC          JFILE(PFA PFB PFC)  
00020A      :  
00030A      :  
00040A      :  
00050A      NAME            JREF (2)
```

A

图 2-57 规定 JREF 键字 (例 2)

图 2-57 与 2-56 是等效的。在图 2-57 中, NAME 出现在 PFA 和 PFB 这两个物理文件中, 指定 JREF(2)表明使用 PFB 中的字段来连接(PFB 是在 JREF 键字上指定的第二个物理文件)。

#### 2.4.30 LIFO (后进先出)

使用这个文件层键字来指定如果从同一个物理文件成员中检索到了带有重复键值的记录, 那么对它们是按后进先出 (LIFO) 的顺序来检索的。

这个键字没有参数。

LIFO 不允许与 FCFO、FIFO、UNIQUE 或 REFACCPTH 键字同时使用。

如果未指定 FCFO、LIFO, 也未指定 FIFO、UNIQUE, 那么有重复键值的记录或是按先入先出或是按后入先出或先修改先出的顺序被检索的, 但是它们被检索顺序是无法确定的。

在有 LIFO 键字的文件中至少要指定一个键字段。当在建立逻辑文件命令 (CRTLF) 上指定了 FILETYPE(\*SRC) 时 LIFO 键字是无效的。

图 2-58 给出如何为一个物理文件指定 LIFO 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                                LIFO  
00020A      R CUSREC                  TEXT (' CUSTOMER RECORD ')  
00030A      CUSNAMEF     10A  
00040A      CUSNAMEM     1A  
00050A      CUSNAMEL     10A  
00060A      K CUSNAMEL  
A
```

图 2-58 规定 LIFO 键字

#### 2.4.31 NOALTSEQ (无交替顺序)

使用这个键字层键字指定在文件层指定的 ALTSEQ 键字对这个键字段是不适用的。如果对键字段指定了 ABSVAL 或 SIGNED, 那么 NOALTSEQ 自动地生效, 不管使用不使用 NOALTSEQ。

这个键字没有参数。

图 2-59 给出如何指定 NOALTSEQ 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                                ALTSEQ(TABLELIB/TABLE1)  
00020A      R DSTR  
00030A      :
```

```

00040A      :
00050A      CODE      1
00060A      NAME      20
00070A      :
00080A      :
00090A      K CODE
00100A      K NAME      NOALTSEQ
A

```

图 2-59 规定 NOALTSEQ 键字

用记录格式 DSTR 的记录是按着 CODE 和 NAME 的组合键字来排序的, CODE 是用交替顺序(TABLELIB 中 ATBLE1)排序的, 而 NAME 是按 EBCDIC 顺序来排序的。NOALTSEQ 避免 NAME 字段排序被替换。

#### 2.4.32 PFILE (物理文件) ——仅用于逻辑文件

使用这个记录层键字来指明通过现在定义的逻辑文件访问哪个物理文件含有的数据。

键字的格式是:

PFILE ([库名/]物理文件名 [……32])

在单格式和多格式逻辑文件中的每一个记录格式, 都要有一个 PFILE 键字。这个键字除它定义的是单格式逻辑文件和多格式逻辑文件之外, 与 JFILE 是相似的, PFILE 键字不允许与 JFILE 同时使用。在一个逻辑文件中的 PFILE 键字中最多可以指定 32 个物理文件名。如果使用了最大值, 那么既可以在一个记录格式 (使用 PFILE 键字) 中指定 32 个物理文件名, 也可将 32 个物理文件名分别在 32 个记录格式中使用, 或者这些文件名可不规则的分布在一些记录格式中。在任何情况下, 物理文件名的最多个数都是 32 个。建立一个逻辑文件时对物理文件个数的限制, 见相应的高级语言指南。

对每一个物理文件, 库名是可选的。如果省略了库名, 那么使用在文件建立时的库列表 (\*LIBL)。

在多格式逻辑文件中, 如果对一个记录指定了多个物理文件名, 那么逻辑文件记录格式中的所有字段必须存在于所有指定的物理文件中。文件的类型不能在 RPG 中外部描述, 这是因为会导致重复的格式名。如果程序要访问 PFILE 键字上指定的一个或多个物理文件的字段, 而不是全部, 则可以做下面中的一个:

- 做一个连接逻辑文件。如果这样做, 用 JFILE 键字代替 PFILE 键字。
- 做一个包含不在第一个物理文件字段的另外一个逻辑文件记录格式。

例如: 如果 FLD1 和 FLD2 在物理文件 PF1、PF2 和 PF3 中, FLD3 仅出现在 PF3 中, 则不能在以 PF1、PF2 为依据的逻辑文件记录格式中指定 FLD3, 为了实现对 FLD3 的访问, 可以做一个包含 FLD3 的另一个逻辑文件记录格式, 也可以使用一个连接逻辑文件。

- 不能使用单格式或多格式逻辑文件将来自不同物理文件的字段组合到一个记录格式

中，使用连接逻辑文件可以达到这个目的。通过使用单格式或多格式逻辑文件中的一个记录格式读记录仅可以得到来自一个物理文件的数据，通过使用逻辑文件一个记录格式写记录时也仅能写到一个物理文件中。

在远程系统上建立的逻辑文件，可以在 PFILE 键字上使用分布数据管理（DDM）文件。详细信息见 DDM 用户的指南。

图 2-60，图 2-61，图 2-62 给出如何指定 PFILE 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R LOGRCD1          PFILE(PF1)  
A
```

图 2-60 规定 PFILE 键字（例 1）

在图 2-60 中，LOGRCD1 可以使用在 PF1 中的字段。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R LOGRCD2          PFILE(PF1 PF2)  
A           :  
A           :  
00020A      R LOGRCD3          PFILE(PF1 PF2 PF3)  
A           :  
A           :  
A
```

图 2-61 规定 PFILE 键字（例 2）

在图 2-61 中，LOGRCD2 必须使用 PF1 和 PF2 所共有的字段，而 LOGRCD3 则必须使用 PF1，PF2，PF3 所共有的字段。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R LOGRCD4          PFILE(PF1)  
A           :  
A           :  
00020A      R LOGRCD5          PFILE(PF2)  
A           :  
A           :  
00030A      R LOGRCD6          PFILE(LIB1/PF6)  
A
```

图 2-62 规定 PFILE 键字（例 3）

在图 2-63 中，LOGRCD4，LOGRCD5 和 LOGRCD6 有唯一字段，LOGRCD6 指定了一

个限定的物理文件名。

#### 2.4.33 RANGE (范围)

在字段层、选择/省略层或在这两个层上指定这个键字。

这个键字格式是：

RANGE (低值 高值)

##### 2.4.33.1 在字段层指定 RANGE

在字段层，这个键字指定当后来显示文件引用这个字段时，对其进行有效性检验。

RANGE 不影响定义的物理和逻辑文件。在显示文件定义一个可输入字段时，可以通过在 29 列上指定 R 和规定 REF 或 REFFLD 键字引用现在定义的字段。显示文件建立期间，OS/400 把来自这个逻辑文件或物理文件中字段的 RANGE 键字和其它的字段属性复制到显示文件的这个字段中。可以通过显示文件的字段指定有效的检验键字来替换 RANGE 键字（如同其它有效检验键字和 CHKMSGZD 键字一样）。详细信息见 3.2.9 的引用（第 29 列）。

在逻辑文件或物理文件中指定这个键字的规则与显示文件的规则相同。详细信息见 3.3.112 (RANGE 范围)。

在浮点字段（35 列为 F）或十六进制字段（35 列为 H）不能指定 RANGE 键字。不要在日期、时间、时间标记（L、T 或 Z 第 35 位）字段指定 RANGE 键字。

##### 2.4.33.2 在选择/省略层指定 RANGE

在选择/省略字段层，当程序使用这个字段的记录格式做输入操作时，按这个键字的范围从物理文件中选择或省略检索到的记录。

下面的规则是适用的：

- 如果定义字符字段，必须指定字符串或十六进制字符串。字符串要在两个单引号之间，见图 2-63。

用一个 X 之后跟单引号括起来的数字 0-9，字母 A-F 的组合规定十六进制字符串。

在单引号中十六进制数字的数目必须是指定字段长度的两倍，见图 2-64。

- 如果定义一个数字字段，必须指定一个数值（数字 0-9，不带单引号），见图 2-63。
- 如果定义一个日期字段，用 DATFMT 键字指定的格式，用在 DATSEP 键字中指定的分隔符。例如，如果为 DATFMT 指定了\*MDY，DATSEP 指定了 ‘/’，那么 RANGE（‘12/15/91’ ‘12/31/91’）是缺省值。
- 如果定义时间字段，用 TIMFMT 键字规定的格式和 TIMSEP 键字规定的分隔符。例如，如果 TIMFMT 为\*ISO，那么 RANGE( ‘11.00.00’ ‘12.00.00’ ) 是缺省值。\*IAO 的缺省分隔符是隔点(.)。
- 如果定义时间标记字段，必须按下列格式指定缺省值：

RANGE('YYYY-MM-DD-HH.MM.SS.UUUUUU' 'YYYY-MM-DD-HH.MM.SS.UUUUUU')。

图 2-63 给出如何为 RANGE 键字指定字符串和数字串。

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

00010A	R RECORD	PFILE (PF1)
A		
00020A	FIELD A	1 0 RANGE (2 5) 1
00030A	FIELD B	1 RANGE ('2' '5')
00040A	FIELD C	
00050A	K FIELD D	
00060A	S FIELD A	RANGE (1 4) 2
A		

图 2-63 规定 RANGE 键字 (例 1)

在图 2-63 中, RANGE (1) 对引用 FIELD A 和 FIELD B 的显示文件, 对这两个字段做有效性检验。对用这个显示文件的用户, RANGE 限制对 FIELD A 或 FIELD B, 仅可以进入 2、3、4 或 5, FIELD A 是数字段而 FIELD B 是字符字段。指定的字段类型依赖于写程序的高级语言。

RABGE (2) 规定 FIELD A 的选择/省略键字, 通过这个逻辑文件检索到物理 PF1 中。FIELD A 的值是 1、2、3 或 4 的记录。

图 2-64 RANGE 键字使用十六进制字符串。

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A R RCD1 PFILE (PF1)
00020A CODEA
00030A FLD1
00040A FLD2
00050A K FLD1
00060A S CODEA RANGE (X'51' X'54')
A

图 2-64 规定 RANGE 键字 (例 2)

RANGE 作为 CODEA (一个字节的字段) 的选择/省略键字。仅当 CODEA 字段的值是从十六进制 51 到十六进制 54 之间的数时, 可检索到物理文件 PF1 的记录。

#### 2.4.34 REF (引用) ——仅用于物理文件

使用这个文件层键字指定字段描述引用的文件名。

这个键字的格式是:

REF ([库名/]数据库文件名 [记录格式名])

REF 从一个预先定义的记录格式中引用这个字段的属性。如果每个字段的描述来自同一个文件, 可以在 REF 键字中一次指定文件名, 不用几个 REFFLD 键字。要引用多个文件, 使用 REFFLD 键字。仅可以指定 REF 键字一次。

这个键字中, 文件名是必须的, 库名和记录格式是可选的。

如果没指定库名，使用建立文件时用的库列表 (\*LIBL)。如果有多个记录格式，要指定记录格式名。如果没指定记录格式名，顺序检索每个记录格式，使用第一个出现的字段名。关于如何选择 REF 和 REFFLD 键字控制这些检索的信息，见附录 A “如何指定 REF 和 REFFLD”。

在这个键字中可以指定一个分布数据管理 (DDM) 文件。

当使用一个 DDM 文件时，数据库文件名和库名是 DDM 源系统的文件名和库名。记录格式名是在目标系统中的远程文件的记录格式名。

图 2-65 和图 2-66 给出如何指定 REF 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               REF(FILE1)  
00020A       R RECORD  
00030A       FLD1      R  
A
```

图 2-65 规定 REF 键字 (例 1)

FLD1 和 FILE1 中的第一个 FLD1 (仅一个) 的属性相同。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               REF(LIB1/FILE1 RECORD2)  
00020A       R RECORD  
00030A       FLD1      R  
A
```

图 2-66 规定 REF 键字 (例 2)

FLD1 与 LIB1 库文件 FIL1 中记录格式 RECORD2 的 FLD1 属性相同。

#### 2.4.35 REFACCPATH (引用访问路径) ——仅用于逻辑文件

使用这个文件层键字来指定这个逻辑文件的访问路径信息是从另外的物理或逻辑文件中复制来的。访问路径的信息包括：键信息、选择和省略信息、交替分配顺序信息、动态选择信息和键顺序信息（用 FIFO、LIFO 和 UNIQUE 键字指定）。

这个键字的格式是：

REFACCPATH ([库名/]数据库文件名)

定义了访问路径的文件名是这个键字的参数值。

有 REFACCPATH 键字的文件不能有键字段，选择或省略字段。

在正定义的文件记录格式中可以有比这个逻辑文件依据的物理文件的字段数多或少的字段。

如果在单格式或多格式逻辑文件指定 REFACCPATH 键字，那么有 REFACCPATH 键字的

文件必须是在 PFILE 键字中以相同的顺序指定的相同物理文件。

在连接逻辑文件中不允许有 REFACCPATH 键字。仅当下面所有的项为真时，才可以把连接逻辑文件作为 REFACCPATH 键字的参数：

- 建立的文件是一个单格式逻辑文件。
- 在 PFILE 键字上指定的物理文件是连接逻辑文件中 JFILE 键字上指定的第一个文件。
- 这个连接逻辑文件指定了键字段而没指定选择和省略字段。

这个键字上不能使用分布数据管理（DDM）文件。

不能与 REFACCPATH 键字一起指定 DYNSLT、ALTSEQ、FCFO、FIFO、LIFO 或 UNIQUE 键字。

图 2-67 给出如何指定 REFACCPATH 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00030A* ORDER HEADER LOGICAL FILE (ORDHDR11)  
00040A                         REFACCPATH(DSTLIB/ORDHDRL)  
00050A      R ORDHDR           PFILE(ORDHDRV)  
A
```

图 2-67 规定 REFACCPATH 键字

#### 2.4.36 REFFLD (被引用字段) ——仅用于物理文件

使用这个字段层键字在下面三个条件之一时引用一个字段：

- 当被引用字段的名字与在 19-28 列上的名字不同时。
- 当被引用字段的名字与在 19-28 列上的名字相同，但是被引用字段的库名、文件名或记录格式名与在 REF 键字中指定的不同时。
- 当被引用的字段与引用字段出现在同一个 DDS 源文件中时。

这个键字的格式是：

REFFLD ([记录格式名/]被引用的字段名 [{\*SRC | 库名/}数据库文件名])。

即使被引用字段的名字与正在定义的字段名相同，也必须给出被引用字段的名字。当被引用文件含有多个记录格式时，使用记录格式名。当被引用字段与正在定义的字段在同一个 DDS 源文件中时使用 \*SRC (而不是数据库文件名)。当没有指定数据库文件名和库名时，\*SRC 是缺省值。

注：当引用同一个 DDS 源文件字段时，引用字段必须在定义字段之前。

如检索一个特定的数据库文件，要指定数据文件名 (如果需要也要有库名)。

在 29 列必须有 R。被引用字段的某些键字不能包括在定义字段中。详细的信息，见 2.3.8 中引用 (第 29 列)。

如果在同一个 DDS 源文件中，文件层指定了 REF，字段层指定了 REFFLD，那么使用 REFFLD 规定。查找的顺序取决于键字 REF 和 REFFLD。详细的信息，见附录 A “如何指定 REF 和 REFFLD”。

可以在这个键字中指定分布数据管理 (DDM) 文件。

当使用 DDM 文件时，数据库文件名和库名是源系统上的 DDM 文件名和库名，字段名和记录格式名是在目标系统上远程文件中的字段名和记录格式名。

IDDU 文件不能做为被引用文件。

图 2-68 给出如何指定 REFFLD 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R FMAT1
00020A      ITEM      5
00030A      ITEM1    R      REFFLD(ITEM)
00040A      ITEM2    R      REFFLD(FMAT1/ITEM)
00050A      ITEM3    R      REFFLD(ITEM FILEX)
00060A      ITEM4    R      REFFLD(ITEM LIBY(FILEX))
00070A      ITEM5    R      REFFLD(FMAT1/ITEM LIBY(FILEX))
00080A      ITEM6    R      REFFLD(ITEM *SCR)
A
```

图 2-68 规定 REFFLD 键字

由于没有指定 REF 键字，故 00030 行和 00040 行的缺省值是查找定义它们的 DDS 源文件。在 00080 行，参数\*SRC 明显的指定了这个源文件。关于各种规范的解释见附录 A “如何指定 REF 和 REFFLD”。

#### 2.4.37 REFSHIFT (引用换档)

使用这个字段层键字指定当一个字段被显示文件或 DFU 文件引用时，为这个字段指定键盘换档。

这个键字格式是：

REFSHIFT (键盘—换档)

当在显示文件中定义一个输入键字时，要引用第 29 列有 R 和规定了 REF 或 REFFLD 键字的字段。在建立显示文件时，OS/400 程序从逻辑文件字段把 REFSHIFT 键字和其它一些属性拷贝到显示文件中去。也可在显示文件或打印文件中指定新的键字来替换原有属性。也可用 DLTEDT 键字删除所有编辑字段。详细信息见 3.2.9 的引用（第 29 列）。

显示文件中的键盘换档（35 列）做为这个键字的值，来代替在数据库文件中指定的数据类型。当从物理文件或逻辑文件引用有 REFSHIFT 键字中的字段时，把 REFSHIFT 键字拷贝到一个新字段。如果为新字段指定的字段属性（例如数据类型）同 REFSHIFT 的规定不匹配，则键字无效。

这个键字对数据类型 A、S、B 和 P 的字段有效。选择与数据类型相匹配的键盘换档做参数。下面是匹配的情况：

- 字符字段 (A): REFSHIFT (A | X | W | N | I | D | M | )
- 数字字段 (S,B,P): REFSHIFT (S | Y | N | D | I | )

详细信息见 3.2.11 中“数据类型/键盘换档（第 35 位）”。

图 2-69 给出如何在物理文件中指定 REFSHIFT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD  
00020A      FIELD A      5      REFSHIFT (X)  
00030A      FIELD N      4P     REFSHIFT (N)  
A
```

图 2-69 规定 REFSHIFT 键字

文件 FILE1 中的 FIELD A 和 FIELD N 字段，有 REFSHIFT 键字。当显示文件引用这些字段时使用 REFSHIFT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                      REF(FILE1)  
00020A      R RECORD  
00030A      FIELD A      R      1 2  
A          FIELD N      R      2 2  
A
```

显示文件引用 FILE1 (REF 键字)，显示文件里的字段 FIELD A 和 FIELD N 引用 FILE1 中的 FIELD A 和 FIELD N。如果对文件 FILE1 中的字段指定 REFSHIFT 键字，那么在显示文件中使用 REFSHIFT 键盘换档，字段有下列属性：

FIELD A 在第 35 列有键盘换档 X。

FIELD N 在第 35 列有键盘换档 N。

#### 2.4.38 RENAME (重命名)

在逻辑文件记录格式中的某个字段名与它依据的物理文件字段名不同时，使用这个字段层键字。

这个键字的格式是：

RENAME (物理文件字段名)。

这个键字的参数是出现在物理文件记录格式中的名字。物理文件中的一个字段可以在记录格式中重命名成多个字段。

在类似下面的情况下，应重命名字段：

- 程序使用同一字段的不同名。
- 要把物理文件中的一个字段映象为逻辑文件中的两个或多个字段。
- 使用的语言不允许两个不同名的字段使用同一个数据存贮区（像 RPGIII）。通过指定 RENAME 键字，则允许两个字段访问相同的数据存贮区域。如果在逻辑文件的记录格式中多次指定在逻辑文件中的同一物理文件的字段（即用 RENAME 或 CONCAT），这些字段的顺序是在更新或插入操作时数据传送到物理文件中的顺序。

这样，在物理字段中最后出现的值是放在物理记录中的值并且用于建立物理字段中所有键值。字段的所有前面的值被复盖且不起作用。

下例显示了如何指定 RENAME 键字。

图 2-70 物理文件 PF1 中的 QTYDUE 字段在逻辑文件中被重命名为 QTY。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00050A      R RCD1          PFILE(PF1)  
00060A      QTY           RENAME(QTYDUE)  
A
```

图 2-70 规定 RENAME 键字（例 1）

图 2-71 在逻辑文件中重命名的字段（QTY）作为一个键字段。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00050A      R RCD2          PFILE(PF2)  
00060A      :  
A          :  
00130A      QTY           RENAME(QTYDUE)  
00140A      K QTY  
A
```

图 2-71 规定 RENAME 键字（例 2）

#### 2.4.39 SIGNED（符号）

使用这个键字层键字让 OS/400 在用数字键字段值排序时要考虑它的符号。

这个键字没有参数。

下面的例子给出了用六个区位十进制数做为键字段情况：

记录	数字键字段（区位十进制）	十六进制表示
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

由缺省值（没有指定排序键字和 ALTSEQ 键字），这个键字段具有 SIGNED 属性。这些记录是按下面的顺序来排列的：

记录	数字键字段（区位十进制）	十六进制表示
----	--------------	--------

3	98-	F9D8
2	00	F0F0
5	20	F2F0
4	97	F9F7
1	98	F9F8
6	99	F9F9

如果指定了 SIGNED 和 DESCEND，记录是按下面的顺序排列的：

记录	数字键字段（区位十进制）	十六进制表示
6	99	F9F9
1	98	F9F8
4	97	F9F7
5	20	F2F0
2	00	F0F0
3	98-	F9D8

对于字符型、日期、时间、时间标记或十六进制型数据类型字段，这个键字是无效的。它不能与 ABSVAL、UNSIGNED、ZONE 或 DIGIT 键字同时使用。

SIGNED（键字层键字）导致 ALTSEQ（文件层键字）无效，如果对一个键字段指定了 SIGNED，即使在文件层指定了 ALTSEQ 键字，NOALTSEQ 自动生效。无论有没有 NOALTSEQ 键字都这样。

图 2-72 给出如何为物理文件指定 SIGNED 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD
00020A      FLDA      7S 2
00030A      FLDB
00040A      K FLDA      SIGNED
A
```

图 2-72 规定 SIGNED 键字

#### 2.4.40 SST（子串）——仅用于逻辑文件

使用这个字段层键字指定已存在的字符、十六进制、区位字段或图形的子集。

这个键字的格式是：

SST（字段名 起始位置 [长度]）

字段名参数指出子串来自哪个字段。这个字段必须在同一逻辑文件的 SST 字段之前，或者 PFILE 或 JFILE 中的物理文件中。

为找到这个字段，系统按如下方式查找匹配字段名：

- 首先，在逻辑文件的 DDS 中 19-28 列上，在指定 SST 字段之前。

2.如果没有找到匹配字段名，系统在 PFILE 或 JFILE 中的物理文件查找，要依据如下顺序：

- 对于单格式或多格式逻辑文件，这个字段必须是 PFILE 键字所有物理文件中共有的字段。
- 如果是连接逻辑文件且在 SST 的字段上有 JREF 键字，该字段必须是在 JFILE 中 JREF 上指定的字段。
- 如果是连接逻辑文件且在 SST 字段上没有 JREF 键字，这个字段必须确切地在 JFILE 中的某一个文件中。

这个子串从 SST 键字上指定的起始位置处开始。长度做为键字的第三个参数，既可以在这个键字中指定，也可以在这个字段的长度上指定（DDS 的 30-34 列）。起始位置是一个必须的参数，而长度是可选的。

注：起始位置和长度值必须是正整数值，子串的长度必须不大于在 SST 字上指定的字段名的长度。

使用 SST 时要遵守下面的规则：

- 如果 SST 中字段是十六进制，则结果字段是十六进制，否则结果字段总是字符型的。  
如果在 DDS 中没指定数据类型，那么为其分配缺省值 A 或 H。
- 结果字段只能用做仅输入字段 (I) 或既不输入也不输出 (N) 字段。
- 结果字段的长度是可选的。必须在字段长度位置上 (DDS) 或键字中指定结果字段的长度，如果在这两处都指定了长度，那么它们必须相等。如果没指定字段长度，那么用键字中的长度参数值。
- 不能对同一字段同时指定 SST、CONCAT、RENAME 或 TRNTBL 键字。
- 这个键字指定的字段不能用 CNOCAT、RENAME、TRNTBL 或 SST 键字来定义。

图 2-73 和图 2-74 给出如何指定 SST 键字。

图 2-73 是在一个连接逻辑文件中指定 SST 键字。

....+....	1....+....	2....+....	3....+....	4....+....	5....+....	6....+....	7....+....	8
A	R	RECORD1			JFILE(PF1 PF2)			
A	J				JOIN(1 2)			
A					JFELD(CITY CITY)			
A	ADDRESS				JREF(2)			
A	CITY	I	SST(ADDRESS 21 10)					
A			JREF(2)					
A	SYEAR	I	SST(SALESDATE 5)					
A	NAME		JREF(1)					
A	CUSTNAME	I	SST(NAME 11 10) JREF(2)					
A	K SYEAR							
A								

图 2-73 在连接逻辑文件中规定 SST 键字

图 2-73 给出：

- CITY 是一个来自这个逻辑格式中 ADDRESS 的一个子串，它与 PF1 的 CITY 连接。
- CUSTNAME 是一个来自 PF2 中 NAME 的子串，这是由于 NAME 在逻辑格式中的有一个不同的 JREF。
- SYEAR 是一个键字段，在 PF1 中必须有唯一的字段名 SALESDATE。
- 有 SST 键字的字段的处理方式（38 列）必须是 I（仅输入）。由于这是一个连接逻辑文件，处理方式的缺省值是 I。

图 2-74 给出如何在一个单格式或多格式逻辑文件上指定 SST 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R REC1                  PFILE(PFA)  
A          LASTNAME             I      SST(NAME 10 10)  
A          K LASTNAME  
A
```

图 2-74 对单格式和多格式逻辑文件规定 SST 键字

LASTNAME 字段 PFA 中的 NAME 的子串，在简单或多格式逻辑文件中的 SST 字段必须在 38 列指定为 I。

#### 2.4.41 TEXT (描述)

使用这个记录层或字段层键字为程序文本提供记录格式或字段的说明（注释）。

这个键字格式是：TEXT（‘说明’）。

说明必须包括在单引号中，长度为 50 个字符，如果多于 50 个字符，那么只有前 50 个字符被高级语言使用。

注：如果对逻辑文件规定了 TEXT 键字且没有规定字段，那么使用物理文件中的 text 键字（如果有）。

图 2-75 给出如何规定 TEXT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R CUSMST            TEXT('Customer Master Record')  
00020A      FLD1                3 0      TEXT('ORDER NUMBER FIELD')  
A
```

图 2-75 规定 TEXT 键字

#### 2.4.42 TIMFMT (时间格式)

使用这个字段层键字为时间字段规定一个格式。这个键字对相应物理文件字段是时间字段或区位字段是有效的。

这个键字的格式是：

## TIMFMT (时间格式)

下表列出有效的时间格式和它们的缺省值。

格式名字	时间格式参数	时间格式和分隔符	字段长度	例 子
Hours:Minutes:Seconds	*HMS	hh:mm:ss	8	14:00:00
International Standards Organization	*ISO	hh. mm. ss	8	14. 00. 00
IBM USA Standard	*USA	hh:mm AM or hh:mm PM	8	2:00 pm
IBM European Standard	*EUR	hh. mm. ss	8	14. 00. 00
Japanese Industrial Standard Christian Era	*JIS	hh:mm:ss	8	14:00:00

- 如果对物理文件没指定 TIMFMT 键字，那么缺省值是\*ISO。
- 如果没有对逻辑文件指定 TIMFMT 键字，那么缺省值是来自物理文件的时间格式。
- 如果指定时间格式参数值\*ISO、\*USA、\*EUR 或\*JIS，那么就不能指定 TIMSEP 键字。这些格式有一个固定的分隔符。
- TIMFMT 键字替換作业属性的时间字段，它不改变系统缺省值。

图 2-76 给出了如何指定 TIMFMT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A  
00020A      R RECORD  
00030A      TIMFLD1      T      TIMFMT(*ISO)  
00040A      TIMFLD2      T      TIMFMT(*USA)  
A
```

图 2-76 规定 TIMFMT 键字

如果当前时间是下午 2 点，系统时间格式是 hhmmss，并且系统时间分隔符是 ‘:’，TIMFLD1 为 14.00.00，TIMFLD2 为 2:00pm。

### 2.4.43 TIMSEP (时间分隔符)

使用这个字段层键字为时间字段指定一个分隔符。这个键字只对时间字段（数据类型为 T）有效。

这个键字格式是：

TIMSEP (\*JOB | ‘时间分隔符’)

时间分隔参数指定的分隔符出现在小时，分钟与秒值的中间。有效值是冒号(:)，隔点(.) 和空格()。参数必须括在单引号内。

如果你指定\*JOB，缺省值是作业属性。

对于物理文件，如果没有指定 TIMSEP 键字，缺省值是作业属性。

对于逻辑文件，如果没有指定 TIMSEP 键字，缺省值是来自物理文件的分隔符。如果没有为物理文件指定 TIMSEP 键字，(在 TIMFMT 键字中指定\*ISO、\*USA、\*EUR 或\*JIS)，缺省值是作业属性。

如果在 TIMFMT 键字中指定\*ISO、\*USA、\*EUR 或\*JIS 时间格式，那么就不能指定 TIMSEP 键字。这些格式有一个固定的分隔符。

如果没有指定 DFT 键字，缺省值是当前时间。

对于一个时间字段，TIMSEP 键字可替换作业属性，但它不改变系统缺省值。

图 2-77 显示了如何指定 TIMSEP 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A  
00020A      R RECORD  
00030A      TIMFLD1      T      TIMSEP(' ')  
00040A      TIMFLD2      T      TIMSEP('.')
```

图 2-77 规定 TIMSEP 键字

如果当前时间是下午 2 点钟，系统时间格式是 hhmmss，时间分隔符是 ‘:’，TIMFLD1 表示为 140000，TIMFLD2 表示为 14.00.00。

#### 2.4.44 TRNTBL (转换表) ——仅用于逻辑文件

使用这个字段层的键字来指定在 PFILE 或 JFILE 键字上指定的物理文件与程序之间传递这个字段时所使用的转换表的名字。这个字段必须是一个字符字段，并且其长度在逻辑文件不可重定义。如果 TRNTBL 键字与 CONCAT 键字一起指定，那么在 CONCAT 键字上指定的那些字段必须都是字符型字段。

这个键字的格式是：

TRNTBL ([库名/]转换表名)

转换表名是必须的参数值，库名是可选的。如果没指定库名，那么 OS/400 使用在文件建立时的库列表 (\*LIBL)。

这个键字仅对只输入 (在 38 列指定 I) 或既不输入也不输出 (在第 38 列上指定 N) 的字符字段有效。

不能对十六进制字段 (35 列上指定 H) 指定 TRNTBL 键字。不要在日期、时间或时间标记字段上 (L、T 或 Z) 指定 TRNTBL 键字。

对同一逻辑文件中的不同字段可以指定多达 99 个不同的转换表。

当从物理文件读这个字段时，发生转换。因此在逻辑文件中指定的所有功能 (例如：字段排序，选择/省略处理和记录的连接) 都依赖被转换后的数据形式。

TRNTBL 键字改变从逻辑文件返回的记录中的数据。ALTSEQ 键字仅改变从逻辑文件返回的记录的顺序。

除下面的情况外，TRNTBL 键字与 CHRID 键字很相似：

- TRNTBL 键字命名所使用的转换表；而 CHRID 键字则不是。
- 当程序读一个逻辑文件时，TRNTBL 键字改变输入到程序上的数据。CHRID 改变指定设备上的显示或打印的数据。程序使用改变的数据时，用 TRNTBL 键字（例如，在一个 COBOL SORT 语句或一个 IF-TAEN-ELSE 语句中）。如果程序处理仅来自一个逻辑文件的数据，则不需要在由程序使用的打印或显示文件中指定 CHRID 键字。

当建立逻辑文件 (CRTLF) 命令上指定了 FILETYPE(\*SRC) 时，TRNTBL 键字是无效的。

注： 1.当使用 TRNTBL 键字时，逻辑文件中字段的长度必须与依据的物理文件中相应

的字段长度相同。

2.在文件建立时，必须对这个转换表有使用的授权。转换表是用生成表 (CRTTBL) 命令建立的。

3.仅当逻辑文件建立时才引用在 TRNTBL 键字中指定的转换表。因此，只要不重建逻辑文件，转换表的更改对逻辑文件是不产生影响的。

图 2-78 给出如何指定 TRNTBL 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1          PFILE(PF1)
00020A      CHAR1             I     TRNTBL(LIB1/TBL1)
00030A      CHAR2             A     I     TRNTBL(LIB2/TBL2)
00040A      NUM1
00050A      NUM2
A
```

图 2-78 规定 TRNTBL 键字

字段 CHAR1 是使用在库 LIB1 中的表 TBL1 来转换的，字段 CHAR2 是使用在库 LIB2 中的 TBL2 表来转换的。为了允许指定 TRNTBL 键字，在逻辑文件中重定义 CHAR2 为字符字段 (35 列为 A)。字段 NUM1 和 NUM 是物理文件 PF1 中的数字字段，对它们不能指定 TRNTBL 键字。

#### 2.4.45 UNIQUE (唯一)

使用这个文件层键字来指定在逻辑文件或物理文件的成员中不允许有重复键值的记录。可以用参数指定重复的空值。它不允许任何可能导致重复键值的插入或增加一个新记录，或是对已有记录的操作，做这个写或更改操作的程序将收到一个错误信息。当一个工作站用户用 DFU 时，这个信息被显示在工作站上。复制文件命令不能完成对文件中有重键值记录的复制。

这个键字格式如下：

**UNIQUE[ (\*INCNUL1 ! \*EXCNUL1) ]**

参数是可选的。在指定参数时，它决定空值是否引起重复。**\*INCNUL1** 是缺省值，它指出当确定是否重复时包括空值。**\*EXCNUL1**，指出在确定是否重复时不包括空值。

当逻辑文件有 **UNIQUE** 键字时，它依据的物理文件成员不能有重复键值。

对物理文件或逻辑文件指定 **UNIQUE** 键字时，必须在建立物理文件（**CRTPF**）或建立逻辑文件（**CRTL1**）时指定 **MAINT(\*IMMED)** 参数值。这样是指在有修改操作后，立即维护访问路径。

如果没有指定 **UNIQUE** 键字，带有重复键值的记录是按指定的顺序检索。如果指定 **FIFO** 键字，它们是按先进先出顺序。如果指定 **LIFO** 键字，它们是按后进先出。如果指定 **FCFO**，它们按先修改先出顺序，如果没有指定 **FIFO**、**LIFO** 或 **FCFO**，则顺序不能确定。关于有重复键值的记录不在同一个文件它们是如何排序的解释，见数据库程序设计书中的内容。

不能与 **FIFO**、**LIFO**、**FCFO** 或 **REFACCP1H** 键字一起指定 **UNIQUE** 键字。

图 2-79 给出如何在一个逻辑文件中指定 **UNIQUE** 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A*  
00020A* SAMPLE LOGICAL FILE (CUSMSTL)  
00030A*  
00040A          UNIQUE  
00050A      R CUSREC          PFILE(CUSMSTP)  
00060A          TEXT('Logical File Master Record')  
00070A      CUST  
00080A      NAME  
00090A      ADDR  
00100A      K CUST  
A
```

图 2-79 规定 **UNIQUE** 键字

#### 2.4.46 **UNSIGNED**（无符号）

使用这个键字段层键字来指定数字字段作为无符号二进制数据处理字符型、日期、时间、时间标记和十六进制字段，缺省为无符号值。

这个键字没有参数。

在物理文件和逻辑文件中不论键字段是什么数据类型，**UNSIGNED** 都有效。**UNSIGNED** 不允许与 **SIGNED** 和 **ABSVAL** 键字同时使用。

在下面的情况下，**UNSIGNED** 键字是缺省值：

- 在文件层对一个区位十进制键字段指定 **ALTSEQ** 时。
- 对区位十进制键字段指定 **ZONE** 或 **DIGIT** 时。
- 对所有的字符、十六进制字段。

注：对浮点字段可以指定 **UNSIGNED**，但结果是无法预测的。

下图给出区位十进制键字段的六个记录。

记录	键字段数字 (区域十进制)	十六进制表示
1	98	F9F8
2	00	F0F0
3	98-	F9D8
4	97	F9F7
5	20	F2F0
6	99	F9F9

如果指定 UNSIGNED, 记录按下面的顺序来排序:

记录	键字段数字 (区域十进制)	十六进制表示
2	00	F0F0
5	20	F2F0
3	98-	F9D8
4	97	F9F7
1	98	F9F8
6	99	F9F9

图 2-80 给出在物理文件中如何指定 UNSIGNED 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORDA
00020A      FLDA      7S 2
00030A      FLDB      5
00040A      K FLDA      UNSIGNED
A
```

图 2-80 规定 UNSIGNED 键字

#### 2.4.47 VALUES (值)

在字段层, 选择/省略层或在这两处指定这个键字。

这个键字的格式是:

VALUES (值-1[值-2…[值 100]])

##### 2.4.47.1 在字段层指定 VALUES

在字段层, 当这个字段被显示文件引用时, 做有效性检验。

VALUES 不影响正在定义的物理文件或逻辑文件, 当在显示文件中定义一个可输入字段时, 能引用这个字段, (通过在 29 列上指定 R 和规定 REF 或 REFFLD 键字)。在显示文件建立期间, OS/400 把在物理文件或逻辑文件中这个字段的 VALUES 及其它属性复制到显示文件的字段中去。可以通过在显示文件中重新指定其它检验来使 VALUES 键字无效。详细内容请看 3.2.9 “引用 (第 29 列)”。

在物理文件和逻辑文件中指定这个关键字的规则与在显示文件中的规则相同。关于如何指定这个关键字的内容见 3.3.163 的“VALUES”。

不能对浮点字段(第 35 列指定为 F)或十六进制字段(第 35 列指定为 H)指定 VALUES 键字。不能对日期、时间、时间标记字段(第 35 列指定为 L、T 或 Z)指定 VALUES 键字。

#### 2.4.47.2 在选择/省略层指定 VALUES

在选择/省略字段层，当程序对指定了选择/省略字段的记录格式送一个输入操作时，这个关键字规定的值选择或省略从物理文件中检索到的记录。

下面的规则是适用的：

- 如果定义的是字符字段，则必须指定字符串或十六进制串。
- 字符串要放在单引号之间，见图 2-81。
- 用 X 之后括在单引号中的数字 0-9 和字母 A-F 的组合来指定十六进制字符串。在单引号中的十六进制的数字个数必须恰好是字段指定长度的两倍，见图 2-82。
- 如果定义的是数字字段，必须为其指定一个数值(不用单引号指定数字 0-9)，见图 2-81。
- 如果定义一个日期字符，用在 DATFMT 中同样的格式和在 DATSEP 中同样的分隔符指定一个有效日期。例如，如果 DATFMT 为\*MDY，DATSEP 指定了‘/’，那么 VALUES(‘12/15/91’ ‘12/31/91’) 是缺省值。
- 如果指定了一个时间字段，用在 TIMFMT 上同样的格式和在 TIMSEP 同样的分隔符定义有效时间。例如：如果 TIMFMT 定义了\*ISO，那么 VALUES(‘11.00.00’ ‘12.00.00’) 是缺省值。\*ISO 的缺省值分隔符是隔点(.)。
- 如果指定了一个时间标记字段，必须按下列格式指定缺省值：

VALUES(‘YYYY-MM-DD-HH.MM.SS.UUUUUU’ ‘YYYY-MM-DD-HH.MM.SS.UUUUUU’)

下例给出如何指定 VALUES 键字。

图 2-81 使用字符和数字值。

00010A	R RECORD1	PFILE(PF1)		
00020A	FIELD A	1 0	VALUES(1 6 9)	1
00030A	FIELD B	1	VALUES(‘A’ ‘B’ ‘C’)	1
00040A	K FIELD A			
00050A	S FIELD B		VALUES(‘A’ ‘B’)	2
00060A	S FIELD A		VALUES(1 6)	2
A				

图 2-81 规定 VALUES 键字(例 1)

1. 对 FIELD A 和 FIELD B 规定了 VALUES 是对引用 FIELD A 和 FIELD B 的显示文件做有效性检验。

2. 对 FIELD A 和 FIELD B 也规定了选择/省略关键字的 VALUES。通过这个逻辑文件记录

并根据下面的字段值从物理文件 PF1 中检索记录:

- FIELDB: 仅当 FIELDB 等于 A 或 B 时选择那些记录。
- FIELDA: 选择那些 FIELDA 等于 1 或 6, 并且还没被选择的记录。

图 2-82 使用十六进制值。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RCD1                  PFILE(PF1)
00020A      CODEA
00030A      FLD1
00040A      FLD2
00050A      K FLD1
00060A      S CODEA                VALUES(X'51' X'54' X'AE')
A
```

图 2-82 规定 VALUES 键字 (例 2)

对 CODEA (一个一位字节的字段) 指定选择/省略的 VALUES。通过这个记录格式, 仅检索那些在物理文件 PF1 中, 并且 CODEA 字段的值是十六进制 51, 十六进制 54 或十六进制 AE 的记录。

#### 2.4.48 VARLEN (变长字段)

使用这个字段层键字定义一个字段为变长字段。当字段数据通常都固定在一个长度范围内, 偶而比此稍长的情况下, 变长字段对于改善存储管理是有用的。在第 30-34 位指定一个最大长度。可以用参数来分配一个固定长度。

这个键字格式如下:

**VARLEN[ (固定长度) ]**

固定长度参数是可选的。用它来规定文件固定部分的长度字节数。如果未指定固定长度参数, 这个字段的数据存贮在文件的变长部分。

固定长度参数的有效值是第 30-34 列指定的最大值。

对于逻辑文件 VARLEN 键字没有参数。

VARLEN 键字对字符字段有效。

当指定 VARLEN 键字时, 能在第 30-34 位指定的最大长度是 32740(如果字段允许空值, 那么最大长度是 32739)。

如果为变长字段指定 DFT 键字, 缺省值长度必须小于等于这个字段的固定长度。如果缺省值大于固定长度, 那么在文件建立后将送出一个错误信息。

如果指定一个十六进制值做为一个变长字段的缺省值, 十六进制字符个数必须等于这个字段指定长度的两倍。

除非为固定长度参数指定一个值, 否则 DFT 键字不允许与 VARLEN 键字用在同一个字段。

不要对日期、时间、时间标记字段 (35 列指定 L, T 或 Z) 指定 VARLEN 键字。

图 2-83 给出如何为物理文件指定 VARLEND 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1  
00020A      FIELD1      100A      VARLEN(30)  
00030A      FIELD2      200A      VARLEN  
A
```

图 2-83 规定 VARLEN 键字

FIELD1 定义为变长字段，它的固定长度为 30，最大长度为 100。FIELD2 也是，它的最大长度 200，但没有固定长度。

#### 2.4.49 ZONE (区位)

使用这个键字段层关键字来指定键字段，仅使用每个字节的区位部分（最左面的 4 位）。数字部分用 0 来填充。

这个关键字没有参数。

这个关键字是针对整个键字段的（不只是这个字段中的一部分），并且仅对字符或十六进制或区位十进制型字符有效。

不能与 ABSVAL、SIGNED 或 DIGIT 关键字一起使用 ZONE。

如果对一个键字段指定 ZONE，则这个字段的值是作为一个无符号的二进制数据来处理的，而不是有符号的，（这是区位十进制的缺省值）。

图 2-84 给出了如何指定 ZONE 关键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      K CODE          ZONE  
A
```

图 2-84 规定 ZONE 关键字

如果 CODE 是一个字节的字段，三个不同记录的字段值如下：

值	十六进制	关键字用数字
A	C1	C
B	C2	C
E	C5	C

# 第三章 显示文件

本章讨论有关显示文件的以下信息：

- 定义
- 位置项
- 键字项

3.2 的“位置项（1 至 44 列）”中给出 DDS 的 1 至 44 列规则和范例。3.3 的“键字项（45 至 80 列）”中给出了定义 DDS 键字的规则和范例。键字按字母顺序说明。

关于如何选择显示文件的位置项和键字，请详见《DB2/400 数据库程序设计》一书。

## 3.1 用 DDS 定义显示文件

为定义一个显示文件，按如下顺序指定各项：

- 1.文件层项（可选）
- 2.记录层项
- 3.帮助层项（可选）
- 4.字段层项（可选）

在文件中至少要指定一个记录格式。一个显示文件中的最大记录格式数为 1024 个。一个记录格式中的最大字段数为 32,763 个。每一记录可显示的最大字段数为 4095 个。每一记录格式中所有字段和指示器的最大总长为 32,763 字节，不包括用法部分（I, O, B, M, H, P）。详见 3.2.13.1 的内容。关于可输入字段的最大数目，请参见《应用显示程序设计》一书。

注：通过建立显示文件命令（CRTDSPF）指定文件名，而不是在 DDS 中规定文件名。

在第一章“简介”中有对文件、记录、帮助和字段层项及规定 DDS 键字语法规则的解释。在附录 B “举例”中有完整的显示文件的例子。

图 3-1 给出了显示文件的编码例子。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00100A* DISPLAY FILE EXAMPLE
00101A*
00102A                               REF (PAYROLL)
00103A      R MENU
00104A      H                      HLPARA(1 1 12 80)
00105A                               HLPRCD (RECORD1 FILEA)
00106A N01
00107A0 02      FLDA      20I 20  2  2DSPATR(HI)
00108A      FLDB      22N 2B  3  2
00109A  72 73
00110A0 60 61 62
```

00111AA	63		DSPATR(HI)
00112A	FLDC	7Y 0B 7 20	DSPATR(RI PC)
00113A	42 43		
00114AO	60 61		
00115AO	62	9 2'	Constant'
00116A	FLDD	R	11 2

图 3-1 显示文件编码例子

### 3.2 位置项 (1 至 44 列)

本部分描述了如何指定显示文件数据描述说明表的前 44 个位置。剩余部分见 3.3 “键字项 (45 至 80 列)”。

图 3-1 给出了显示文件的一些位置项。

#### 3.2.1 顺序号 (1 至 5 列)

用这些位置来指定表中每一行和列的顺序号。顺序号是可选的且仅为文档所用。

#### 3.2.2 格式类型 (第 6 列)

此位置键入 A 来指定此表为 DDS 格式。格式类型是可选的且仅为文档所用。

#### 3.2.3 注释 (第 7 列)

在此位置键入星号 (\*) 来标识这一行为注释行。八至八十列为注释的内容。空行 (7 至 80 列没有字符) 亦被认为是注释。DDS 中注释可在任何位置出现且只保存在源文件中。注释只在源计算机打印输出中，但不在扩展源计算机打印输出中。

#### 3.2.4 条件限定 (7-16 列)

7 至 16 列是一个多字段区，在此区内可指定可选指示器。可选指示器是从 01 至 99 的 2 位数字。在程序中可将可选指示器置为 on (十六进制 F1) 或 off (十六进制 F0) 来选择一个字段或一个键字。可通过指示器选择字段，从而在不同的输出操作来显示不同的数据，而不必为每一种组合字段定义不同的记录格式。

条件是由二列九个指示器的“与”分组，且在字段或键字被选定之前这些批示符必须全起作用 (指定 N 为无效，不指定 N 为有效)。每个条件最多可指定九个指示器，每一字段或键字最多可指定九个条件。因此，每一字段或关键字最多可指定 81 个指示器。当某条件需一个以上指示器且条件满足前必须是 ON 或 OFF 时，要规定“与”条件。即指定了第一个指示器后，“与”上第二个，再“与”上第三个，如此下去，在条件满足且字段或键字被选择之前，所有的提示器必须全部有效。在最后一组 (或只此一组) 指示器指定时，在同一行上要指定字段或键字。

可为一个字段或键字指定几个条件，这样只要满足了其中任一条件就选择该字段或键字。这叫做“或”(OR) 关系。在“或”关系中，如果满足第一个条，或者是第二个条件，

或者是第三个条件，如此下去，就选择该字段或键字。“或”关系中的条件可只有一个指示器构成，或由几个指示器“与”起来构成。多个指示器可“与”起来构成一个条件。多个条件“或”起来可为程序提供几种选择字段或键字的方式。

- 第 7 列 (AND)

如果需多于三个指示器来构成一个“与”条件，可在下一行或下几行中指定指示器。可在第二行或下几行位置的第 7 列键入 A 来继续“与”条件，或此处为空白，因为缺省值即为 A。

- 第 7 列 (OR)

如果指定的几个条件是“或”关系，则每个条件必须各占一行，且除第一个条件外，第 7 列必须为“O”。为第一个条件指定“O”会产生一条警告信息，且此位置被假定为空白。

- 第 8、11、14 列 (NOT)

如果想把一个指示器置为 OFF，只需在此指示器之前位置指定“N”(位置 8、11 或 14)。

### 3.2.5 多键字字段条件

若要条件限定一个字段，字段名(或常量)和最后一个(或仅有的一个)指示器必须在同一行上。如果不是选择字段的输出操作，为该字段指定的键字全不起作用，不考虑这些键字是怎样条件限定的。例如，在图 3-1 中，指示器 01 置为 off 或指示器 02 为 on 时，选择 FLDA。如果 FLDA 没选择，与该字段相关的所有键字，比如，DSPATR (H2) 都被忽略。

若需条件限定一个或多个键字，最后一个(或仅有的)指示器必须出现在键字同一行上。如果条件适用于多于一行的键字，必须对这些指示器使用键字条件作用于所有键字。关于 DDS 语法规则详见第一章“简介”。

显示尺寸条件名：若使程序在非 24 行×80 列显示尺寸的显示设备上打开文件，可在文件层上指定 DSPSIZ(显示尺寸)，然后有条件的使用键字及在 DSPSIZ 中指定“显示尺寸条件名”来分配字段。如果没有指定 DSPSIZ，则程序只能在显示设备上以 24×80 的显示尺寸打开文件。

下表列出每种显示设备的显示尺寸条件名。

设 备	显示尺寸	显示尺寸条件名
3179 3180 3196 3197 (Models C1 and C2) 3476 3487 (Models HA, HC, HG, and HW) 3488 (See Note 2) 3486 (Models BA and BG) 5251 (Models 11 and 12) 5291 5292	24 x 80 个字符 (1920 个字符)	*DS3
3180 3197 (Models D1, D2, W1, and W2) 3477 (Models FA, FC, FD, and FG) 3487 (Models HA, HC, HG, and HW)	27 x 132 个字符 (3564 个字符)	*DS4
注 1：可不用*DS3, *DS4 而自己定义一个显示尺寸条件名。见 DSPSIZ 键字说明，弄清怎样指定用户 定义条件名。		
注 2：取决于显示设备所连的显示器。		

图 3-2 说明了如何指定 DSPSIZ 键字和显示尺寸条件名。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
      A           1           2
00010A                               DSPSIZ(27 132 *LARGE 24 80 *NORMAL)
00020A       R RECORDA
00030A       FIELD A      10  0   1  2
00040A       FIELD B      10  0  1120
00050A   *NORMAL          1  49
00060A       FIELD C      10  0  27  1
00070A   *NORMAL          15  1
      A

```

图 3-2 规定 DSPSIZ 键字和显示尺寸条件名

在图 3-2 中，主显示尺寸的显示尺寸条件名是\*LARGE 1 (52 至 34 列)，次显示尺寸的显示尺寸条件名为\*NORMAL 2 (66 至 75 列)。FIELD A 在两个显示尺寸都出现在第一行第 2 列。FIELD B 在主显示尺寸中出现在第 1 行第 120 列。\*LARGE 的缺省值在次显示尺寸出现在第 1 行第 49 列 (在位置 9 至 16 指定为\*NORMAL)。FIELD C 在主显示尺寸出现在第 27 行第 1 列，在次显示尺寸出现在第 15 行第 1 列。只有次显示尺寸 (此例中\*NORMAL) 可用于条件限定字段位置。

使用显示尺寸条件名的方式类似于使用可选指示器，只是显示尺寸条件名不出现在程序中也不出现在输出记录中。如果显示文件用某个显示尺寸打开，则相应的显示尺寸条件为 ON。使用显示尺寸条件名时，需遵循以下规则：

- 指定 DSIZ 来确定主显示尺寸和次显示尺寸。如果没有指定 DSIZ，缺省为 DSIZ(\*DS3)。
- 只能为某个条件确定一个显示尺寸条件名。不能把多个显示尺寸条件名“与”(AND) 及“或”(OR) 起来，也不能把显示尺寸条件和可选指示器“与”(AND) 及“或”(OR) 起来。
- 显示尺寸条件名必须从第 9 列开始。
- 显示尺寸条件名可由用户定义。详见 3.3.42 “DSIZ (显示尺寸)” 的键字说明。
- 可在第 8 列指定 N 来为主显示尺寸指定“非”(NOT) 条件。

注：第 8 列为 N 意味着在其余的显示尺寸条件之间是或 (OR) 的关系。例如，当在 DSIZ 中指定\*DS3 为次显示尺寸时，N\*DS4 即等价于\*DS3。

绝不能用显示尺寸条件名来代替记录中某一字段的行或位置顺序。字段在显示文件中是由主显示定位排好序。如果次定位改变主定位排好的顺序，则在文件建立时会产生严重错误。

例如，FLD1 和 FLD2 分别由主显示定位位于第 2 行第 2 列和第 4 行第 2 列。在次显示尺寸中用显示尺寸条件名来使 FLD2 显示在 FLD1 前面。

在次显示尺寸中指定某一字段的位置时，只能指定 8 至 16 列 (条件限定) 和 39 至 44 列 (位置)。

如果没有为一个键字指定有效条件名，则在 DSIZ 中指定的主条件名为缺省值。

图 3-3 给出显示尺寸条件名和主显示尺寸的正确的和不正确的组合，此时在 DSIZ 中同时指定了两种显示尺寸，但第一个规定不同。

图 3-3 有效的显示尺寸条件规定		
显示尺寸条件名 (1)	24 x 80 DSIZ(*DS3...) 或 DSIZ(24 80...) 主显示尺寸	27 x 132 DSIZ(*DS4...) 或 DSIZ(27 132...) 主显示尺寸
*DS3	错误(2)	有效
*DS4	有效	错误(2)
N*DS3	有效	错误(3)
N*DS4	错误(3)	有效

注：1.有关这些显示尺寸条件名的用户定义名，见 DSIZ 说明。

2.该显示尺寸条件名有错误，因为显示尺寸为主显示尺寸。

3.这些显示尺寸条件名有错误，因为主和次定位隐含着相同的显示尺寸。一个由非 (NOT) 条件限定的条件名隐含着或 (OR) 关系。例如，N\*DS4 隐含着\*DS3。

图 3-4 和图 3-5 给出键字（此例中为 MSGLOC）的显示尺寸条件限定。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00030A DSPSIZ(*DS3 *DS4)  
00040A *DS4 MSGLOC(26)  
A
```

图 3-4 显示尺寸条件限定（例 1）

在图 3-4 中，显示附条件名指定为\*DS4，所以信息在 27×132 显示器上为第 26 行，在 24×80 显示器（缺省）中为第 25 行。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00080A DSPSIZ(*DS4 *DS3)  
00081A MSGLOC(26)  
A
```

图 3-5 显示尺寸条件限定（例 2）

在图 3-5 中，由于 DSPSIZ 键字中主显示尺寸 (\*DS4) 是缺省值，所以即使没有指定显示尺寸条件名，信息行在 27×132 显示器上仍为第 26 行，在 24×80 显示器上仍为第 25 行。

### 3.2.6 名字或规范的类型（第 17 列）

在此位置指定一有效值来标识 19 至 28 列的名字类型。显示文件的有效项为：

项	含义
R	记录格式名
H	帮助说明
空白	字段名

图 3-1 给出如何对名字类型编码。

关于名字类型，详见 3.2.8 “名字” 部分。关于帮助说明，详见 3.3.57 “HELP (帮助)” 部分。

### 3.2.7 保留（第 18 列）

任何文件类型都不用此位置。除非用作注释，否则此位置空白。

### 3.2.8 名字（19 至 28 列）

用这些位置规定记录格式名和字段名。关于在 DDS 中指定记录或字段名时的规则，参

见 1.2 中“语法规则”。

名字必须从第 19 列开始。

图 3-1 给出了如何指定记录格式名和字段名。

#### 3.2.8.1 记录格式名

当 17 列为 R 时, 19 至 28 列的名字为记录格式名。一个显示文件可有多个记录格式名, 但该文件中的每个记录格式名在此文件中都必须唯一。

#### 3.2.8.2 字段名

如果 17 列为空白, 则 19 至 28 列的名字为字段名。记录格式中的字段名必须唯一。

#### 3.2.8.3 常量字段

常量字段是未命名字段 (19 至 28 列必须空白)。以下规则适用于常量字段:

- 位置 17 至 38 必须是空白。
- 要求有字段位置 (39 至 44 列)。
- 可用指示器对字段进行条件限定 (7 至 16 列)。
- 可用显示尺寸条件名指定次显示定位 (8 至 16 列)。只能指定显示尺寸条件名和定位。就是说, 第 7 列、17 至 38 列、以及 45 至 80 列必须为空白。
- 常量本身用以下项之一在 45 至 80 列定义:
  - 显式 DFT 键字 (在一对单引号中指定常量值做 DFT 的值),
  - 隐式 DFT 键字 (不使用 DFT 键字, 仅用一对单引号中指定值)
  - DATE 键字 (不指定值, 见 DATE 键字说明)
  - TIME 键字 (不指定值, 见 TIME 键字说明)
  - SYSNAME 键字 (不指定值, 见 SYSNAME 键字说明)
  - USER 键字 (不指定值, 见 USER 键字说明)
  - MSGCON 键字 (规定信息描述、信息文件、库名以及信息描述的长度)。

#### 3.2.8.4 如何确定记录格式中的字段顺序

记录格式中规定的字段顺序与程序编译时程序所用的字段顺序是一致的。(没有命名的字段不出现在程序中)。

在 39 至 44 列指定的命名和未命名字段位置决定了字段在显示屏上的顺序。详见 3.2.14 “位置 (39 至 44 列)” 隐式字段 (38 列为 H) 和程序到系统字段 (38 列为 P) 在显示器上不出现。

#### 3.2.9 引用 (第 29 列)

将此位置定为 R, 能利用 OS/400 程序的引用功能, 把一个先前定义的命名字段 (称为被引用字段) 的属性拷贝给要定义的字段。被引用字段可提前在当前定义的显示文件中定义, 也可在先前建立的数据库文件中, (该数据库文件由 REF 或 EFFLD 键字指定)。字段属性包

括字段的长度、数据类型和小数点位置，以及 ALIAS（别名）、FLTPCN（浮点精度）、CCSID（编码字符集标识）、TEXT、编辑和有效性检验键字。

如果 29 列没有指定 R，对此字段就不能使用引用功能，必须另行规定属性。

在文件层、记录层、帮助层上的 29 列，必须为空白。

如果引用字段名和要定义的字段名相同，则只需将 29 列指定为 R（在 19 至 28 列写字段名）。如果要定义的字段名和被引用字段名不同，则必须用 REFFLD（引用字段）键字来指定被引用字段名。

可以用定义被引用字段的文件名做为 REF 或 REFFLD 的参数。参见 3.3.112 “REF（引用）”，3.3.113 “REFFLD（引用字段）” 和附录 A，“何时指定 REF 和 REFFLD”，弄清 OS/400 程序如何识别被引用字段。

可以不必拷贝先前定义字段的所有属性给定义的字段。若需替换引用字段的某些属性，按如下方式为要定义的字段指定属性：

要替换 EDTCDE（编辑编码）或 EDTWRD（编辑字）键字，需为要定义的字段指定 EDTCDE 或 EDTWRD。可通过指定 DLTEDT（删除编辑）键字为要定义的字段删除这些键字。

要替换 CHECK（检验）、COMP（比较）、RANGE（范围）和 VALUES（值）有效性检验键字，及 CHKMSGID（检验信息标识）键字，可为所定义的字段重新指定这些有效性检验键字。可通过指定 DLTCHK（删除检验）键字为所定义的字段删除这些键字。

当替换一些属性说明时，其它属性也受到影响，具体如下：

- 如果为字段定义键盘转换属性，字段长度或小数点位置，那么编辑和有效性检验键字都不能从被引用字段拷贝过来。
- 如果要把先前定义的数据类型重定义为字符型（35 列为 M、A、X 或 W）则小数点位置不被拷贝。但如果 35 列为 N、D 或 I，并且 36 和 37 列（小数点位置）为空白，则所定义的字段与被引用字段的小数点位置相同。对 D 类型来说，小数点位置必须为零。

显示文件不支持压缩十进制和二进制字段。这样，如果引用了这些类型的字段，所赋的类型的带键盘转换的区位十进制，具体如下：

- 如果所定义的字段编辑有效，则键盘转换仅为数字型（35 列为 Y）。
- 如果所定义的字段编辑无效，则键盘转换为有符号数字型（35 列为 S）。

当引用字段包含 REFSHIFT（引用转换）键字时，为 REFSHIFT 所指定的值作显示文件键盘转换。然而，如果为新字段指定的数据类型与 REFSHIFT 键字所指定的键盘转换不兼容，该键字就不拷贝给新字段。

注：显示文件建立后，可删除或修改被引用文件，而不会影响显示文件中的字段描述。把显示文件删除再重新建立，可将引用文件中的修改反映到新的显示文件中。

### 3.2.10 长度（30 至 34 列）

必须为每一个命名的字段指定长度，除非从被引用字段中把长度拷贝过来。长度就是当 I/O 操作使用一个字段时要向程序传送或从程序接收的数据字节数。该长度称为字段的程序

长度。

一个字段出现在显示器上时的长度称为该字段的显示长度。显示长度要大于或等于程序长度。字段的显示长度由键盘转换（35 列）和其它字段说明，例如小数点位置（36 和 37 列），以及编辑功能确定的。详见 3.2.11 “数据类型/键盘转换（35 列）” 内容。

显示长度不包括字段的起始和结束属性字符。但当设计显示布局进行字段定位时，必须考虑这些属性字符。在一个记录中，如果字段间只有一个位置间隔时，则前一字段的结尾属性字符会与后一字段的起始属性字符重叠。详见 3.2.14 “位置（39 至 44 列）” 内容。

字符字段的最大长度等于显示尺寸减一。（留有起始属性字符空间）。数字（区位十进制）字段的最大长度为 31 位。单精度浮点数字段的最大长度为 9 位数字，双精度浮点数字段的最大长度为 17 位数字。

不能为常量字段确定字段长度。关于常量字段长度的解释，详见 DATE、DFT、MSGCON 和 TIME 键字说明。

如果定长度，必须右对齐。前置零可选。图 3-2 给出不正确的和正确的字段长度说明。

...+....1....+....2....+....3....+....4....+....5
00010A           FIELD1        7
A
00020A           FIELD2        7
A
00030A           FIELD3        R    +7
A

注：FIELD1 的字段长度不正确，FIELD2 和 FIELD3 指定的字段长度是正确的。

图 3-6 不正确和正确的长度说明

如果使用了被引用字段，则可通过指定新值或增、减量来替换字段的长度。增加长度值，用+n，n 为增加的长度。减少长度值，用-n，n 为减少的长度。例如，一个数字字段的+4，表示该字段的长度比被引用字段的长度多 4 位。

在某些情况下，如果指定长度值，由数据库文件字段确定的键字并不包括在显示文件中。详见 3.2.9 中“引用”内容。

字段不能占据显示器的第一个位置。第一个位置保留给属性字符。例如，在 24×80 的显示器上，定义一个有符号数字段时，不能规定 39 至 41 列（行位置）为 1，42 至 44 列（列位置）为 1，因为该字段起始位置为 1 是不行的。

### 3.2.11 数据类型/键盘转换（35 列）

35 列为显示文件的数据类型/键盘转换属性。它并不决定程序中使用字段的数据类型。36 至 37 列（小数位）决定字段的数据类型。详见 3.2.12 “小数位”的内容。

键盘转换属性自动转换有数据输入键盘的 5250 工作站，而对所有键盘，限制工作站用

户为某一字段输入内容。但键盘转换属性不能转换带有类打字机一键盘的 5250 工作站。键盘转换属性不能限制程序向某一字段写入的方式。程序可以向一个数字字段写入字母字符，在大多数情况下，也可读该字段并且接收这些字符。所有限制都只与程序设计所用语言有关。

### 3.2.11.1 显示文件有效项

以下为显示文件的有效项：

项	含义	允许数据类型
空白	缺省	
X	仅为字母	字符
A	字母数值转换	字符
N	数值转换	字符或数值
S	有符号数值	数值
Y	仅为数值	数值
W	片假名（仅为日本国用）	字符
I	禁止键盘输入	字符或数值
D	仅为数字	字符或数值
M	仅为数值字符	字符
数据类型		
F	浮点	数值

图 3-1 和图 3-7 给出如何指定键盘转换属性。

键盘转换属性在以下部分详细说明。

缺省（空白）：如果 35 列为空白，36 和 37 列（小数位）内容决定字段的数据类型，具体如下：

- 如果 36 和 37 列为有效值，则数据类型为区位十进制，键盘转换属性为有符号数值（S），除非同时指定编辑键字。如果指定了编辑键字，则键盘转换属性为仅为数值型（Y）。
- 如果 36 和 37 列没有值，则数据类型为字符，键盘转换属性为字母数值转换（A）。

如果对被引用字段指定了 REF SHIFT 键字，则使用所指定的值。否则，在显示文件中压缩十进制或二进制数据转换成区位十进制。这些转换（转换成压缩十进制或二进制及反过来）仅发生在程序中。

仅为字母（X）：两种键盘都为小写。只能键入字符 A 至 Z，逗号（,），句点（.），破折号（—）和空格（ ）。当键入小写字符 a 至 z 时，送往程序的是大写字符。参考 3.3.21.2 “LC（小写状态）” 内容，弄清如何允许在 5250 工作站上类打字机的键盘键入小写字符。

字母数值转换（A）：两种键盘均为小写允许进入所有字符。

数值转换（N）：每种键盘都转换为允许数值输入：数据输入键盘为大写，打字机类键盘为小写。所有字符都有效。

当满足以下条件时，数值转换字段的显示长度比 30 至 34 列规定的长度多一：

- 该字段是一个非编辑的、可输入的字段。

- 小数位字段值大于零。

显示长度中额外位置为小数位。

注：有小数位（36 和 37 列）的输入数值转义字段由数据管理按仅数值字段处理的，除非没有编辑属性。详见 3.2.11.1 “仅为数值（Y）” 键盘属性部分。

有符号数值（S）：每种键盘都被转换为允许数值输入：数据输入键盘为大写，打字机类键盘为小写。该字段只能键入 0 至 9 这十个数字（没有空格，没有加号，没有减号）。要跳出该字段，按换行键、Field+键、Field - 键或光标移动键。如果不对该字段输入任何数据，按回车（Enter）键。

当选择有符号数值（S）或仅为数值（Y）时，应考虑以下区别：

- 有符号数值限制了可键入的字符为数字 0 至 9。
- 当指定了 EDTCDE 或 EDTWRD 键字时，不能同时指定 35 列为 S。
- 仅数值可除去非数字字符，有符号数值根本就不允许输入这些字符。

对仅输入字段，字段的显示长度比 30 至 34 列规定的长度大一。最右端位置保留为减号。

当 OS/400 向程序传递一个有符号数值字段内容时，需考虑以下问题：

- 程序所见的总是一个数值的，右对齐的，零填充的字段。
- 除非规定了键字 CHECK (RZ) 字段按右对齐，空格填充的格式显示。如果指定了 CHECK (RZ)，则该字段按右对齐，零填充显示。
- OS/400 程序不进行十进制定位。
- OS/400 程序不从字段中去掉字符（对仅数值字段要去掉）。

当显示一个可输入的有符号数值字段且没指定 CHECK (RZ)，OS/400 程序按缺省值消掉无用的零（对有符号数值字段，EDTCDE 和 EDTWRD 键字是不合法的）。

负数按如下处理：

- 输入时，必须键入数值，然后按 Field - 键。该数字在显示字段中为右对齐，最右端有一负号。OS/400 在向程序传递数值之前把最右边的有效数字转换成十六进制 D<sub>n</sub>，n 为最后一个有效数字。例如，如果键入 12345 后按 Field - 键，则显示为 12345，程序见到的为 X ‘F1F2F3F4D5’。
- 输出时，OS/400 程序把最右位的十六进制 D 转换为 F，为显示用把负数转换为正数，并在显示字段的最右位(附加位)显示一个负号。例如，若程序所见为 X ‘F1F2F3F4D5’，则显示的数字为 12345 - 。

关于有符号数值字段的范例及输入的样本数据，见图 3-7。

仅数值（Y）：每种键盘都被转换为允许数值输入：数据输入键盘为大写，类打字机键盘为小写。只能键入数字 0 到 9，加号 (+)，减号 (-)，句点 (.)，逗点 (,) 和空格 ( )。按任意键即可跳出该字段。

当满足下述两个条件时（程序长度在 30 至 34 列确定），仅为数值字段的显示长度比程序长度大一：

- 字段为未编辑的，可输入字段。
- 36 和 37 列（小数位）值大于零。显示长度的附加位为十进制小数点。

当 OS/400 向你的程序传递字段内容时，需考虑以下几个方面：

- 程序见到的为一个数值的，对准小数位的字段。
- 要想在小数点右侧键入数字，36 和 37 列必须大于零，还必须键入小数点。
- 不能键入最大位数的数字，小数点和符号，因为字段的显示长度等于程序长度加一。可按 Field+ 键或 Field- 键来避免键入符号字符。
- OS/400 程序消除 0 至 9 及符号之外的所有字符（不论是键入的还是由 EDTWRD 键字提供的）。
- OS/400 程序在小数点对准前把嵌入的空格（十六进制 40）转换为零（十六进制 F0）。（嵌入的空格为字段中任意有效数字之间的空格）。前置空格，后续空格，零，正号，负号不作有效数字对待。编辑字中嵌入的&号在小数对准之前也被转换为零。
- 在小数点对准前和 PANGE、COMP、CMP、VALVES、CHECK (VN)、CHECK (M10)、CHECK (M11)、CHECK (VNE) 键字有效性检验之前，所有的非数值字符都被消除。由 EDTWRD 键字提供的数值字符 (0 到 9) 不消除。CHECK (M10F) 和 CHECK (M11F) 键字的有效性检查在非数值字符消除之前进行。
- 在输入缓冲区中的字段长度为程序长度。

当 OS/400 程序显示仅数值字段时，如果指定了 EDTCDE 或 EDTWRD 键字，该键字就起作用了。可为仅数值字段指定 EDTCDE 或 EDTWRD 键字。显示长度等于程序长度加上由编辑码或编码字指定的编辑字符数。

负数按如下处理：

用户可在输入时，用以下两种方式之一键入负数：

—键入数字，再键入负号。负号 (-，十六进制 60) 在显示器上键入的地方显示出来。

—键入数字，然后按 Field- 键。

- 如果没有指定 CHECK (RZ) 或 CHECK (RB)，则在最右位置显示一个花括号 {}。如果在 36 和 37 列指定了不为零的小数位，在工作站上将出现一条错误信息，如果指定了 CHECK (RZ) 或 CHECK (RB)，则键入的数字为右对齐。任何一种情况下都不出现负号。
- 如果指定了 EDTCDE 键字来显示一个负号，并没有指定 CHECK (RZ) 或 CHECK (RB)，则在最右位置显示一个花括号 {}。在工作站上不会显示错误信息。输出时最右位置有一个负号。
- 如果指定了 EDTCDE 键字来显示一个负号，并且也指定了 CHECK (RZ) 或 CHECK (RB)，则最右边有效数显示为十六进制 Dn (负数)。输出时有负号。
- 向程序传递一个负数时，OS/400 程序把最右边有效数字由十六进制 Fn (正数) 转换为十六进制 Dn (负数)，此处 n 为最右边的有效数字。

输出时符号出现在显示位置最右边，并占用显示长度中的一个位置。见 3.3.44“EDTCDE (编辑编码)”和 3.3.46 “EDTWRD (编辑字)”的内容。

注：OS/400 程序检查仅数值字段的每一个字符，消除非数字字符，正号 (+)，负号 (-)，逗点 (,) 和小数点 (.)，以及非有效数位，并把嵌入空格转换为零。如果必须处理输入字段中的某一有效数字，这种检查和消除会延迟响应时间。

片假名 (W): 本字段属性指定日本片假名键盘转换所有字符都是有效的。

禁止键盘输入 (I): 有此键盘转换属性的字段不接收键盘键入的任何数据，如按任何键都会发生错误警告。可按字段前进键在字段开始处定位光标。这个字段可用来使用特别设备（比如光笔）进行输入。具有此属性的字段使用 Field+, Field- 和 Dup 键都是合法的，功能类似于显示属性保护 (DSPATR (PR)) 无效的输入字段。

当满足下列条件时，禁止键盘输入字段的显示长度比 30 至 34 列规定长度大一：

- 字段为未编辑的，可输入的字段。
- 小数位字段值大于零。

显示长度中附加位置为小数点所用。

仅数字 (D): 每种键盘都允许数值输入转换：数据输入键盘为大写，打字机类键盘为小写。仅数字键盘转换定义了一个字符或数值字段，该字段只允许键入数字 0 至 9。不能键入特殊字符或空格。

只有配置在 6040 或 6041 本地控制器，或者 5294 或 5394 控制单元上的设备，才支持仅数字键盘转换。当把一个仅数字字段送往一个没有配置在合法类型控制器上的设备时，该字段被当作字母数值（键盘转换为 A）处理。这样，由于可向该字段键入任何字母数值字符，在应用程序中会出现十进制数据错误。

允许使用 Field Exit, Field+ Exit 和 Dup 键。Field+ Exit 被当作无符号 Field Exit 处理。

禁止使用 Field- Exit 键。

小数位 (36 和 37 列) 仅支持空白和零。如果 36 和 37 列为空白，该字段被认为是字符字段。如果指定为 0，则该字段认为是数字字段。

D 字段只能输入一个正的整数值。

仅为数字字段的显示长度总为由 30 至 34 列规定的字段长度。

仅数字字段不支持抑制零属性。EDTCDE 和 EDTWRD 键字为不合法，OS/400 程序也不象在有符号数值字段中那样按缺省值消除无用零。

不能向此字段中输入空格。但可在输入字段的部分内容后把光标移出。这时，当 OS/400 向程序传递内容时，应考虑以下几方面：

- 对数字字段，OS/400 程序把前置空格转换为零，并且在把字段内容传给应用程序之前做右对齐。
- 对字符字段，空格按十六进制 40 传给应用程序，该字段中是右对齐的。
- 输入缓冲区中的字段长度是程序长度。

在数据库文件中，如果字段数据类型为数值或字符 (S, B, P 或 A)，可在 REFSHIFT 键字中指定 D 键盘转换。对数值字段，小数位必须为零。

仅为字符数值 (M): 每种键盘都被转换为允许数字输入：数据输入键盘为大写。打字机类键盘为小写。M 键盘转换定义了一个字符字段，它仅允许键入数字 0 到 9，正号 (+)，负号 (-)，逗点 (,), 句点 (.) 和空格。

允许使用 Field Exit、Field+ Exit、Field- Exit 和 Dup 键。Field+ Exit 按无符号 Field Exit 处理。Field- Exit 按如下处理：

- 如果没有指定 CHECK (RZ) 或 CHECK (RB)，最右位置变为花括号。

- 如果指定了 CHECK (RZ) 或 CHECK (RB)，最后键入字段的字符必须为数字（否则发出键盘错误）。最右端数字 n 由十六进制 Fn (正) 转换为十六进制 Dn (负)。

M 字段的显示长度为 30 至 34 列指定的编码长度。字段长度必须包含符号字符或小数点所占的位置。

该字段按空格填充显示，除非指定了 CHECK (RZ) 或 CHECK (RB) 键字。在这种情况下，字段按右对齐、零填充或空格填充显示。

当 OS/400 向程序传递字段内容时，应考虑以下几方面：

- 程序见到的为字符字段。
- 输入缓冲区中的字段长度为程序长度。
- 字段内容直接传给程序。OS/400 程序既不把嵌入空格转化为零，也不消除非数字字符，比如符号和小数点。

在数据库文件中，如果字段数据类型为字符 (A)，可在 REFSHIFT 键字中指定 M 键盘转换。

浮点 (F): 每种键盘都被转换为允许数字输入：数据输入键盘为大写，打字机类键盘为小写。可在浮点字段中键入任何字符组合（但只有数字 0 到 9，符号字符 (+或-)，E 或 e，小数点 (.) 和逗点 (,) 是合法的）。在浮点字段中如果键入其它字符将发出一条出错信息。

一个浮点值包括五部分：

- 有效值符号
- 有效值
- 指数字符
- 指数符号
- 指数值

下面举例说明浮点值的五个部分：

图中各部分如下：

有效值符号：

—正值用 +，负值用 -。

—输出时正值的有效值符号不显示。

—输入时，有效值符号是可选的。如果没有键入+或-，则设定为正值。

有效值：

—数字 0 到 9 和小数点 (.) 或逗点 (,) 为合法字符。

—输出时，有效值数字位数由 30 至 34 列指定的长度决定。小数点或逗点的定位由 36 和 37 列规定的小数位指定。

—输入时，必须键入有效值。只有数字 0 到 9 是合法的。小数点或逗点为可选。如果没有指定小数点位置，则设定为右端。

指數字符：

—E 或 e 为合法的。

—输出时，总会显示指數字符。

—输入时，如果浮点值包括指数，则必须键入指數字符。

指數符号：

—正值用+，负值用-。

—输出时，总会显示指數符号。

—输入时，指數符号可选。如果没有键入+或-，则假定为正。

指數：

—数字 0 到 9 为合法的。

—输出时，指數总为三个数字。

—输入时，如果键入指數字符 (E 或 e)，必须至少键入一个数字。最多可键入三个数字。

注：

- 1.显示浮点值时，嵌入空格被消除。输入时，在浮点值之前或之后都可键入空格。在一个浮点数中，有效值和指數字符之间允许有空格。
- 2.如果在显示的浮点字段中没有键入值，则假定为正零。
- 3.浮点字段中负零值是合法的。只显示小数点左侧的第一个零。负号显示在第一个零右侧。
- 4.浮点字段中正零是合法的。有效值符号 (+) 不显示。只显示小数点左边第一个零。
- 5.在浮点字段中可键入定点值。

浮点字段的显示长度比 30 至 34 列规定的长度大七个位置。附加的七个位置为有效值符号，小数点或逗点，指數字符，指數符号和三位指數数字所用。

图 3-7 和图 3-8 给出键入的数据如何传递给程序。

```
|...+.1...+.2...+.3...+.4...+.5...+.6...+.7...+.8  
00100A*  
00200A* KEYBOARD SHIFT ATTRIBUTES  
00300A      CHARA      5   I   2   2  
00400A      CHARB      5   I   3   2CHECK(RB)  
00500A      CHARC      5   I   4   2CHECK(RZ)
```

00600A	CHARD	5X I 4 30
00700A	CHARE	5M I 4 40
00800A	CHARF	5D I 4 50
00900A	SIGN1	5 0I 5 2
01000A	SIGN2	5 2B 5 30
01100A	NBR1	5Y 0I 6 2
01200A	NBR2	5Y 2I 6 15
01300A	NBR3	5Y 2B 6 30EDTCDE(L)
01400A	NBR4	5N 2I 6 40
01500A	NBRZ	5 H
01600A	FLPT	7F 4I 7 2
	A	

图 3-7 数据类型和键盘转换编码的例子

在图 3-8 中使用了三个特殊字符：

\_意思是未键入任何字符。

X 表示空格。

}内部表示为十六进制 00。

除非有提示，否则只能通过按命令功能键来输入数据。**SIGNIT** 第四个输入是个例外。

它是由按 Field Exit 键完成该输入。

下表列出图 3-7 定义的字段。

图 3-8 数据类型和键盘转换编码的例子

注：K 的内部表示为十六进制 D2。所有非数值字符（包括区位部分含十六进制 D 的字符）都被删除而不放值。例如，5\_KAK\_成为 5020}。

键盘类型：AS/400 系统有两种键盘，一种是打字机类键盘，另一种为数据输入键盘。显示站可配置这两种键盘中的任意一种。

打字机类键盘：打字机类键盘可用大写或小写。用大写方式时键入大写符号（标有两个符号的键）。用小写方式时键入小写符号（标有两个符号的键）。当键盘是用大写方式时用字母键（对只有一个符号的键）键入大写字符。用小写方式时键入大写字母字符，除非指定了检验小写（CHECK (LC)）关键字。如果指定了 CHECK (LC) 键字，并且将键盘置为小写方式，则可输入小写字母字符 a 到 z。

注：对打字机类键盘来说，任何键盘转换属性都不引起自动大写方式。

下图解释打字机类键盘。

数据输入键盘：数据输入键盘可在数字方式（大写）或字母方式（小写）状态。当键盘

在大写方式时键入的是大写符号（标有两个符号的键）。在这种键盘上，数字 0 到 9 在字母键上是大写符号。键盘处在小写方式时键入的是小写符号（有两个符号的键）。字母字符 A 到 Z 都是小写符号并且总是大写状态。数据输入键盘不支持小写字符 a 到 z，即使指定了 CHECK (LC) 键字也不行。

下图解释数据输入键盘。

注：在打字机类键盘上用小写方式键入数字 0 到 9，而在数据输入键盘上用大写方式（数字）。因此，如果一个字段有数值键盘转换属性（数值转换或仅为数值），则打字机类键盘为小写方式，而数据输入键盘为大写方式。在这两种情况下，都可以不用按转换键就能键入数值字符。

### 3.2.12 小数位 (36 和 37 列)

使用这些列来指定区位十进制字段的小数位，还可以指定出现在程序中字段的数据类型。

如果这些列为空白，OS/400 程序为该字段分配字符数据类型。如果在这些位置有数字，OS/400 程序将该字段分配区位小数数据类型。所指定的数字为小数点右侧的位数。这项须小于或等于字段长度，最大为 31 位。

显示文件不支持压缩十进制和二进制字段。因此，当使用引用功能引用这些类型的字段时，分配它数据类型为带有如下键盘转换的区位十进制。

- 如果对定义的字段编辑属性有效，键盘转换是仅数值 (35 列为 Y)。
- 如果对定义的字段编辑属性无效，键盘转换为有符号数值 (35 列为 S)。

如果使用了引用字段，则可以替换或修改这些列的内容，指定新值来代替小数位。要修改小数位，在加号 (+) 或减号 (-) 后写上修改的数字。例如，+4 表示相对于引用字段在小数点右侧增加四位。

图 3-7 给出如何指定小数点位置字段。

### 3.2.13 用途 (38 列)

用这列来指定一个命名的字段是仅输出、仅输入、既输入又输出、隐藏、程序到系统或者信息字段。常量（未命名的）字段此位置无内容。

#### 3.2.13.1 用途有效项

显示文件的有效项如下：

项	含义
空格或 0	仅输出
I	仅输入
B	输入输出都可
H	隐藏（特殊的输出/输入字段）
M	信息（特殊的输出字段）
P	程序到系统（特殊的输出字段）

注：

1. 仅输入和输出/输入字段为可输入字段。
2. 仅输出和输出/输入字段为可输出字段。
3. 如果该列为空白，缺省为仅输出。

仅输出：仅为输出字段在程序向显示器写记录时，从程序向设备传递数据。如果对这个字段指定了 OVRDTA 键字，则可用 DFT（缺省）键字为其指定初始值。如果 OVRDTA 键字无效，则使用该字段的初始值。如果 OVRDTA 键字有效，则使用该字段的数据，数据是从输出缓冲区取出的。

仅输入：当程序读记录时，仅输入字段将数据从设备传给程序。输入字段可初始化为缺省值（由 DFT 键字指定）。如果没修改该字段，该字段又被选择用于输入，则缺省值传给程

序。在显示器上输入字段由缺省值规定加下划线。可用修改输入缺省键字 (CHGINPDFT) 或显示属性下划线键字 (DSPATR (UL)) 去掉下划线。

**输出/输入:** 输入输出字段在程序向显示器写记录时由程序送出, 程序从显示器读记录, 且该字段被选作输入时传给程序。在程序显示可修改数据时经常使用输出/输入字段。可用 DFT 键字为该字段指定初始值。指定了 DFT 后, 还要指定 OVRDTA, 以表明该字段显示的数据是来自输出缓冲区 (OVRDTA 有效), 还是 DFT 规定的 (OVRDTA 无效)。输出/输出字段在显示器上由缺省规定下划线。

**隐藏:** 隐藏字段是命名的数值或字母数字字段, 它不在显示器上出现。程序可用输出操作向该字段传送数据, 也可用输入操作从该字段检索数据, 但不能显示和修改该字段内容。

以下规则适用于隐藏字段:

- 隐藏字段总是命名的。
- 不能给隐藏字段确定位置。
- 象其他命名字段一样可以为它指定长度, 数据类型和小数点位置。
- 一个显示文件可指定多个隐藏字段。

既然隐藏字段不显示, 也就不考虑该字段为可输入及输出属性, 尽管程序可向该字段传送数据, 也可从该字段接收数据。

隐藏字段在涉及子文件的应用中很有用。例如, 子文件记录可在隐藏字段中包含记录的键字信息。隐藏字段不可见, 但该字段可由子文件记录返回给程序, 这样程序就能把记录返回给数据库。

**信息字段:** 信息字段是一个命名的, 仅输出的字符字段。以下规则适用子信息字段:

- 可使用可选指示器来选择信息字段, 但在处理期间, 一次只能显示一条信息。一次操作只有选择的第一个信息字段被显示, 其它都忽略。
- 显示一个信息字段时, 该记录的其它字段都按正常方式处理。设备进入错误状态 (锁住键盘, 光标闪烁, 信息以高亮 (HI) 显示。) 在按了 Reset 键后, 继续正常处理。
- 当程序向信息字段传送值后, 信息正文才建立起来。
- 信息在显示器上的位置为信息行 (除非 MSGLOC 键字有效, 否则信息为显示器的最后一行)。
- 信息字段的长度在 24×80 工作站上要小于 79 个位置, 在 27×132 工作站上要小于 131 个位置。所有超出的信息文本都将从尾部截断。
- 信息字段不支持帮助键 (Help)。当按下 Help 键时, 关于该信息的帮助信息不显示。
- 对信息字段以下键字是有效的:

ALIAS	REFFLD	
INDTXT	TEXT	OVRDTA

- 如果一个字段是子文件记录格式的一部分, 不能在 38 列指定为 M。

**注:** 对一个不包含可输入字段的记录进行输入操作是合法的。这允许按下一个功能键作为对一个输出记录的响应。

**程序到系统字段:** 程序到系统字段是一个命名的、数值的或字母数字值的仅输出字段, 用来在程序和系统之间传递数据。它不出现在显示器上, 程序可用输出操作向该字段传送数

据，但工作站用户无法见到字段的内容。由于程序到系统字段不显示，因此不认为它们是可输出字段，即使用程序可以由它们传送数据。

以下规则适用于显示文件中的程序到系统字段：

- 程序到系统字段总是命名的。
- 为程序到系统字段定位是不合法的。
- 象其它命名字段一样，要给它指定长度、数据类型和小数点位置。
- 程序到系统字段必须做为同一记录格式的 CHCACCEL、CHCCTL、CHKMSGID、CHOICE、ERRMSGID、GRDATTR、GRDBOX、GRDCLR、GRDLIN、HTML、MNUBARCHC、MSGID、PSHBTNCHC、SFLCHCCTL、SFLMSGID、SFLSIZ、WDWTITLE 或 WINDOW 键字的一个参数。P—用法字段作为任何其它键字的参数都不合法。该字段必须在上述键字中至少一个中指定，否则会发出严重错误信息。
- 不象 ICF 文件中的 P—用法字段，显示文件中 P—用法字段可在缓冲区中任何位置出现。在 ICF 文件中，P—字段必须在所有数据字段（B—用法字段）之后指定。
- P—用法字段可指定为 MSGID 键字的信息标识。MSGID 中的信息文件或库名，提供定义的字段属性（比如长度）。详见 3.3.95 “MSGID（信息标识符）”。
- 包含 P—用法字段的记录必须在系统得知 P—用法字段所包含的内容之前写入。

程序到系统字段仅允许下列键字：

ALIAS      TEXT      INDTXT      REFFLD

### 3.2.14 位置（39 至 44 列）

使用这些列来为每个字段指定在显示器上的准确位置。不能为隐藏、程序到系统或信息字段指定位置。位置的有效与否是依据 DSPSIZ 键字和显示尺寸条件名。见 3.3.42 “DSPSIZ（显示尺寸）” 的举例说明。

行（39 至 41 列）：用这些位置指定字段的起始行。该项必须为右对齐，前置零可选，最大行数为 24 或 27。关于显示尺寸条件名，详见 3.2.5 的图 3-3。

列（42 至 44 列）：使用这些位置来指定字段在确定行中的起始列位置，它为右对齐，前置零可选，最大列数对 3180 设备为 132，对其余所有显示设备为 80。

对记录中非第一个字段，可在 42 至 44 列指定一个增量来指定列数，该增量表明了在前一字段的末尾和所定义字段的起始列之间留出的空格数。增加值必须在 1 到 99 之间。（增加值为 0 是不合法的）。

起始属性字符：每个显示的字段都有一个属性字符，它定义该字段在显示器上的显示属性。属性字符不显示，但会在紧换字段之前占据一个位置。由于有起始属性字符，就不能在显示器的第一个位置（第一行第一列）指定做为一个字段起始位，除非指定了 SLNO（起始行）键字且起始行大于一。

如果一个字段起始于第一列，则起始属性字符出现在前一行的最后一个位置。如果此字段为某记录的第一个字段，则前一行为该记录区的一部分，显示为空白行。使用该行的任何记录格式都不能和其它记录同时显示。显示最后一行会引起删除另外的行（除非为最后显示的记录指定了 CLRL(\*NO) 键字）。

**结尾属性字符:**由结尾属性字符指出一个字段在显示器上的末尾位置,除非该字段和下一字段间只有一个空格。在这种情况下,第二个字段的起始属性字符就起到了第一个字段的结尾属性字符的作用。任何情况下,一个记录中字段间必须至少有一个空格。当记录的某一字段的结尾是那一行的最后一个位置,则它的结尾属性字符是下一行的第一个位置。然而,下一行不被认为是第一个记录的一部分,这两行上的记录可同时显示。

**重叠字段:**在一个记录格式中,一个字段可与其它字段的一部分或它们的属性字符重叠;然而,在显示器上一次只能显示其中一个字段。运行中,在处理记录内重叠字段时,OS/400程序按行和列顺序查看每个字段。如果OS/400程序发现某一个字段的条件满足要求,或没指定可选指示器,则选择一个字段显示,而其它剩余的重叠字段都被忽略。第一个没有规定可选指示器的重叠字段使搜索停止,其后的重叠字段则不用显示。在下面例子中,如果指示器01为ON,则显示FIELD1字段。如果指示器01为OFF,而02为ON,则仅显示FIELD2字段。这些都没被选择时,显示FIELD3字段。

图 3-9 给出了如何定义重叠字段

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A 01      FIELD1      10      1  5TEXT(' ONE')  
00020A 02      FIELD2      5       1  5TEXT(' TWO')  
00030A          FIELD3      2  0   1  5TEXT(' THREE')  
A
```

图 3-9 规定重叠字段

如果使用不当,此性能可能导致用户和程序间的通讯出现问题。下述例子中,记录中只指定了一个输入字段(FIELD4),根据字段位置的规定,该字段与前一输出字段重叠。工作站用户无法键入任何数据,因为显示的总是FIELD1字段。其它三个字段,包括FIELD4,则不能显示。

图 3-10 给出不正确的字段定义。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A          FIELD1      10      1  5TEXT(' ONE')  
00020A 21      FIELD2      5       1  5TEXT(' TWO')  
00030A 12      FIELD3      2  0   1  5TEXT(' THREE')  
00040A          FIELD4      5Y 2I  1  5TEXT(' FOUR')  
A
```

图 3-10 不正确的字段规定

**显示长度:**在设计显示的时候必须考虑某些类型字段的显示长度增加。在下述情况下OS/400程序增加显示长度:

- 对带编辑的仅数值字段,显示长度由编辑字或程序长度及编辑码确定。

- 对仅输入有符号的数值字段，显示长度比程序长度多一。
- 对数值转换字段和无编辑仅数值字段，当它们有输入属性且有小数点时，显示长度比程序长度多一。
- 浮点字段的显示长度比由 30 至 34 列规定的长度大七。这七个附加位置为有效值符号、小数点或逗点、指数字符、指数符号和三位指数数字所用。

对非零小数位的无符号数值字段（如图 3-10 中的 FIELD4），系统要求在数据输入小数值时键入一个小数点字符。在图 3-9 中，FIELD4 中的 123 不需要小数点字符，但 1234 就需要 (123.4)。此字段的显示长度为 6。

### 3.3 键字项 (45 至 80 列)

本部分包括定义显示文件有效的键字项。这些键字在 45 至 80 列输入 (功能项)。见 1.2 “语法规则”有关指定键字的一般规则。

以下为显示文件有效的键字：

ALARM	EDTMSK	LOWER	SFLCSRPRG
ALIAS	EDTWRD	MDTOFF	SFLCSRRRN
ALTHELP	ENTFLDATR	MLTCHCFLD	SFLCTL
ALTNNAME	ERASE	MNUBAR	SFLDLT
ALTPAGEDWN	ERASEINP	MNUBARCHC	SFLDROP
ALTPAGEUP	ERRMSG	MNUBARDSP	SFLDSP
ALWGPH	ERRMSGID	MNUBARSEP	SFLDSPCTL
ALWROL	ERRSFL	MNUBARSW	SFLEND
ASSUME	FLDCSRPRG	MNUCNL	SFLENTER
AUTO	FLTFIXDEC	MOUBTN	SFLFOLD
BLANKS	FLTPCN	MSGALARM	SFLINZ
BLINK	FRCDTA	MSGCON	SFLLIN
BLKFOLD	GETRETAIN	MSGID	SFLMLTCHC
CAnn	GRDATR	MSGLOC	SFLMODE
CCSID	GRDBOX	NOCCSID	SFLMSG
CFnn	GRDCLR	OPENPRT	SFLMSGID
CHANGE	GRDLIN	OVERLAY	SFLMSGKEY
CHCACCEL	GRDRCD	OVRATR	SFLMSGRCD
CHCAVAIL	HELP	OVRDTA	SFLNXTCHG
CHCCTL	HLPPARA	PAGEDOWN	SFLPAG
CHCSLT	HLPBDY	PAGEUP	SFLPGMQ
CHCUNAVAIL	HLPCCLR	PASSRCD	SFLRCDNBR
CHECK	HLPCMDKEY	PRINT	SFLRNA
CHGINPDFT	HLPODOC	PROTECT	SFLROLVAL
CHKMSGID	HLPEXCLD	PSHBTNCHC	SFLRTNSEL
CHOICE	HLPFULL	PSHBTNFLD	SFLSCROLL
CHRID	HLPID	PULLDOWN	SFLSIZ
CLEAR	HLPPNLGRP	PUTOVR	SFLSNGCHC

CLRL	HLPRCD	PUTRETAIN	SLNO
CMP	HLPRTN	RANGE	SNGCHCFLD
CNTFLD	HLPSCHIDX	REF	SYSNAME
COLOR	HLPSEQ	REFFLD	TEXT
COMP	HLPSHELF	RETKEY	TIME
CSRINPONLY	HLPTITLE	RETCMDKEY	UNLOCK
CSRLOC	HOME	RETLCSTS	USER
DATE	HTML	RMVWDW	USRDFN
DFT	INDARA	ROLLDOWN	USRDSPMGT
DFTVAL	INDTXT	ROLLUP	USRRSTDSP
DLTCHK	INVITE	RTNCSRLOC	VALNUM
DLTEDT	INZINP	RTNDDTA	VALUES
DSPATR	INZRCD	SETOF	VLDCMDKEY
DSPMOD	KEEP	SETOFF	WDWBORDER
DSPRL	LOCK	SFL	WDWTITLE
DSPSIZ	LOGINP	SFLCHCCTL	WINDOW
DUP	LOGOUT	SFLCLR	WRDWRAP
EDTCDE			

### 3.3.1 ALARM (声音报警)

使用此记录层键字可使 OS/400 程序在显示此记录时激活声音报警。报警为短声。

此键字没有参数。

如果错误信息 (ERRMSG) 或错误信息标识 (ERRMSGID) 键字有效，则即使规定了 ALARM 键字也无效。

若要在 ERRMSG 或 ERRMSGID 键字有效时发出声音警报，见 3.3.93“MSGALARM(信息警报)” 内容。

此键字可规定可选指示器。

图 3-11 给出如何指定 ALARM 关键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00005A      R CUST
00010A 01          ALARM
A
```

图 3-11 规定 ALARM 键字

### 3.3.2 ALIAS (替换名)

使用此字段层键字来为字段指定另一个名字。当程序编译时，此名送给程序来取代 DDS 中的字段名。所用的高级语言编译程序决定是否使用 ALIAS 名。关于高级语言对 ALIAS 的支持情况，参见相应的高级语言参考手册。

此键字格式为：

### ALIAS (替换名)

关于 ALIAS 的命名约定，参见 1.2 “语法规则”。

替换名必须和记录格式中其它所有替换名及 DDS 字段名不同。如果发现重名，则在字段名或替换名上出现错误。

替换名不能在 DDS 中或任何其他 OS/400 功能中使用（例如，作为键字字段名，作为 REFFLD 键字的字段名，或作为拷贝文件 (CPYF) 命令中使用的字段名）。

当引用一个有 ALIAS 键字的字段时，ALIAS 键字被拷入，在要引用字段中明确的指定了 ALIAS 键字。

此键字不能用可选指示器。

图 3-12 给出如何指定 ALIAS 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00070A          FIELD A          25A      1 2ALIAS(CUSTOMERNAME)  
A
```

图 3-12 指定 ALIAS 键字

FIELDA 的替换名为 CUSTOMERNAME。

### 3.3.3 ALTHELP (替换帮助键)

使用此文件层键字来把一个命令注意键 (CA) 做为一个替换的帮助键。当按下帮助键或 CA 键时，都会调用帮助功能。

此键字的格式为：

ALTHELP[ (CAnn) ]

可选参数的值为 CA01 到 CA24。如果没有规定参数，则缺省为 CA01。

与此键字同时必须还在文件层或至少一个记录中指定 HELP 键字。只有使用了 HELP 键字的记录使用 ALTHELP 键字才有效。如果 HELP 键字是在文件层上指定的，则对该文件中所有记录都适用，这样，ALTHELP 也对该文件中所有记录都适用。如果 HELP 键字是在记录层指定的，则 ALTELP 键字只对指定了 HELP 键字的记录适用。关于指定 HELP 键字，详见 3.3.5“HELP (帮助)”内容。

如果在 HELP 键字中指定了响应指示器，则响应指示器被置为 ON，当按下 Help 键或 CAnn 键时控制转给程序。

如果在 HELP 键字中指定了可选指示器，这些批示器对 ALTHELP 键字也适用。就是说，在指示器条件为真时 HELP 键和 CAnn 键都有效。

文件中若规定不带参数（缺省为 CA01）的 ALTHELP 键字时，**不能**指定以下键字：

ALTPAGEDWN (CF01)

PSHBTNCHC (... CF01)

ALTPAGEUP (CF01)

SFLDROP (CA01)

CA01

SFLDROP (CF01)

CF01

SFLEENTER (CA01)

MNUCNL (CA01)	SFLENTER (CF01)
MNUBARSW (CA01)	SFLFOLD (CA01)
MOUBTN (... CF01)	SFLFOLD (CF01)

同样，文件中若规定 ALTHELP (CAnn) 键字 (nn 是相同的数字) 也不能指定下述键字：

ALTPAGEDWN (CFnn)	PSHBTNCHC (... CFnn)
ALTPAGEUP (CFnn)	SFLDROP (CAnn)
CAnn	SFLDROP (CFnn)
CFnn	SFLENTER (CAnn)
MNUCNL (CAnn)	SFLENTER (CFnn)
MNUBARSW (CAnn)	SFLFOLD (CAnn)
MOUBTN (... CFnn)	SFLFOLD (CFnn)

**不能**在有 ALTHELP 键字的文件中指定 RETKEY 和 RETCMDKEY 键字。

可选指示器对此键字无效。

图 3-13 给出如何指定 ALTHELP 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               ALTHELP
A                               HELP(01 'HELP KEY PRESSED')
A           R RECORD
A           FIELD1      20A      5  5
A
```

图 3-13 指定 ALTHELP 键字

CA01 键是一个替换帮助键。如果用户按下 CA01 键或 Help 键，指示器 01 为 ON，控制返给应用程序。

### 3.3.4 ALTNAMEx (替换记录名)

使用此记录层键字来为记录确定一个替换名。替换名可在使用程序描述文件时为 I/O 操作来指定。替换记录名语法必须是使用的高级语言编译认为是有效的。

该键字的格式为：

ALTNAMEx ('替换名')

关于如何确定 ALTNAMEx 键字，详见附录 F “系统/36 环境考虑” 内容。

### 3.3.5 ALTPAGEDWN/ALTPAGEUP (替换的翻下页/翻上页键)

使用这两个文件层键字来把命令功能键(CF)定义为替换翻下页/翻上页键。当按 Page 键或 CF 键时，就调用页功能。选择翻下页/翻上页键只对文件中用 ALTPAGEDWN/ALTPAGEUP 键

字定义的显示起作用，而对系统显示（比如帮助信息）不起作用。

这两个键字的格式为：

ALTPAGEDWN[ (CFnn) ]

ALTPAGEUP[ (CFnn) ]

可选参数的有效值为 CF01 到 CF24。如果没有指定参数，则 ALTPAGEDWN 缺省为 CF08，ALTPAGEUP 缺省为 CF07。

若想使程序能够处理用户按下 Page 键或替换的 CFnn 键，还应该同时指定 PAGEDOWN/PAGEUP 键字，而且 OS/400 程序不能移动显示器上的文本行。如果没有指定 PAGEDOWN/PAGEUP 键字，当按下一个 Page 键或替换的 CFnn 键时，会显示一条信息来表明此键无效，且 OS/400 程序不能移动显示器上的文本行。如何指定 PAGEDOWN/PAGEUP 关键字，详见键字说明。

注：通过这个键字的说明，PAGEDOWN 是指 PAGEDOWN 或 ROLLUP 键字，PAGEUP 是指 PAGEUP 或 ROLLDOWN 键字。

如果在 PAGEDOWN 或 PAGEUP 中指定了响应指示器，则在按下 Page 键或替换 CFnn 键中任一键时，响应指示器置为 ON，并返回给程序，OS/400 程序不能移动显示器上的文本行。

如果在 PAGEDOWN 或 PAGEUP 中指定了可选指示器，则这些可选指示器按所给顺序同样适用于 ALTPAGEDWN 或 ALTPAGEUP 键字。就是说，在指示器条件为真时，按下 Page 键或 CFnn 键之一时，控制权都将返给程序，OS/400 程序不能移动显示器上的文本行。

文件中有不带参数（缺省 CF08）的 ALTPAGEDWN 键字时不能指定下述键字：

ALTHELP (CA08)	PSHBTNCHC (... CA08)
ALTPAGEUP (CF08)	SFLDROP (CA08)
CA08	SFLDROP (CF08)
CF08	SFLENTER (CA08)
MNUCNL (CA08)	SFLENTER (CF08)
MNUBARSW (CA08)	SFLFOLD (CA08)
MOUBTN (... CA08)	SFLFOLD (CF08)

文件中有不带参数（缺省为 CF07）的 ALTPAGEUP 键字时，不能指定下述键字：

ALTHELP (CA07)	PSHBTNCHC (... CA07)
ALTPAGEDWN (CF07)	SFLDROP (CA07)
CA07	SFLDROP (CF07)
CF07	SFLENTER (CA07)
MNUCNL (CA07)	SFLENTER (CF07)
MNUBARSW (CA07)	SFLFOLD (CA07)
MOUBTN (... CA07)	SFLFOLD (CF07)

类似地，文件中有 ALTPAGEDWN (CFnn) 或 ALTPAGEUP (CFnn) 键字不能指定下

述键字 (nn 为相同数字):

ALTHELP (CAnn)	SFLDROP (CAnn)
CAnn	SFLDROP (CFnn)
CFnn	SFLENTER (CAnn)
MNUCNL (CAnn)	SFLENTER (CFnn)
MNUBARSW (CAnn)	SFLFOLD (CAnn)
MOUBTN (... CAnn)	SFLFOLD (CFnn)
PSHBTNCHC (... CAnn)	

还有，在 ALTPAGEDWN 和 ALTPAGEUP 中不能指定同一个命令功能键 (CF)。

文件中有 ALTPAGEDWN 或 ALTPAGEUP 键字不能指定 RETKEY 或 RETCMDKEY。

ALTPAGEDWN 和 ALTPAGEUP 键字只用在有可分页区(子文件或 PAGEDOWN/PAGEUP 键字)的文件中。

可选指示器对这两个键字是无效的。

图 3-14 给出如何指定 ALTPAGEDWN 和 ALTPAGEUP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               ALTPAGEUP
A                               ALTPAGEDWN
A     R SUBFILE               SFL
A     FIELD1      20A    5   5
A     R CONTROL              SFLCTL(SUBFILE)
A                           SFLSIZ(30)
A                           SFLPAG(10)
A                           SFLDSP
A
```

图 3-14 指定 ALTPAGEDWN 和 ALTPAGEUP 键字

CF07 和 CF08 键为替换的 Page 键。显示子文件过程中，如果用户按 CF07/CF08 键或 Page 键，OS/400 程序就把子文件分页显示。如果 OS/400 程序在分页时不在页首或页尾，则会显示一条信息说明此时该键无效 (没有指定 PAGEDOWN/PAGEUP 键字)。

### 3.3.6 ALWGPH (允许图形)

这个文件或记录层键字允许在 5292 - 2 型彩色显示站上同时显示图形和由记录格式规定的字母数值内容。若在其它型显示器上显示文件，即使指定了此键字也无效。

此键字没有参数。

当把一个此键字有效的记录写向 5292 - 2 型彩色显示站时，如果此时该显示站不是处于图形显示方式，则会被置为图形显示方式。不管该键字对显示站上的其它记录是否有效都会这样。只要有选择了 ALWGPH 键字的记录显示，显示站一直保持在图形显示方式状态。显

示没有 ALWGPH 键字的记录不会结束图形显示方式。

为结束图形显示方式，必须从显示器上把所有规定了 ALWGPH 键字的记录删除（或将这些记录中的该关键字置为无效）。图形显示结束后，已显示的所有图形都不删除，只不过不显示罢了。没有删除的图形显示（通过使用 GDDM\*功能），在显示下一个含 ALWGPH 键字的记录时会再一次显示出来，包括由系统请求菜单调用第二个交互式作业中的记录。

在图形显示方式中：

- 显示站自动被置为缩减行间距方式，此时行与行之间空格较少。不管工作站用户是否从键盘上（本地方式）选择这种方式都这样做，而且用户也不能超越这一步。
- 当显示 DDS 记录格式时，已在显示器上的所有图形仍旧保留下来，作为按记录格式显示的字母数字字符的背景。

可选指示器对此键字有效。

此键字不能和 SFL 或 USRDFN 键字一起指定此关键字。

图 3-15 给出如何指定 ALWGPH 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1          ALWGPH  
00020A          23 2' Enter account number:  
00030A      ACCT      5   B    +2  
A
```

图 3-15 规定 ALWGPH 键字

在图 3-15 中，RECORD1 可用图形方式显示。常量“Enter account number:”和字段 ACCT 出现在显示器上。任何使用 GDDM 例程显示的图形都出现在字母数字后面。换句话说，当图形的线或图与字母数值交叉时，部分图形被字母数值遮盖。

图 3-16 给出在可选指示器 01 为 ON 时，RECORD2 用图形方式显示。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD2  
00020A  01          ALWGPH  
00030A          1 34' Sample Title:  
A
```

图 3-16 用可选指示器的 ALWGPH 键字

在图 3-16 中，如果可选指示器 01 为 OFF，则所有已显示的图形都不再显示（没有删除），只有通过 DDS 指定的字母数值（比如“Sample Title”）出现在显示器上。在后面的输出操作中，如果可选指示器 01 为 ON，图形会和字母数值一起重新出现。

### 3.3.7 ALWROL (允许滚动)

在显示所定义的记录格式时，使用此记录层的键字允许用户程序在显示器上开个窗口对数据进行翻页。窗口由程序中定义的起始行和结束行之间的显示行组成。翻页的行数和翻页的方向都在用户程序中定义。

此键字没有参数。

当用户程序向此记录格式做输出操作时，OS/400 程序对已在窗口中的数据进行分页，并可向上或向下翻页，从而显示出整个记录。分页时超过起始行或结束行的数据全部丢失。在分页完毕后，用户程序不能对前页部分或全部在窗口中的记录格式进行输入操作。

在 COBOL 语言中使用 ALWROL 功能，用 WRITE ROLLING 语句。

ALWROL 键字不允许显示站用户对数据进行分页，它只允许用户程序在显示器上对数据进行分页。为允许显示站用户对数据分页，可指定 ROLLUP 或 ROLLDOWN 键字，或指定一个子文件，使子文件页与子文件长度不等。详见 3.3.149 “SFLROLVAL（子文滚动值）” 和 3.3.147 “SFLRCDNBR（子文件记录数）” 内容。

为防止删除分页记录，和 ALWROL 键字起指定 OVERLAY 键字或 CLRL（清除行）键字。

在指定 ALWROL 键字时，不要在字段上指定 PUTRETAIN 键字。如果同时指定它们，不管是否选择了 PUTRETAIN 键字，在用户程序做输出操作时，OS/400 程序将发送 CPF5014 信息。

如果和下述键字一起指定 ALWROL，则必须指定可选指示器：

ERRMSG           ERRMSGID           PUTOVR  
PUTRETAIN（在记录层）

用户程序不能选择这些键字之一的同时发出输出操作试图使用 ALWROL 功能（OS/400 程序发出 CPF5014）。

不能和下述键字的任何一个同时指定 ALWROL 键字：

ASSUME   KEEP   SFL   SFLCTL   USRDFN

如果在记录中和 DSPMOD 键字一起指定了 ALWROL 键字，则在文件生成时会出现一条警告信息。运行时，如显示方式改变 ALWROL 键字被忽略。

在用 PASSRCD 键字指定的记录格式中不能再指定 ALWROL 键字。

可选指示符对此键字无效。

图 3-17 给出如何指定 ALWROL 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      1   R RECORD1           ALWROL OVERLAY    2
00020A          FLDA        79   I 23  2CHECK(LC)
00030A  44          3   ERRMSG('Record not found' 44)
A
```

图 3-17 规定 ALWROL 键字

1. 应用程序对 RECORD1 执行输出操作，在第 23 行第 2 列显示 FLDA。在随后的输出

操作中，程序对 RECORD1（本例中 FLDA）分页，进行向上或向下翻页显示，或完全关闭显示。正常情况为上翻一行。原来输入的数据显示在第 22 行，新的输入字段显示在第 23 行。显示站用户不能向第 22 行的字段键入数据，程序也不能读第 22 行字段内容。第 22 行的字段会由随后的输出操作推向上屏，直到它超出窗口（在程序中定义）的起始行或显示器的第一行。

## 2.OVERLAY 键字防止分页记录被删除。

3.图中显示了 ERRMSG 键字来说明 ERRMSG 是如何影响 ALWROL 功能的。程序不能把可选指示器 44 置 ON 的同时发出请求 ALWROL 功能的输出操作。如果程序这么做了，OS/400 程序会发出一条通知信息 CPF5014。

### 3.3.8 ASSUME（假定）

使用此记录层键字使 OS/400 程序假定在打开文件时该记录已出现在显示器上。这样的记录也可以在另一个显示文件中用 KEEP 键字定义。在打开这个文件（规定 ASSUME 的文件）之前，另一个文件被关闭。

此键字没有参数。

在显示文件中必须为至少一个记录格式指定 ASSUME 键字，这样在打开文件时，OS/400 程序不会删除显示。另外，指定 ASSUME 键字的同时指定 OVERLAY 键字，可在用户程序打开文件后发出第一个输出操作时，防止 OS/400 程序删除显示。

如果使用了 ASSUME 键字，则该记录中必须至少有一个字段是可以显示的。如果多个记录包含 ASSUME 键字，它们必须占据唯一的一些显示行。

为使 OS/400 程序能正确地处理数据，用户程序必须指定包含此键字的记录格式名。

如果所定义的记录格式是在共享文件中，在建立显示文件 (CRTDSPF)、修改显示文件 (CHGDSFP) 或替换显示文件 (OVRDSPF) 命令中使用 SHARE(\*YES) 参数，就不需要 ASSUME 键字。

不能和下述键字一起指定 ASSUME:

ALWROL	SLNO	CLRL	USRDFN
SFN	USRDSPMGT		

如果在一记录中和 DSPMOD 键字一起指定了 ASSUME 键字，则在文件生成时会发出一条警告信息。运行过程中，当显示文件改变时 ASSUME 键字被忽略。

有 ASSUME 键字的文件以有 KEEP 键字的文件的显示尺寸打开。

可选指示器对此键字无效。

图 3-18 给出如何指定 ASSUME 键字。

```
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8  
00005A      R RECORD          ASSUME  
A
```

图 3-18 规定 ASSUME 键字

### 3.3.9 AUTO (自动)

AUTO 键字与 CHECK 键是等价的，如下：

AUTO (RA)	CHECK (ER)
AUTO (RAB)	CHECK (RB)
AUTO (RAZ)	CHECK (RZ)

此键字的格式为：

AUTO ([RA [RAB | RAZ]])    AUTO (RAB | RAZ)

建议用 CHECK 键字。关于如何使用这些键字，详见 3.3.21.2 “键盘控制” 中 CHECK 键字的说明。

### 3.3.10 BLANKS (空白)

此字段层键字用于数字可输入字段时，使用户程序可以辨别在显示器上该字段何时为空白，何时为零。在这两情况下，用户程序所见均为零。当显示器上该字段为空白时，BLANKS 键字将响应指示器置为 ON。在输入操作后，用户程序可测试这个指示器来确定该字段（程序值为零）在显示器上确实为空白。此字段可全由空白（十六进制 40）或空零（十六进制 00）构成。该字段对显示站用户来说仍为空白。如果指示符为 OFF，则在显示器上该字段为零。

该键字对字符字段同样有效，但通常没有为字符字段指定该键字的必要。用户程序可直接测试字符字段来确认显示器上显示的是什么。

关于在系统/36 环境使用 BLANKS 键字的问题，参见附录 F “系统/36 环境考虑” 内容。

该键字的格式如下：

BLANKS (响应指示器[ ‘说明’ ])

和 BLANKS 键字相关的响应指示器在记录中应是唯一的。就是说，这个指示器不能用在其它键字中，比如 CHANGE、DUP 及 VLDCMDKEY、功能键的键字、或者同一记录中其它字段的 BLANKS 键字。这是因为在输入操作时，如果字段包含非空白字符，OS/400 程序总是把响应指示器置为 OFF。OS/400 程序这样做就保证了在字段显示为空白时，响应指示器置为 ON，在不出现全空白时，响应指示器置为 OFF。

在程序编译时产生的清单中包括了可选的说明内容，它说明使用的指示器。这个说明在文件和程序中不起作用，仅仅是个注释。说明要撇号括起。如果两撇号之间的字符多于 50 个，则在程序清单中截为 50 个字符。

可选指示器对此键字无效。

#### 3.3.10.1 指定 BLANKS 键字

图 3-19 给出如何指定 BLANKS 键字。

注：图 3-20、图 3-21 和图 3-22 给出了限定 BLANKS 键字的几种情况。

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A                QTY1                5Y 0B 5 2BLANKS(01 'ON=QTY1 IS ALL BLANKS')
00040A                QTY2                5Y 0B 6 2BLANKS(02 'ON=QTY2 IS ALL BLANKS')

00050A	QTY3	5Y 0B 7 2BLANKS(03 'ON=QTY3 IS ALL BLANKS')
A		

图 3-19 规定 BLANKS 键字

这里共显示了三个数字字段 (QTY1、QTY2 和 QTY3)。如果显示站用户向字段内键入数值并且按了 Enter 键，则有以下情况：

键入字段的值	传递给程序的值	响应指示器
100	00100	OFF
0	00000	OFF
空白	00000	ON

注：如果显示站用户按了 Field Exit 键或 Erase Input 键，则该字段显示为空白，因为其内容为空 (null)。

### 3.3.10.2 限制 BLANKS 键字

在某些情况下，BLANKS 键字不能置响应指示器为 ON，而是限制它的使用。图 3-20、图 3-21 和图 3-22 说明了这些情况。

注：当字段为字符时情况不同，但对字符字段没有用 BLANKS 的必要。

在图 3-20 中，当显示全为空白（十六进制 40）或空零（十六进制 00）的输出/输入字段，且规定了影响字段显示的某些键字时，响应指示器置为 ON。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* 规定了 OVRATR:
00020A      R REC1                  PUTOVR
00030A      FLD1      10 0B 2 2BLANKS(50) OVRATR
00040A 78          DSPATR(HI)

00050A* 规定了 PUTRETAIN:
00060A      R REC2                  PUTRETAIN OVERLAY
00070A      FLD2      10 0B 2 2BLANKS(50)
00080A      R REC3                  OVERLAY
00090A      FLD3      10 0B 2 2BLANKS(50)
A                      PUTRETAIN
A
```

图 3-20 规定 BLANKS 键字 (I/O 字段为空白或空零)

对此例中所有记录格式来说，响应指示器 50 在程序第一次读字段时置为 ON（如果该字段在显示器上显示为空白）。然而，在随后的显示中，只有显示站用户再一次把该字段清为空白，响应指示器 50 才再一次被置为 ON。如果工作站用户没有再一次把该字段清为空白，则响应指示器一直为 OFF。

图 3-20 和图 3-22 给出字段首先被显示，然后被删除的情况。

在图 3-21 中，当一个可输入字段被另一字段覆盖时，导致第一个字段被删除，响应指示器没被置为 ON (即使输入缓冲区中的字段仍为全空白或全空零)。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R REC4          OVERLAY  
00020A 15    FLD4           10 0B 2 2  
00030A      FLD4           10 0B 2 5BLANKS(50)  
A
```

图 3-21 规定 BLANKS 键字 (字段因覆盖被删除)

在图 3-21 中，如可选指示器 15 在 REC4 第一次显示时为 OFF，则显示 FLD4 而不显示 FLD4。在读 REC4 时，如果 FLD4 为空白则响应指示器 50 被置为 ON，当 REC4 再一次显示时可选指示器 15 被置为 ON，则 FLD4 覆盖 FLD4 并删除它。在读 REC4 时响应指示器 50 又变为 OFF。(发生这种情况是因为 OS/400 程序在显示记录格式时把响应指示器置为 OFF，并不为不出现在显示器上的字段恢复为 ON，即使该字段因前面的 I/O 操作成为了全空白或全空零)。

在图 3-22 中，经初始显示后输出/输入字段不在随后的输出/输入操作中再次显示。(即使在输入缓冲区内该字段为全空白或全空零)。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R REC5          ERASEINP OVERLAY  
00020A 20    FLD5           10 0B 2 2BLANKS(50)  
00030A*  
00040A      R REC6          ERASEINP OVERLAY MDTOFF  
00050A 20    FLD6           10 0B 2 2BLANKS(50) DSPATR(MDT)  
A
```

图 3-22 规定 BLANKS 键字 (没有重显示)

在图 3-22 中，如果第一次显示 REC5 或 REC6，可选指示 21 为 ON，则显示 FLD5 或 FLD6。在读 REC5 或 REC6 时，如果 FLD5 或 FLD6 为空白，则响应指示器 50 置为 ON。然而，如果在第二次显示时可选指示器 20 为 OFF，则不显示 FLD5 或 FLD6。

### 3.3.11 BLINK (闪烁)

使用此记录层键字在显示所定义的记录时光标闪烁。在下一次对没有规定 BLINK 键字的记录进行输出操作时闪烁清除。

此键字没有参数。

可选指示器对此键字有效。

图 3-23 给出如何指定 BLINK 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00020A          R MASTER           BLINK
A

```

图 3-23 规定 BLINK 键字

### 3.3.12 BLKFOLD (空白折叠)

使用此字段层键定义命名的仅输出字段(但不是信息或程序到系统字段)使它输出时“溢出”到后续显示行上。此键字引起在数据中空白处进行“折叠”，而不是在显示行的结尾处。它使很长的`文本字段较易阅读。缺省做法是对数据在物理行结束处折叠。

此键字没有参数。

使用此键字时，字段长度并没有增加。因此，输出数据的一部分被截断是可能的。

不能在浮点字段中指定 BLKFOLD 键字。

可选指示器对此键无效。

图 3-24 给出如何指定 BLKFOLD 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A          FIELD1        638      2 1BLKFOLD
A

```

图 3-24 规定 BLKFOLD 键字

### 3.3.13 CAnn (命令注意)

使用此文件层或记录层键字来指定可以使用的功能键 (CA01 到 CA24)。它可作为一个命令注意 (CA) 键使用。没有输入数据从设备上传送过来。可用 01-99 做响应指示器。

关于在系统/36 环境的文件中指定 CAnn 键字所需特殊考虑的问题，参见附录 F“系统/36 环境考虑”内容。

此键字的格式为：

CAnn[(响应指示器[‘说明’])]

如果指定了此键字并且显示站用户按下了规定的功能键，发生以下事件：

- 输入缓冲区中所有其它功能键响应指示器都被置 OFF (十六进制 F0)。
- 使用 CAnn 键字指定的响应指示器被置为 ON (十六进制 F1)。
- 更新 OS/400 数据管理反馈区。
- 已在输入缓冲区中的数据保持不变，除非响应指示符 (如果指定了) 被置为 ON。
- 控制权返回到应用程序。

如果指定了响应指示器而且键已按下，响应指示器被置为 ON 且控制返回给用户程序。

(说明信息是与指示器相对应的，被高级语言编译器用做程序文本)。

如果显示站用户按下一个功能键，但该键没被指定为命令功能键 (CF) 或命令注意键，OS/400 程序会给显示站用户显示一条信息，提示该键在此时无效。

可在同一显示文件中组合使用 CA 和 CF 键字，但不能把同一键号同时指定给 CA 和 CF。例如，同一显示文件中包含 CA02 和 CF02 就不合法。

注：文件层的 CA 和 CF 键字可扩展到记录层，这就必须考虑键号。例如，如果在文件层指定了 CA02，又在记录层指定了 CF02，则 CF02 是错误的。

如果指定的 1 到 9 内的键号，则键号必须有前置零（例如 CA04）。

可选指示器对此键字是有效的。

### 3.3.13.1 有效性检验问题的考虑

当显示站用户按下 CF 键时，MDT 为 ON 的字段数据送到输入缓冲区，然后进行有效性检验，数据中存在的错误被检查出来，往显示器上发送错误信息。因为有效性检验是在数据被放到输入缓冲区后才做的，按下 CF 键后再按有效 CA 键会导致不正确的数据传给用户程序。只要用户程序在 CA 键按下时不处理输入数据，这种情况就不会成为问题。

这种情况可用以下两种方式避免：

- 不使用 CA 键。指定 CF 键仅对数据进行有效性检验。
- 如果使用 CA 键，不要指定下述有效性检验键字：

CHECK (M10)	CHECK (M11)
CHECK (VN)	CHECK (VNE)
COMP (EQ, NE, LT, NL, GT, NG, LE, GE)	
RANGE	VALUES

### 3.3.13.2 处理期间有效的功能键

根据一般规则，最后一次输出操作决定哪些功能键是有效的。但对此规则有以下例外情况：

- 当一个操作没有向显示器传送数据时，各种功能键的有效性没有改变。这样的操作包括：
  - 一对子文件记录进行的输入操作。
  - 一对子文件记录的更新。
  - 一对子文件控制记录进行的输出操作，它仅是清除、删除或初始化子文件，而不显示子文件或子文件控制记录。
- 选择 ERRMSG（错误信息）或 ERRMSGID（错误信息标识）键字来显示一条错误信息的输出操作，在显示错误信息时也可以选择 CA 或 CF 键。
- 如果对子文件指定了 MNUCNL(菜单取消)。MNUBARSW(菜单条开关)或 SFLDROP (子文件丢失) 键字，对 SFLDROP 键字规定的 CA 或 CF 键的有效性由最后一次输出操作决定。然而，只要显示子文件，CA 或 CF 键（有效时）仅起到一个丢失（Drop）键的作用。
- 如果对子文件指定了 SFLFOLD (子文件折叠) 键字，则为 SFLFOLD 键字确定的 CA 或 CF 键字有效性就由最后一次输出操作决定。然而，只要显示子文件，有效的 CA 或 CF 键只起到一个折叠（Fold）键的作用。

- 如果使用 SFLDROP 或 SFLFOLD 的两个子文件同时显示,在 SFLDROP 和 SFLFOLD 键字中应指定相同的功能键。如果不同,只有对最近显示的子文件指定的功能键有效。按下功能键影响有光标的子文件。如果光标不在子文件中,则功能键影响上一个子文件。
- 如果使用 SFLENTER (子文件送入) 的两个子文件同时显示,只有在最近显示的子文件中为 SFLENTER 键字指定的 CA 或 CF 键起到输入键 (Enter) 的作用。

注: 以下键字起到类似 CA 键的作用:

CLEAR、HELP、HOME 和 PRINT (同时指定响应指示器)。

图 3-25 示意了如何指定 CAnn 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00011A                                CA01(91 'End of Program')
00012A                                CA02(92)
00013A                                CA03
A
```

图 3-25 规定 CAnn 键字

### 3.3.14 CFnn (命令功能)

使用此文件层或记录层键字来确定键 CF01 到 CF24 功能键。它和不传送修改了的数据的命令注意可以使用的键 (CA) 相反,这些功能键用作命令功能键 (CF) 来传送修改了的数据。响应指示器用 01 到 99。

在系统/36 环境的文件中规定 CFnn 键字所需特殊考虑的问题,参见附录 F “系统/36 环境考虑”内容。

此键字的格式为:

CFnn[ (响应指示器[ ‘说明’ ]) ]

如果规定了此键字,且显示站用户按下了指定的功能键,则发生以下情况:

- 输入缓冲区中所有其它功能键的响应指示器置为 OFF (十六进制 F0)。
- 在 CFnn 键字中规定的响应指示器置为 ON (十六进制 F1)。
- OS/400 数据管理反馈区被更新。
- 从各设备接收的数据放入输入缓冲区中。
- 控制权返给用户程序。

如果指定了响应指示器且按下了功能键,则响应指示器置为 ON,并与输入数据一起返回给用户程序。如果没有指定响应指示器,则仅有输入数据返给用户程序。(与响应指示器有关的说明信息是被高级语言编译器使用,有助于程序文档。)

如果显示站用户按下一个功能键,而此键尚未被定义为 CF 键或 CA 键,则 OS/400 程序会显示一条信息,提示用户这个键此时无效。

在同一显示文件中可以组合使用 CF 和 CA 键字,但不能把一个键号同时指定给命令注意键 (CA) 和命令功能键 (CF)。例如,同一显示文件中有 CF01 和 CA01 是无效的。

注：文件层的 CA 和 CF 键扩展至记录层。这时要考虑键号。例如，如果在文件层指定了 CA02，又在记录层指定了 CF02，则 CF02 无效。

如果指定 1 到 9 范围内的键号，在键号中必须写上前置零（例如 CF03）。

可选指示器对此键字是有效的。

### 3.3.14.1 处理时有效的功能键

根据一般规则，最后一次输出操作决定哪些功能键是有效的，但对这条规则有以下例外情况：

- 当一个操作没有向显示器送数据时，各功能键的有效性没有改变。这样的操作包括：
  - 一对子文件记录进行的输出操作。
  - 一对子文件记录的更新。
  - 一对子文件控制记录进行的输出操作。
- 仅仅是清除、删除或初始化子文件，而不显示子文件或子文件控制记录。
- 由选择 ERRMSG 或 ERRMSGID 的输出操作，也可以选择有效的 CA 或 CF 键，同时显示错误信息。
- 如果为子文件指定了 SFLDROP 键字，则为 SFLDROP 指定的 CA 或 CF 键的有效性就由最后一次输出操作决定。只要显示子文件，CA 或 CF 键仅起到一个丢失键（Drop）的作用。
- 如果为子文件指定了 SFLFOLD 键字，则为 SFLFOLD 指定 CA 或 CF 键的有效性就由最后一次输出操作决定。只要显示子文件，CA 或 CF 键在有效时只起到一个折叠键（Fold）的作用。
- 如果使用 SFLDROP 或 SFLFOLD 的两个子文件同显示，对 SFLDROP 和 SFLFOLD 应指定同一个功能键。如果不同，则只有为最近显示的子文件指定的功能键是有效的。按下功能键后将影响有光标的子文件。如果光标没有在子文件中，则功能键影响上一个子文件。
- 如果使用 SFLENTER 的两个子文件同时显示，则只有在最近显示的子文件中为 SFLENTER 键字指定的 CA 或 CF 键起到输入键(Enter)的作用。按下输入键时光标的位置决定了哪个子文件受到影响。

注：ROLLUP 和 ROLLDWN 键字起到类似 CF 键的作用。

图 3-26 给出了如何指定 CFnn 键字。

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8	
00010A	CF01(91 'End of Program')
00020A	CF02(92)
00030A	CF03
A	

图 3-26 指定 CFnn 键字

### 3.3.15 CHANGE (修改)

使用此记录层或字段层键字在如下条件下把为一个输入操作定义的响应指示置为 ON：

- 此键字是在记录层定义的，且记录格式中所有可输入字段的修改数据标记(MDT)置为 ON。
- 此键字是在可输入字段规定的，而且此字段的修改数据标记 (MDT) 置为 ON。

关于如何在系统 36 环境文件中指定 CHANGE 键字，详见附录 F “系统 36 环境考虑” 内容。

此键字的格式为：

CHANGE (响应指示器[ ‘说明’ ])

在显示站用户向一个可输入字段输入数据时，或者用户程序为该字段的输出操作选择了显示属性 (DSPATR (MDT)) 键字时，该字段 MDT 被置为 ON。如果 MDT 是用 DSPATR (MDT) 键字置为 ON 的，字段中的数据可能并没有修改，即使 MDT (从而是为 CHANGE 键字指定响应指示器) 被置为 ON。注意即使显示站用户向该字段内键入与初始显示相同的数据（比如向空白字段键入数据然后再把该字段清为空白），MDT 还是被置为 ON。

注：在按下命令注意键 (CAnn, Help, Pring, Home 或 Clear) 时，没有把 CHANGE 响应指示器置为 ON。

在 OS/400 程序检测到有效性错误并且再次显示有错误信息的记录时，所有因向字段中键入数据而被置为 ON 的 CHANGE 键字响应指示器一直保持 ON，直到所有有效性检验通过，记录数据被传送给用户程序为止。

在程序编译期间产生的清单中包括了可选说明，用来解释所用的指示器。这个说明在文件程序中不起作用，仅仅是个注释。说明要用两个撇号括起。如果两个撇号间的字符多于 50 个，在程序打印输出时截断为 50 个。

可选指示器对此键字是有效的。

图 3-27 给出如何在字段层指定 CHANGE 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          FLDX           5   B  8  2CHANGE(67 'FLDX was changed')
A
00020A          FLDY           3   I  8  30CHANGE(68 'FLDY was entered')
A
```

图 3-27 在字段层规定 CHANGE 键字

图 3-28 给出了如何在记录层规定 CHANGE 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R REC1
00020A
00030A*
00040A          FIELD1        10   B  3  2
00050A          FIELD2        5    B  5  2
00060A          FIELD3        6    B  7  2
00070A          FIELD4        3    I  9  2DFT('ABC')
A
```

图 3-28 在记录层规定 CHANGE 键字

### 3.3.16 CHCACCEL (选项加速说明)

使用此字段层键字来为下拉记录中的单选选择字段对某一选项的加速键规定说明。

注：CHCACCEL 只描述加速键的说明。它不能定义功能键。

此键字的格式为：

CHCACCEL (选项号 加速说明)

选项号参数指定此键字用的单选字段的选项号，有效值为 1 到 99。

加速说明参数规定标识加速键的描述。此参数可以用下面两种格式之一规定：

- 以字符串形式：‘加速说明’。
- 以程序到系统字段形式：&字段名。

规定的字段必须和选择字段存在于同一记录中，而且必须定义为用法为 P 的字符字段。说明放在距选项说明最大长度右端三个空格处。加速说明的最大长度由最长的选项说明长度决定。这两者合起来不能超出文件规定的最小显示尽寸的宽度。

CHCACCEL 键字仅允许用于在记录层指定键字 (PULLDOWN) 的下拉记录的单选选择字段。(在同一字段规定 SNGCHCFLD 键字。)

可选指示器对此键字是无效的。

图 3-29 给出了如何规定 CHCACCEL 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R PULLEDIT          CF04 CF06
A
A
A      F1           2Y 0B  1  2SNGCHCFLD
A
A
A
A
A
A
A
A
A      F6           2A  P
```

图 3-29 规定 CHCACCEL 键字

在此例中，Choice1 和 2 分别有加速键 CF04 和 CF06。当显示下拉菜单时，字符说明‘F4’出现在‘Undo’右边，其间有三个空格；字段 F6 的说明出现在‘Mark’右边，其间有三个空格。最长的 Choice 说明决定了所有 Choice 说明的长度。对 ACCEL 说明也是这样。于是 ACCEL 说明起始于距最长 Choice 右端三个空格处。

### 3.3.17 CHCAVAIL (可用选项的颜色/显示属性)

使用此字段层键字来指定在菜单条、按钮、选择字段或子文件选择表中单选或多选中可用选项所用的颜色和显示属性。

键字的格式为：

CHCAVAIL ([颜色][显示属性])

必须至少规定一个参数。

颜色参数指明了在彩色显示站上显示的，使用键字 MNUBARCHC、CHOICE 或

**PSHBTNCHC** 的字段的选项说明颜色。选项说明也能来自子文件选择表中单选或多选的说明。参数是以 (\*COLOR 值) 的表达式来规定的。

颜色参数的合法值为：

值	含义
BLU	蓝色
GRN	绿色
PNK	粉红色
RED	红色
TRQ	蓝绿色
YLW	黄色
WHT	白色

如果没有指定颜色参数，菜单条中可用选项的缺省颜色为绿色。选择字段中可用选项的缺省颜色为绿色。在单色工作站上此参数无效。

显示属性参数表明了字段的 **MNUBARCHC** 或 **CHOICE** 键字所指定的选项说明的显示属性。该参数以 (\*DSPATR 值 1<值 2〈值 3……>>) 的表达式规定。

显示属性的有效值为：

值	含义
BL	闪烁
CS	列分隔符
HI	高亮度
ND	不显示
RI	反象
UL	下划线

菜单条中缺省显示属性为高亮度。选择字段中缺省显示属性为正常（或低）亮度。

注：显示属性 CS, HI 和 BL 可使在 5292, 3179, 3197 型 C1 和 C2, 3477 型 FC, 3486, 3487 型 HC 和 3488(1)工作站上的字段显示为彩色字段。使用显示属性 HI, RI 和 UL 时不显示分隔线。关于 COLOR 键字的使用，详见 3.3.30 “COLOR (颜色)” 内容。

**CHCAVAIL** 键字只允许用在规定了一个或多个 **PSHBTNCHC**, **CHOICE** 或 **MNUBARCHC** 键字的字段中。它也可用在子文件控制记录使用 **SFLSNGCHC** 或 **SFLMLTCHC** 键字的子文件中。

可选指示器对此键字有效。

(1)取决于连接显示设备的监视器

图 3-30 给出如何指定 CHCAVAIL 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R RECORD          MNUBAR
A      F1           2Y 0B  1  2
A                      MNUBARCHC(1 PULLFILE 'File      ')
A                      MNUBARCHC(2 PULLEDIT 'Edit      ')
A                      CHCAVAIL((*COLOR YLW))
A
```

图 3-30 指定 CHCAVAIL 键字（例 1）

在图 3-30 中，彩色显示器上，菜单条的选项显示为黄色。对单色显示器，菜单条缺省为高亮、绿色。

在图 3-31 中，选择字段的选项显示下划线。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A  
A      R RECORD  
A          F1      2Y 0B  2  5SNGCHCFLD CHECK(ER)  
A                      CHOICE(1 'Choice number 1')  
A                      CHOICE(2 'Choice number 2')  
A                      CHCCTL(1 &CHCCTL1)  
A                      CHCCTL(2 &CHCCTL2)  
A                      CHCAVAIL((*DSPATR UL))  
A
```

图 3-31 指定 CHCAVAIL 键字（例 2）

在图 3-32，在彩显中单选的选择列表显示为黄色，可用的选项有下划线。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A      R SFLREC           SFL  
A      CTLFLD        1Y 0H    SFLCHCCTL  
A      R SFLCTRLCD       SFLCTL(SFLREC)  
A                      SFLSNGCHC  
A                      :  
A                      :  
A                      CHOICE(1 'Choice number 1')  
A                      :  
A                      :  
A                      CHOICE(2 'Choice number 2')  
A                      :  
A                      :  
A                      CHCAVAIL((*DSPATR UL))  
  
A                      CHCAVAIL((*COLOR YLW))
```

图 3-32 指定 CHCAVAIL 键字（例 3）

### 3.3.18 CHCCTL（选项控制）

在选择字段中使用此键字来控制该字段选项的可用性。

此键字的格式为下述二者之一：

CHCCTL(选项号 &控制字段 [信息标识符 [信息库/] 信息文件])

或

CHCCTL(选项号 &控制字段 [&信息标识 [&信息库/] &信息文件])

选项号参数是必须的，它指定了使用此键字的选项。有效为 1 到 99。

也要求有控制字段参数，它指定了一个一字节的数字隐藏字段名，在输出时，此字段包含该选项的控制值。此字段必须与你定义的字段在同一记录中，并且必须定义为 Y(数字型)，长度为 1，小数位为 0，用法为 H 的属性。在多选的选择字段输入时，选择字段指明选择了哪个字段。

下图为隐藏字段的控制值及它在输入输出时的含义：

图 3-33 隐藏字段的控制值

控制值	输出含义	输入含义
0	可用	未选择
1	已选择	已选择
2	不可用，除非选项的帮助有效，否则光标不能置于选项上	
3	不可用，允许光标置于选项上	
4	不可用，即使选项的帮助有效，也不能置光标于选项上	

注：上述对光标的限制仅适用于连在支持不可编程工作站增强接口控制器上的显示器。如果用的是其它显示器，则对光标无限制。

信息标识和信息文件参数是可选的，在用户选择了不可用选项时指定显示的信息。如果没有指定这些参数，系统发出一条缺省信息 CPD919B。如果信息标识用一个字段，则该字段必须在所定义的记录中，且定义为数据类型为 A，用法为 P，长度为 7 的属性。

使用了信息标识就要求有信息文件参数。如果没有规定库参数，在程序运行时就使用 \*LIBL 来找信息文件。如果信息库或信息文件使用一个字段，则该字段必须在所定义的记录中，且必须定义为数据类型为 A，用法为 P，长度为 10。

如果为某一字段指定了 CHCCTL 键字，则必须对该字段用相同的选项号指定 CHOICE 或 PSHBTNCHC 键字。

可选指示符对此键字无效。

图 3-34 给出如何指定 CHCCTL 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A : 
A : 
A F1 2Y 0B 3 35SNGCHCFLD
A CHOICE(1 '>Undo ')
A CHOICE(2 '>Mark ')
A CHOICE(3 '>Copy ')
A CHCCTL(1 &CTLUNDO MSG1112 QUSER/A)
A CHCCTL(2 &CTLMARK &MSG &LIB/&MSGF)
A CHCCTL(3 &CTLCOPY)
A CTLUNDO 1Y 0H
```

A	CTLMARK	1Y 0H
A	CTLCOPY	1Y 0H
A	MSG	7A P
A	MSGF	10A P
A	LIB	10A P
A		

图 3-34 指定 CHCCTL 键字

在使用一个连在支持不可编程工作站增强接口控制器上的图形显示站时,选择字段看起来如下图:

PICTURE 12



### 3.3.19 CHCSLT (被选择的选项颜色/显示属性)

使用此字段层键字来指定在菜单条或选择字段中显示已选择的选项时,使用的颜色或显示属性。还可使用此键字在一个有 PULLDOWN(\*NOSLTIND)键字的下拉菜单中,规定选择字段选中选项的颜色和显示属性。

此键字的格式为:

CHCSLT ([颜色][显示属性])

必须至少指定一个参数。

颜色参数指定了在彩色工作站上由 MNUBARCHC 或 CHOICE 键字指定的选项说明的颜色。此参数的定义为\*COLOR 值的表达式。

颜色参数的合法值为:

值	含义
BLU	蓝色
GRN	绿色
PNK	粉红色
RED	红色
TRQ	蓝绿色
YLW	黄色
WHT	白色

如果没有指定颜色参数,菜单条中已选择选项的缺省颜色为白色。在不显示选择字符的下拉菜单中,选择字段的已选项缺省颜色是白色。在单色显示器中颜色参数被忽略。

显示属性参数指明了字段的 MNUBARCHC 或 CHOICE 键字所指定的选项说明的显示属性。该参数用 (\*DSPATR 值 1<值 2<值 3……>>) 表达式。

显示属性的合法值为:

值	含义
---	----

BL	闪烁
CS	列分隔符
HI	高亮
ND	不显示
RI	反象
UL	下划线

菜单条中选中选项的缺省显示属性为正常(或低)亮度。在不显示选择字符的下拉菜单的选择字段中的已选项，它的缺省显示属性为高亮。

注：显示属性 CS, HI 和 BL 可使在 5292, 3179, 3197 型 C1 和 C2, 3477 型 FC, 3486, 3487 型 HC 和 3488 (2) 工作站上显示的字段为彩色字段。使用显示属性 HI, RI 和 UL 时不显示分隔线。关于 COLOR 键字的使用，详见 3.3.30 “COLOR” 内容。

如果为某字段指定了这个键字，则必须也为它指定 MNUBARCHC 键字或 CHOICE 键字。如果此字段指定了 CHOICE 键字而不是 MNUBARCHC 键字，则包含此字段的记录必须指定 PULLDOWN(\*NOSLTIND)。

当对子文件控制记录规定 CHCSLT，那么也应同时规定 SFLSNGCHC 或 SFLMLTCHC 键字。

可选指示器对此键字有效。

(2)取决于连接显示设备的监视器

图 3-35 给出如何指定 CHCSLT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R RECORD           MNUBAR
A      F1                 2Y 0B  1  2
A                           MNUBARCHC(1 PULLFILE 'File      ')
A                           MNUBARCHC(2 PULLEDIT 'Edit      ')
A                           CHCSLT((*COLOR PNK) (*DSPATR RI))
A
```

图 3-35 指定 CHCSLT 键字

在此例中，如果在彩色显示器上选择了某选项，菜单条以粉红色反象方式显示。

(3)取决于连在显示设备上的监视器。

### 3.3.20 CHCUNAVAIL (不可用选项的颜色/显示属性)

使用此字段层键字来指定在选择字段或按钮字段中显示不可用选项所用的颜色或显示属性。

此键字的格式为：

CHCUNAVAIL ([颜色][显示属性])

必须至少规定一个参数。

颜色参数指定了在彩色工作站上由 CHOICE 键字指定的选项说明的颜色。指定此参数用 \*COLOR 值的表达式。

颜色参数的合法值为：

值	含义
BLU	蓝色
GRN	绿色
PNK	粉红色
RED	红色
TRQ	蓝绿色
YLW	黄色
WHT	白色

如果没有指定颜色参数，选择字段中不可用选项的缺省颜色为蓝色。此参数在单色显示器上被忽略。

显示属性参数指明了字段中 CHOICE 或 PSHBTNCHC 键字所指定选项说明的显示属性。规定该参数用 (\*DSPATR 值 1<值 2<值 3……>>) 的表达式。

值	含义
BL	闪烁
CS	列分隔符
HI	高亮
ND	不显示
RI	反象
UL	下划线

在单色显示器上，选择字段中不可用选项的缺省显示属性为正常（或低）亮度。而且不可用选项的每个字符被星号(\*)复盖。

注：显示属性 CS, HI 和 BL 可使在 5292, 3179, 3197 型 C1 和 C2, 3486, 3487 型 HC, 和 3488(3)工作站上显示的字段为彩色字段。使用显示属性 HI, RI 和 UL 时不显示分隔线。关于 COLOR 键字用法，详见 3.3.30 “COLOR” 内容。

此键字仅用于有一个或多个 CHOICE 或 PSHBTNCHC 键字的字段中。当它用在子文件控制记录中时，也同时要指定 SFLSNGCHC 或 SFLMLTCHC 键字。

可选指示器对此键字有效。

(3)取决于连在显示设备上的监视器。

图 3-36 给出如何指定 CHCUNAVAIL 键字：

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R RECORD
A      F1      2Y 0B  2  5SNGCHCFLD CHECK(ER)
A                      CHOICE(1 'Choice number 1')
A                      CHOICE(2 'Choice number 2')
A                      CHCCTL(1 &CHCCTL1)
A                      CHCCTL(2 &CHCCTL2)
A                      CHCUNAVAIL((*COLOR TRQ))
A      :
A      :
A
```

图 3-36 指定 CHCUNAVAIL 键字

在此例中，选择字段的不可用选项在彩色显示器上显示为蓝绿色。

### 3.3.21 CHECK (检验)

根据所指定的参数不同，用此键字可完成以下功能：

功能	有效参数值
有效性检验	AB, ME, MF, M10, M11, M11F, VN, VNE
键盘控制	ER, FE, LC, RB, RZ
光标控制	RL, RLTB

此键字的格式为：

CHECK (有效性检验码[……])  
CHECK (键盘控制码[……])  
CHECK (光标控制码)

对下述其它 DDS 键字，建议用 CHECK 键字替换：

CHECK(ER)等价于 AUTO(RA)  
CHECK(LC)等价于 LOWER  
CHECK(RB)等价于 AUTO(RAB)  
CHECK(RZ)等价于 AUTO(RAZ)

下述 CHECK 键字实现的功能也可用修改输入缺省键字 (CHGINPDFT) 在文件、记录或字段层上指定：

CHECK 键字	CHGINPDFT 等价键字
CHECK(FE)	CHGINPDFT(FE)
CHECK(LC)	CHGINPDFT(LC)
CHECK(ME)	CHGINPDFT(ME)
CHECK(MF)	CHGINPDFT(MF)

可选指示器仅对 CHECK(ER) 和 CHECK(ME) 有效。

#### 3.3.21.1 有效性检验

在字段层上使用 CHECK 键字来指定 OS/400 程序或设备对键入可输入字段（仅输入或输出/输入）的数据的有效性进行检验。

CHECK 使用一种或多种编辑/检验算法来确认数据是否有效。如果没有满足某一特定的编辑/检验算法，则显示一条错误信息。

注：关于定义用户指定的信息。详见 3.3.23 “CHKMSGID (检验信息标识)” 内容。

有效的编辑/检验码有：

编辑/检验码	含义
AB	允许空白
ME	强制输入
MF	强制填充
M10	IBM 模 10 自检算法
M10F	IBM 模 10 自检算法
M11	IBM 模 11 自检算法
M11F	IBM 模 11 自检算法
VN	有效名
VNE	有效扩展名

**AB (允许空白):** 如果对一可输入字段的某些有效性检验失败, 可在文件层、记录层或字段层指定此码, 它允许输入全空白来满足有效性检验。这使得在工作站用户把光标定位在某一字段上但把它置为空白(比如, 通过按下 Erase Input 键, Field Exit 键或空格键)时, 仍能把数据传送给程序。例如, FLD1 是一个指定了 CHECK(M10 ME) 的可输入字段。如果工作站用户意外地键入该字段, 则必须满足 M10 算法。若指定了(CHECK(M10 ME AB)) 就允许用户把该字段置为空白来满足有效性检验。

如果在文件层指定了这个键字, 则它对文件中所有规定了有效检验码的可输入字段有效, 同样, 在记录层指定了有效性检验键字, 则此键字适用于记录中所有规定了有效检验码的可输入字段。在字段层上, 它总是和其它有效性检验键字(CHECK(M10, M10F, M11, M11F, VN, VNE), CHKMSGID, COMP, RANGE 或 VALUES)一起指定。

如果对某一字段指定了 SFLROLVAL 或 SFLRCDNBR 键字, 就不应再指定 CHECK(AB)。

CHECK(AB)可用在做引用的数据库文件中。

如果和其它有效性检验功能一起指定 CHECK(AB), 则注意处理过程按如下顺序进行:

1. 为下述任何情况:

a. 在 35 列指定键盘转换属性(比如字母数字转换或仅为数字), 可以限制键进输入某些字符。

b. 如果键盘转换属性是数字转换, 则数据类型(字符或数字)由 36 至 37 列(小数位)设定, 并限制键进输入特定的字符。

c. 如果指定了 CHECK(FE), CHECK(MF) 和 CHECK(ME) 键字, 则限制键进输入。

2. 为下述情况之一:

a. 如果指定了 CHECK(AB) 键字, 数据管理向程序传送输入数字(字符字段为空白, 数字字段为零)。不再做进一步的有效性检验。

b. 如果没有指定 CHECK(AB), 则数据管理在向程序传送数据之前要执行下述有效性检验功能: CHECK(VN), CHECK(VNE), CHECK(M10), CHECK(M10F), CHECK(M11), CHECK(M11F), COMP(.....), RANGE(.....), VALUES(.....)。

不能对浮点字段(35 列为 F)指定 CHECK(AB) 键字。

可选指示器对此键字无效。

**ME (强制输入):** 此码指定了必须至少有一个字符数据(可为空白)键入字段中。如果目前显示器上的字段没被修改, 显示站不能强行进行强制输入。若需强行做强制输入, 必须为显示记录中至少一个字段指定 DSPATR(MDT) 键字, 然后对记录中其它字段, 强行使用 CHECK(ME) 键字。然而, 因为设备不能确定用户是否向指定了 DSPATR(MDT) 和 CHECK(ME) 的字段中键入了数据, 所以还必须指定 DSPATR(ND), 这样该字段才不被显示。

可选指示器对此键字有效。

**MF (强制填充):** 本码指定若改变字段的某一部分, 字段中每一位必须输入一个字符。空白视为有效字符。

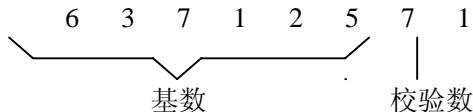
本码不能与键盘控制码(RB 或 RZ)或 WRDWRAP 键字共同指定。

可选指示器对此键字无效。

**M10/M10F 或 M11/M11F (IBM 模 10 或模 11 算法):** 用此码指明键入到字段的数据必须满足 IBM 模 10(M10 或 M10F) 或模 11(M11 或 M11F) 自检算法。若指定了 CHECK(M10) 或 CHECK(M11), 在按下 Enter 键或功能键时, 自检算法确认在该字段中有一个合法的模 10 或模 11 数字。若指定了 CHECK(M10F) 或 CHECK(M11F), 在用户向字段中键入数据时, 自检算法确认该字段中含有一个合法的模 10 或模 11 数字。不能为同一字段同时指定模 10 或模 11 自检算法, 也不能为同一字段同时指定同一种算法的两种格式。

自检字段包含两部分: 基数和一个校验位。校验位是字段中的最右一位。基数和校验位

共同构成数据库中一个字段（例如帐号）。下图是一个八位自检字段的例子：



如何使用 CHECK(M10), CHECK(M10F), CHECK(M11)和 CHECK(M11F)的内容，  
详见《应用显示编程》一书。

注：

- 1.OS/400 程序支持的数字字段的最大长度为 31 位。
- 2.不能在指定 CHECK(M10), CHECK(M10F), CHECK(M11)和 CHECK(M11F)同时指定 COMP(EQ)关键字。
- 3.不能为浮点字段 (35 列为 F) 指定 CHECK(M10), CHECK(M10F), CHECK(M11) 和 CHECK(M11F)关键字。
- 4.在有 USRDSPMGT 键字的文件中不能再指定 CHECK(M10F)或 CHECK(M11F)关键字。
- 5.在有 CHKMSGID 或 WRDWRAP 键字的字段中不能再指定 CHECK(M10F)或 CHECK(M11F)关键字。

对基数的每一位，都有一个模 10 权因子和一个模 11 权因子。位置是从最右位数字算起的（不包括校验位）。

位置 1, 3, 5, ……, 31 的模 10 权因子为 2。位 2, 4, 6, ……, 30 的模 10 权因子为 1。位 1, 2, ……, 31 的模 11 权因子分别为 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, ……, 2, 3, 4, 5, 6, 7, 2。

按如下步骤计算模 10 自检数：

- 1.把基数的个位及隔位数字乘以 2。
- 2.把乘积中的数字和基数中的没乘 2 的数字加起来得到一个和。
- 3.从下一个比此和大的且以零结尾的数中减去此和。

差即为自检位。

例：

基数:	6	1	2	4	8
个位及隔位:	6		2		8
乘以权因子 2:	×2		×2		×2
乘积:	12		4		16

未乘 2 的数字: 1 4

乘积数字加上未乘 2 的基数数字之和:  $1 + 2 + 4 + 1 + 6 + 1 + 4 = 19$

下一个大于此和

且以 0 结尾的数字: 20

做减法: - 19

自检: 1

按以下步骤计算模 11 自检数：

1.为基数的每位数字分配一个权因子。权因子由个位数字起到高位分别为：2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, 2, 3, ……。例如，基数 991246351 分配的权因子如下：

基数 991246351

权因子 432765432

2. 把每位数字与其权因子相乘。

3. 把乘积相加。

4. 把相加所得和除以 11。

5. 用 11 减去余数。

差即为自检数。

例：

基数	13739
权因子	$\times 6 \times 5 \times 4 \times 3 \times 2$
把每位数字与权因子相乘	6 15 28 9 18
乘积相加	$6 + 15 + 28 + 9 + 18 = 76$
相加和除以 11	$76 / 11 = 6$ 余数 10
用 11 减去余数	$11 - 10 = 1$
自检数	1

注：如果第 4 步中余数为 0，则自检数为 0。如果余数为 1，则此基数无自检数，必须确保定义为自检字段的字段中没有使用这样的基数。

可选指示器对 CHECK(M10), CHECK(M10F), CHECK(M11) 或 CHECK(M11F) 键字无效。

VN (有效名)：用此码来指定键入字段中的数据必须是有效的简单名。首字符必须是\$, #, @或 A 到 Z。其余字符必须是字母数字 (\$, #, @, A 到 Z, 0 到 9, 或者下划线 (\_)), 且不能嵌入空格。

若为某字段指定了 CHECK(VN) 键字，则该字段必须为字符字段（键盘转换为 A, N, X, W 或 I），且必须为可输入字段（用途为 I 或 B）。

CHECK(VN)不能和任何以下键字同时指定：

CHECK(M10)	CHECK(VNE)
CHECK(M10F)	COMP
CHECK(M11)	RANGE
CHECK(M11F)	VALUES

可选指示器对此键字无效。

VNE (有效扩展名)：用此码来指定键入字段的数据必须为有效的扩展名。

若为某字段指定了 CHECK(VNE) 键字，则该字段必须为字符（键盘转换为 A, N, X, 或 I），可输入的（用途为 I 或 B），且最大长度为 255 个字符的字段。

如果名字不是用双引号括起的，则：

- 第一个字符必须是 A 到 Z, a 到 z, #, \$ 或 @
- 其余字符必须是 A 到 Z, a 到 z, #, @, \_ 或句点
- 小写字符被转换为大写

如果名字是用双引号括起的，则：

- 除以下字符外都可以使用：

十六进制 OO 到 3F	(设备控制)
十六进制 FF	(设备控制)
十六进制 4O	(空格)
十六进制 5C	(*)
十六进制 6F	(?)
十六进制 7D	(')
十六进制 7F	(")

- 小写字母仍为小写
- 系统在不需要双引号时就把它们去掉。(如果此名字语法满足不加双引号名字的要求, 所有字符都大写。)

CHECK(VNE)不能与以下键字同时使用:

CHECK(M10)	CHECK(VN)
CHECK(M10F)	COMP
CHECK(M11)	RANGE
CHECK(M11F)	VALUES

可选指示器对此键字无效。

### 3.3.21.2 例子

图 3-37 给出了如何指定有效性检验 (CHECK) 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00060A      R RECORD1                  CHECK(AB)
00070A      FIELD11      10  B  1  2TEXT('CHECK(AB) not propagated to +
00080A                      this field')
00090A      FIELD21      10  B  1  22CHECK(VN)
00100A                      TEXT('CHECK(AB) is propagated to +
00110A                      this field')
00120A      FIELD31      10  B  1  42CHECK(VNE)
00130A                      TEXT('CHECK(AB) is propagated to +
00140A                      this field')
00150A*
00160A      R RECORD2
00170A      FIELD12      10  B  2  2CHECK(VN)  CHECK(AB)
00180A      FIELD22      10  B  2  22CHECK(VN AB)
00190A      FIELD32      1  B  2  42CHECK(AB)  VALUES('A' 'B' 'C')
00200A      FIELD42      10  B  2  62CHECK(VN)
00210A      FIELD52      10  B  3  2CHECK(VNE)
00220A      FIELD62      10  B  3  22CHECK(VNE AB)
00230A      FIELD72      10  B  4  1CHECK(ME MF)
00240A      FIELD82      8  OB  4  22CHECK(M10)
00250A      FIELD92      10  OB  4  42CHECK(M11)
A
```

图 3-37 指定有效性检验 CHECK 键字

### 3.3.21.3 键盘控制

如果 CHECK 键字用键盘控制码, 则它控制特定的数据入口特征。有效的控制码是:

键盘控制码	含义
ER	记录结束, 等价于 AUTO(RA)
FE	字段退出检验

LC	小写，等价于 LOWER
RB	右对齐，空格填充；等价于 AUTO(RAB)
RZ	右对齐，零填充；等价于 AUTO(RAZ)

**ER (记录结束):** 用此码可使工作站用户不必按输入 (Enter) 键。工作站用户向字段最后一个位置键入字符 (包括空格)，记录就由设备送出，就象按下了输入 (Enter) 键一样。如果同时为字段指定了 DSPATR(SP)，则在工作站用户选择该字段时，记录就由设备送出。此项功能在用户为记录最后一个字段键入数据时使用。

可选指示器对此键字有效。

**FE (字段退出检验):** 此码指定了用户要按字段退出键才能往前进入下一个输入字段。如果不按字段退出键，即使字段低位字符已经键入，光标仍留在那个位置。如果用户按下了其它键，将发生错误。

如果想为记录格式中所有可输入字段指定 CHECK(FE)，可在记录层指定 CHGINPDFT(FE)。如果想为文件中所有可输入字段指定 CHECK(FE)，可在文件层指定 CHGINPDFT(FE)。

字段退出键包括 Field Exit, Field +, Field - 和光标移动键。哪些键是有效的字段退出键取决于所用键盘的结构。

此码仅用于工作站用户可键入数据的输入字段。

可选指示器对此键字无效。

**LC (小写):** 为仅输入或输出/输入字段指定 CHECK(LC) 键字，允许工作站用户键入小写的 a - z。工作站用户键入字符的方式 (大写或小写)，与在显示器上显示的，及返给用户程序的方式相同。

若想为记录中所有字符可输入字段指定 CHECK(LC)，可在记录层指定 CHGINPDFT(LC)。若想为文件中所有字符可输入字段指定 CHECK(LC)，可在文件层指定 CHGINPDFT(LC)。

用户程序可显示既包含大写字符，又包含小写字符的字段。

如果指定了此键字，小写 a - z 仍为小写。如果没有指定，小写字符 a - z 转换为大写。

CHECK(LC) 对数据入口键盘无效。数据入口键盘不支持小写字符 a - z。

可选指示器对此键字无效。

图 3-38 给出了如何指定 CKECK(LC) 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00100A          NAME        30   I  3  2CHECK (LC)
A
```

图 3-38 指定 CHECK(LC) 键字

**RB (右对齐空格填充):** 此码把键入字段中的数据移到最右边，并把其余位置用空格填充。

不必为有符号数字字段指定 CKECK(RB)。右对齐空格填充是它的缺省值。当它的值为零时，则显示为全空格。

OS/400 程序在把数字字段传给用户程序时，把空格转换为零。

可选指示器对此键字无效。

**RZ (右对齐零填充):** 此码把键入字段中的数据移到最右边，并把其余位置用零填充。

对有符号数字字段，若没有指定 CHECK(RZ)，则缺省为 CHECK(RB)。

可选指示器对此键字无效。

编程时使用 CHECK(RB) 和 CHECK(RZ) 应考虑如下问题：

- 按下 Field Exit, Field + 或 Field - 键就是用右对齐功能。如果使用光标移动键退出一个右对齐字段，则该字段并没有右对齐，它还和原来键入的一样。
- 大于 15 个字符的右对齐字段会使键盘输入减慢。
- DUP 键把右对齐字段从光标处到字段结尾用重复字符填充，该字段并不右对齐。
- 不能为包含 WRDWWRAP 键字的字段再指定 CHECK(RB) 或 CHECK(RZ) 键字。

图 3-39 给出了如何指定右对齐空格填充(RB)和右对齐零填充(RZ)的 CHECK 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A* 仅数字
00030A          DATA1      7Y 0I 2 2TEXT('No right-adjust')
00040A          DATA2      7Y 0I 3 2CHECK(RZ)
00050A* 有符号数
00060A          DATA3      7S 0I 4 2TEXT('CHECK(RB) is the default')
00070A          DATA4      7S 0I 5 2CHECK(RZ)
00080A* 字符
00090A          DATA5      7   I 6 2TEXT('No right-adjust')
00100A          DATA6      7   I 7 2CHECK(RB)
A
```

图 3-39 指定右对齐零填充和空格填充的 CHECK 键字

在指定右对齐零填充或空格填充的 CHECK 键字时，按如下显示的填充：

DATA1	1. 1 2 3 _ _ _ _	Field Exit	1 2 3 _ _ _ _	
	2. 1 2 3 - _ _ _	Field Exit	1 2 3 - _ _ _	
	3. 0 _ _ _ _ _	Field Exit	0 _ _ _ _ _	看注.
	4. _ _ _ _ _ _	Field Exit	_ _ _ _ _ _	看注.
DATA2	1. 1 2 3 _ _ _ _	Field Exit	0 0 0 0 1 2 3	
	2. 1 2 3 - _ _ _	Field Exit	0 0 0 1 2 3 -	
	3. 0 _ _ _ _ _	Field Exit	0 0 0 0 0 0 0	
	4. _ _ _ _ _ _	Field Exit	0 0 0 0 0 0 0	

注：OS/400 程序在把数字字段传给用户程序时，将空格转换为零。因此，此字段以全零返回给用户程序。

DATA3	1. 1 2 3 _ _ _ _	Field Exit	_ _ _ _ 1 2 3 _	
	2. 1 2 3 _ _ _ _	Field-	_ _ _ _ 1 2 3 -	
	3. 0 _ _ _ _ _	Field Exit	_ _ _ _ _ 0 _	看注
	4. _ _ _ _ _ _	Field Exit	_ _ _ _ _ _	看注.
DATA4	1. 1 2 3 _ _ _ _	Field Exit	0 0 0 0 1 2 3 _	

2.	1 2 3 _ _ _	Field-	0 0 0 0 1 2 3 -
3.	0 _ _ _ _	Field Exit	0 0 0 0 0 0 _
4.	_ _ _ _ _	Field Exit	0 0 0 0 0 0 _

#### 字符

DATA5	1.	1 2 3 _ _ _	Field Exit	1 2 3 _ _ _
	2.	1 2 3 - _ _	Field Exit	1 2 3 - _ _
	3.	0 _ _ _ _	Field Exit	0 _ _ _ _
	4.	_ _ _ _ _	Field Exit	_ _ _ _ _
	5.	A B C _ _ _	Field Exit	A B C _ _ _
DATA6	1.	1 2 3 _ _ _	Field Exit	_ _ _ 1 2 3
	2.	1 2 3 - _ _	Field Exit	_ _ _ 1 2 3 -
	3.	0 _ _ _ _	Field Exit	_ _ _ _ 0
	4.	_ _ _ _ _	Field Exit	_ _ _ _ _
	5.	A B C _ _ _	Field Exit	_ _ _ A B C

#### 3.3.21.4 光标控制

CHECK 键字配合光标控制码使用时，指定了光标由右向左移动。这种特性是为由右向左读信息的语言设计的。

OS/400 程序并不保证由右向左读的文件仅为光标可由右向左移动的显示站打开。因此，同一系统中所有的工作站都应具有同样的语言能力和同样的由右向左读的能力。

可用于光标控制的有效光标控制码为：

光标控制码	含义
RL	字段中光标由右向左移动
RLTB	字段与字段之间光标由右向左，由上向下移动

由右向左能力有如下限制：

- 模检验的校验位为字段中最右一位
- 片假名不能使用由右向左用法
- CHECK(RL)和 CHECK(RLTB)不能用于用户定义的记录中（有 USRDFN 键字）
- CHECK(RL)仅适用于字符字段
- 不能为有 WRDWRAP 键字的字段指定 CHECK(RB)或 CHECK(RZ)键字

如有以下条件，会显示警告消息：

- 由右向左字段同时也允许磁卡读操作符标识数据 (DSPART(OID)键字)
- 由右向左字段的范围超过一行
- 由右向左字段同时也为自检字段（规定了键字 CHECK(M10)或 CHECK(M11)）
- 由右向左字段同时又指定 CHECK(RZ)或 CHECK(RB)

可选指示器对光标控制码无效。

**RL** (由右至左): 在文件层、记录层或字段层上使用 CHECK(RL)键字来指定光标在可输入字符字段内由右向左移动。在文件层上指定 CHECK(RL)使得光标在此文件的所有可输入字符字段中都由右向左移动。在记录层指定 CHECK(RL)使光标在该记录的所有可输入字符字段中都由右向左移动。在字段层指定 CHECK(RL)使光标在与之相关的字段中由右向左

移动。

图 3-40 给出如何在文件层指定 CHECK(RL) 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                                CHECK (RL)  
00020A      R DSPLY  
          A
```

图 3-40 在文件层指定 CHECK(RL) 键字

图 3-41 给出如何和编辑检验一起指定 RL 光标控制。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A      :  
A      :  
A      R RECORD1           CHECK (RL AB)  
A      :  
A      :  
A      R RECORD2  
A      INPFLD      4   I  4 10CHECK (RL MF)  
A      :  
A      :  
A
```

图 3-41 和编辑检验一起指定 RL 光标控制

注：若要和编辑/检验码一起指定 RL 光标控制码，则必须在这一层上编辑/检验码是合法的。在上例中，CHECK(RL AB)是在记录层指定的，因为 AB 在记录层是合法的。CHECK(RL MF)是在字段层指定的，因为 MF 仅在字段层是合法的。

RLTB（由右向左，由上向下）：仅能在文件层使用 CHECK(RLTB) 键字。它指定了光标从一个可输入字段向另一个可输入字段移动的方向。CHECK(RLTB)指定了在退出一个字段时，光标在显示器上由右向左，由上向下，直到下一个可输入字段。只能和编辑/检验码 AB 一起指定 RLTB 光标控制码，因为其它编辑/检验码在文件层是不合法的。

注：指定 CHECK(RLTB) 并不改变初始显示中光标所在的字段。

图 3-42 示意了如何指定 CHECK(RLTB) 关键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                                CHECK (RLTB)  
00020A      R PROMPT  
          A
```

图 3-42 指定 CHECK(RLTB) 键字

### 3.3.22 CHGINPDFT（修改输入缺省值）

在文件层、记录层或字段层上使用此键字来为可输入字段修改一项或几项输入缺省值。没有参数值时，此键字为可输入字段（仅输入或输出/输入）消除下划线。有参数值时，此键字按指定的显示属性或键盘控制来影响可输入字段。

此键字的格式为：

CHGINPDFT[ (输入缺省 1 输入缺省 2……) ]

合法的参数值有：

参数值	等价的 DDS 键字	含义
无参数	DSPATR(UL) (指定了但未选择)	去掉下划线
BL	DSPATR(BL)	闪烁字段
CS	DSPATR(CS)	列分隔符
HI	DSPATR(HI)	高亮度
RI	DSPATR(RI)	反象
UL	DAPATR(UL)	加下划线
FE	CHECK(FE)	字段退出
LC	CHECK(LC)或 LOWER	小写状态
ME	CHECK(ME)	强制输入
MF	CHECK(MF)	强制填充

注：如果为某字段指定了 DSPATR(UL)，则 CHGINPDFT 键字不能控制该字段是否加下划线。

上述等价的 DDS 键字仅适用于输出字段。对输入和输出/输入字段来说，在指定了等价的 DDS 键字后，还必须指定 DSPATR(UL) 值不选择它。这是因为在没有指定 CHGINPDFT 时，DSPATR(UL) 按缺省值作用于输入及输出/输入字段上。

通常以下面两种方式使用此键字：一是允许记录格式或文件中所有可输入字段进行小写数据输入，二是为某一记录格式或文件中的所有可输入字段指定列分隔符。

在文件层，此键字作用于文件中所有的可输入字段。在记录层，此键字作用于记录格式中所有的可输入字段。在字段层，此键字仅作用于指定了此键字的那些字段。

如果在多个层上指定了 CHGINPDFT 键字，则低层中的键字替换高层中的键字。因此，若在文件层指定了 CHGINPDFT(BL)，而在记录层指定了 CHGINPDFT(HI)，则除了这个记录格式之外，文件中所有的可输入字段都要闪烁。而在这个记录格式中，所有可输入字段高亮度显示。

CHGINPDFT 键字可以和任何 CHECK 或 DSPATR 键字一起指定。若在文件层、记录层或字段层指定了 CHGINPDFT 键字，则可通过在字段层指定 CHECK 或 DSPATR 键字来为某个字段增设检验码或显示属性。例如，若在记录层指定了 CHGINPDFT(CS)，又在字段层指定了 DSPATR(HI)，则该字段显示时有列分隔符且是高亮度显示。而且，在字段层指定的 CHECK 或 DSPATR 键字控制该字段的检验码或显示属性。例如，若在记录层指定了 CHGINPDFT(CS)，在字段层指定了有可选指示器的 DSPATR(CS)，则可选指示器的状态控制该字段的列分隔符。

若想显示一个 UL, RI 和 HI 都有效的字段，则不管是由 CHGINPDFT 键字，还是由 DSPATR 键字，还是二者组合指定的，该字段都不显示。

若在文件或记录层上指定 CHGINPDFT(LC)，则该键字不适用于数字字段。若为数字字段指定 CHGINPDFC(LC)，则会被忽略。

CHGINPDFT(MF) 不允许和 CHECK(RB), CHECK(RZ), AUTO(RAB), AUTO(RAZ) 或 WRDWRAP 键字一起指定。

可选指示器对此键字无效。

图 3-43 给出如何指定 CHGINPDFT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1          CHGINPDFT
00020A      FLD1      10   B  1  2
00030A      FLD2      10       2  2
00040A      R RECORD2          CHGINPDFT (CS)
00050A      FLD3      10   I  3  2
00060A      FLD4      10   B  4  2
00070A      FLD5      10   B  5  2
00080A  01           DSPATR (CS)
00090A      FLD6      10       6  2
00100A      R RECORD3          CHGINPDFT (CS)
00110A      FLD7      10   I  7  2
00120A      FLD8      10   I  8  2
00130A  02           DSPATR (HI)
A
```

图 3-43 指定 CHGINPDFT 键字

对 RECORD1, RECORD2 和 RECORD3, CHGINPDFT 是在记录层上指定:

对 RECORD1, CHGINPDFT 去掉了 FLD1 的下划线。

对 RECORD2, CHGINPDFT 起到以下作用:

- FLD3 和 FLD4 有列分隔符
- 只有选择了 DSPATR(CS), FLD5 才有列分隔符
- FLD6 (仅输出字段) 无列分隔符

对 RECORD3, CHGINPDFT 起到以下作用:

- FLD7 和 FLD8 有列分隔符
- 若选择了 DSPATR(HI), FLD8 还被高亮度显示

### 3.3.23 CHKMSGID (检验信息标识)

用此字段层键字来标识在检测到有效性错误时出现的错误信息。如果没有指定 CHKMSGID 键字，则使用系统提供的信息。相关的有效性规则由 CHECK(M10), CHECK(M11), CHECK(VN), CHECK(VNE), CMP, COMP, RANGE 或 VALUES 键字指定。

此键字的格式为:

CHKMSGID (信息标识 [库/]信息文件 [&信息数据字段])

信息标识参数指定了显示在信息行上的说明信息说明。

信息文件和库参数标识了包含信息说明的信息文件。库名是可选的。如果没有指定库名，则使用在运行时有效的库列表 (\*LIBL) 来找信息文件。

信息数据字段参数指定了包含显示于信息行上的信息替换说明的字段名。该参数的格式为&字段名，这里的字段名就是包含信息替换说明字段的名字。该字段名必须存在于记录格式中，且该字段必须定义为用法为 P 的字符字段（数据类型 A）。

CHKMSGID 仅允许用于指定了 CHECK(M10), CHECK(M11), CHECK(VN),

CHECK(VNE), CMP, COMP, RANGE 或 VALUES 键字的字段。该字段必须是可输入的（用法 B 或 I）。

可选指示器对此键字无效。

图 3-44 给出了如何指定 CHKMSGID 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               MSGLOC(20)
00020A      R RECORD1
00030A      FIELD1      10A  B  4  2CHECK (VN)  CHKMSGID (USR1234 +
00040A                           QGPL/USRMSGS &MSGFLD1)
00050A      MSGFLD1     12A  P
00060A      FIELD2      1A   I  4  20VALUES ('A' 'B' 'I')
00070A                           CHKMSGID (XYZ9999 APPLMSGS)
00080A      FIELD3      3S  0B  4  25RANGE (023 199)
A
```

图 3-44 指定 CHKMSGID 键字

在从显示屏幕上读 RECORD1 时：

- 如果 FIELD1 不含有效名，则会在第 20 行显示信息 USR1234，该信息存在于库 QGPL 的 USRMSGS 信息文件中，替换说明指定在 MSGFLD1 字段中。
- 如果输入到 FIELD2 的数据不是字母 A, B 或 I，则在第 20 行显示存在于 \*LIBL/APPLMSGS 的信息 XYZ9999。
- 如果输入到 FIELD3 的数据小于 023 或大于 199，由于没有指定 CHKMSGID 键字，则在第 20 行显示系统提供的信息 CPF5224（字段值不在有效范围内）。

### 3.3.24 CHOICE (选项)

使用此字段层键字来为一个选择字段定义一个选项。

此键字的格式为：

CHOICE (选项号 选项说明 [\*SPACEB])

选项号参数为此选项定义了一个标识数字。这个参数是必需的。选项号返回给应用程序指明选择字段中的哪个选项被选择了。在非图形显示器上，该选项号也在选项说明的左侧显示出来。选项号的有效值为大于 0 小于 99 的正整数。在同一选择字段中出现相同的选项号是不允许的。

选项说明参数定义出现在选择字段中的选项说明。这个参数是必需的。该参数可以用下面两种形式指定：

- 字符串：‘选项说明’
- 程序到系统字段：&字段名

所指定的字段必须和选择字段在同一记录中，且定义为用法为 P 的字符字段。

选择字段中所有选项的说明必须和文件中指定的最小显示尺寸相适合。因此，选项说明的最大长度由以下而定：

- 选择字段的位置
- 显示于选项左边的最长的选项号的长度
- 选项说明本身的长度

- 选择字段的列数
- 列与列之间的间隔宽度

如果最小显示尺寸  $24 \times 80$ , 则上述各项的和必须小于或等于 80。如果最小显示尺寸为  $27 \times 132$ , 则上述各项和必须小于或等于 132。

在选择项文本中, 可用大于号字符 (>) 为该选项指定一个助记符。>号右侧的字符即为助记符。助记符仅用在该显示器连在支持不可编程工作站增强接口的控制器上的基于字符的图形显示器上, 它是用广播按钮来做选择的。由于系统不是同时支持选择字段的用数字选择也可用助记符选择字段时, 在使用数字选择的显示中忽略助记符。

如何指定助记符, 举例如下:

选项说明	显示结果
'>File'	File
'F>inish'	Finish
'Save>As...'	Save As...
'X> = 1'	X = 1

为把>指定为说明中的一个字符, 则必须把>写两次, 就象必须在说明中的撇号字符要写两次才能得到一个撇号字符一样。例:

选项说明	显示结果
'X>>=1'	X>=1
'X>>>=1'	X>=1

注: 不可能把>指定为助记符。

助记字符只能是单字节字符, 且不能是空格。选项说明中只能有一个助记符, 且多个选项不能指定同一助记符。

\*SPACEB 参数是可选的, 该参数提示应在此选项之前插入一个空格 (或一条线)。此参数用来指定连续排号的选项逻辑组。

对垂直选择字段排在一列中的选择字段来说, 如果选项号是不连续的, 则在非连续选项间会自动插入一个空格。对水平选择字段 (排在多个列中的选择字段) 来说, 则不会有这种情况。

若为某字段指定了 CHOICE 键字, 则必须同时指定 SNGCHCFLD 或 MLTCHCFLD 键字。

可为一个选择字段指定多个 CHOICE 键字。可指定 CHOICE 键字的最大个数, 取决于选择字段的位置和显示尺寸。所有选项必须适合于为文件所指定的最小显示尺寸。

可选指示器对比键字有效。当 CHOICE 键字被置为无效时, 选项列表就会压缩。

图 3-45 给出如何指定 CHOICE 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECORD
A          F1           2Y 0B  1  2SNGCHCFLD
A  01                  CHOICE(1 '>Undo      ')
A                      CHOICE(2 &MARKTXT)
A                      CHOICE(3 '>Copy      ')
A          MARKTXT     12A  P
A
```

图 3-45 指定 CHOICE 键字

在此例中，为单选字段 F1 定义了三个选项。选项 2 的文本包含在字段 MARKTXT 中，运行时选项 2 的助记符必须包含在由应用程序提供的文本中。如果在写记录时指示器 01 为 OFF，仅显示选项 2 和 3。

### 3.3.25 CHRID (字符标识)

如果在显示文件的 CHRID 参数值和工作站的 CHRID 参数值不同时，使用此字段层键字来指定对某一命名字段进行翻译。它对于要显示或键入扩展字符（例如 u 的变音ü，C 的变音，）时很重要。

此键字没有参数。

如果没有为某字段指定 CHRID 键字，而且显示文件的 CHRID 值也不是\*JOBCCSID，则在该字段中显示数据的字符集与所用设备的字符集是一样的。数据如何显示是不可预测的，这取决于原始代码页中的代码点是如何向设备使用的代码页映象的。

不能为常量字段、数字字段（在 36 至 37 列指定了小数位的字段），信息字段（38 列为 M），隐含字段（38 列为 H）或程序列系统字段（38 列为 P）指定 CHRID 键字。

如果显示文件的 CHRID 值为\*JOBCCSID，则 CHRID 键字无效。

不能同时指定 CHRID 键字和 DUP 键字。

如果对某字段同时指定 CHRID 和 DFT 键字，则该字段的初始（缺省）值不翻译，但翻译输入该字段的数据。

虽然能用可选指示器对指定了 CHRID 键字的字段进行条件限定，但可选指示器对此键字无效。

图 3-46 给出如何指定 CHRID 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1  
00020A      TITLE      40      1 20CHRID  
A
```

图 3-46 指定 CHRID 键字

字段 TITLE 是个命名字段。指定了 CHRID 键字后，依据在“应用显示编程”一书所说明的条件，在输出和输入时都会进行字符翻译。

### 3.3.26 CLEAR (清除)

使用此文件层或记录层键字来指定用户程序在工作站用户按上 Clear 键时是否接收控制。一般是 OS/400 程序会将有关的响应指示器置 ON。

关于在系统/36 环境的文件中指定 CLEAR 键字所需考虑的问题，参见附录 F “系统/36 环境考虑”内容。

此关键字的格式为：

**CLEAR[ (响应指示器) ['说明']]**

按 Clear 键如同按命令注意键（没有从设备传递数据）。OS/400 程序并不清除显示，必须由用户程序来实现期望的功能（比如从显示器上清除字段或记录）。

如果没有指定此关键字而用户按下了 Clear 键，OS/400 程序会显示一条信息，指出这个键此时不合法。

注：在使用类打字机键盘的显示站上，要按下 CMD 键，再按 Shift 键和 Field Exit 键上方的左箭头，才能激活 Clear 键。在使用数据输入键盘的工作站上，按下 CMD 键，再按 Shift 键和顶行最右端的空格键，才能激活 Clear 键。

可选的说明在程序编译时建立的打印输出中，用来解释指示器的用法。这个说明在文件或程序中无任何功能，仅仅是个注释。内容要用撇号括起。若在两撇号间的字符多于 50 个，则在程序打印输出时截断为 50 个。

可选指示器对此关键字有效。

图 3-47 给出如何指定 CLEAR 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00101A                                CLEAR(10 'Clear key pressed')  
A
```

图 3-47 指定 CLEAR 键字

### 3.3.27 CLRL (清除行)

用此记录层关键字来指定 OS/400 程序在显示该记录前清除（删除）指定的行。只有指定的行才被清除。

注：同时指定 OVERLAY 时，其它记录仍保留在显示器上。

如何在系统/36 环境使用 CLRL 键字，详见应用显示编程一书。

此关键字的格式为：

**CLRL (nn | \*END | \*NO | \*ALL)**

可用以下几种方式之一指定 CLRL 键字：

- 指定 nn，nn 为 1 到 27 之间的整数。它规定要清除的行数，起始并包括显示该记录区间的第 1 行。如果同时为此记录指定了 SLNO（起始行数）关键字，则删除起始于显示该记录格式的有效起始行数。

注：如果指定了 nn，则该记录必须至少定义了一个字段。

- 指定\*NO 表示在显示所定义的记录格式之前不清除任何行。显示该记录时会将显示器上原有数据覆盖。
- 指定\*ALL 表示在显示所定义的记录格式之前清除显示器上所有的行。此记录格式中必须至少定义一个字段。

如果某记录格式起始于第一列的字段，则该记录格式的起始属性字符为前一行的最后一列。前一行号是此记录格式的起始行号。这条规则也适用于在 DDS 中定义的第一行第一列

字段的 SLNO 格式。

如果指定 CLRL 键字的记录格式中有一个或多个可输入字段，任何被覆盖的记录都不再被 OS/400 程序识别。就是说，所有可输入字段都不能再键入数据，所有向这些记录中写入的输入操作都将导致错误，它们也能通过选择 ERASE 键字来清除。

如果为一个没有可输入字段的记录格式指定了 CLRL (nn) 键字，则被覆盖记录中的可输入段还是可输入的。就是说，被覆盖记录中的可输入字段仍可输入，向这些记录写的输入操作还是有效的。如果为有 CLRL 键字定义的记录指定 ROLLUP 或 ROLLDOWN 键字，则这两个键字被忽略。有 CLRL 键字而没有可输入字段的记录在被其它有 OVERLAY 键字的记录覆盖时不能被完全清除。覆盖记录所需的行被清除，而覆盖记录不需要的行仍在显示屏上。

在覆盖记录被写入显示器时，可使用 CLRL(\*NO) 键字来防止清除被覆盖记录。如果使用了这个键字，所有已在显示器上且将被覆盖的记录都不会被清除。新记录完全或部分地覆盖了这些记录。如果要显示常量和需重复地送往显示器的数据，使用 CLRL(\*NO) 键字就会有改善性能。把送常量作为另一个记录格式，而对包含数据的记录格式使用 CLRL(\*NO) 键字，这样会减少向显示器发送记录所需的时间。

如果没有指定 CLRL 键字，也没有指定 OVERLAY 或 PUTOVR 键字，则会清除整个屏幕。

如果使用 CLRL 键字，同时 PUTOVR 或 PUTRETAIN 有效，则清除任何行都会和 PUTOVR 或 PUTRETAIN 相冲突。PUTOVR 或 PUTRETAIN 键字要求被复盖的字段还要留在显示器上，而 CLRL(nn) 或 CLRL(\*END) 键字则要求首先清除这些字段。如果某一字段因使用 CLRL(nn) 或 CLRL(\*END) 键字而不能用于输入，则在 PUTOVR 有效时可输入字段仍为可输入的。然而，如果程序试图读这样的记录，OS/400 程序会发出信息。

如果指定了 CLRL 键字，还必须在建立显示文件 (CRTDSPF) 或修改显示文件 (CHGDSFP) 命令中指定 RSTDSP(\*YES)。否则，如文件暂时挂起，显示器上的数据可能会丢失。

### 3.3.27.1 防止清除被覆盖的记录

CLRL 键字不能和以下键字同时定义：

ASSUME SFLCTL KEEP USRDFN SFL

如果为有 DSMPMOD 键字的记录指定 CLRL 键字，在文件建立时会显示一条警告信息。运行期间，当显示方式改变时会忽略 CLRL 键字。

对由 PASSRCD 键字指定的记录格式，不能再指定 CLRL 键字。

可选指示器对此键字无效。

图 3-48 给出如何指定 CLRL 键字。

00010A	R RECORD1	CLRL(5)
00020A	FLD1	5 3 2
00030A	FLD2	10 0B 5 2

00040A	FLD3	10 I 6 2
00050A*		
00060A	R RECORD2	CLRL (*NO)
00070A	FLD1	5 2 2 2
00080A	FLD2	5 H
00090A	FLD3	10 I 4 2
00100A*		
00110A	R RECORD3	CLRL (*END)
00120A	FLD1	5 B 5 2
00130A	FLD2	5 I 8 2
	A	

图 3-48 指定 CLRL 键字

在显示 RECORD1 之前清除了第 3、4、5、6 和 7 行。在 RECORD2 中，没有清除任何行，在显示此记录时，将覆盖任何已显示的信息。显示 RECORD3 之前清除了第 5 到 24 行。

### 3.3.28 CMP (比较)

此键字与 COMP 键字是等价的。

此键字的格式为：

CMP (关系运算符 值)

建议使用键字 COMP。关于如何使用这两个键字，见 3.3.31 “COMP (比较)” 的内容。

### 3.3.29 CNTFLD (连续输入字段)

使用此字段层键字定义一个连续输入字段。连续输入字段是一组相关字段的集合，在字段数据输入和编辑时，工作站控制器把它们当作一个字段来对待。如果显示设备不是连到支持不可编程工作站增强接口的控制器上，在编辑连续输入字段时，它的每一段都分别处理。

图 3-49 为使用连续输入字段建立的一个矩形文本输入字段。

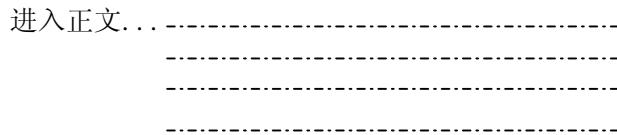


图 3-49 矩形连续输入字段

对终端用户来说，文本输入格式比占用多个显示行的单个输入字段更有吸引力。尽管最后一行没有占满全部列的宽度，在矩形中还是不允许再有其它字段。连续输入字段允许在一个窗口中定义多行输入字段。

此键字的格式为：

CNTFLD (列的宽度)

必须指定一个参数。

列的宽度参数指定了连续字段所占据的列数。此值必须在显示器或窗口列的宽度之内，还必须小于字段的长度。

有 CNTFLD 键字的字段必须是数据类型为 A 的可输入字段。此字段不能定义在子文件中。

不能与 CNTFLD 键字一起指定以下键字：

AUTO(RAB, RAZ)      CHECK(AB, AF, RB, RZ, RLTB)

CHOICE                  DSPATR(OID SP)            EDTMSK

用 CNTFLD 键字定义的字段必须和其它字段至少有 2 个空格的间隔。

可选指示器对此键字无效。

CNTFLD 键字由于用组成这个特别字段的段总数而减少了可用的输入字段的个数。例如：60 个字符的输入字段如果有 CNTFLD(10)就显示为 6 行，每行 10 个字符。控制器把每行（或段）看作一个输入字段，这样，可输入字段数减少为 6。

图 3-50 给出如何指定 CNTFLD 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A         R RECORD  
00020A         F1                90A B 3 4CNTFLD(30)
```

图 3-50 指定 CNTFLD 键字

在此例中定义了多行输入字段。此输入字段连续 3 行，列的宽度为 30。

### 3.3.30 COLOR (颜色)

此字段层键字指定了某字段在彩色显示器（3179, 3197 型 C1C2, 377 型 FC, 3486, 3487 型 HC, 3488(4) 或 5292 彩色显示器）上显示时的颜色。如果在单色显示器上为某字段选择了该键字，则该键字被忽略。只能为 COLOR 键字指定一个参数值，但可为每个字段指定多个 COLOR 键字。

此关键字的格式为：

COLOR (GRN | WHT | RED | TRQ | YLW | PNK | BLU)

有效的参数值有：

值	含义
GRN	绿色
WHT	白色
RED	红色
TRQ	蓝绿色
YLW	黄色
PNK	粉红色
BLU	蓝色

因为绿色是彩显的缺省颜色，所以若想使某字段保持绿色，则需指定 COLOR (GRN)。若指定了 DSPATR (HI), DSPATR (CS) 或 DSPATR (BL)，但没指定 COLOR (GRN) 那就会改变字段的颜色。

可选指示器对此键字有效。

若为一个字段指定了多个 COLOR 键字，则必须为每个 COLOR 键字指定可选指示器。如果对某一个输出操作有多个 COLOR 键字有效，OS/400 程序使用在 DDS 中指定的第一个 COLOR 键字（见 3.3.30.2 图 3-52）。不能为同一字段多次指定同一种颜色。

在一个显示文件中可指定的 COLOR 键字的数目受限一个叫作屏幕属性数组的系统内部存贮区的最大尺寸。整个显示文件的屏幕属性数组最大尺寸为 32763 字节。在文件中每指定一个 COLOR 键字就占用这个数组中一定的存贮空间。

如果在一个文件中使用了多个 COLOR 键字，特别是还有条件限制，则应考虑这些键字共需多少内部存贮空间。如果超出了 32763 字节，则在建立文件时会发生 CPF0673 信息（文件中指定了太多的 COLOR 或 DSPATR 键字）。为确定某个 COLOR 键字需多少存贮空间，可使用如下算法：

$$(\text{键字条件数}) \times 2 + 2 + 2^9 = \text{屏幕属性数组中此键字所需字节数}$$

例如，假设一个文件有 8 个字段，每个字段有 9 个 COLOR 键字，每个 COLOR 键字使用 3 个可选条件。用上述算法，每个 COLOR 键字需要屏幕属性数组中的 520 个字节：

$$3 \times 2 + 2 + 2^9 = 520 \text{ 字节}$$

因为每个字段有 9 个 COLOR 键字，文件中有 8 个字段，所以需要总的存贮空间为 37440 字节 ( $520 \times 9$  键字  $\times$  8 字段)。因为 37440 大于 32763，所以在建立文件时就会发出 CPE0673 信息。

COLOR 键字与 DSPATR 键字共同使用：

在 COLOR 和 DSPATR 的某些组合使用方式中，两个键字同时有效。这些组合方式为：

COLOR	DSPATR
任意	RI (反象)
任意	UL (下划线)
RED	BL (闪烁字段)
RED	BL 和 RI
RED	BL 和 UL
RED	RI 和 UL
GRN	RI 和 UL
TRQ	RI 和 UL
PNK	RI 和 UL

例如，如果 COLOR (YLW) 和 DSPATR (RI) 同时有效，则字段显示为黄背景黑字符。

在某些 COLOR 与 DSPATR 组合使用方式中，一些参数值被忽略。这些组合方式列于下表中。

COLOR	DSPATR	效果
-------	--------	----

任意	ND (不显式)	所有颜色都被忽略
任意	HI (高亮)	忽略 HI
任意	CS (列分隔符)	忽略 CS (注 1)
GRN	BL	忽略 BL (注 2)
WHT	BL	忽略 BL (注 2)
TRQ	BL	忽略 BL (注 2)
YLW	BL	忽略 BL (注 2)
PNK	BL	忽略 BL (注 2)
BLU	BL	忽略 BL (注 2)
RED	RI、BL 和 UL	忽略 UL (注 3)
YLW	RI 和 UL	忽略 RI
BLU	RI 和 UL	忽略 RI
WHT	RI 和 UL	忽略 RI

注：

1. 即使没有指定 DSPATR (CS)，蓝绿色和黄色也有列分隔符。(在彩色显示器上，列分隔符显示为字符间的小蓝点。如果显示站用户将彩色显示站置为缩减行间距方式，列分隔符就会消失)。
2. 可闪烁的颜色只有红色。
3. 可输入字段的下划线也消除了，而在 AS/400 系统中按缺省是有下划线的。例如，如果输出操作同时选择了 COLOR (YLW) 和 DSPATR (HI)，则字段显示为黄色但没有高亮。

### 3.3.30.1 彩色显示器的 DSPATR 键字

如果指定了 DSPATR 键字但没指定 COLOR 键字，则字段在彩色显示器上按下表所示的颜色显示，但不按所指定的显示属性显示。

图 3-51 彩色显示器的 DSPATR 键字

DSPATR(CS)	DSPATR(HZ)	DSPATR(BL)	在彩色显示器上的颜色
			绿色 (正常)
×			蓝绿色 (1)
	×		白色
		×	红色，不闪烁
	×	×	红色，闪烁
×	×		黄色 (1)
×		×	粉红色
×	×	×	蓝色

注:

1. 蓝绿色和黄色字段显示时有列分隔符（总为蓝色的），除非工作站用户把彩色显示站置为缩减行间距方式。

例如，如果为某字段指定了 DSPATR(HI)，而没有指定 COLOR 键字，则该字段在彩色显示器上显示为白色但并未高亮显示。

为单色显示器选择的 COLOR 键将被忽略。

图 3-52 给出为一个字段指定 COLOR 和 DSPATR 键字的情况。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD           1
00020A                  1 2' Column Heading'
00030A                  DSPATR(HI)
00040A      2  FIELD1       5   3 2
00050A      3  FIELD2       5   I 5 2COLOR(YLW)
00060A      4  FIELD3       5   7 2DSPATR(BL)
00070A      5  FIELD4       5   I 9 2
00080A  42                   COLOR(YLW)
00090A  43                   COLOR(TRQ)
00100A  44                   COLOR(BLU)
A
```

图 3-52 指定 COLOR 和 DSPATR 键字

1. 在彩色显示器上，常量字段的列标题是白色的，在单色显示器上，则为高亮显示。
2. 在所有显示器上，FIELD2 都是绿色的。
3. 在彩色显示器上，FIELD2 是黄色的并带有蓝色的列分隔符。在所有显示器上，该字段都带有下划线，因为这是一个可输入字段。
4. 在彩色显示器上，FIELD3 是红色的并且不闪烁，在单色显示器上，FIELD3 是闪烁的。
5. 在彩色显示器上，FIELD4 可显示为下述颜色之一：
  - 绿色，如果没有指示器为 ON。
  - 黄色，如果指示器 42 为 ON（不管其它指示器如何设置）
  - 蓝绿色，如果指示器 43 为 ON 而指示器 42 为 OFF。
  - 蓝色，如果只有指示器 44 为 ON。

在单色显示器上，FIELD4 是绿色的。在所有显示器上，FIELD4 都加下划线。

图 3-52 给出在彩色和单色显示器上将一个字段指定为可输入字段的一种方法。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD
00020A 2   1  FIELD A       5   B 2 2COLOR(TRQ)
00030A  44                      ERMSG('Record not found' 44)
```

图 3-53 在彩色和单色显示器上指定一个字段

1. 在彩色显示器上, FIELD A 是蓝绿色的, 并带有蓝色的列分隔符; 在单色显示器上, FIELD A 为绿色的。
2. 如果显示 FIELD A 时可选指示器 44 为 ON, 则 ERRMSG 键字有效并发生以下情况:
  - 在彩色显示器上, FIELD A 是蓝绿色的并且是反向显示的。(因为有 COLOR 键字, 所以 FIELD A 并没有高亮显示)。错误信息 “Record not found” 用白色显在信息行上。
  - 在单色显示器上, FIELD A 为高亮反象显示。错误信息 “Record not found” 在信息行上高亮显示。

### 3.3.31 COMP (比较)

使用此字段层键字来指定 OS/400 程序把工作站用户键入仅输入或输出/输入字段的数据同指定的数值进行比较。用关系运算符构成比较判别式。如果键入的数据不符合规定的合法性检验, OS/400 程序就显示一条出错信息。注意 OS/400 程序只有在工作站用户修改了字段内容或者用 DSPATR (MDT) 把该字段的 MDT 置为 ON 时才进行检验。

注: 关于定义用户指定的出错信息, 详见 CHKMSGID 键字说明。

这个键字的格式为:

COMP (关系运算符 值)

每个 COMP 键字只能指定一种操作, 每个字段只能指定一个 COMP 键字。

所用的关系运算符有:

关系运算符	含义
EQ	等于
NE	不等于
LT	小于
NL	不小于
GT	大于
NG	不大于
LE	小于或等于
GE	大于或等于

根据数据类型不同 (小数位), 所用的比较值是数值或者是字符。数值由数字 0 到 9 和符号 (+或-) 表示。字符值必须在一对撇号中。

注: 如果所定义的字段是数值的, 则根据 36 和 37 列所指定的小数位进行对准, 前导及结尾的空格都用 0 填充。如果没有输入小数点, 则假定小数点在最后一位右侧。例如, 一个长度为 5 (由 34 列指定), 小数位为 2 (由 37 列指定) 的数值字段, 1.2 被解释为 001.20, 100 被解释为 100.00。

不能为浮点字段 (35 列为 F) 指定 COMP 键字。

可选指示器对此键字无效。

图 3-54 给出如何指定 COMP 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      FIELD2      6 0I 10 10COMP(EQ +021920)  
00020A      FIELD1      3   I 11 11COMP(EQ 'ABC')  
A
```

图 3-54 指定 COMP 键字

### 3.3.32 CSRINPONLY (光标移至仅输入位置)

在文件或记录层上使用此键字来限制光标只能向可输入字段位置移动。此键字只对使用箭头键移动的光标有效。

此键字没有参数。

在定义帮助用此键字时要小心。用户可能无法将光标定位在帮助有效区内。

关于 CSRINPONLY 键字，详见《应用显示程序设计》一书。

可选指示符对此键字是合法的。

图 3-55 给出如何指定 CSRINPONLY 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          CSRINPONLY  
A          R RECORD1  
A          1 10' ONE--:'  
A          FIELD1    10A  I  1 20TEXT(' ONE')  
A          2 10' TWO--:'  
A          FIELD2    10A  I  2 20TEXT(' TWO')  
A          3 10' THREE--:'  
A          FIELD3    10A  I  3 20TEXT(' THREE')  
A          4 10' FOUR--:'  
A          FIELD4    10A  I  4 20TEXT(' FOUR')  
A          5 10' OUT--:'  
A          FIELD5    10A  O  5 20TEXT(' OUT')  
A
```

图 3-55 指定 CSRINPONLY 键字

图 3-55 中 RECORD1 定义为输入、输出和常量字段。因为指定了 CSRINPONLY 键字，用户只能把光标定位在 FIELD1、FIELD2、FIELD3 或 FIELD4 上。光标无法进入 FIELD5 或其它显示区。

### 3.3.33 CSRLOC (光标定位)

使用此记录层键字来指定对所定义的记录格式进行输出操作时给光标定位。用户程序在光标定位后做输出操作。

此键字的格式为：

CSRLOC (字段名 1 字段名 2)

键字中的参数值指定了两个字段名，其内容分别为光标位置的行号（字段名 1）和列号（字段名 2）。字段名 1 和字段名 2 都是三字节的，区位十进制的隐藏字段，用户程序用这些字段来告诉 OS/400 程序在何处定位光标。

如果输出操作锁住了键盘，则光标不会定位在预期的位置上。只有用户程序发出一个输入或输出操作来解锁键盘，光标才会移到预期的位置上。如果用户程序把光标定位字段的值设在显示设备合法值范围之外，则该键字会被忽略。

对任意一个输出操作，CSRLOC 键字会覆盖其它任何光标定位规定，比如 DSPATR(PC) 和 SFLRCDBR(CURSOR)。除非用户程序在 DSPATR(PC)、CSRLOC 或 SFLRCDBR(CURSOR) 有效时发出另一个输出操作，或者有此键字的记录被覆盖 (OVERLAY 键字) 或删除 (ERASE 键字)，否则此键字将保持有效。

在输入操作中，光标位置是由 I/O 反馈区内容或 RTNCSRLOC 键字的相应参数确定的。详细信息请看应用显示程序设计一书的内容。

每个记录格式只能指定一个 CSRLOC 键字。

CSRLOC 键字对以下记录格式是不合法的：

- 子文件记录格式（由 SFL 键字标识）
- 用户定义记录格式（由 USRDFN 键字标识）

可选指示器对此键字有效。显示尺寸条件名是不合法的。

图 3-56 给出如何指定 CSRLOC 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1          CSRLOC(LINNBR POSNBR)
00020A      TITLE      40   B  1  2
00030A      PAGE       5Y 0B  1 60
00040A      TEXT       1760 B  2  1
00050A      LINNBR     3  OH
00060A      POSNBR     3  OH
A
```

图 3-56 指定 CSRLOC 键字

应用程序在向 RECORD1 发出输出操作之前设置了 LINNBR 和 POSNBR 的内容。显示记录时，字段标题、页和说明都显示出来。光标可定位在 RECORD1 的一个字段上，工作站用户可在此处进行修改。

### 3.3.34 DATE (日期)

使用此字段层键字来把当前作业时间用 6 字节长的常量(仅输出)字段显示。可以指定字段的位置, DATE 键字还可有选择地同时指定 EDTCDE、EDTWRD、COLOR、DSPATR 或 TEXT 键字。17 到 38 列必须是空格。

此键字没有参数。

如果为 DATE 字段指定了 EDTCDE(Y), 则显示长度为 8 个位置。指定 EDTCDE(Y) 把日期由 mmddyy 形式改为 mm/dd/yy 形式, 其中斜线(/)表示在运行时作业属性的 DATSEP, 而月、日、年的顺序由作业属性 DATFMT 确定。(DATFMT 可为 MDY、DMY、YMD 或 JUL, M 为月, D 为日, Y 为年, JUL 为儒略日期。DATSEP 可为斜线(/), 破折号(—), 句点(.)或逗点(, )。)

可选指示器对此键字无效, 虽然可选指示器可用来条件限定用此键字的字段。

图 3-57 给出如何指定 DATE 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A 20           1 56DATE  
00020A 21           1 56DATE EDTCDE(Y)  
A
```

图 3-57 指定 DATE 键字

如果可选指示器 20 为 ON, 日期按无编辑显示; 如果可选指示器 20 为 OFF 而可选指示器 21 为 ON, 则日期按有编辑显示。

### 3.3.35 DFT (缺省)

使用此字段层键字来为常量字段(未命名字段)指定常量值, 或为命名字段指定缺省值。

此键字的格式为:

DFT ('值') | '值'

可规定文字的最多字符数由显示此字段的显示器的尺寸决定, 具体如下:

显示器尺寸	最大字符数
24×80	1919
27×132	3563

常量字段: 常量字段值可用一对撇号括起来指定。(关于其它指定常量字段的方法, 见 DATE、MSGCON 和 TIME 键字)。可省略 DFT 键字本身和括号来简化 DDS。不管是显式还是隐式地指定 DFT 键字, OS/400 程序都把指定的值作为常量字段来显示。关于常量字段的说明见第 3.2.8 “名字(19 至 28 列)” 的内容。

命名字段: 对仅输入字段, 每次显示字段时都显示指定值。工作站用户随即可能修改显示的值, 并返回给用户程序。

对仅输出和输出/输入字段指定了 DFT 键字, 还必须在记录层指定 PUTOVR, 在字段层指定 OVRDTA。只在每一次输出操作才显示指定值。在其后的输出操作中则显示程序值。详见第 3.3.109 “PUTOVR”。

键字 DFTVAL、EDTCDE 和 EDTWRD 不能与 DFT 一起指定。  
对浮点字段使用 DFT 键字是不合法的。  
可选指示器对此键字无效，尽管可用可选指示器来条件限定指定了该键字的字段（常量字段或命名字段）。

图 3-58 给出如何指定 DFT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A           HOTTYP      1   I  7  9DFT('D')
00040A                   VALUES('D' 'S')
00050A           8  9' ON'
00060A       01           12  1' HOTEL NAME: '' TERRACE INN ''
00070A                   TEXT('Constant field is +
00080A                   conditioned, not the implicit +
00090A                   DFT keyword')
00100A       02           12  1' HOTEL NAME: '' RIVER VIEW INN ''
00110A                   TEXT('Either '' TERRACE INN' or +
00120A                   '' RIVER VIEW INN' could +
00130A                   appear in line 12, position 1')
```

图 3-58 指定 DFT 键字

常量字段 ON 没有可选指示器，所以一直显示。

如果指示器 01 为 ON，则显示如下内容： HOTEL NAME: ‘TERRACE INN’

如果指示器 02 为 ON 而指示器 01 为 OFF，则显示如下内容：

HOTEL NAME: ‘RIVER VIEW INN’

如果要为多个显示尺寸指定一个常量字段，并且要为不同显示尺寸修改字段位置但不修改字段内容，那么就不要重复指定这个常量值。图 3-59 给出该怎样处理这种情况。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A           DSPSIZ(*DS3 *DS4)
A               :
A               :
00080A           22  2' Constant data'
00090A           26  2
A
```

图 3-59 指定有多个显示尺寸的常量字段

在 24×80 显示器上，常量字段 ‘Constant data’ 显示在第 22 行第 2 列，而在 27×132 显示器上，显示在第 26 行第 2 列。

### 3.3.36 DFTVAL (缺省值)

使用此字段层键字来为一个输出字段指定缺省值。在第一次输出操作时，如果可选指示

器为 ON 或没有指定，则会显示此值，否则会显示程序值。在随后的输出操作中，总显示程序值。

此关键字的格式为：

DFTVAL ('值')

此关键字只对仅输出字段 (O) 或输出/输入字段有效。

此关键字仅用于初始化命名字段，对常量字段则不允许使用。

因为一条 DDS 语句所含字符不能超过 5000 个，这个关键字和其它关键字一起指定时，其所含字符必须少于 5000 个。

不能在子文件格式（关键字 SFL）中使用此关键字。

不能为指定了 DFT、EDTCDE 或 EDTWRD 的字段再指定 DFTVAL 关键字，也不能为浮点字段指定此关键字。

可选指示器对此关键字有效。

图 3-60 给出如何指定 DFTVAL 关键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECORD1
A 50           PUTOVR
A          FIELD1      3A  B 12 01DFTVAL('AAA') OVRDTA
A          FIELD2      3D 00 12 050VRS
A 10           DFTVAL('000')
A          FIELD3      3D 00 12 09DFTVAL('000') OVRDTA
A
```

图 3-60 指定 DFTVAL 关键字

在图 3-60 中，在显示记录之前，应用程序把 FIELD1 赋值为“ZZZ”，把 FIELD2 赋值为“999”，把 FIELD3 赋值为“456”。在第一次输出操作时，如果指示器 10 为 ON 则显示“AAA 000 000”；如果指示器 10 为 OFF 则显示“AAA 999 000”。

工作站用户向 FIELD1 中键入“XXX”。在第二次输出操作中，如果指示器 50 为 ON 则显示“XXX 999 456”；如果指示器 50 为 OFF 且第一次输出操作时指示器 10 为 ON，则显示“AAA 000 000”。如果指示器 50 为 OFF 且第一次输出操作时指示器 10 为 OFF，则显示为“AAA 999 000”。

### 3.3.37 DLTCHK (删除检验)

用此字段层关键字可使 OS/400 程序忽略为某一引用字段指定的所有合法性检验及 CHKMSGID 关键字。此关键字只有在 29 列为 R 时才是合法的。

此关键字没有参数。

如果指定了任何新的合法性检验关键字，则 DLTCHK 就是不必要的了。新的合法性检验关键字取代引用字段的合法性检验关键字。

可选指示器对此关键字无效。

图 3-61 给出如何指定 DLTCHK 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00030A                               REF(FILE)  
00040A       R RECORD  
00050A       CODE      R      I  3 20DLTCHK  
A
```

图 3-61 指定 DLTCHK 键字

### 3.3.38 DLTEDT (删除编辑)

使用此字段层键字可使 OS/400 程序忽略为引用字段指定的 EDTCDE 或 EDTWRD 键字。此键字只有在 29 列为 R 时才合法。

此键字没有参数。

如果指定了新的编辑键字，则 DLTEDT 就是不必要的了。新的编辑键字取代引用字段的编辑键字。

可选指示器对此键字无效。

图 3-62 给出如何指定 DLTEDT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00040A                               REF(FILEA)  
00050A       R RECORD  
00060A       AMT      R      B  5 20DLTEDT  
A
```

图 3-62 指定 DLTEDT 键字

### 3.3.39 DSPATR (显示属性)

使用此字段层键字来为所定义的字段指定一个或多个显示属性。可为同一字段指定多个 DSPATR 键字，也可为同一键字指定多个属性。但对每一个字段来说，每种属性（例如 UL）只能指定一次。

此键字格式为下述二者之一：

DSPATR (属性 1[属性 2[属性 3[...]]])

DSPATR (&程序到系统字段)

如果为同一字段指定多个属性，不管是用一个键字指定的，还是用多个键字分别指定的，所指定的每一个属性（在显示该字段时有效）都会影响该字段。例如，如果想使一个字段按反象且高亮度显示，可指定 DSPATR(RI HI)；或 DSPATR(RI) 及 DSPATR(HI)。

还可以使用程序到系统字段参数，该命名字段必须在记录格式中定义，且数据类型为字母数字（35 列为 A），长度为 1，用法为 P（38 列为 P）。程序用这个 P-字段来为指定 DSPATR 键字的字段设置显示属性。

这个 P-字段名用于定义记录中的多个字段，每个字段只允许一个 DSPATR P-字段。P-字段中包含了显示属性并标识此字段受保护。详见第 3.3.39.3 “有效的 P-字段值”。

以下为 DSPATR 键字第一种格式的有效属性：

显示属性	含义
BL	闪烁字段
CS	列分隔符
HI	高亮度
ND	不显式
PC	定位光标
RI	反映象
UL	下划线

对仅输入字段：

显示属性	含义
MDT	显示时设置修改数据标记
OID	操作符标识
PR	保护字段不能输入内容
SP	选择光笔

注：

1. 如果在 5250 显示站上为同一字段指定了 UL、HI 和 RI 显示属性，结果与指定 ND 是一样的。
2. 如果指定了 OID，则不应再指定 SP，无论 OID 还是 SP 都不能置为可选，除非和其它显示属性一起指定。
3. 显示属性 BL、CS、HI、RI 和 UL 还可在文件层、记录层或字段层上指定为 CHGINPDFT 键字的参数值。
4. 显示属性 CS、HI 和 BL 可使字段在 5292、3477 型 FC、3487 型 HC、3179、3197 型 C1 和 C2 和 3488(5)彩色显示站上显示为彩色字段。详见第 3.3.30 “COLOR (颜色)” 内容。
5. 如果使用一台 IBM PS/2\*来仿真一台 5250 显示站，并且直接修改 EBCDIC 屏幕缓冲区，则需要设置 MDT 属性。详见《IBM 个人计算机增强 5250 仿真程序技术参考》手册。
6. 如果使用的是 PS/2 计算机及 VGA 监视器，则由于使用缓冲方式的硬件特殊限制，UL 属性并不起作用。

可选指示器对此键字是合法的，除非只单独指定了 OID 或 SP 属性。

图 3-63 给出了编码举例和显示样本，随后将对每一种属性进行详细说明。

PICTURE 14

图 3-63 按不同显示属性显示的五字节字段

### 3.3.39.1 对所有字段都适用的显示属性

**BL (闪烁):** 用此属性来指定字段在显示时闪烁。

**CS (列分隔符):** 用此属性来指定在显示一个字段时其左右两边都有竖直条。如果为非显式字段指定了此属性，即使分隔符之间没有字符，也显示分隔符，可用分隔符来精确地指定字段中光标的位置，指定另外空白字段的长度。

**HI (高亮度):** 用此属性来指定字段在显示时按高亮度显示。

**ND (不显式):** 用此属性来指定某一字段不显示；显示该字段的位置为空白。此属性可用于口令或其它安全敏感数据。如果执行打印功能（通过指定 PRINT 键字），则不会打印不显式字段。

**PC (定位光标):** 用此属性来把光标定位在所定义字段的第一个字符的位置上。可为几个字段指定此属性，光标会定位在此属性的第一个被选择的字段上。注意记录中各字段是按行/列顺序显示在显示器上，而不是按指定时的顺序显示。

**RI (反映象):** 用此属性规定字段在显示时，它的映象与显示屏上其它部分是反象的。

屏幕为黑底白字还是白底黑字取决于显示字段前显示器的状态。这项设置由工作站用户来控制。

**UL (下划线):** 用此属性来指定在显示字段时加上下划线。所有可输入字段按缺省值是加下划线的。用 CHGINPDFT 键字可改变缺省值而不加下划线。（如果指定了 CHGINPDFT，则要为可输入字段加下划线时必须指定 DSPATR(UL)）。如果指定 DSPATR(UL)也指定了可选指示器，并且可选指示器为不符合，（没有选择 DSPATR(UL)），则显示字段时不加下划线。

### 3.3.39.2 仅输入字段的显示属性

**MDT** (置修改数据标记): 用此属性来指定 OS/400 程序在字段写入显示器时将修改数据标记置为 ON。该属性保证了在从显示器读记录时该字段从设备送出。

注: OS/400 为输出/输入字段保存输出数据, 还为指定了 DFT 键字的字段保存初始化数据。如果字段中没有输入新的(修改了的)数据, 则所保存的数据将由一次输入操作返回。

**OID** (操作符标识): 用此属性来指定 OS/400 程序允许磁条阅读器 OID 数据进入到此字段中。如果这是个非显式字段, 还必须指定 DSPATR(ND)属性。

指定 DSPATR(OID)键字的字段和其它可输入字段作用一样, 数据可从键盘或磁条阅读器输入。可指定 DSPATR(OID)键字(但并非必须)来指明可用一个磁条阅读器来输入数据。也可从键盘上输入数据, 除非指定了键盘转换禁止键盘输入(I)。如果为一个字段同时指定了 DSPATR(OID)和 DSPATR(SP)两个键字, 则会忽略 DSPATR(SP)。

**PR** (保护): 用此属性来指定工作站用户不能向所定义的可输入字段中键入数据。此属性仅用于可输入字段。仅输出字段和常量字段其定义就是受保护的。

**SP** (由光笔选择): 用此属性来指定该可输入字段可由光笔选择。工作站用户可向光笔字段中键入数据, 除非为该字段的位置 35(数据类型/键盘转义)指定为 I(禁止键盘输入)。

在字段第一次显示时, 字段的内容由用户程序(输出/输入字段)设定, 或由 DDS 设定(指定了 DFT 键字的仅输入字段或字符串)。如果工作站用户没有键入新的数据, 则此输出数据通过一次输入操作返回用户程序。

可用光笔选择的字段至少有三个字节长。建议此字段内容如下:

- 一个切换字符, 可为十六进制 6F(?); 若工作站用户用光笔选择该字段, 也可为十六进制 6E(>)。
- 空格(十六进制 40)
- 目的字符, 可为任意字符, 比如星号(\*)。
- 另一个空格
- 可使工作站用户识别该字段的附加数据(一个或多个字符)。

此属性仅对配有光笔进行选择的工作站才有用。

### 3.3.39.3 合法的 P-字段值

DSPATR P-字段不支持以下显示属性:

显示属性	含义
MDT	显示时设置修改数据标记
OID	操作符标识
PC	定位光标
SP	由光笔选择

图 3-64 有效的 P 字段值(非保护)

十六进制	限制彩色	全彩色
20	正常	绿
21	反像	绿, 反像

22	高亮	白
23	高亮, 反像	白, 反像
24	下划线	绿, 下划线
25	下划线, 反像	绿, 下划线, 反像
26	下划线, 高亮	白, 下划线
27	不显示	不显示
28	闪烁	红
29	闪烁, 反像	红, 反像
2A	闪烁, 高亮	红, 高亮
2B	闪烁, 高亮, 反像	红, 高亮, 反像
2C	闪烁, 下划线	红, 下划线
2D	闪烁, 下划线, 反像	红, 下划线, 反像
2E	闪烁, 下划线, 高亮	红, 下划线, 闪烁
2F	不显示	不显示
30	列分隔符	青绿色, 列分隔符
31	反像, 列分隔符	青绿色, 列分隔符, 反像
32	高亮, 列分隔符	黄, 列分隔符
33	高亮, 反像, 列分隔符	白, 反像, 列分隔符
34	下划线, 列分隔符	青绿色, 下划线, 列分隔符
35	下划线, 反像, 列分隔符	青绿色, 下划线, 反像, 列分隔符
36	下划线, 高亮, 列分隔符	黄, 下划线, 列分隔符
37	不显示	不显示
38	闪烁, 列分隔符	粉红
39	闪烁, 反像, 列分隔符	粉红, 反像
3A	闪烁, 高亮, 列分隔符	蓝
3B	闪烁, 高亮, 反像, 列分隔符	蓝, 反像
3C	闪烁, 下划线, 列分隔符	粉红, 下划线
3D	闪烁, 下划线, 反像, 列分隔符	粉红, 下划线, 反像
3E	闪烁, 下划线, 高亮, 列分隔符	蓝, 下划线
3F	不显示	不显示

图 3-65 有效的 P- 字段值 (保护)

十六进制	限制彩色	全彩色
A0	正常	绿
A1	反像	绿, 反像
A2	高亮	白
A3	高亮, 反像	白, 反像
A4	下划线	绿, 下划线
A5	下划线, 反像	绿, 下划线, 反像
A6	下划线, 高亮	白, 反像
A7	不显示	不显示
A8	闪烁	红
A9	闪烁, 反像	红, 反像

AA	闪烁, 高亮	红, 高亮
AB	闪烁, 高亮, 反像	红, 高亮, 反像
AC	闪烁, 下划线	红, 下划线
AD	闪烁, 下划线, 反像	红, 下划线, 反像
AE	闪烁, 下划线, 高亮	红, 下划线, 闪烁
AF	不显示	不显示
B0	列分隔符	青绿, 列分隔符
B1	反像, 列分隔符	青绿, 列分隔符, 反像
B2	高亮, 列分隔符	黄, 列分隔符
B3	高亮, 反像, 列分隔符	白, 反像, 列分隔符
B4	下划线, 列分隔符	青绿, 下划线, 列分隔符
B5	下划线, 反像, 列分隔符	青绿, 下划线, 反像, 列分隔符
B6	下划线, 高亮, 列分隔符	黄, 下划线, 列分隔符
B7	不显示	不显示
B8	闪烁, 列分隔符	粉红
B9	闪烁, 反像, 列分隔符	粉红, 反像
BA	闪烁, 高亮, 列分隔符	蓝
BB	闪烁, 高亮, 反像, 列分隔符	蓝, 反像
BC	闪烁, 下划线, 列分隔符	粉红, 下划线
BD	闪烁, 下划线, 反像, 列分隔符	粉红, 下划线, 反像
BE	闪烁, 下划线, 高亮, 列分隔符	蓝, 下划线
BF	不显示	不显示

图 3-66 给出如何为仅输入字段指定 DSPATR(SP)键字 (建议数据内容为字符串)。

```
|...+....1....+....2...+....3....+....4....+....5....+....6....+....7....+....8
00110A      SPFLD      50I  I  5  4' ? * OPTION 1'
00120A          DSPATR(SP)
A
```

图 3-66 为仅输入字段指定 DSPATR(SP)键字

字段 SPFLD 中不能键入任何数据。如果用光笔选择了该字段，则返回字段 SPFLD 的数据为：

>\_\*OPTION\_1              其中\_代表空格。

图 3-67 给出在工作站用户用光笔选择一个字段时，字段的 MDT 位和第一个字段都被修改了。如果选择了某一字段，MDT 位置为 ON，并把字段第一个字符改为>。如果再次选择了该字段，则 MDT 位置为 OFF，第一个字符变为?。

通过指定开关字符，用户程序可防止在光笔选择了字段时第一个字符变为>或?。如果在用户程序对记录格式进行一次输入操作时 MDT 位为 ON，则该字段内容作为一个用户修

改了的字段返回给用户程序。

如果用 DSPATR(MDT)来将用光笔选择的字段 MDT 置为 ON，则应在其它记录格式中省略 MDTOFF 键字，或者应在显示有 MDTOFF 有效的记录格式之前读取该字段。

图 3-67 给出使用该属性的情况。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R LIGHTPEN  
00020A      FLD1      10  I 5 2' > * $12.50'  
00030A          DSPATR(SP MDT)  
00040A*  
00050A      R RCD2          OVERLAY MDTOFF  
00060A      FLD1      10  B 11 2  
A
```

图 3-67 指定 DSPATR(MDT)键字来把由光笔选择字段的 MDT 置为 ON

如果程序显示 LIGHTPEN，然后显示 RCD2，然后读 LIGHTPEN，工作站用户没有用光笔选择 FLD1，则由显示 RCD2 时把 FLD1 的 MDT 置为 OFF。

而且，FLD1 的开关字符按? 返回，即使并未选择该字段，开关字符显示为>。MDT 和开关字符都处于相反状态。

图 3-68 给出如何指定使用 P-字段用法的 DSPATR 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A      R RECORD          2DSPATR(&PFLD1)  
A      FLD1      5A      2 6DSPATR(&PFLD2)  
A      FLD2      5A      2 6DSPATR(&PFLD2)  
A      PFLD1      1A  P  
A      PFLD2      1A  P  
A
```

图 3-68 定义有 P 字段的 DSPATR

### 3.3.40 DSPMOD (显示方式)

对于 3148、3477 或 3197 的 D1、D2、W1、W2 显示站，用这个记录层键字定义使用两种显示方式中的那一种。**\*DS3(24×80)** 和 **\*DS4 (27×132)** 都可用于 3180、3477、3487 的 HA、HC、HG 和 HW 显示站，也可用于 3197 的 D1、D2、W1 或 W2 显示站。

这个键字的格式为：

DSPMOD (条件名)

只有当已用键字 DSPSIZ 同时定义了 24×80 和 27×132 两种显示尺寸时，键字 DSPMOD 才有效。有键字 DSPSIZ 定义的两种显示尺寸中的第一种就是缺省显示方式，记录将按此显

示方式显示，除非用键字 DSPMOD 指明了要使用第二种显时方式。

注：这个键字是运行时键字而非编译时键字。

只能用没有定义可选指示器的方式来用此键字定义使用缺省显示尺寸。

只有在与 6040 或 6140 控制器本地连接或与 5294 或 5394 控制器远程连接的 3180-2 或 3197 的 D1、D2、W1 或 W2 上，才允许以  $27 \times 132$  方式显示。如果没用这些控制器，将忽略键字 DSPMOD。

当有 DSPMOD 键字的记录引起显示方式改变时，所有正在显示的键字的记录，只要 DSPMOD 键字是活动的，这个记录的显示方式就会被保存在显示器上。设置 DSPMOD 无效或写入另一不带 DSPMOD 的记录将引起显示方式变回设备的主显示方式。

如果显示方式改变，忽略下列键字：

ALWROL	OVERLAY
ASSUME	PROTECT
CLRL	PUTOVR
ERASEINP/INZINP	PUTRETAIN
ERRMSG	SFLMSG
ERRMSGID	SFLMSGID
KEEP	

在建立文件时若对同一记录规定以上键字和 DSPMOD，则会产生建立时警告信息。不过在运行过程中不会给出诊断信息。

这个键字对用户定义的记录（USRDFN 键字）无效。

不能在子文件记录（SFL 键字）中定义该键字。

子文件将根据与之相关的子文件控制记录中的 DSPMOD 来显示。

可选指示器对这个键字有效。如果在处理过程中可选指示器为 ON，那用定义的显示方式来显示记录。但如果在处理过程中可选指示器是 OFF，则用缺省显示方式来显示。

注：转换显示方式类似于不用 OVERLAY 显示记录。

图 3-69 和图 3-70 给出如何定义 DSPMOD 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               DSPSIZ(*DS3 *DS4)
A       R RECORD1             DSPMOD(*DSP4)
A       R RECORD2
A       R RECORD3
A   03                         DSPMOD(*DS4)
A
```

图 3-69 定义 DSMPMOD 键字（例 1）

这个 DSPMOD 键字将会产生如下结果：

(1) 如果写 RECORD1，则 RECORD1 以\*DS4 方式显示。

- (2)如果写 RECORD2，显示器清屏且 RECORD2 以\*DS3 方式显示。
- (3)如果写 RECORD3 且指示器 03 为 OFF, RECORD3 以\*DS3 方式显示。RECORD2 仍保留在显示器上。
- (4)如果写 RECORD3 且指示器 03 为 ON，则显示器清屏且 RECORD3 以\*DS4 方式显示。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               DSPSIZ(24 80 *NORM +
A                               27 132 *WIDE)
A       R RECORD1
A   03                         DSPMOD(*WIDE)
A
```

图 3-70 规定 DSPMOD 键字（例 2）

### 3.3.41 DSPRL（从右到左显示）

用这个文件层键定义显示文件中的记录是从右到左写而不是从左到右写。

这个键字没有参数。

可选指示器对这个键字无效。

此键字只用在双向设备上。

图 3-71 给出如何定义 DSPRL 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               DSPRL
A       R RECORD
A           FIELD1      20A      5 5') emaN remotsuC('
A
```

图 3-71 规定 DSPRL 键字

### 3.3.42 DSPSIZ（显示尺寸）

用这个文件层键字定义用户程序所能打开的显示文件的显示尺寸。

这个键字的格式如下：

```
DSPSIZ(*DSW[*DSX])
DSPSIZ (行 列 [条件名 1] [行 列 [条件名 2]])
```

键字 DSPSIZ 是可选的。如果没有给一个显示文件定义这个键字，则这个显示文件在显示设备上只能以 24×80 的显示尺寸打开。可以用两种方法定义这个键字：

利用 IBM 提供的显示尺寸条件名。以任何顺序用\*DS3 和\*DS4 来定义，至少需要一个参数。但不能两次都定义同一个参数。

指定行和列允许用户定义的显示尺寸条件名。不用 IBM 提供的显示尺寸条件名，而是

自己利用行和列位置来定义显示尺寸（只有  $24 \times 80$ ,  $27 \times 132$  是有效的）。（可参见图 3-73, 图 3-75 和图 3-76）。

可以自己定义显示尺寸条件名而不使用 \*DS3 或 \*DS4。定义的显示尺寸条件名必须 2 到 8 个字符长，第一个字符必须是星号 (\*)。要在字段层上的 DDS 语句的 7-16 列指定用户定义条件名。（见图 3-75）。如果没指定用户定义条件名，则必须使用 IBM 提供的显示尺寸条件名。

如果不只定义了一个参数值，可参见本章稍后的“主和辅显示尺寸”。

可选指示器对这个键字无效。

### 3.3.42.1 主和辅显示尺寸

不管使用 IBM 提供的显示尺寸条件名还是直接规定行和列，第一个定义的显示尺寸就是主显示尺寸。如果还用了第二个显示尺寸，那么它就是辅显示尺寸。图 3-72 给出主和辅显示尺寸的例子。



图 3-72 用 DSPSIZ 定义主和辅显示尺寸

当为 DSPSIZ 不只定义了一个显示尺寸时，可在记录层或文件层的 DDS 语句的 7 至 16 列定义显示尺寸条件名。然后即可利用这些显示尺寸条件名限制键字状态和字段的位置。当主和辅显示尺寸都已定义时，显示文件对两种显示尺寸都有效。若想进一步了解这方面的知识，请参见 3.2.4。

注：如果为 DSPSIZ 定义了显示尺寸条件名，则不能使用 IBM 提供的显示尺寸条件名来定义。

$27 \times 132$  方式只能用在连到本地 6040 或 6041 控制器上或远程 5294、5394 控制器上的 3180-2 或者 3197D1、D2、W1 或 W2 设备上。如果不使用这些控制器， $27 \times 132$  方式的显示方式将被忽略。

下表给出有效的显示尺寸。

显示尺寸	显示设备	意义
*DS3 或 $24 \times 80$	3179 3180 3196 3197 3476 3486 3487 (型号 HA, HC, HG, and HW) 3488 5251 (型号 11 and 12) 5291 5292	24 行 $\times$ 80 共 1920 个位置
*DS4 或 $27 \times 132$	3180 3197 (型号 D1, D2, W1, and W2) 3477 (型号 FA, FC, FD, and FG) 3487 (型号 HA,	27 行 $\times$ 132 共 3564 个位置

	HC, HG, and HW) 3488 (使用 6040 或 6041 本地控制器, 27 x 132 显示属性的 5294 或 5394 远程控制器.)	
--	---	--

被指定为主显示尺寸的应该是显示文件最常用的显示尺寸。当实际显示尺寸是辅显示尺寸时需要附加的处理开销。

显示尺寸条件名可以提高显示文件的利用率，使之可用任意尺寸显示。举例来说，当使用子文件时，既可为  $27 \times 132$  方式定义每页 24 个记录，也可为  $24 \times 80$  方式定义每页 22 个记录。

图 3-73 和图 3-75、图 3-76、图 3-77 给出如何定义 DSPSIZ 键字以及显示尺寸条件名。

图 3-73 给出如何用 DSPSIZ 键字定义主和辅显示尺寸。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A*
00020A*                               1      2
00030A                               DSPSIZ(27 132 24 80)
00040A      R RECORDA
00050A      FIELDA     10 0   1 2
00060A      FIELDB     10 0   1 81
00070A      FIELDC     10 0   25 1
A
```

图 3-73 规定 DSPSIZ 键字（例 1）

在图 3-73 中，主显示尺寸 1 是  $27 \times 132$  方式，辅显示尺寸 2 是  $24 \times 80$  方式。FIELDB 超出 80 列位置，FIELDL 超出 24 行，所以数据描述处理器在用  $24 \times 80$  的辅显示尺寸时在扩展打印输出中给出\*NOLOC 的位置。

如果数据描述处理器给输入字段作了\*NOLOC 的标记，这个字段在运行时将使输入缓冲区数据返回到用户输入缓冲区。字段本身不显示。工作站用户不能输入或修改这些字段。对仅输出字段不作任何处理。

图 3-74 给出图 3-73 的编译清单

PICTURE16

图 3-74 编译清单

图 3-75 是用键字 DSPSIZ 规定主和辅显示尺寸的又一个例子。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               DSPSIZ(27 132 *WIDE 24 80 *NORMAL)  
00020A      R RECORDA  
00030A      FIELD A      10  0   1  2  
00040A      FIELD B      10  0   1  81  
00050A  *NORMAL                  1  50  
00060A      FIELD C      10  0   25  1  
00070A  *NORMAL                  23  1
```

A  
(

图 3-75 规定 DSPSIZ 键字（例 2）

图 3-75 和图 3-73 类似，唯一不同的是在辅显示尺寸（用户定义成\*NORMAL）上为 FIELDDB（第 1 行第 50 列）和 FIELDC（第 23 行第 1 列）定义了有效的辅助位置。

图 3-76 给出当一个文件以不同显示尺寸打开时如何给字段重定位。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               DSPSIZ(24 80 27 132)
00020A      R RECORDA
00030A      FIELD1      10  0  23  2
00040A
00050A
00060A  *DS4      26  2
A
```

图 3-76 规定 DSPSIZ 键字（例 3）

在图 3-76 中，FIELD1 在两种显示尺寸上都有有效位置。它出现在每一种显示尺寸最后一行的下一行上。

图 3-77 给出如果没规定显示尺寸条件名，字段的显示位置依赖于 DDS 中+号计算后的值。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               DSPSIZ(*DS4 *DS3)
00020A      R RECORD1
00030A      FIELD1      21      2 70
00040A      FIELD2      10      +10
A
```

图 3-77 规定 DSPSIZ 键字（例 4）

在图 3-77 中，用键字 DSPSIZ 规定的每个显示尺寸为每个字段都计算了所在的行与位置。如果相加得到的位置值超出 80 列，字段的位置将根据显示尺寸来定义。图 3-78 是图 3-77 的编译清单。

图 3-78 编译清单

特殊情况：下面是定义 DSPSIZ 键字时可能遇到的特殊问题：

DSPSIZ(\*DS3 \*DS4) 显示尺寸\*DS4 的所有字段位置和显示尺寸\*DS3 的一样。

一个记录的全部字段可能都不适合用于文件时打开的显示尺寸。在这种情况下，不显示任何字段。记录将被处理成适合这些字段显示的大一点的显示尺寸。记录在被删除或覆盖之前将一直保持活动。用户程序能访问这些记录（活动的）。输入请求会被送到显示设备，工作站用户必须做出反应来满足请求。

一个子文件记录的所有字段必须适合定义的子文件页，并且一个整页的显示尺寸必须和在处理时所用的显示尺寸适合。可以有显示尺寸条件名的 SFLPAG (子页) 键字来规定有效的显示尺寸。

下面这些记录不占用显示空间：

—没有定义字段的记录（不同于没有选中字段的记录）。

—只有隐藏字段、信息字段和程序到系统字段的记录。

—规定了 CLRL 键字且没有可输入字段的记录。

（这些记录可保持显示，但不能被 OS/400 程序的输入操作识别，或者这些记录可用 ERASE 键字清除）。

为了实施及程序设计的要求，这些字段假定在位置 00 上（从第 0 行到第 0 行）。在输出操作时，位置 00 的记录覆盖另一个位置 00 的记录。当发生重叠时，先前的记录被忽略而停止活动。新的在位置 00 的记录是活动的且可被程序读取。

如果一个记录格式中的两个字段在相同的位置（行和列都同），则它被看作重叠字段。重叠字段在操作时不显示。OS/400 程序处理每一个字段时都要检查，以保证它不覆盖先前处理的字段。

如果一个字段发生了重叠，它会被当作可选字段而不被选择。为了完成处理时的检查，数据描述定义必须保证记录中的所有字段都按主位置顺序，即使规定了条件名也要这样。举

例来说，假如记录格式仅定义了一个输入字段，根据字段位置的规定，这个字段与前一个输出字段发生重叠，则工作站用户不能输入任何数据，因为该输入字段从没显示过。

注：在显示文件中所见到的主位置顺序不能因规定不同的辅显示尺寸而改变位置。（否则将会产生严重错误且文件不能生成）。

### 3.3.43 DUP（复制）

用此字段层键字启动键盘上的复制键。当光标停在输入字段时按下复制键，这指出本字段的数据是用前一个输入操作送入的记录来复制。实际上复制是由程序来完成的。（参见本节稍后的“复制键编程”）。

如何在 System/36 环境使用的文件中定义 DUP 键，请参见附录 F。

键字的格式为：

DUP[（应答指示器[‘说明’]）]

不能在浮点字段上定义 DUP 键字。

使用此键字的数字字段要有一个应答指示器。数字字段的十六进制 1C 不会返回到缓冲区，但十六进制 F0 会返回剩余的字段位置。

可选指示器对这个键字有效。

复制键编程：当按下复制键字时，OS/400 程序会按下述方式处理字段：

如果是字符型字段，字段中显示的数据会按原样返回给程度。十六进制 1C 放在光标位置及右边剩余位置上。（十六进制 1C 在显示时是星号）。若定义了应答指示器，则该指示器为 ON。

如果是数字型字段且定义了应答指示器，十六进制 F0 被放在光标位置和字段的剩余位置上。如果没指定应答指示器，则返回程序的为多个十六进制 1C。

在程序中可用下列过程复制全部字段（无论字符型的还是数字型的）：

1. 在 DDS 中为显示中的每个输入字段定义两个字段：

a. 定义一个字段为输入字段，且为此字段定义一个带应答指示器的 DUP 键字。可以定义一个带选择指示器的字段，在此字段第一次显示时可选指示器为 OFF。这样做可避免工作站用户在字段第一次显示时使用 DUP 键。

b. 定义另一个字段为隐藏字段（38 列为 H）。

2. 第一次输出操作时，把 DUP 的指示器设为 OFF，这样做避免工作站用户用 DUP 键。

3. 第一次输入操作时，把输入字段移到隐藏字段。这样做可以保存输入的值以便稍后使用。

4. 在每一个随后的输出操作中，把 DUP 键字的指示器设为 ON。这样做使工作站用户能够使用 DUP 键字。

5. 在每一个随后的输入操作中，检查 DUP 的应答指示器。如果应答指示器为 OFF，输入数据应被放到隐藏字段。如果应答指示器为 ON，则可以使用隐藏字段中的值。

注：在子文件中用 DUP 键字时，在步骤 3 和 5 之后应执行更新操作以把隐藏字段的值保存在子文件中，当再一次读子文件时返回这个值。

6. 重复步骤 4 和 5，以使用复制键输入后来的数据。

你也可每次复制字符字段的一个字符，先把它们保存在数组中，然后每次从数组中移出一个字符且检查复制键是否指示 16 进制 1C。

可以每次复制数字字段的一个数字，方法是把该字段定义字符型，并且在移出十六进制 1C 后把它移到数字字段。

检查是否按下了复制键：

- 对数字型字段，应答指示器是必须的。
- 对字符型字段，应答指示器是可选的。

如果按下了复制键，字段将在光标位置及剩余位置保留十六进制 1C。

有效性检查的限制：有效性检查关键字（CHECK、COMP、RANGE 和 VALUES）能跟 DUP 键字一起规定。但当按下复制键时，有效性检查关键字将失去作用。

如果记录中另一个字段的有效性检查失败，OS/400 程序会再次读显示。无论工作站用户是否重新键入复制字符或者按下复制键，复制应答指示器仍把当前状态返回到程序。

图 3-79 给出如何定义 DUP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1
00020A      FLDA      5   I  3  2
00030A  15          DUP(16 'FLDA was duped')
00040A      FLDAH     5   2H
A
```

图 3-79 规定 DUP 键字

FLDA 是一个可输入的字符和十六进制数的字段，工作站用户可按复制键。在程序中，必须测试 FLDA 中的数据以保证它只包括从 0 到 9 的数字。在 RPGIII 中，可用 TESTN 操作码。在 COBOL 中，可用数字类测试。在 BASIC 中，可在把 FLDA 赋值给 FLDAH 语句之前规定一个 ON CONV 语句。在 PL/1 中，可在把 FLDA 赋值给 FLDAH 语句之前规定一个 ON 语句的错误条件。

字段 FLDAH 是一个隐藏数字字段，长度和 FLDA 一样。当应用程序读 RECORD1 且发现应答指示器 16 为 OFF 时，程序把 FLDA 送到 FLDAH 并且使用 FLDAH（数字型字段）。当应用程序读记录 RECORD1 且发现应答指示器 16 为 ON 时，按下复制键且 FLDA 包含多个十六进制 1C。程序使用 FLDAH 且不改变它的内容（它的内容是先前从显示设备接收的输入内容）。

### 3.3.44 EDTCDE (编辑码)

利用这个字段层关键字编辑数字输出字段。

这个关键字的格式为：

EDTCDE (编辑码[\* | [浮动货币符号]])

根据编辑的定义，编辑包括如下字段显示时的变化：

取消前置零。

把字段用“,”和“.”做标记，这样给出小数点位置和千分号。

负值可与负号或右边的 CR 表示出来。

零值可显示为 0 或空白。

在最高有效位的左边放若干星号，提供星号保护。货币符号（相当于系统值 QCURSYM）放在紧靠最高有效位的左边，（称作浮动货币符号）。要用固定货币符号，可使用 EDTWRD 键字。

字段可用用户定义的编辑码做进一步的编辑。可参见 3.3.44.2 的“用户定义的编辑码”。

EDTCDE 可以满足大部分编辑要求。当它不能满足要求时，可使用 EDTWRD。

只有当字段的第 35 列是“Y”或空白时，EDTCDE 键字才有效。这个键字的作用是把第 35 个位置的缺省值变成 Y。

不能对一个字段同时定义 EDTCDE 和 EDTWRD 键字。如果先为一个字段在数据库文件中定义了 EDTCDE 键字，那么则不需在显示文件中还为这个字段定义 EDTCDE 键字。可在 29 列定义 R 来引用先前定义的字段。被引用字段的编辑码包括在显示文件中。如果显示文件的字段定义了长度、数据类型或者小数位，而被引用字段的编辑没包括在显示文件中，那就必须重新为显示文件定义编辑方式。

键字 DFT 和 DFTVAL 不能和 EDTCDE 同时定义。

可选指示器对 EDTCDE 无效。

所有类型的文件定义编辑码和编辑字的规则都是一样的。可以定义两种编辑方式：一种是 OS/400 编辑码，另一种是用户定义编辑码。

### 3.3.44.1 OS/400 编辑码

OS/400 编辑码如下：

1 至 4

A 至 D

J 至 Q

X 至 Z

注：OS/400 系统硬件用指示符 F 操作，这与使用编辑码 X 等价。编辑码 X 使空的键盘转换（35 列）置为缺省的仅数字型（属性 Y）。字段的显示长度是由键盘转换决定的，而不是由编辑码 X 决定的。（缺省的仅数值型 Y 属性可为十进制数增加 1 位）。如果用编辑码 X 定义 DATE 和 TIME 键字，不显示分隔符。

星号填充或浮动货币符号：可用编辑 1 至 4、A 至 D、J 至 Q 有选择地定义星号填充或浮动货币符号。

当定义星号填充时，每个被取消的零的位置都打印一个星号（\*），零平衡字段打印成一个完全由星号组成的字段。

当定义浮动货币符号时，符号出现在第一个有数位的左侧。当用编辑码抑制零平衡时，在零平衡位上不打印货币符号。（所定义的符号必须和货币符号的系统值（QCURSYM）匹配。也必须与文件建立时的符号匹配，但不必非与文件所用的符号匹配）。

注：在文件建立之后如果编辑码改变，将使用文件建立时所定义的编辑码。除非文件重新生成，否则不能使用新的编辑码。

下表是 OA/400 编辑码提供的功能总结。

图 3-80 编辑码

图 3-80 OS/400 编辑码

注：

1.QDECFTM 系统值决定小数点符号、(美国用“.”)、千分号(美国用“,”)以及压缩零的类型(取决于“.”和“,”的位置)。

2. 编辑码 Y 取消 3 至 6 位长字段最左端的零，也取消 7 位长字段最左端的两个零。编辑码 Y 还可在月、日、年之前按如下格式插入 “/”：

nn/n  
nn/nn  
nn/nn/n  
nn/nn/nn  
nnn/nn/nn

如果键字 DATE 规定了 EDTCDE(Y)，运行所用的分隔符是运行时作业属性 DATSEP。“/”是 DATSEP 的缺省值。如果在文件建立时 DATFMF 是 JUL，日期的格式通常是 nn/nnn，这时 EDTCDE(Y) 无效。

3. 编辑码 Z 去掉数值型字段的符号 (+或-)，在写字段之前，符号位被转换成十六进制的 F。

注：如果想更多了解 QDE CFMT 系统值，请参看工作管理手册。

#### 3.3.44.2 用户定义的编辑码

编辑码 5 至 9 是用户定义的编辑码，用户定义编辑码比 OS/400 编辑码有更多的编辑方式。例如，可能需要编辑包括字符的数字（如电话号码）或不只一个小数点的数字。就可使用用户定义编辑码来实现。这些编辑码为 QEDIT5、QEDIT6、QEDIT7、QEDIT8、QEDIT9。这些编辑码能被 DDS 或高级编辑语言以数字（5、6、7、8 或 9）引用。

用户定义编辑码是 OS/400 的目标，必须在显示文件生成之前存在。它是用建立编辑描述命令 (CRTEDTD) 生成的。当生成一个有用户定义编辑码的文件时，编辑信息可从先前生成的描述中抽取出来。在显示文件生成之后改变用户定义编辑不影响该显示文件，除非该显示文件重新生成。

下表给出编辑码、未被编辑的源数据、编辑后的输出情况。零压缩和小数符号都由系统值 QDECIMFT 决定。日期分隔符由作业属性 DATSEP 决定。图中，QDECIMFT 为 X (空白)，DATSEP 为 “/”。

图 3-81 编辑码例子

图 3-81 有效的编辑码、源数据及编辑后的输出

注：

1. 图中 X 代表空白。
2. 编辑码 Y 取消 3 至 6 位长日期字段最左端的零和 7 位长字段最左端的两个零。若想

了解更多信息，请参看 3.3.44.1 的图 3-86 注中的第 2 项。

3. 编辑码 Z 去掉加减号及前置零。

图 3-82 给出如何定义 EDTCDE 键字。

00010A	PRICE	5	2	1	10EDTCDE(J)
00020A	SALES	7	2	2	10EDTCDE(K \$)
00030A	SALARY	8	2	3	10EDTCDE(1 *)
A					

图 3-82 规定 EDTCDE 键字

PRICE 的显示长度为 7，原因是使用了编辑码 J，使字段包含了一个小数点和一个结尾的减号。它被编辑成：

ddd.dd-

其中的 d 代表一位数字。

SALES 的显示长度为 11，原因是使用了编辑码 K 和浮动货币符号。它被编辑成：

\$dd,ddd.dd-

SALARY 的显示长度是 10，因为使用了编辑码 1 和用星号填充，它被编辑成：

ddd,ddd.dd

### 3.3.45 EMTMSK (编辑屏蔽)

用此字段层键字和 EDTCDE 或 EDTWRD 一起为字段定义编辑屏蔽。当显示用此键字的字段时，字段的用户定义区被保护。当工作站与支持增强数据流的控制器相连时，EDTMSK 键字才有效。

此键字的格式为：

EDTMSK (编辑屏蔽)

必须规定一个参数，编辑屏蔽由两个字符组成，一个是&，另一个是空格符。&表示字段被保护的部分，空格表示字段不被保护的部分。编辑屏蔽必须与编辑的显示字段编辑之后的长度相同。不被保护的位数必须等于字段的程序长度。用户必须小心保护非数字数据，因为如果字段改变了，被保护数据不能返给用户。

有键字 EDTMSK 的字段一定是用法 I 或用法 B。它必须有 EDTCDE 或 EDTWRD 键字的规定。

下列键字不能和 EDTMSK 在同一字段上定义：

AUTO (RAB, RAZ)

CHECK (AB, MF, RP, RZ, RLTB)

CHOICE

CNTFLD

DSPATR (OID SP)

可选指示器对此键字无效。

由于 EDTMSK 键字用来组成某字段的总段数的原因，它减少了可用的输入字段数。例如 EDTMSK (' & & ')，由 3 段组成，这样就把可用的输入减少了 3。

如果键字 ENTFLDATR 和 EDTMSK 一起定义，则结果不可预料。

图 3-83 给出如何定义 EDTMSK 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECORD
A          F1      11  0B  3  4EDTWRD('0(  ) - ')
A                      EDTMSK(' & & & ')
A          F2      6   0B  4  4EDTCDE(Y)
A                      EDTMSK(' & & ')
A
```

图 3-83 规定 EDTMSK 键字

在此例中，字段 1 的破折号和圆括号被保护。字段 2 中的数据分隔符也被保护。

### 3.3.46 EDTWRD (编辑字)

如果用键字 EDTCDE 不能实现设想的编辑，可使用这个字段层键字。

这个键字的格式是：

EDTWRD ('编辑字')

编辑字定义字段值显示的格式，通过直接插入字符解释数据，例如小数点、逗号、浮动或固定货币符号，以及信用平衡标识。编辑字也可用来压缩前置零以及提供星号填充保护。

如果一个字段先在数据文件中定义了 EDTWRD，则无需在显示文件中再定义 EDTWRD。此时可在 29 列写 R 以指明引用先前定义的字段。被引用字段编辑方式包括在显示文件中。但是，如果为显示文件字段定义了长度、数据类型、小数位，则为引用字段定义的编辑不包含在显示文件中，此时必须重新在显示文件字段中定义编辑方式。

#### 3.3.46.1 编辑字的组成部分

编辑字由三部分组成：主体、状态和扩充。

图 3-84 给出编辑字的三部分。

PICTURE 18

图 3-84 编辑字的三部分

主体是数字从数据字段传到输出记录数字的空间。主体从编辑字的最左端开始。它所包含的空格的数目（每有一个零或一个星号就加 1）等于所编辑数据字段的位数。如果零或星号是编辑字的第一个字符，主体所含的空格数等于数据字段的位数。主体在可被数字替代的最右端字符处结束。

状态显示数据字段的加减号（+、-）。状态延伸到主体右端的 CR 或减号（-）。这两个符号只在字段为负时才打印。没有 CR 或减号的编辑字没有状态位。

扩充位置不能由编辑操作修改。扩充从状态位右端的第一个字符开始，（若没定义状态，则扩充从主体右端的第一个字符开始）到编辑字的最右字符结束。

### 3.3.46.2 编辑字主体格式

下列字符在编辑字主体中有特殊意义：

空格：一个空格可被数据字段相关位置的字符代替。一个空格位置指定的一个数字位置。

&：每一个&符在被编辑字段中产生一个空格符。&符不显示。注意当定义一个输出/输入字段的编辑字时，在编辑字空格符之间定义&会引起数据错误，这是由于在一个仅数字字段中嵌入的空格符会被转换成零。

零：一个零能停止零压缩。把零放在零压缩的最右位置上表示结束零压缩。零要被数据字段相关位置的字符代替，除非那个位置也为零。数据中任何出现在停止零压缩符右面的零都被显示。停止零压缩符的位置被当成一个数位。但当它是第一个字符时，它就不能再代表数位。除非它是 EDTWRD 的第一个字符，否则至少要取消开头的一个零。而它也不能再算作一个数字，因为此时字段中空格数等于字段中数字的个数。每个被压缩的零都由一个空格符代替。零用星号代替（星号保护）。把星号放在停止零压缩的最右位置。每个被压缩的零都用一个星号替代。

注：如果你用的显示文件是在 V2.1 版本前生成的，如果重编译它，在使用 EDTWRD 键字的地方可能出现不同的输出。

星号（\*）：零前面的星号当成星号保护，这种情况下，零被做为常量打印。任何出现在停止零压缩符右边的星号（\*）或零都是常量。

货币符号：如果在紧靠零压缩码左边放一个货币符号，那么货币符号将插入到第一个有效数字的左侧。用这种方式的货币符号称作浮动货币符号。

当在编辑字的最左端放一个货币符号时，它是固定的且每次都在相同的位置打印。用这种方式的货币符号称作固定货币符号。

货币符号不是数字可替代位置。货币符号必须与系统值 QCURSYM 相匹配。

小数点和逗号：小数点和逗号打印时放在编辑字中出现的同一相对位，除非它们在最高有效数字的左侧时，它们用空格符或星号替代。

所有在编辑字中在有效位右侧的其它字符都被打印。而在最高有效数字左侧的字符将被空格或星号替代（如果使用了星号保护）。

如果一个常数要在最左位置打印，此常数必须先放一个先置零且该字段长度加 1。

### 3.3.46.3 编辑字状态格式

下列字符在编辑字状态部分出现时具有特殊的含义：

&：在被编辑的输出字段中产生空格符。&本身不能放在被编辑的输出字段中。

CR 或减号(-)：如果被编辑输出字段的符号是正号(+)，加号的位置将为空格。如果符号是(-)，原样保留。

### 3.3.46.4 编辑字扩充格式

编辑的扩充部分的字符是必需的。扩充部分不能包括空格。如果被编辑的输出字段需要空格，可在主体中定义&符。

定义有效编辑字：用以下规则定义有效编辑字：

关键字 EDTWRD 对仅数字字段（在 35 规定了 Y）有效。

不能对同一字段同时规定 EDTWRD 和 EDTCDE。

关键字 DFT 和 DFTVAL 不能和 EDTWRD 一起规定。

把编辑字用撇号(')括起来。

编辑字中的空格和数字位置的停止零压缩字符的总数必须和字段的长度相等。

如果停止零压缩符是编辑字中的第一个字符，则空格的总数应和字段长度相等或比字段长度少 1。

当使用浮动货币符号时，货币符号不能算作一个数位。例如，如果为一个长 7 位，有 2 位小数的字段定义了货币符号，编辑字为：

EDTWRD ('---\$0.-- ')

其中 ‘-’ 代表空格。

如果想对负数给出一个负号，要在编辑字中包括符号，把减号(-)或字母 CR 放在最后一个数字可替换字符的右侧。只有当数字是负数时，减号和 CR 才打印。

可选指示器对此关键字无效。

图 3-85 给出对字段的程序值和显示值用同一编辑字的例子。

PICTURE 19

图 3-85 编辑字例子

图 3-86 给出如何规定 EDTWRD 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00020A          FIELD A        7 2 5 2EDTWRD(' $0. ')  
A
```

图 3-86 规定 EDTWRD 键字

### 3.3.47 ENTFLDATR (进入字段属性)

用这个字段层、记录层、文件层的键字定义字段先导属性，无论光标字段在何处，先导属性修改规定的属性。当在字段层和记录层都作了规定时，用字段层定义。当工作站与支持增强数据流的控制器相连时，键字 NETFLDATR 才有效。

此键字的格式为：

ENTFLDATR[ ([颜色][显示属性][光标可见]) ]。

对于此键字参数是可选的。

颜色参数定义了在彩色工作站上光标进入一个字段时该字段显示的颜色。该参数用 \*COLOR 值的表达式格式。

颜色参数的有效值是：

值	含义
BLU	蓝色
GRN	绿色
PNK	粉红色
RED	红色
TRQ	蓝绿色
YLW	黄色
WHT	白色

如果没规定颜色参数，缺省值为白色。

显示属性参数定义了当光标进入字段时该字段的显示属性。该参数用(\*DSPATR 值 1<值 2<值 3……>>)的表达式格式。

显示属性的有效值是：

值	含义
BL	闪烁
CS	列分隔符
HI	高亮度
NO	不显示
RI	反象显示
UL	下划线

隐含显示属性是 HI。

注：显示属性 CS, HI 和 BL 能使在 5292, 3179, 3197-C1 和 C2, 3487 - HC, 3488 (6) 工作站中的字段为彩色显示。当使用显示属性 HI、RI 和 UL 时，分隔线不出现，更进一步了解这方面的知识，可参看 3.3.30 中的“COLOR (颜色) 键字。”

光标可见参数允许用户规定当光标进入一个字段时，它是否被用户见到。\*CURSOR 表示光标可见，而\*NOCURSOR 表示光标不可见。缺省值是\*CURSOR。当在键字 ENTFLDATR 中规定\*NOCURSOR 时，被定义的字段在 35 列必须是 I。如果字段不是 I 型数据，则隐含值用光标可见参数。

有 ENTFLDATR 键字的字段必须是一个输入字段，对有 DSPATR(PR)规定的字段中忽略 ENTFLDATR 键字。

可选指示器对此键字有效。

如果 ENTFLDATR 与 EDTMST 一起规定，则结果不可预测。

(6)依据显示设备连接的监视器。

图 3-87 给出如何定义 ENTFLDATR 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A  
A      R RECORD  
A      F1          10A  B  3  4ENTFLDATR  
A      F2          10A  B 13  4ENTFLDATR((*COLOR RED))  
A      F3          10I  B 16  4ENTFLDATR(*NOCURSOR (*DSPATR HI RI))
```

图 3-87 规定 ENTFLDATR 键字

在此例中，F1 的颜色是白色，属性是高亮度，光标可见。F2 的颜色是红色，属性是高亮度，光标可见。F3 的颜色是白色，属性为高亮度反象显示，光标不可见。

#### 2.3.48 ERASE (清除)

这个记录层键字与键字 OVERLAY 一起使用，当写入记录时，定义为参数值的记录从显示器上删除。

该键字的格式是：

ERASE (记录名 1[记录名 2…[记录名 20]])

用做参数的记录必须在文件中存在。

ERASE 可能不只定义一次，每次定义 ERASE 时必须一起定义 OVERLAY。

如果对一个输出操作 ERASE 和 CLRL 两键字都起作用，此时尽管定义了 CLRL，定义了 ERASE 的记录仍会被删除。

已经显示的记录，如果有 CLRL 定义的非输入字段，则该记录不能被另一记录上的 ERASE 删除 (ERASE 无效)。

如果指定的记录并没有显示，则 ERASE 功能对该记录无效。

可选指示器对这个键字有效。

注：这个功能要求额外的数据传输，仅用在不想删除所有其它记录格式时才使用。也可以定义有可选指示器的 OVERLAY 键字，这样可在删除所有其它格式时有所选择。

图 3-88 给出如何规定 ERASE 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00020A      R REC1  
A      :  
A      :  
A      R REC2          OVERLAY  
A      :  
A      :  
A      R REC4          OVERLAY  
A      ERASE (REC1)  
A
```

图 3-88 规定 ERASE 键字

### 3.3.49 ERASEINP (清除输入)

用这个记录层键字与 OVERLAY 键字一起使用来删除已经显示的非保护的可输入字段。(非保护的可输入字段是键字 DSPATR (PR) 无效的字段)。这些字段在所定义的记录格式显示前被删除。现在定义的记录格式中的可输入字段不能被删除。

在 System/36 环境文件中, 如何使用 ERASEINP 键字请参看应用显示程序手册。

这个键字的格式为:

ERASEINP[ (\*MDTON | \*ALL) ]

使用\*ALL 参数可以删除所有已经显示的可输入字段。要只删除所有修改数据标识(MDT)为 ON 的可输入字段, 使用\*MDTON 参数, 定义 ERASEINP(\*MDTON)或 ERASEINP 具有和按下删除输入键一样的效果。

定义 REASEINP 键字时必须同时定义 OVERLAY 键字。

如果在同一记录格式中既定义了 MDTOFF, 也定义了 ERASEINP, 将会产生两种情况:

- 除非定义了 MDTOFF(\*ALL), 否则 ERASEINP(\*ALL)包含了 MDTOFF(\*UNPR)。
- 如果和 MDTOFF(\*ALL)一起定义了 ERASEINP 或 ERASEINP(\*MDTON), 最终结果如同一起定义了 ERASEINP(\*ALL)和 MDTOFF(\*ALL)。

如果键字 ERASEINP 和 PROTECT 对一个输出操作同时有效, OS/400 程序将首先依据 ERASEINP 中的参数删除可输入字段, 然后从键盘上保护所有显示的可输入字段。

ERASEINP 能减轻线路通信压力, 因为可重用已经显示的记录格式, 无须重新发送显示。

如果 ERASEINP 键字和 DSPMOD 键字在同一记录中规定, 在文件生成时将出现警告信息。在运行时间, 当显示方式改变时将忽略 ERASEINP 键字的存在。

可选指示器对此键字有效。

图 3-89 给出如何规定 ERASEINP 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A*  
00020A*  
00030A*  
00040A      R RECORD1          OVERLAY  
00050A      :                  ERASEINP  
00060A      :  
00070A      :  
00080A*    清除屏幕上所有修改过的输入属性字段  
00090A      :  
00100A      :  
00110A      :  
00120A      R RECORD2          OVERLAY  
00130A      :                  ERASEINP (*ALL)  
00140A      :  
00150A      :  
00160A*    不管修改过没有, 清除屏幕上所有非保护的输入属性字段  
00180A  
00190A  
00200A      R RECORD3          OVERLAY  
00210A  32                  ERASEINP (*MDTON)
```

00220A N32			ERASEINP (*ALL)
00230A	FIELD1	5 I	DSPATR (PR)
00240A	:		
00250A	:		
00260A	:		
00270A*	不清除 FIELD1		
A			

图 3-89 规定 ERASEINP 键字

### 3.3.50 ERRMSG (错误信息) 和 ERRMSGID (错误信息标识)

用这两个字段层键字标识显示在信息行上的与此字段有关的信息。

如果这两个键字中的任一个和 DSPMOD 键字在一个记录中同时定义，在建立文件时将显示警告信息。在运行时，当显示方式改变时这两个键字将失效。

可选指示器对这两个键字有效。

#### 3.3.50.1 ERRMSG 键字

此键字的格式为：

**ERRMSG** ('信息内容' [应答指示器])

对 ERRMSG 规定信息内容参数，可选地定义应答指示器。信息内容是要显示的信息。  
(不支持帮助键，当按下帮助键时不能显示帮助信息)。

如果定义了应答指示器，它应该和限定 ERRMSG 条件的可选指示器相同。在错误信息显示之后进行的输入操作中，OS/400 程序关闭应答指示器。如果应答指示器和可选指示器相同，它们将被同时关闭。此规则的一个例外就是应答指示器也用于某些其它键字，如 CHANGE、CAnn 或 CFnn。在这种情况下，应答指示器的开和关是基于在 CHANGE 或 CFnn 功能的结果。当定义了一个应答指示器时，信息前 50 个字符用作指示器说明。分开的应答指示器说明对键字 ERRMSG 无效。

#### 3.3.50.2 ERRMSGID 键字

这个键字的格式是：

**ERRMSGID** (信息标识符 [库名/]信息文件 [应答指示器] [&信息数据])

ERRMSGID 的参数指定了：

将要显示信息的信息标识符

信息文件和库（库是可选的）

应答指示器（可选）

信息数据字段名（可选）

如果定义了应答指示器，该指示器应和限制键字 ERRMSGID 条件的可选择指示器相同。在显示出错信息后的输入操作中，OS/400 程序关闭应答指示器。不过，如果也在其它键字如 CHANGE、CAnn 或 CFnn 定义了应答指示器，则应答指示器的开和关依赖于 CHANGE、CAnn 或 CFnn 功能结果。

注：ERRMSGID 中不能定义指示器说明。

如果定义了信息数据字段，则字段中包含信息的替换说明。该字段必须存在于记录格式中，必须定义为用方法为 P 的字符型字段。详细信息参看 CL 参考手册中的发送程序信息 CL 命令。

图 3-90 给出如何规定 ERRMSG 和 ERRMSGID

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R CUSMST  
00020A      :  
00030A      :  
00040A      :  
00050A      QTYORD      10A I 5 3  
00060A 61          ERRMSG('No stock available' 61)  
00070A 62          ERRMSG('Partial stock available' +  
00080A          62)  
00090A 63          ERRMSGID(MSG2000 CONSOLEMMSG 63 +  
00100A          &RPLTXT)  
00110A      RPLTXT      78A P  
A
```

图 3-90 规定 ERRMSG 和 ERRMSGID 键字

选择键字的优先级：对同一个字段可以多次定义 ERRMSG 和 ERRMSGID。在程序处理中，用可选指示器来选择要显示的某个信息。

在同一个输出操作中，即使几个字段的信息都在起作用，一次也只能显示一个信息。是程序来选择首先显示哪个字段的信息。

如果在输出操作中几个键字对一个字段同时有效，首先显示的信息如下：

ERRMSG（如果选择了多个 ERRMSG 键字，显示程序选择的第一个信息）。

ERRMSGID（如果选择了多个 ERRMSGID 键字，显示程序选择的第一个信息）。

仅当没有错误信息键字显示时，信息字段才能显示。

键字 SFLMSG 或 SFLMSGID 的优先列表，请参看 3.3.141.3 的内容。

信息显示时产生的条件：用 ERRMSG 和 ERRMSGID 显示的信息与检测字段有效时，由 OS/400 程序显示的很相似。

当由 ERRMSG 或 ERRMSGID 显示信息时，所有在显示中的字段都被保存，包括与信息有关的字段。除了可选指示器外，忽略输出缓冲区中的数据（就是说，任何从程序来的新数据都不能显示）。

显示信息时有效的功能键为：

在文件层上定义的功能键，为信息显示的记录格式定义的功能键，如果在信息显示时被选择则有效。

信息显示时会产生如下条件：

对所有错误：

—信息为高亮度。

—光标闪烁并且锁住键盘直到工作站用户按 RESET 键。

对与可输入字段有关的错误：

—所有产生错误的字段都反象显示。

如果出错字段属性同时为下划线(UL)和高亮度(HI)，或同时为下划线(UL)和 COLOR(BLU, WHT 或 YLW)，则显示不为反象。

—光标重定位在出错的第一个显示字段上。

对与仅输出字段有关的错误：

—字段的显示属性不变

—光标不重新定位（不改变位置）

注意：有一些显示属性可使字段在 IBM 彩色显示站上表现为彩色字段。参见本章 3.3.30 的“COLOR”。

反象显示字段的恢复：由于系统查到了输入错误或由于规定了 ERRMSG 和 ERRMSGID 键字，都可使字段反象显示。通常，OS/400 程序可在下一次送到显示的 I/O 操作时恢复图像，一般是程序的下一次请求。恢复在执行请求功能之前完成。下面是不能使 OS/400 程序恢复反相显示的程序请求特例：

带有取消的输入请求（用 NOWAIT 取消读操作）

当 KEEP 键字有效时关闭请求

任何对子文件记录的请求（没有数据送往设备）

不显示子文件控制记录或子文件记录的子文件控制记录格式的输出操作（例如：清屏、删除、子文件初始化）。

### 3.3.50.3 限制和注意事项

下面这些适用于 ERRMSG：

在 ERRMSG 和 ERRMSGID 键字有效时，先执行键字的处理。如果这两个键字都无效，则记录按通常方式处理。

当 RMVWDW 键字有效时，不显示错误信息。

ERRMSG 和 ERRMSGID 对仅输出、仅输入或输出/输入字段有效。不能为常数字段、隐藏字段、系统到程序字段和信息字段规定这两个键字。

对于输入及输出字段，假如记录中有正在显示的字段，ERRMSG 和 ERRMSGID 才能生效。

ERRMSG 和 ERRMSGID 不能在子文件记录格式中定义。怎样显示子文件的出错信息，请参看 3.3.141 中的“SFLMSG 和 SFLMSGID”。

如果从最后一次输出操作起变量起始行号（由 SLNO(\*VAR) 定义的）已改变，则 ERRMSG 和 ERRMSGID 被忽略。

如果定义了 ERRMSG 和 ERRMSGID，也应在建立显示文件 (CRTDSPF) 和修改显示文件 (CHGDSPF) 命令中规定 RSTDSP(\*YES)。否则，在文件挂起时，在显示上的数据可能会丢失。

在引起显示方式改变的输出操作中，ERRMSG 和 ERRMSGID 被忽略。

### 3.3.51 ERRSFL (错误子文件)

用这个文件层键字定义使用系统提供的错误子文件显示的信息。在错误子文件中显示的信息应该是系统有效性检查信息和与下例键字有关的信息：

ERRMSG	CHECK(VN)
ERRMSGID	CHECK(VNE)
SFLMSG	COMP
SFLMSGID	RANGE
CHECK(M10)	VALUES
CHECK(M11)	

与下列输入错误有关的有效性检查信息，也在错误子文件中显示：

浮点运算错误

### 小数位错误

这允许你在记录写往显示器时对所有给出的出错信息翻页，也允许在从显示器读记录时对所有有效性检查错误信息翻页。系统在信息行上显示错误子文件，如果信息行和已经显示在屏幕上的记录发生重叠，则忽略 ERRSFL 键字。

本键字没有参数。

可选指示器对这个键字无效。

图 3-91 给出如何规定 ERRSFL 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               MSGLOC(24)
00020A                               ERRSFL
00030A      R RCD1
00040A      FIELD1      5A   B  2  3
00050A  10                         ERMSGID(MSG0001 MSGF1 10 &MDTA)
00060A      FIELD2      5A   B  5  7
00070A                           ERMSG('ERROR MSG 1' 11)
00080A      FIELD3      4S   I  7  7RANGE(1000 9999)
00090A                           CHKMSGID(MSG0002 MSGF1 &MDTA1)
00100A      FIELD4      10A  B  8  7CHECK(VN)
00110A      MDTA        78A  P
00120A      MDTA1       4A   P
A
```

图 3-91 规定 ERRSFL 键字

在这个例子中，当从显示器读 RCD1 时，任何先前错误子文件中的信息被清除。然后，如果 FIELD3 没有 1000 至 9999 范围的值，并且 FIELD4 没有有效的名字时，则系统把信息 MSG0002 和与 CHECK(VN) 有关的系统信息放到错误子文件中并且在显示器的第 24 行显示错误子文件。用户可按下 Page Up 和 Page Down 键浏览出错信息。

当再次从显示器读 RCD1 时，先前在错误子文件中的信息被清除。然后，如果 FIELD3 和 FIELD4 有效，则控制转给应用程序。如果 FIELD1 和 FIELD2 无效，当应用程序在指示器 10 和 11 为 ON 时，把 RCD1 写到显示上，系统则把信息 MSG0001 和 ERROR MSG1 放在出错子文件中并且在显示器的第 24 行显示出错子文件。用户可按下 Page Up 和 Page Down 键来浏览出错信息。

### 3.3.52 FLDCSRPRG (光标前进字段)

用这个字段层键字定义字段，使这个字段出现时光标移到本字段。如果工作站不是连在支持增强数据流的控制器上，则 FLDCSRPRG 键字无效。

此键字的格式为：

FLDCSRPRG (字段名)

这个键字必须有一个参数。

RPG 中的字段名要为输入字段，不能是子文件。

字段名是在同一记录格式中定义的输入字段名。

可选指示器对这个键字有效。

键字 FLDCSRPRG 不允许和 SNGCHCFLD 和 MLTCHCFLD 一起定义。

图 3-92 给出如何规定 FLDCSRPRG 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
R RECORD
    F1      10A  B   3  4FLDCSRPRG (F3)
    F2      10A  B   13 4FLDCSRPRG (F1)
    F3      10A  B   16 4FLDCSRPRG (F2)
```

图 3-92 规定 FLDCSRPRG 键字

在这个例子中，光标从字段 FIELD1 移到 FIELD3，然后从 FIELD3 移到 FIELD2，最后从 FIELD2 移到 FIELD1。

### 3.3.53 FLT FIXDEC (浮点到定点十进制)

用这个字段层键字可使输出浮点字段（用法 B 或 O）以定点十进制方式显示。

这个键字没有参数。

浮点数首先转换成带指数为零的相等数字。如果结果数字（数字和指数）不超出字段的长度和小数位，则取消此数字的指数部分并显示，对齐小数点位。如果数字超出字段的长度和小数位，则该数字以标准浮点格式 n.nnnnnnE+nnn 形式显示。当规定了 FLT FIXDEC 时，字段的显示长度为 DDS 的长度加 2（符号位和小数点），字段最小长度为 6。

如果浮点数字太大或太小以至于整个数字或小数部分不适于用 FLT FIXDEC 定义的定点形式，则用浮点形式以下方式表示有效数字。（有效数字是指数字字母 E 左侧包括正负号和小数点在内的数字串）：

全部有效十进制数位：DDS 全部数位减 5

小数部分有效数位：DDS 全部数位减 6

FIT FIXDEC 对数据的输入格式没有作用。数字输入时既可用定点格式也可用浮点格式。但当显示时，FLT FIXDEC 将用来决定显示格式。

可选指示器对这个键字无效。

图 3-93 给出如何规定 FLT FIXDEC 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECFMT1
A          FIELD1      10F 3B  1  2FLT FIXDEC
A
A          FLTPCN(*DOUBLE)
A
```

图 3-93 规定 FLT FIXDEC 键字

本例中的输出数字将被转换成如下形式：

输出数字	显示形式
-4.99994321000000E-004	, -4.9999E-004'
-5.00010000000000E-004	, -0.001'

```

-2.69123400000000E-002      ,      -0.027'
-0.00000000000000E+000      ,      0.000'
0.00000000000000E+000      ,      0.000'
2.71828182845900E++003     ,      2718.282'
3.14159000000000E-052     , 3.1416E-052'
9.87654321012345E+006     , 9876543.210'
9.9999999960000E+006     , 1.0000E+007'

```

### 3.3.54 FLTPCN (浮点精度)

用这个字段层键字可定义浮点字段的精度。

这个键字的格式是：

FLTPCN(\*SINGLE | \*DOUBLE)

有效的参数是\*SINGLE (单精度) 和\*DOUBLE (双精度)。这个键字仅对浮点字段 (F) 有效。单精度字段可以精确到 9 位，双精度字段可以精确到 17 位。如果定义的字段长度多于 9 位 (单精度) 或 17 位 (双精度)，则会产生出错信息，且文件不能生成。

可选指示器对这个键字无效。

图 3-94 给出如何规定 FLTPCN 键字

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00090A          FIELD A        17F 4   1  5FLTPCN(*DOUBLE)
A

```

图 3-94 规定 FLTPCN 键字

FLELDA 是双精度浮点字段。

### 3.3.55 FRCDTAL (强制数据)

用这个记录层键字可以无须等待下一个输入/输出操作而立即显示一个记录格式。当缓冲区部分满时，键字 FRCDTA 可以用来清除缓冲区。

注：如在每个写语句之后都用此键字，将出现性能问题。

此键字没有参数。

当键字对记录格式有效时，记录格式如同在建立显示文件 (CRTDSPF) 命令或修改显示文件 (CHGDSPF) 命令中规定了 DFRWRT(\*NO)一样显示。

当 DFRWRT(\*YES)有效并且程序在作输入操作之前要作几个输出操作时，可以使用这个键字。当定义了 DFRWRT(\*YES)时，直到有输入操作时才显示记录格式。在程序完成这个处理过程时，工作站用户可能要等很长时间。这时可以为最先显示的记录格式定义 FRCDTA。这个记录格式告诉工作站用户，等待是正常的。对于一步接一步的说明，请参看下面的例子。

FRCDTA 为每个记录格式仅可定义一次。

可选指示器对这个键字有效。

图 3-95 给出如何规定 FRCDTA 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A*下面的记录格式在程序信息中显示
00020A          R INPROG           LOCK

```

```

00030A          FRCDTA
00040A          12 21' Please wait; +
00050A          operations in progress'
    A
00060A* 下面的记录格式使用屏幕的上部分
00070A      R RCD1          OVERLAY
00080A          1 34' Sample Title'
00090A      FLD1      8  0  3  2
00100A      FLD2      20   4  2
    A
00110A*下面的记录格式使用屏幕的中部
00120A      R RCD2          OVERLAY
00120A      FLDA      8   11  2
00140A      FLDB      18   12  2
    A
00150A*下面的记录格式使用屏幕的下部
00160A      R RCD3          OVERLAY
00170A      FLDC      8   B 15  2
00180A      FLDD      8   B 16  2
    A

```

图 3-95 规定 FRCDTA 键字

用三个记录格式 (RCD1、RCD2 和 RCD3) 来建立一个显示。每一个记录格式都是显示的一部分。当准备这三个记录格式时，记录格式 INPROG 为用户对等待作准备。程序作如下操作：

显示记录格式 INPROG，由于规定了 FRCDTA，这使正在处理的信息立即显示。由于规定 LOCK 键字，键盘锁住。

继续处理来准备其它三个记录格式 (RCD1、RCD2 和 RCD3)，然后显示它们。由于它们覆盖记录格式 INPROG，使之被清除。

读记录格式 RCD3。解锁键盘并使工作站用户响应完整的显示。

### 3.3.56 GETRETAIN (保留)

用这个记录层键字和键字 UNLOCK 一起定义，使 OS/400 程序不删除用 UNLOCK 说明的输入字段。

本键字没有参数。

当连续的记录包含多处相同的输入时，用 GETRETAIN 可减少输入的次数。当规定 GETRETAIN 时，工作站用户只需改变某些输入字段而无需每次都键入整个记录。

当使用 GETRETAIN 时必须定义不带任何参数的 UNLOCK 键字。定义 UNLOCK(\*MDTOFF) 提供和 GETRETAIN 一样的功能。请参看 3.3.159 的“UNLOCK”。

可选指示器对这个键字无效。

图 3-96 给出如何规定 GETRETAIN 键字。

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

00101A	R REC1	GETRETAIN
A		UNLOCK
A		

图 3-96 规定 GETRETAIN 键字

### 3.3.57 **HELP** (帮助)

用这个文件层或记录层键字使帮助键可用。

若想了解如何在 System/36 环境下定义 HELP 键字, 请参看附录 F。

这个键字的格式为:

**HELP[ (应答指示器[ ‘说明’ ]) ]**

如果定义了应答指示器, 当按下 HELP 键时, 应答指示器为 ON 并返回用户程序。没有从设备传来的输入数据, 处理过程和使用命令注意键类似。

可选的说明包含在程序编译时产生的编译清单中, 它解释所用的指示器。这个说明在文件或程序中只有注释的作用。撇号(‘)是必需的。如果在撇号之间有多于 50 个字符, 则在程序清单中只保留前 50 个字符。

当按下帮助键时会产生如下动作:

- 如果光标在信息子文件位置或定义了ERRMSG 或ERRMSGID 键字的字段, 将显示信息的二级文本。
- 联机帮助信息与整个显示或部分显示有关。这个功能用 H 定义 (参见 3.3.58 “HLPARA”) 或使用文件层键字 HLPRCD、HLPPNLGRP 或 HLPDOC 来定义。
- 控制转交给用户程序。当文件中没有 H 定义和文件层键字 HLPDOC, HLPPNLGRP 或 HLPRCD 时, 发生此种情况。

如果没有定义此键字时按下了帮助键, OS/400 程序给出错误信息, 指明此时帮助键无效。

HLPRTN 键字允许使用可选指示器, 用来选择什么时候显示帮助信息和什么时候把控制交给用户程序。详细内容请参看 3.3.68 的“HLPRTN”。当在 HELP 键字中定义了应答指示器, 则在文件中不必定义 H, 也可规定键字 HLPRCD、HLPPNLGRP、HLPDOC 或 HLPRTN。

如果文件中包含 H 规范或键字 HLPRCD、HLPPNLGRP、HLPDOC 或 HLPRTN, 则必须规定无应答指示器的 HELP 键字。

可选指示器对这个键字有效。

图 3-97 给出了如何规定 HELP 键字。

00024A	R RECORD1	HELP
A		
A		

图 3-97 规定 HELP 键字

### 3.3.58 **HLPARA** (帮助区)

用这个帮助层键字可以在显示中定义一个矩形区域。当按下帮助键时, 如果光标在这个矩形区域中, 则显示 H(帮助)规范所指定的联机帮助信息。(使用了 HLPDOC, HLPPNLGRP 或 HLPRCD 键字)。

这个键字可使用下列格式:

**HLPARA** (顶行, 左列, 底行, 右列)

或 **HLPARA(\*RCD)**

或 **HLPARA(\*NONE)**

或 **HLPARA(\*FLD 字段名 [选项号])**

或 **HLPARA(\*CNST 帮助标识)**

当定义行和列参数时请参照如下规则:

- 行和列值必须在显示尺寸之内。
- 如果对辅显示尺寸没有定义 HLPARA，则可用主显示尺寸定义的 HLPARA（前提是这个 HLPARA 对辅显示尺寸也有效）。如果为主显示尺寸定义的 HLPARA 对辅显示尺寸无效，则使用 HLPARA(\*NONE)。
- 顶行的值不能大于底行的值，左列不能大于右列的值。
- 如果在记录中定义了键字 SLNO(n)，则要调整顶行和底行的值，并且诊断出在建立时间任何错误。如果在记录中定义了键字 SLNO(\*VAR)，则顶行和底行在处理时调整。

值\*RCD 指明帮助区是包含 H 定义的记录区。这个区域包括了被该记录占用的每一行的全部显示位置。

**HLPARA(\*RCD)** 对子文件控制 (SFLCTL) 或用户定义 (USRDFN) 记录格式无效。

如果在 H 规范上定义了 **HLPARA(\*RCD)**，则包含 H 规范的记录格式必须至少包含一个主显示尺寸的字段。隐藏字段 (38 列为 H)，信息字段 (38 列为 M) 和系统到程序字段 (38 列为 P) 以及规定了 SFLPGMQ 和 SFLMSGKEY 的字段是不显示的。

特殊值\*NONE 指明没有和 H 规范定义的帮助信息有关的帮助区。如果帮助信息是用 UIM(HLPPNLGRP 键字) 定义的，则在帮助键按下时它不能做为指定项帮助来显示，但可以做扩充帮助来显示。如果帮助信息是用 DDS (HLPDOC 键字) 定义的，则当按下帮助键时它不能做主帮助显示，但可以做辅助帮助显示，这时要在另外的帮助显示中用 Page Up 或 Page Down 键来做。当帮助信息是在文本中定义时 (HLPDOC 键字)，\*NONE 值无用，因为此时即使按下帮助键也不会显示帮助信息。

特殊值\*FLD 指出帮助区就是字段区。如果字段只占一行，则帮助区包括这行的第一个字符到最后一个字符中间的所有字符。

如果一个字段不只占用了一行，则帮助区包括字段中的所有行的全部长度。例如，一个字段从第 3 行第 4 列开始，在第 5 行第 10 列结束，则帮助区为从第 3 行的第 1 列到第 5 行的第 80 列区域。如果一个字段是选择字段或续入字段，则帮助区是由选择字段或续入字段占用的矩形区域组成。

字段名参数指出定义帮助区的字段名。字段必须在有 H 规范的记录中。

如果定义了选项号参数，帮助区是定义字段的选项区域。当指定一个选项号后，字段名必须是菜单条字段名或选项字段，所定义的选项号必须也在那个字段的 MNUBARCHC 或 CHOICE 键字中规定。选项号的有效值是大于 0 小于 100 的正整数。

\*CNST 指出帮助区是常数字段区。这个区域包括字段的开头和结尾的属性字节。

帮助标识参数是一个数字，它用来标识定义帮助区的常数字段。常数字段必须存在有 H 规范的记录中，必须有用同一个帮助标识定义的 HLPID 键字。

在 H 规范中必须至少定义一个 HLPARA 键字。当为每个 H 规范指定多个 HLPARA 键字时，必须使用显示尺条件方式。

当为一个记录定义了多外 H 规范，帮助区可以发生重叠。当定义了多个 H 规范时，使用带有下列两个特点的第一个 H 定义：

用 HLPARA 键字规定帮助区包含当前光标位置。

键字 HLPRCD、HLPPNLGRP、HLPDOC 上的可选指示器在应用记录写往显示器时有效。

下列规则适用于 H 规范：

在 17 列的 H 是 H 规范开始的标志。在 DDS 中，H 规范必须在记录层键字之后，在这个记录的第一个字段之前。

每个 H 规范必须正好有一个 HLPRCD、HLPPNLGRP 或 HLPDOC 键字，有一个 HLPBDY 或 HLPEXCLD 键字，还至少有一个 HLPARA 键字。

H 规范结束的标志是另一个 17 列的 H 或第一个字段。

不能在子文件 (SFL 键字) 记录格式中使用 H 规范。在与信息子文件 (SFLMSGRCRD 键字) 有关的子文件控制格式中不允许使用 H 规范。

可选指示器对这个键字无效。

图 3-98 和图 3-99 给出如何规定 HLPARA 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               HELP
A                               HLPRCD (DFTHELP)
A           R RECORD1
A           H                   HLPARA (1 5 3 15)
A                               HLPDOC (FLDHELP DOC1 FOLDER1)
A
A           H                   HLPARA (*RCD)
A                               HLPRCD (HELPRCD1)
A
A           H                   HLPARA (*NONE)
A                               HLPRCD (HELPRCD2)
A           FIELD1      10A    2   5
A           FIELD2      40A    10  10
A
```

图 3-98 规定 HLPARA 键字 (例 1)

在此例中，第一个 H 规范中的 HLPARA 键字指明联机帮助信息文本 DOC1 使用从第 1 行第 5 列到第 3 行第 15 列的区域，当帮助键按下时如果光标在上述区域内，则文本 DOC1 在标号 FLDHELP 处开始显示。

第二个 H 规范中的 HLPARA 键字指明记录 RECORD1 占有的区域(从 1 第 2 行到第 10 行)和联机帮助信息记录 HELPRCD 相关。当按下帮助键时如果光标在第 2 行到第 10 行的任何位置 (除在第一个 H 规范中定义的区域之外)，则显示 HELPRCD1。

第三个 H 规范中的 HLPARA 键字指明没有区域和联机帮助信息 HELPRCD2 相关。HELPRCD 只有在显示联机帮助信息时，按 Page Up 和 Page Down 键才能显示。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               HELP HLPRCD (DFTHELP)
A           R RECORD
A           H                   HLPARA (*FLD F1 1)
```

```

A          H          HLPRCD (UNDOHLP HLPLIB/HLPFILE)
A          H          HLPARA (*FLD F1 2)
A          H          HLPRCD (MARKHLP HLPLIB/HLPFILE)
A          H          HLPARA (*FLD F1 3)
A          H          HLPRCD (COPYHLP HLPLIB/HLPFILE)
A          H          HLPARA (*FLD F2)
A          H          HLPRCD (F2HLP HLPLIB/HLPFILE)
A          H          HLPARA (*CNST 1)
A          H          HLPRCD (TITLEHLP HLPLIB/HLPFILE)
A          1 37' Title' HLPID(1)

A          F1         2Y 0B 10 2SNGCHCFLD
A          CHOICE(1 'Choice 1')
A 01      CHOICE(2 'Choice 2')
A          CHOICE(3 'Choice 3')
A          F2         10A B 10 30
A

```

图 3-99 规定 HLPARA 键字（例 2）

前三个 H 规范中的 HLPARA 键字指明了被 Choice1, Choice2, Choice3 占有的区域分别和联机帮助信息相关。如果没选选项 2，则选项 3 上移一行，选择了的帮助区自动地随之移动。

第四个 H 规范中的 HLPARA 键字指明 F2 区域与联机帮助信息相关。这个区域为第 10 行的第 29 列到第 40 列。

第五个 H 规范的 HLPARA 键字指明常量标题区和联机帮助信息相关。这个区域在第 1 行的第 36 列到第 42 列。

### 3.3.59 HLPBDY (帮助边界)

当显示联机帮助信息时用这个帮助层键字限制联机帮助信息的可用性。

这个键字没有参数。

如果没有定义 HLPBDY 键字，与活动的 H 规范（所有显示记录累加）有关的联机帮助信息都可被用户访问。定义 HLPBDY 键字可以通过定义边界把表分成子表。每个子表包含帮助边界之间定义的 H 规范。有 HLPBDY 键字的 H 规范要在边界之前考虑。如果用 DDS (HLPRCD 键字) 定义联机帮助信息，当按下帮助键时用户只能访问包含所选用 H 规范的子表中的帮助信息。如果用 UIM (HLPPNLGRP 键字) 定义帮助信息，则子表决定扩充帮助。扩充帮助包括文件层键字 HLPPNLGRP，之后是包含为指定项帮助选择的 H 规范的子表。

可选指示器对这个键字有效。

图 3-100 给出如何规定 HLPBDY 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          HELP
A          HLPTCD (DFTHELP)

```

```

A          R RECORD1
A*
A* This is H-spec 1
A*
A          H           HLPARA(1 5 3 15)
A          HLPARD(HELPARD1)
A*
A* This is H-spec 2
A*
A          H           HLPARA(*NONE)
A          HLPARD(HELPARD2)
A          HLPBDY
A*
A* This is H-spec 3
A*
A          H           HLPARA(4 5 6 15)
A          HLPARD(HELPARD3)
A          HLPBDY
A*
A* This is H-spec 4
A*
A          H           HLPARA(8 5 10 15)
A          HLPARD(HELPARD4)
A          HLPBDY
A          FIELD1      10A     1 10
A

```

图 3-100 规定 HLPBDY 键字

RECORD1 的 H 规范的列表分成二个或三个子表，依据是帮助键按下时指示器 90 的状态。如果指示器 90 OFF，则分成两个子表，第一个子表包含 H 规范 1 和 2，第 2 个子表包含 H 规范 3 和 4。如果指示器 90 为 ON，则分成三个子表，第 1 个子表包含 H 规范 1 和 2，第 2 个子表包含 H 规范 3，第 3 个子表包含 H 规范 4。

### 3.3.60 HLPCLR (帮助清除)

用这个记录层键字清除活动的帮助规范的列表。当此记录显示时，只有定义在当前记录格式上或文件层上的联机帮助信息才是可访问的。

这个键字没有参数。

如果没规定 HLPCLR 键字，则所有记录的帮助规范被累积起来显示并保持活动，直到包含帮助规范的记录被从显示上清除或者完全被另一个记录覆盖为止。

可选指示器对这个键字有效。

定义 HLPCLR 键字的记录必须至少包含一个帮助规范。因为对一个有 USRDSPMGT 键字的复盖操作忽略帮助规范，使用有 USRDSPMGT 和 PUTOVR 键字的 HLPCLR 会导致没有联机帮助信息可显示。为了显示联机帮助信息，当 PUTOVR 起作用时要关闭 HLPCLR 的

可选指示器。

图 3-101 给出如何规定 HLPCLR 键字。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
A	R RECORD1	USRDFN
A		HLPCLR
A	H	HLPPARA(1 10 1 30)
A		HLPPRCD(RECORDA FILE1)
A	H	HLPPARA(4 10 4 30)
A		HLPPDOC(A B C)
A		

图 3-101 规定 HLPCLR 键字

当显示 RECORD1 时，只有与在 RECORD1 中定义的两个帮助规范有关的联机帮助信息才能显示。

### 3.3.61 HLPCMDKEY (帮助命令键)

使用这个记录层键字可在应用帮助记录格式中，当按下命令注意键 (CA) 或命令功能键 (CF) 之后把控制转给应用程序，这个键字是在应用帮助记录格式中规定的。为了使控制权能返回，必须在应用记录格式和应用帮助记录格式中都定义命令键。

这个键字没有参数。

在有 HLPCMDKEY 键字的帮助记录中或在文件层上必须定义一个 CA 或 CF 键。如果没有定义 CAnn 或 CFnn 键，则会给出警告信息（错误级别为 10）。如果在文件层规定的所有的 CAnn 和 CFnn 键及同一帮助记录的 HLPCMDKEY 有可选指示器，也发出 10 级警告信息。如果在应用帮助记录格式上为 CA 或 CF 键定义了应答指示器，则会给出警告信息（错误级别 10）并且应答指示器会被忽略。

如果在帮助记录上定义了这个键字，当工作站用户按下一个既在应用记录上定义也在应用帮助记录上定义的命令键，会发生如下情况：

如果命令键是 CAnn 键，则没有数据从应用记录格式传给应用程序。

如果命令键是 CFnn 键，则数据从应用记录格式传给应用程序。

命令键返回应用程序。

命令键必须既在应用记录格式上定义，也在应用帮助记录格式上定义。如果只在应用记录格式上定义，则命令键不能在显示帮助记录格式时使用。如果只在应用帮助记录格式上定义，则命令键只会起键的作用，并且控制不能返回给程序。

键字 HLPCMDKEY 不能在子文件上定义 (SFL 键字)，也不能在子文件控制 (SFLCTL 键字) 或用户定义 (USRDFN) 记录格式上定义。

在有键字 USRDSPMGT 的文件中不能规定 HLPCMDKEY 键字。

可选指示器对这个键字无效。

图 3-102 给出如何规定 HLPCMDKEY 键字。第一个记录是应用记录格式，第二个记录是应用帮助记录格式。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
00010A	R APPRCD	CA01

00020A		CA03
00030A		CF12(12)
00040A		CA04
00050A		HELP
00060A	H	HLPRCD (HELPRCD)
00070A		HLPARA (1 1 24 80)
00080A		8 2' THIS IS THE APPLICATION'
00090A		9 2' RECORD FORMAT'
00100A	INPUT1	10 B 12 10
00110A	INPUT2	10 B 13 10
00120A	INPUT3	10 B 14 20
00130A*		
00140A	R HELPRCD	HLPCMDKEY
00150A		CA01(11)
00160A		CF03
00170A		CF05
00180A		CF12
00190A		5 8' SPECIFY COMPANY NAME'
00200A		6 9' SPECIFY STREET'
00210A		7 10' SPECIFY CITY, STATE, ZIP'
	A	

图 3-102 规定 HLPCMDKEY 键字

如果用户是在应用帮助屏幕，会发生如下情况：

如果按下 CMD1 键，控制会返给应用程序，但没有数据从应用记录格式传送到应用程序。因为应答指示器 11 是在帮助记录格式上定义的而不是在应用记录格式上，所以这个应答指示器为 OFF。

如果按下 CMD12 键，控制返给应用程序，数据从应用记录格式传送到应用程序。应答指示器 12 为 ON。

如果按下 CMD5 键，功能和按下 ENTER 键一样。为了把控制返给程序必须既在应用帮助记录格式上，也在应用记录格式上定义相同的 CA 或 CF 键。如果按下 CMD3 键，功能类似于 ENTER 键，为了把控制返给应用程序，必须在应用帮助记录格式和应用记录格式中都规定相应的 CA 和 CF 键。

如果按下了 CMD4 键，会产生指明 CMD4 键不允许使用的信息。为了使控制能够返回应用程序，必须既在应用记录格式上定义也在应用帮助记录格式上定义相同的 CA 或 CF 键。

### 3.3.62 HLPDOC (帮助文本)

用这个文件层或帮助层键字定义一个文本，用它作显示器上特定位置的帮助信息内容。

这个键字的格式是：

HLPDOC (联机帮助信息正文标号名 文本名 文件夹名)

联机帮助信息正文标号参数相对于一个联机帮助文本标号，为文本的起始显示位置做个标记。

文本名参数指出包含联机帮助信息的联机文本。

文件夹名参数指出包含文本的文件夹。因为一个文件夹能存在于另一个文件夹中，并且给出的任何文件夹或文本名，在它所在的文件夹中都是唯一的，可能需要把几个文件夹名结合起来以标识一个文本/文件夹。在 HLPDOC 键字上定义的文件夹名可能只是个简单的文件夹名，它遵循和文本名一样的语法规则，或者可把文件夹名规定成一个联合的名字。

参见 1.2 “语法规则” 以了解文本、文件夹和标号的命名规定。

如果以下两个条件成立，则显示在 H 规范层上 HLPDOC 键字中规定的文本内容：

光标位于 H 规范的帮助区（用 HLPARA 定义的）

H 规范是活动的（H 规范层的 HLPDOC 键字的可选指示器决定 H 规范是否为活动。）

当包含当前光标位置的活动记录没有帮助区时，显示在文件层上用 HLPDOC 键字定义的文本。

HLPDOC 不能和 HLPBDY, HLPNLGRP 或 HLPRTN 一起定义。

可选指示器对这个键字有效。

图 3-103 给出如何规定 HLPDOC 键字

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8	
A	HELP
A	HLPDOC(START GENERAL.HLP HELP.F1)
A R REC1	OVERLAY
A H	HLPDOC(LBL1 HELP#1 HELP.F1)
A	HLPARA(10 3 12 50)
A H	HLPARA(15 9 17 61)
A 90	HLPDOC(LBL2 HELP#2 HELP/FLD)
A H	HLPARA(15 9 17 61)
A N90	HLPDOC(LBL3 HELP#3 HELP.F1/FLD)
A	

图 3-103 规定 HLPDOC 键字

由 HELP 键字使 HELP 键能够使用。

在文件层上的 HLPDOC 键字指明，如果按下帮助键并且光标不在 H 规范层上 HLPARA 键字定义的帮助区中，则在 HELP.F1 文件夹中的文本 GENERAL.HLP 从帮助符号 START 处开始显示。

在 H 规范层：

第一个 H 规范指明，如果按下帮助键并且光标在第 10 行、第 11 行或第 12 行的 3 到 50 列之间，文件夹 HELP.F1 中的文本 HELP#1 从 LBL1 帮助标号处开始显示。

第二个 H 规范指明，如记录 REC1 的指示器 90 为 ON，按下了帮助键，并且光标在第 15、16 或 17 行的 9 到 61 列之间，则在文件夹 HELP/FLD 中的文本 HELP#2 从帮助标号 LBL2 处开始显示。

第三个 H 规范指明，如果记录 REC1 的指示器 90 为 OFF，按下帮助键，并且光标在第 15、16 或 17 行的 9 到 61 列之间，那么文件夹 HELP.F1/FLD 中的文本 HELP#3 将从帮助标号 LBL3 处开始显示。

### 3.3.63 HLPEXCLD (禁止帮助)

用这个帮助层键字指明和这个帮助规范有关的联机帮助信息不能作为扩充帮助显示，但

是可以作为指定项的帮助。

这个键字没有参数。

如果没有定义这个键字，扩充帮助由与文件层的以及所有活动的帮助规范的 HLPPNLGRP 键字有关的联机帮助信息组成。

这个键字只允许在定义了 HLPPNLGRP 键字的 H 规范上使用。在 HLPPNLGRP 键字中每个参数至少有一个没有定义 HLPEXCLD 键字。如果所有定义了特殊帮助屏组名的帮助规范都被禁止，则在运行时，如果按下了帮助键并且光标位于和此帮助屏组名有关的帮助区中，则会给出错误信息。

可选指示器对这个键字有效。

图 3-104 给出如何规定 HLPEXCLD 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECORD1
A
A          H           HLPARA(1   12   8   14)
A           HLPPNLGRP(R1   PNLA)
A          H           HLPARA(1   18   8   19)
A           HLPPNLGRP(R2   PNLA)
A          H           HLPARA(1   35   8   37)
A           HLPPNLGRP(R1   PNLA)
A           HLPEXCLD
A          H           HLPARA(1   49   8   50)
A           HLPPNLGRP(R2   PNLA)
A           HLPEXCLD
A
```

图 3-104 规定 HLPEXCLD 键字

在图 3-104 中，HLPEXCLD 键字可阻止帮助模块 R1 和 R2 作为扩充帮助显示两次。

### 3.3.64 HLPFULL (全屏帮助)

用这个文件层键字指出应用程序的帮助屏组帮助正文用整屏的方式显示，而不是用窗口显示。

这个键字没有参数。

如果没有规定这个键字，联机帮助信息只在一个窗口中显示，除非为用户配置文件定义了\*HLPFULL 选项。

当定义 HLPFULL 键字时，必须在文件层或在帮助层上定义 HLPPNLGRP 键字。

可选指示器对这个键字无效。

图 3-105 给出如何定义 HLPFULL 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          HELP
A           HLPPNLGRP(GENERAL LIBA/PNL1)
A           HLPFULL
```

```

A          HLPTITLE(' Sample Screen')
A          R REC001H
A          H          HLPARA(4 10 4 29)
A          HLPPNLGRP(NAMETAG LIBA/PNL1)
A          1 10' Sample Screen'
A          NAME1      20A B 2 10
A

```

图 3-105 规定 HLPFULL 键字

当按下帮助键并且光标位于第 4 行的第 10 列到第 29 列之间，则 LIBA/PNL1 的帮助模块 NAMETAG1 用全屏方式显示。如果在光标位于其它位置时按下帮助键，则 LIBA/PNL1 的帮助模块 GENERAL 以全屏方式显示。

### 3.3.65 HLPID (帮助标识)

用这个常数字段层键字为字段层帮助中的常数定义一个标识符。所定义的标识符可用在 HLPARA 键字上来联结帮助正文和这个常数字段。

这个键字的格式是：

HLPID (帮助标识)

帮助标识参数是必需的并且只能是 1 到 999 之间的一个数字。这个数字在你所定义的记录中必须是唯一的。

可选指示器对这个键字无效。

图 3-106 给出如何规定 HLPID 键字

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECORD           HELP
A          H          HLPARA(*CNST 1)
A          HLPRCD(HLPCNST1 LIB1/FILE1)
A          H          HLPARA(*CNST 2)
A          HLPRCD(HLPCNST2 LIB1/FILE1)
A          2 4' Constant field 1' HLPID(1)
A          4 4' Constant field 2' HLPID(2)
A

```

图 3-106 规定 HLPID 键字

在这个例子中，如果光标位于显示中正文“Constan field 1”中并按下了帮助键，库 LIB1 的文件 FILE1 中的记录 HLPCNST1 作为帮助正文显示。如果光标位于“Consten field 2”中且按下了帮助键，则库 LIB1 的文件 FILE1 中的记录 HLPCNST2 作为帮助正文显示。

### 3.3.66 HLPPNLGRP (帮助屏组)

用这个文件层或帮助层键字定义当按下帮助键时要显示的 UIM 定义的联机帮助的信息源。

这个键字的格式是：

### HLPPNLGRP (帮助模块名 [库名/]屏组名)

帮助模块名为 1 至 32 个字符，第一个字符的有效值为字母 A 到 Z，之后的字符的有效值为字母 A 到 Z，数字 0 到 9，和 (/) 和 (\_)，如果有一个 (/) 或 (\_) 字符，则帮助模块名必须用撇号 (') 括起来。

屏组名定义了包含帮助模块的 UIM 屏组目标。当显示文件建立时它不是必需存在，如果没规定库名，则用\*LIBL 来查找屏组目标。

每个帮助规范必须包括 HLPRCD，HLPDOC 或 HLPPNLGRP。但一个显示文件不能同时包含 HLPPNLGRP 和 HLPRCD 键字，也不能同时包含 HLPPNLGRP 和 HLPDOC 键字。

如果在文件层上定义了 HLPPNLGRP 键字，也必须在文件层上定义 HELP 键字。如果文件中没有帮助规范，则必须在文件层上规定 HLPTITLE 键字。

如果在帮助层上定义了 HLPPNLGRP 键字，则必须在文件层或在当前记录上定义 HELP 和 HLPTITLE 键字。

可选指示器对这个键字有效。

图 3-107 给出如何规定 HLPPNLGRP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               HELP
A                               HLPPNLGRP(GENERAL LIBA/PANEL1)
A                               HLPTITLE(' Sample Screen')
A                               HLPSCHIDX(QHSS1)
A           R REC001
A           H                   HLPARA(4 10 4 29)
A                               HLPPNLGRP(NAMETAG LIBA/PANEL1)
A           H                   HLPARA(5 10 5 19)
A                               HLPPNLGRP(OPTION1TAG PANEL2)
A                               1 10' Sample Screen'
A           NAME1      20A B 4 10
A           OPTION1    10A B 5 10
A
```

图 3-107 规定 HLPPNLGRP 键字

如果光标位于第 4 行的 10 到 29 列之间，当按下帮助键时，显示库 LIBA 中的 UIM 屏组 PANEL1 的帮助模块 NAMETAG。如果光标位于第 5 行的 10 到 19 列之间，则显示库列表中的 UIM 屏组 PANEL2 中的帮助模块 OPTION1TAG。如果在光标位于其它位置时按下帮助键，则显示库 LIBA 中的屏组 PANEL1 中的帮助模块 GENERAL。

#### 3.3.67 HLPRCD (帮助记录)

用这个文件层或帮助层键字来定义，在按下帮助键时将要显示的联机帮助信息的记录格式。

这个键字的格式为：

### HLPRCD (记录格式名 [[库名/]文件名])

记录格式可在定义的文件中或在 HLPRCD 规定的文件中。如果没定义文件名，则记录格式必须在定义的文件中。

文件名参数指出包含记录格式的文件。如果没有指定库名，在程序运行时使用当前库列表(\*LIBL)。

如果以下两个条件为真，则显示在 H 规范层 HLPRCD 键字上定义的记录：

光标位于 H 规范用 HLPARA 定义的帮助区。

H 规范是活动的。(H 规范层 HLPRCD 键字的可选指示器决定 H 规范是否活动)。

当包含当前光标位置的活动记录不在帮助区时，显示在文件层 HLPRCD 键字上规定的记录。

可选指示器对此键字有效。

图 3-108 给出如何规定 HLPRCD 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               HELP
A                               HLPRCD(DFTHHELP HELPFILE)
A       R RECORD1
A       H                   HLPARA(1 1 24 80)
A   99                         HLPRCD(ERRHELP)
A       H                   HLPARA(1 1 1 80)
A                               HLPRCD(HELPRCD1 HELPFILE)
A       FIELD1      10A      1 10
A
```

图 3-108 规定 HLPRCD 键字

当指示器 99 为 ON，按下帮助键时，显示在记录 ERRHELP(此记录必须在显示文件中)的联机帮助信息。如果指示器 99 为 OFF 且按下帮助键时，光标位于第一行，则显示 HELPFILE 中的记录 HELPRCD1。否则当按下帮助键时显示在文件层 HLPRCD 键字上定义的记录中的联机帮助信息。

### 3.3.68 HLPRTN (帮助返回)

用这个文件层或记录层键字可在按下帮助键时把控制返给应用程序。如果没定义 HLPRTN，则显示当前光标位置有关的联机帮助信息。(参见 3.3.58 的“HLPARA”)。

如何在 System/36 环境下定义 HLPRTN 键字，可参看附录 F。

HLPRTN 键字的格式为：

**HLPRTN (应答指示器['说明'])**

如果定义了应答指示器，则应答指示器设为 ON 并返给应用程序。此时没有从设备传来的输入数据。处理过程类似于命令注意键。

可选的说明用来解释指示器的扩充用途，它在程序编译产生的打印输出中。这个说明在文件或程序中只有注释的作用。撇号是必需的。如果在撇号之间的字符多于 50 个，则多出的字符在打印输出中被减掉。

HLPRTN 在文件层或记录层中有比 HLPPNLGRP 或 HLPDOC 高的优先级。文件中的 HLPRTN 键字要在任何其它生效的帮助键字之前处理。

当规定了 HLPRTN，控制可能返给也可能不返给应用程序，这取决于是否使用可选指示器：

如果定义的 HLPRTN 不带指示器，当按下帮助键时，控制返给应用程序。如果在有 H

规范文件或记录中定义了不带指示器的 HLPRTN，则生成时会产生警告信息。

如果定义的 HLPRTN 带有指示器，在记录显示时如果指示器为 ON，控制返给应用程序。如果指示器为 OFF，则使用 H 规范。

可选指示器对 HLPRTN 键字有效。

图 3-109、图 3-110 说明如何使用 HLPRTN 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               HELP
A                               HLPRCD(GENERAL)
A     R RECORD1
A   02                         HLPRTN
A     H                           HLPARA(1 1 3 10)
A                               HLPRCD(HELPREC1)
A     FIELD1        10A B 2 2
A     R RECORD2                 HLPRTN
A     FIELDA       5A B 10 7
A
```

图 3-109 规定 HLPRTN 键字（例 1）

当显示 RECORD1 时若指示器 02 为 ON，则在按下帮助键时控制会返给用户程序。若指示器 02 是 OFF，则在按下帮助键时，显示联机帮助信息记录 HELPREC1 或 GENERAL，具体显示哪一个取决于当时光标的位置。当显示 RECORD2 时，按下帮助键控制会返给到用户程序。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               HELP
A   01                         HLPRTN
A                               HLPRCD(GENERAL)
A     R RECORD1
A     H                           HLPARA(1 1 3 10)
A                               HLPRCD(HELPREC1)
A     FIELD1        10A B 2 2
A     R RECORD2
A     FIELDA       5A B 10 7
A
```

图 3-110 规定 HLPRTN 键字（例 2）

如果可选择指示器 01 为 ON，当按下帮助键时，控制会直接返给用户程序而不考虑光标的位置。

### 3.3.69 HLPSCHIDX (帮助查询索引)

用这个文件层键字使索引查找功能（帮助显示上的 F11）能够使用，并且定义用于索引

查找的查询索引目标。

这个关键字的格式是：

**HLPSCHIDX** ([库名/]查询索引目标)

用 **CRTSCHIDX** 命令生成的查询索引目标包含在按下功能键启动索引查询功能时能使用的数据。

如果没有定义库名，则使用\*LIBL，当显示文件创建时查询索引目标不是必须存在的。

只有在文件中至少定义了一个 HLPPNLGRP 键字时，**HLPSCHIDX** 才是有效的。

**HLPSCHIDX** 键字不能和 **HLPSHELP** 键字一起定义。

可选指示器对这个关键字无效。

图 3-111 说明如何规定 **HLPSCHIDX** 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               HELP
A                               HLPTITLE(' Sample Screen')
A                               HLPPNLGRP(GENERAL LIBA/PANEL1)
A                               HLPSCHIDX(LIBA/SEARCH1)
A           R REC001H
A           H                               HLPARA(4 10 4 29)
A                               HLPPNLGRP(NAMETAG LIBA/PNL1)
A           1 10' Sample Screen'
A           NAME1      20A B 4 10
A
```

图 3-111 规定 **HLPSCHIDX** 键字

### 3.3.70 HLPSEQ (帮助顺序)

用这个记录层关键字来定义用 Page 键处理的正文记录的顺序。

这个关键字的格式为：

**HLPSEQ** (组名 序号)

组名是 1 到 10 个字符，用于联接在帮助文件的主帮助格式和辅助帮助格式。在联机帮助信息显示时按下 Page Up 或 Page Down 键，可显示在有同一帮助组名的帮助文件中做为联机帮助信息显示的当前记录格式。

序号是 0 到 99 之间的数字值，用来给帮助组内的记录格式排序。这个次序决定用作辅助联机帮助信息的记录格式的顺序。

一个帮助组中不允许存在相同的序号。

没有 **HLPSEQ** 键字的帮助记录格式被当作组中唯一的记录。

不能在子文件 (SFL 键字) 或用户定义 (USRDFN 键字) 记录格式上使用 **HLPSEQ**。

可选指示器对这个关键字有效。

图 3-112 说明如何规定 **HLPSEQ** 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A           R RECORD1                  HLPSEQ(HGROUP1 10)
A                           5 1' Help text ...
A
```

图 3-112 规定 HLPSEQ 键字

RECORD1 在帮助组 HGROUP1 中，它的顺序号为 10。

### 3.3.71 GLPSHELF (帮助书架)

用这个文件层键字使信息查询功能(帮助显示上的 F11)显示书架列表或者某一个书架。这个键字的格式是：

**HLPSHELF** (书架 | \*LIST)

在用户书架中规定要显示的一个书架名或者定义\*LIST 来显示书架列表。书架名可以为 8 个字符长。

只有在文件中至少定义了一个 HLPPNLGRP 键字时 HLPSHELF 才有效。

HLPSHELF 不能和 HLPSCHIDX 一起定义。

可选指示器对这个键字无效。

图 3-113 说明如何规定 HLPSHELF 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                      HELP
A                      HLPTITLE(' Sample Screen')
A                      HLPPNLGRP(GENERAL LIBA/PANEL1)
A                      HLPSHELF(BKSHFNM)
A          R REC001H
A          H                      HLPPARA(4 10 4 29)
A                      HLPPNLGRP(NAMETAG LIBA/PNL1)
A          1 10' Sample Screen'
A          NAME1      20A B 4 10
A
```

图 3-113 规定 HLPSHELP 键字

在此例中，在帮助（在屏组中定义）上按 F11 键时，显示书架 BKSHFNM 的内容。

### 3.3.72 **HLPTITLE** (帮助标题)

用这个文件层或记录层键字为全屏方式显示的屏组联机帮助信息定义一个缺省的标题。这个标题应是在按下帮助键时所显示的名字。只有在全屏帮助显示，且是在帮助原文中没有指定帮助标题时才使用这个键字。

这个键字的格式是：

**HLPTITLE** ('正文')

正文最长可为 55 个字符。

如果在文件中规定了 HLPTITLE 键字，则文件必须在文件层或在帮助层至少定义一个 HLPPNLGRP 键字。

如果在文件层上定义了 HLPPNLGRP 键字但在文件中没有帮助规定，此时需要在文件层上定义 HLPTITLE 键字。

如果没有在文件层上定义 HLPTITLE 键字，则在包含帮助规范的每个记录上都要定义

一个 HLPTITLE 键字。在没有帮助规范的记录中 HLPTITLE 键字无效。

可选指示器对文件层的 HLPTITLE 键字无效，但是可以在记录层的 HLPTITLE 键字上使用，此时如果记录中包含多个 HLPTITLE 键字，就要为每个键字定义可选指示器。在为每个键字都定义可选指示器的情况下，一个记录中最多可有 15 个 HLPTITLE 键字。在运行时，使用第一个有效的 HLPTITLE 键字。如果全部 HLPTITLE 的可选指示器都关闭，则会给出错误信息。

图 3-114 说明如何规定 HLPTITLE 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               HELP
A                               HLPPNLGRP(GENERAL LIBA/PANEL1)
A      R REC001
A 90                         HLPTITLE(' Sample Screen 1')
A N90                        HLPTITLE(' Sample Screen 2')
A      H                         HLPARA(4 10 4 29)
A                               HLPPNLGRP(NAMETAG LIBA/PANEL1)
A      H                         HLPARA(6 10 6 19)
A 10                          HLPPNLGRP(OPTION2TAG PANEL2)
A 90                         1 10' Sample Screen 1'
A N90                        1 10' Sample Screen 2'
A      NAME1        20A B 4 10
A 10             OPTION2       10A B 6 10
A
```

图 3-114 规定 HLPTITLE 键字

记录中有两个标题，所以在记录级上定义 HLPTITLE 键字，并且使用了指示器。指示器 90 的状态决定了哪一个标题既在应用显示时显示又在联机帮助信息显示时显示。使用一个指示器就保证了有一个 HLPTITLE 键字是有效的。

### 3.3.73 HOME (Home 键)

用这个文件层或记录层键字定义通过应用程序识别和处理 Home 键。

如果想了解如何在 System/36 环境的文件中定义 HOME 键字，请参看附录 F。

这个键字的格式为：

HOME [ (应答指示器[ ‘说明’ ]) ]

不管是否定义了 HOME 键字，当光标不在起点按下了 Home 键时，光标会返回起点处。

当按下 Home 键时如果光标已在起点处，OS/400 系统会像按下命令注意键时一样（这时没有数据从设备输入）把控制返给应用程序。如果没定义 HOME 键字，OS/400 程序显示信息指明此时 Home 键无效。

起点是下列情况之一（以优先级排序）：

最后一个输出操作规定的光标位置；

第一个不被保护的输入字段；

第一行第一个位置。

‘说明’参数是可选的，它在程序编译时产生的说明指示器用途的打印输出中。它在文

件或程序中只起到注释的作用。撇号是必需的。‘说明’在程序打印输出中只保留前 50 个字符。

可选指示器对这个键字有效。

图 3-115 说明在按下 Home 键时，如果光标在起始点如何定义 HOME 键字才能使在应答指示器 95 为 ON 时，控制返给程序。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00030A                                HOME(95 'Home key')  
A
```

图 3-115 规定 HOME 键字

### 3.3.74 HTML (超级文本标志语言)

这个字段层键字用在与 5250 数据流一起送出的有超级文本标志语言的非命名字段。如果数据流是送往 AS/400 的 5250 入口设备，HTML 标志将在 HTML 浏览器上处理。如果数据流不是送往 AS/400 的 5250 入口设备，则忽略 HTML 键字。

键字的格式为：

HTML ('值')

或 HTML (&程序—系统字段)

HTML 键字的参数是必须的。它必须是在单引号之内的有效 HTML 标志或是一个程序到系统字段。程序到系统字段要是合法长度且必须是字母数字的 (35 列为 A)，在编译 DDS 时不检查 HTML 参数的语法，接收 HTML 的浏览器在运行时检查语法是否正确。

下列键字不许和 HTML 一起规定：

COLOR	EDTCDE	OVRATR	USER
DATE	EDTWRD	PUTRETAIN	NOCCSID
DFT	HLPID	SYSNAME	
DSPATR	MSGCON		

可选指示器对此键字无效，但在这个常数字段可用指示器。

HTML 键字不允许在子文件记录的字段中使用。

图 3-116 说明如何规定 HTML 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
  
A  
A      R RECORD  
A          7 20HTML('<TITLE>')  
A          7 20HTML(&TAG)  
A          7 21HTML('</TITLE>')  
A      TAG      20A  P
```

图 3-116 规定 HTML 键字

HTML 是一个标志语言，标志的次序决定了它们的处理的顺序。在 HTML 文本中的行和列没有意义。这时行和列决定 HTML 标志送往浏览器的顺序，如果包括在 HTML 的常数

字段有相同的行和列，将按在 DDS 中出现的顺序处理。如何解决 HTML 字段的重叠现象，请看应用显示程序设计的第六章。如果在 CRTDSPF 中的 ENHDSP（增强显示）为\*NO，将忽略 HTML 键字，这就使用户可以不必重编译来将 HTML 键字关闭。

### 3.3.75 INDARA（指示器区）

用这个文件层键字可从缓冲区（也叫记录区）中移出可选指示器和应答指示器并把它们放入一个 99 字节的单独的指示器区。

这个键字没有参数。

定义 INDARA 键字有以下好处：

- 当可选指示器和应答指示器都使用时可简化 COBOL 编程。如果同一个指示器既用作应答指示器又用作可选指示器，则两个指示器的值总是一样的，这与在 DDS 中定义它们的顺序无关。
- 帮助 RPGIII 程序员使用程序描述的 WORKSTN 文件。

如果定义了 INDARA，可在 DDS 中增加修改或删除可选指示器和应答指示器，这时只要重新生成文件而无需重新编译高级语言程序。这是因为缓冲区中的字段位置没有改变，这样级别检查数据没有改变。不过，如果程序想使用新的指示器，此时程序仍是需要改写和重新编译的。

如果要定义 INDARA，有一些高级语言可能需要在应用程序中定义所用的独立的指示器区。

如果定义了 INDARA，在应用程序发出 I/O 操作命令时，LOGINP 和 LOGOUT 键字不记录应答指示器和可选指示器。这是因为应答指示器和可选指示器不在输入或输出缓冲区中出现。

可选指示器对这个键字无效。

图 3-117 说明如何规定 INDARA 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                         INDARA  
00020A                         CF01(01 'End program')  
00030A      R PROMPT  
00040A      ACTNBR      10   B  2  2  
00050A  41                      ERRMSG('Account number +  
00060A                           not found' 41)  
A
```

图 3-117 规定 INDARA 键字

此例中定义了 INDARA；从记录格式 PROMPT 的缓冲区中移走选择指示器 41 和应答指示器 01 和 41，并把这几个指示器放入另外的指示器区。只有名为 ACTNBR 的输入/输出字段仍保留在记录格式 PROMPT 的缓冲区。

### 3.3.76 INDTXT (指示器说明)

用这个文件层、记录层或字段层键字把指明目的或用途的说明和具体的应答或可选指示器联系起来。可为每个应答和可选指示器定义一次 INDTXT。

这个键字的格式是：

### INDTXT (指示器‘指示器说明’)

如果规定了这个键字，指示器说明是一个必需的参数值，这个说明出现在程序编译时的打印输出中，用来解释指示器的用途。这个文本在文件或程序中仅有注释的功能。撇号是必需的。如果在两个撇号之间定义的字符超过 50 个，则在程序打印输出中只保留前 50 个字符。

可选指示器对这个键字无效。

注：INDTXT 本身并不能使指定的指示器出现在输出或输入记录区，它仅仅提供一个和指示器有关的说明。如果指示器没有在其它地方（如可选指示器或应答指示器）定义，则说明丢失时没有任何提示。同样，一旦一个指示器有了说明（是用 INDTXT 键字定义或者是应答指示器说明），则不能再为这个指示器规定其它说明。

图 3-118 说明如何使用 INDTXT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               INDTXT(02 'Alternate month')
00020A          R MASTER
00030A          MTH           2 10
00040A  02      ALTMTH        2 10
A
```

图 3-118 规定 INDTXT 键字

INDTXT 说明了指示器 02 的用途。在高级语言的编译打印输出中，“Alternate month”将做为注释与指示器 02 一起打印。

### 3.3.77 INVITE (申请)

用这个文件层或记录层键字为稍后的读操作申请设备。为把一个申请送到指定的设备，应用程序要把一个有效的 INVITE 键字的输出操作发往设备。如果所用的记录格式有输出字段，则数据在申请到设备之前就发往设备。

这个键字没有参数。

如果显示文件可以有多个申请设备并且应用程序确实要从申请的设备操作中读取数据，则必须使用 INVITE 键字。这是因为从申请设备到一个多设备显示文件的读操作只是从所申请到的一个设备返回记录。如果你想所有被申请的设备都能返回数据，则必须从被申请设备进行读操作之前，往每一个设备发送一个有效的 INVITE 键字的输出操作。即使多设备文件中只申请到一个设备，此设备在进行读操作之前也必须用 INVITE 来申请。

INVITE 也可使你能生成一个能对读操作作出应答的可得到设备的子集。举例来说，如果文件当前可得到十台设备但只有三台设备被申请，则申请设备的下一个读操作只从三台设备之一返回一个记录。即使其它设备也有可用数据也是这样。

INVITE 能对性能有一些改进。通常在程序送出一个输入操作时给设备发出一个读请求。但 INVITE 允许在发出输出操作命令时请求读数据。在输出操作结束之后，程序可进行其它处理，此时设备正发送数据，OS/400 程序在处理这些数据。如果设备是远程设备，那么这样做可能是一个性能的明显的改善。

关于申请操作所用的指令，以及如何在程序申请设备来做读操作，请参看相应的高级语言手册。

INVITE 不能在文件层和记录层上同时规定，并且不能和子文件键字 (SFL) 一起使用。可选指示器对这个键字有效。

### 3.3.77.1 特殊考虑

使用 INVITE 键字要特殊考虑如下问题：

发往具体设备的输入操作不需要申请。带有指定记录格式名或设备的输入操作直接送到一台设备上。如果该设备在输入操作时有一个没完成的申请，则在完成输入操作后申请被删除。

一旦一个申请被送往设备，唯一的有效操作（除对被申请设备进行读操作外）如下：

- 对具体设备的输入操作。
- 带有试图取消申请数据的输出操作。如果取消成功，则数据可写进。如果 INVITE 在输出操作时有效，则重新申请设备。如果取消不成功（因为数据已被系统接收），则输出操作失败。此时程序必须执行一个输入操作来处理数据。这个输入操作取消对此设备的申请。

当从申请设备操作往显示文件中读数据时，只考虑从未完成申请的设备来的数据。输入操作等待从任何被申请设备中来的数据。（参见 CRTDSPF 和 CHGDSPP 命令中的 WAITRCD 参数）。如果在等待时间结束之前没有任何申请设备的响应，则发出一个提示信息且不返回任何数据。所有被申请设备保持被申请状态。

如果显示文件可得到的多个设备中不只一个有没完成的申请，则从被申请设备的读操作从被申请设备中返回下一个可用的记录。在这个输入操作之前如果从多个设备上收到记录，则其它的记录被保留用于以后的输入操作。

当从被申请设备到显示文件的一次读操作从被申请设备向程序返回一个记录时，则对此设备的申请被删除。其它带有未完成申请的设备继续被申请。当在稍后对被申请设备进行读操作时，如果想从这个设备接收数据则必须对之重新进行申请。

如果没有设备申请或者虽然申请了设备但是作业被控制选项取消，则从被申请设备到显示文件的一个读操作会产生一个提示信息，且不会有数据返给程序。所有被申请设备继续被申请。

如果想申请一台设备却没有数据给此设备，则要发出一个输出操作，此操作使用包含有效的 INVITE 键字的无可输出属性字段的记录格式。

在从被申请显示设备接收第一个记录后，设备不必重新申请，直到显示的所有输入字段的记录格式读到用户程序为止。如果在读操作中定义了记录格式名和设备名，则程序也能读那些其它的记录格式。

如果显示文件在 CRTDSPF 和 CHGDSPP 命令中规定了延迟写选项(DFRWRT(\*YES))则规定有效的 INVITE 键字的输出操作会导致在设备被申请之前延迟显示输出。

图 3-119 说明如何规定 INVITE 键字。

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8	
00010A 01	INVITE
00020A R RCD1	
00030A FLD1 10 2 2	
00030A FLD2 5 2 24	
A	

图 3-119 规定 INVITE 键字

只有在指示器 01 为 ON，INVITE 才有效。

### 3.3.78 INZINP (初始输入)

用这个记录层键字初始化输出/输入字段，在 PUTOVR 和 ERASEINP(\*ALL)都有效时不用送初始化数据来显示。

这个键字没有参数。

INZINP 对处理远程工作站的数据入口的应用程序特别有用。

下面是如何使用这个键字：

1.一个输出操作显示一个 OVRDTA 或 OVRATR 有效的输出/输入字段。系统初始化输入保存区成字段的程序值。例如，如果程序把 NAME1 设为 Bob，则输入保存区包含名字 Bob。

2.一个输入操作把数据从工作站放到输入缓冲区。如果工作站用户没有键入输入字段并且 DSPATR (MDT) 键字对字段无效，则字段不能从工作站返回数据。系统从输入保存区取得数据并把数据放入程序使用的输入缓冲区。这样，所有输入字段在输入缓冲区中都有数据。这些数据或是从工作站接收来的，或是从输入保存区接收来的。

3.在另一个输出操作中会产生如下情况：

如果 INZINP 和 OVRDTA 没有生效，则输入保存区不会改变，即使程序改变了字段值也是如此。举例来说，如果程序把字段值改为名字 Tom，程序保存区中的值仍然是工作站用户输入的，或是先前的程序值 Bob。

如果 INZINP 有效，输出保存区将存有当前程序值。当前程序值被送往字段的显示，条件是 OVRDTA 键字对这个字段有效。

如果 OVRDTA 键字对一个输出操作无效，程序必须清除规定了 OVRDTA 字段的输出缓冲区来保证输入保存区和显示器上的字段（在输出操作之后都是空白符）相匹配。

注：如果 ERASEINP(\*ALL)键字对输出操作有效，则可输入字段从显示设备上清除，但是输入保存区没有清除。有关输入保存区的内容，请参看图 3-120 和图 3-121。有 DFT 规定的字段如没有被选中做显示时用 DFT 规定的值初始化。这些值一直被保留，除非应用程序选择这些字段来显示，才会修改这些数据值。使用这个键字时要在记录层上定义 PUTOVR、OVERLAY 和 ERASEINP(\*ALL)。

下表给出有和没有 INZINP 键字时，键字 ERASEINP(\*ALL)和 PUTOVR 产生的影响。

图 3-120 输入/输出字段的 INZINP

OVRATR 或 OVRDTA 键字	INZINP 键字	输入保存区内容
没规定	不提供	前面的内容
规定但无效	规定但无效	前面的内容
规定但无效	规定有效	程序值（不送往显示）
OVRATR 规定且有效	不提供	程序值（送往显示）
OVRATR 规定且有效， OVRDTA 没规定或无效	不提供	前面的值（不送往显示）

图 3-121 仅输入字段的 INZINP

OVRATR 键字	INZINP 键字	输入保护区的内容
没规定	不提供	前面的内容
规定且有效	不提供	前面的内容
规定但无效	不提供	字符字段：空格

		数字字段：零
--	--	--------

依下列步骤把输入保存区置为空格或零以匹配在工作站上被 ERASEINP(\*ALL)清除的字段：

1. 为 INZINP, ERASEINP(\*ALL), PUTOVR 和 OVERLAY 定义相同的可选指示器。
  2. 为所有输出/输入字段定义 OVRDTA 或 OVRATR。(如果不想往设备传送数据或属性, 可把这些关键字的指示器关闭。如果让 OVRATR 有效, 也要让 OVRDTA 有效)。
  3. 为所有只输入字段规定 OVRATR 键字 (如果不往设备传送属性, 可把 OVRATR 键字的选择指示器关闭)。
  4. 在输出操作前, 把所有输出/输入字段都置成空格 (字符型字段) 或零 (数字型字段)。
- 如果在记录中规定了 DSPMOD 又规定了 INZINP, 则生成文件时会产生警告信息。在运行时, 当显示方式改变时, 忽略 INZINP。

图 3-122 给出如何规定 INZINP 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R FMT1
00020A  77          PUTOVR OVERLAY ERASEINP (*ALL)
00030A  77          INZINP
00040A          7 8' CUSTOMER NUMBER'
00050A      CUSNBR    6   I  7 26
00060A N77          OVRATR
00070A          9 12' CUSTOMER NAME'
00080A      NAME     25   B  9 30
00090A N77          OVRATR
00100A          10 2' CUSTOMER ADDRESS LINE 1'
00110A      ADDR1     25   B 10 30
00120A N77          OVRATR
00130A          11 2' CUSTOMER ADDRESS LINE 2'
00140A      ADDR2     25   B 11 30
00150A N77          OVRATR
00160A          12 9' NEW CREDIT LIMIT'
00170A      LIMIT     4  0I 12 30
00180A N77          OVRATR
A

```

图 3-122 规定 INZINP 键字

这个例子解释如下：

1. 对第一个输出操作, 用户程序把指示器 77 关闭。这时 PUTOVR、ERASEINP 和 INZINP 都无效。结果是：字段 NAME、ADDR1 和 ADDR2 都送往工作站。

输入保存区中有：

CUSNBR	全为空格
NAME	程序值 (在显示器上显示)
ADDR1	程序值 (在显示器上显示)

ADDR2 程序值（在显示器上显示）

LIMIT 全为零

2. 对第二个输出操作，用户程序打开指示器 77。结果如下：

所有接收输入字段从工作站上清除。

没有字段被送往工作站。

输入保存区中含有：

CUSNBR 都是空格

NAME 程序值（不显示）

ADPR1 程序值（不显示）

ADPR2 程序值（不显示）

LIMIT 都是零

注：如果在第二个输出操作前把 NAME1、ADDR1 和 ADDR2 都置成空格，则输入保存区中全为空格和零。

### 3.3.79 INZRCD (初始记录)

用这个记录层关键字定义如果记录还没有显示，则该记录在定义这个记录名的程序送出输入操作之前写往显示。如果这个记录已经显示，则忽略 INZRCD 关键字。这个隐含的输出操作是 OS/400 程序初始的，它的目的仅是在执行输入操作时使显示器格式化显示。

这个关键字没有参数。

这个关键字不能用于输出操作。

如果没定义 INZRCD 关键字，当程序试图读一个不在显示器上的记录时会收到一个错误信息。

在处理 INZRCD 关键字时，存在如下特殊情况：

对于仅输出字段，没有用户数据可用。字段在显示器上以空白符出现。任何规定的编辑方式都无效，BLKFOLD 关键字在显示上不起作用。

对输出/输入字段，没有用户数据可用，字段在显示器上以空白出现，输入保存区以和只输入字段一样的方式初始化（根据数据类型不同初始化成空格或零）。

常量和仅输入字段与用明显的输出操作一样的显示方式。

隐藏字段在输入操作时以空白符和零返回。

信息和系统程序字段被忽略。

因为没有输出缓冲区来作运行记录，LOGOUT 关键字被忽略。

因为没有已经显示的记录格式，ERRMSG 和 ERRMSGID 被忽略。

SFLMSG 和 SFLMSGID 也被忽略。

所有其它选择的关键字和字段都作处理。

注：为了执行 INZRCD 功能，在发出有此关键字规定的输入操作时，用户程序必须定义一个记录格式名。用于输入操作的记录格式必须定义 INZRCD 关键字。

可选指示器对这个关键字无效。

图 3-123 给出如何规定 INZRCD 关键字。

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

00274A R REC4 INZRCD

A

图 3-123 规定 INZRCD 关键字

### 3.3.80 KEEP (保持)

用这个记录层键字在显示文件关闭防止显示被删除。只要在显示中的一个记录定义了 KEEP，整个显示都被保留。文件关闭时缺省的是删除整个显示。

另外，在显示器上的第一个有 KEEP 属性的记录被 OS/400 程序保存以备随后的程序使用。随后程序的第一个输入操作中定义的记录名不是被保持的记录名，就可以使用这个记录名。这个键字可使在程序结束之后仍保持显示的数据，或把数据作为随后程序的输入来用。

这个键字没有参数。

这个键字不能和以下键字一起定义：

ALWROL

CLRL

SLNO

如果在定义了 DSPMOD 的记录上定义的 KEEP，则在文件建立时会发出警告信息。在运行时间，当显示方式改变时忽略 KEEP 键字。

可选指示器和应答指示器对这个键字无效。

图 3-124 给出如何规定 KEEP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00061A      R REC46      KEEP  
A
```

图 3-124 规定 KEEP 键字

### 3.3.81 LOCK (锁)

用这个记录层键字规定在输出操作完成后键盘保持锁住。通常键盘在输出操作之后自动解锁。当几个连续的输出操作都包含输入字段时，使用 LOCK 键字。

这个键字没有参数。

如果没有定义这个键字，当几个连续的输出操作把数据送往显示时，工作站用户可往字段中键入数据。在这种情况下光标位置可被改变，键入数据丢失。

注：输出操作后缺省是键盘解锁。如果在键盘锁住时发出一个输入操作，则键盘自动解锁。

这个键字与其它影响输出操作的键字互不影响。

可选指示器对这个键字无效。

图 3-125 给出如何规定 LOCK 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00011A      R REC1      LOCK  
A
```

图 3-125 规定 LOCK 键字

### 3.3.82 LOGINP (日志输入)

用这个记录层键字定义：每次 OS/400 程序对此记录执行输入操作时，这个记录格式的

输入缓冲都写进作业日志中。日志的数据包括记录格式定义的输入字段值，隐藏字段值和应答指示器。如果此记录格式是一个子文件记录格式，日志的数据也包括输出字段。（如果为文件定义了 INDARA 键字，则不为应答指示器作记录）。这个键字可用于调试或其它例外功能。用户程序不能读这个作业日志。

这个键字没有参数。

在下列条件下，OS/400 程序会忽略 LOGINP:

a)在记录格式中没有输入字段、隐藏字段和应答指示器。

b)记录格式是一个信息子文件的子文件记录格式。

可选指示器对这个键字无效。

图 3-126 给出如何规定 LOGINP 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00072A          R REC24           LOGINP  
A
```

图 3-126 规定 LOGINP 键字

### 3.3.83 LOGOUT (日志输出)

用这个记录层键字定义：每次 OS/400 程序执行对记录的输出操作时，此记录格式的输出缓冲区都被写到作业日志中。日志的数据包括输出字段、隐藏字段和指示器。（如果为文件定义了 INDARA，则不作可选指示器的日志）。LOGOUT 键字可用于调试和其它例外功能。用户程序不能读这个作业日志。

这个键字没有参数。

在下面情况下 OS/400 程序忽略 LOGOUT:

a)在记录格式没有可输出字段、隐藏字段和可选指示器。

b)记录格式是信息子文件的子文件记录格式。

可选指示器对这个键字无效。

图 3-127 给出了如何规定 LOGOUT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00021A          R REC25           LOGOUT  
A
```

图 3-127 规定 LOGOUT 键字

### 3.3.84 LOWER (小写)

LOWER 键字和 CHECK(LC)是相同的。建议使用 CHECK 键字。请参见 3.3.21 的内容。

### 3.3.85 MDTOFF (关闭修改数据标记)

用这个记录层键字和 OVERLAY 键字一起把正在显示的记录格式的输入字段修改数据标记 (MDT) 关闭。

这个键字的格式是：

MDTOFF[(\*UNPR | \*ALL)]

当程序送一个输出操作到定义的记录格式时，修改数据标记被关闭。

要只对非保护字段关闭修改数据标记（无有效的 DSPATR(PR)键字），可定义\*UNPR 参数（也是缺省值）。若想对所有输入字段关闭修改数据标记，可定义参数\*ALL。

程序可以选择字段 DSPATR(MDT)键字，也可为字段所在的记录格式选择 MDTOFF(任何参数)。如果这样，这些字段用修改数据标记(MDT)为 ON 显示。

除非定义了 MDTOFF(\*ALL)，否则 ERASEINP(\*ALL)将隐含 MDTOFF(\*UNPR)。

如果 ERASEINP(\*MDTON)键字是和 MDTOFF(\*ALL)一起定义的，最终结果和同时定义了 ERASEINP(\*ALL)、MDTOFF(\*ALL)一样。如果 ERASEINP 没定义参数，结果也同上面一样的。

可选指示器对这个键字有效。

MDTOFF 对子文件记录格式无效，对定义了 OVERLAY 的所有其它记录格式都有效。

图 3-128 给出如何规定 MDTOFF 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1          OVERLAY MDTOFF
00020A      FLD1      6   B   2   2
00030A      FLD2      6   B   3   2
00040A*
00050A      R RECORD2          OVERLAY MDTOFF (*UNPR)
00060A      FLD21     6   B   4   2
00070A      FLD22     6   B   5   2
00080A*
00090A      R RECORD3          OVERLAY MDTOFF (*ALL)
00100A      FLD31     6   B   6   2
00110A      FLD32     6   B   7   2
00120A      FLD33     6   B   8   2DSPATR(PR)
A
```

图 3-128 规定 MDTOFF 键字

RECORD1 和 RECORD2 有同样的 MDTOFF 键字定义。当 RECORD1 或 RECORD2 显示时，每个已经显示的输入字段的 MDT 被关闭，除非字段有生效的 DSPATR(PR)键字。(当 FLD3 显示时是这样的字段)。当 RECORD3 显示时，即使 DSPATR(PR)对字段生效，已经显示的输入字段的 MDT 也要关闭。

### 3.3.86 MLTCHCFLD (多选择选项字段)

用这个字段层键字定义一个字段作为多选项字段。多选项字段包含一个固定的选择数，用户可以选择多个选项。字段以选项的横排或竖排列表出现，每个选项的左边都有一个输入字段，或在一组检测框中。

如果看到每个选项左端是一个输入字段，检测框选择字符缺省值为“/”。信息 CPX5A0C 包括国家规定的选项字符。这个值可以修改，可以是大写或小写的国家规定选择字符。当显示文件建立时这些字符即被定义。

这个键字的格式是：

```
MLTCHCFLD[*RSTCSR | *NORSTCSR]
[*NOSLTIND | *SLTIND]
```

```

[[*NUMCOL  列数 |
  (*NUMROW  行数)]
  [(*GUTTER  间隔宽度) ]])

```

参数是可选的。如果没定义参数，多选项选择字段的选择项将竖行排列。允许用户用箭头键把选择光标移出这个字段。在选项和选择指示器之间在显示时会有三个空格。

**RSTCSR** 参数定义是否允许用箭头键把选择光标移出选择字段。**\*RSTCSR** 规定箭头键不能把选择光标移出字段，**\*NORSTCSR** 规定用箭头键可把光标从选择字段移出。缺省值为**\*NORSTCSR**。

注：如果选择字段是包含下拉窗口的唯一字段，则会发生由**\*RSTCSR** 引起的例外限制。在这种情况下，如果选择光标在最左或最右的列内，左箭头键和右箭头键将关闭当前下拉式窗口，并且打开新的下拉式窗口，新的窗口跟当前选项的左或右的菜单选择项有关。

**\*RSTCSR** 参数在不是与不可编程工作站的增强控制器相连的显示器上被忽略：

**SLTLND** 参数指出选择指示器（比如检测框）是否显示。**\*NOSLTIND** 规定不显示选择指示器。缺省值为**\*SLTLND**。

**\*NUMCOL** 规定选择字段用横排的多列显示，如下所示：

choice1	choice2	choice3
choice4	choice5	choice6
choice7	choice8	choice9

参数中的列数部分指定在选择字段中有多少列。列数是正整数值，并且整个多选项选择字段在显示中必须能全都放进去。

**\*NUMROW** 规定选择字段用竖排的多列显示。如下所示：

choice1	choice4	choice7
choice2	choice5	choice8
choice3	choice6	choice9

参数中的行数规定选择字段中包括多少行。它是一个正整数并且整个多选项字段在显示时要能全都放进去。

**GUTTER** 参数是可选的，它规定在多选项选择的每个列之间的空格数。只有定义了**\*NUMCOL** 或**\*NUMROW** 时才定义 **GUTTER**，并且后面要紧接写**\*NUMxxx #**参数。它必须是至少为 2 的正整数。如果没定义**\*GUTTER**，缺省为三个空格。

有 **MLTCHCFLD** 键字定义的字段必须有一个或多个 **CHOICE** 和 **CHCCTL** 键字来定义字段的选项。

有 **MLTCHCFLD** 键字的字段必须定义成一个输入字段，它的类型为 **Y**，长度为 2。此字段的位置是显示中第一个选择项或最上面一个检测框的最上面的位置。如果在 **PULLDOWN** 键字中使用**\*NOSLTIND**，并且设备是连接到支持不可编程工作站的增强型控制器上，则字段的位置是第一个选择说明的第一个字符。在输入操作中，字段包含选择的选项号码，如没有选择选项时，号码为 0。在输出操作中，字段值被忽略。

如下键字可在有 **MLTCHCFLD** 键字规定的字段上定义：

ALIAS	CHGINPDFT
AUTO(RA)	FLDCSRPRG
CHANGE	DSPATR(RI UL BL CS HI ND PC)(3)
CHCAVAIL	ERRMSG
CHCUNAVAIL	ERRMSGID
CHCSLT(1)	INDTXT
CHCCTL	OVRDATA

CHECK(ER,FE)(2)	OVRATR
CHOICE	PUTRETAIN
	TEXT

注：

- 1.当定义 PULLDWN(\*NOSLTIND)时，只有在不显示选择指示器的下拉菜单中显示多选项选择字段时，CHCSLT 才有用。
- 2.CHECK(FE)只能用于连到不支持增强型接口控制器上的显示器上。
- 3.如果为定义了 MLTCHCFLD 键字的字段定义 COLOR 或 DSPATTR 键字，则它们只能用在基于字符显示的选择字段的输入字段中。

可选指示器对这个键字无效。

图 3-129 给出如何规定 MLTCHCFLD 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A          R RECORD
A          F1           2Y 0B  3 35MLTCHCFLD
A  01           CHOICE(1 '>Undo      ')
A           CHOICE(2 &MARKTXT)
A           CHOICE(3 '>Copy      ')
A           CHCCTL(1 &CTLONE1 MSG1111 QUSER/A)
A           CHCCTL(2 &CTLTW01 &MSG1 &LIB/&MSGF)
A           CHCCTL(3 &CTLTHR1)
A          CTLONE1     1Y 0H
A          CTLTW01     1Y 0H
A          CTLTHR1     1Y 0H
A          MSGF         10A P
A          LIB          10A P
A          MARKTXT     10A P
A
```

图 3-129 规定 MLTCHCFLD 键字

用于 MLTCHCFLD 的每个 CHOICE 键字都需要定义 CHCCTL 字段。

在输入中，CHCCTL 键字的隐藏字段指出选项是否被选中。0 表示选项未被选中；而 1 则表示选项已被选中。在输出操作中，隐藏字段控制选项的可用性，也用于为选项设置缺省值。0 表示选项可用，1 表明选项用缺省值选中，2 表示选项不可用。其它值被截短。

### 3.3.87 MNUBAR (菜单条)

用这个记录层键字定义菜单条。一个菜单条是一些选项的水平列表，列表之后是些可选的行。这些选项代表一组用户可选择的有关动作。举例来说，当一个用户选择了一个菜单条后，会在下拉菜单中出现一组相关动作。一个菜单条记录包含一个字段，字段中有一个或多个 MNUBARCHC 键字用以定义菜单条选项，由系统产生分隔行。

这个键字的格式是：

MNUBAR ([\*SEPARATOR | \*NOSEPARATOR])

参数是可选的。用来定义是否在菜单条选项的最后一行之后放置分隔行，\*SEPARATOR 表示需要放置，而\*NOSEPARATOR 则表示不需要分隔行。缺省值是\*SEPARATOR。

注：如果定义了\*NOSEPARATOR，则不可在这个记录上定义 MNUBARSEP 键字。

如果一个记录规定了 MNUBAR 键字，则该记录必须有一个且仅能有一个菜单条字段一个或多个 MNUBARCH (键字的字段)，并且不能包含除菜单条之外的任何可显示字段。

以下键字允许在有 MNUBAR 的记录中使用：

CAnn	HLPTITLE	MNUCNL
CFnn	HOME	OVERLAY
CLEAR	INDTXT	PAGEDOWN/PAGEUP
CLRL	INVITE	PRINT
CSRLOC	KEEP	PROTECT
DSPMOD	LOCK	ROLLUP/ROLLDOWN
HELP	MNUBARDSP	TEXT
HPCLR	MNUBARSEP	UNLOCK
HPCMDKEY	MNUBARSW	VLDCMDKEY
	HPRTN	

注：如果菜单条是由系统显示的，则忽略以上键字（例如，MNUBARDSP 是在记录中规定而是不在菜单条记录上规定时）。

可选指示器对这个键字无效。

图 3-130 给出了如何规定 MNUBAR 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A      R RECORD1          MNUBAR
A      MNUFLD      2Y 0I  1  2
A                      MNUBARCHC(1 RCDFILE 'File')
A                      MNUBARCHC(2 RCDEDIT 'Edit')
A                      MNUBARCHC(3 RCDVIEW 'View')
A                      MNUBARCHC(4 RCDOPT 'Options')
A                      MNUBARCHC(5 RCDHELP 'Help')
A                      :
A                      :
A
```

图 3-130 规定 MNUBAR 键字

在此例中，RECORD1 被定义成一个菜单条记录。当 RECORD1 显示时，在字段 MNUFLD 上定义的菜单条选项作为菜单条显示。菜单条选项之后是分隔行。在基于字符的显示上，分隔行由蓝色的“/”组成。在和支持增强型接口的控制器相连的图形显示上，分隔行是一条线。

### 3.3.88 MNUBAR CHC (菜单条选项)

用这个记录层键字为一个菜单条字段定义一个选项。菜单条选项表示成一组相关动作，可被用户选择。当用户选中一个选项时，在下拉式菜单中出现一组动作。

这个键字的格式是：

## MNUBARCHC (选项号 下拉记录 选项正文[&返回字段])

选项号参数是必需的，它指定一个标识号。选项号返回给应用程序用以指出选中了哪个选项。选项号的有效值为 1 到 99 之间的整数。在一个菜单条字段中不允许存在两个同样的选项号。

下拉记录参数也是必需的，它指明在用户选中选项时被显示的下拉式记录的名字。这个记录必须存在于文件中并且必须有 PULLDOWN 键字。

选项正文参数是必需的，它用于定义选项的说明正文。这个参数可以用以下两种格式之一定义：

作为一个字符串：‘选项正文’

作一个系统到程序字段：&字段名

字段名必须在菜单条记录中，而且必须定义成一个用法为 P 的字符型字段。

选项正文必须能在最小显示尺寸中的一行显示。因为第一个菜单条选项的正文在第 3 列开始显示，并且在正文末总有一个空格，所以如果最小显示尺寸是  $24 \times 80$  则选项正文最长为 76，如果最小显示尺寸是  $27 \times 132$ ，则说明最长为 128。

当用字符串或系统到程序字段做选项正文显示时，尾部空格被删除且在与选项之间插入三个空格。菜单条在显示时所占行数由此菜单条的所有 MNUBARCHC 上的选择正文参数长度的总和决定，再加上每个选项的三个空格。选项正文的长度或者是字符串的长度，不算尾部空格数，或是程序到系统字段的长度。菜单条字段行最大为 12 行（包括分隔行）。

在选择正文中，可用大于号 (>) 为选项定义一个助记符，在 > 右侧的是助记符：例如

选项正文	显示结果
‘>File’	File
‘F>Finish’	Finish
‘Save>AS…’	Save AS…
‘X>=1’	X=1

在正文本中如有>符，则必须把>定义两次，这和你想定义撇号时也定义两次一样。

选项正文	显示结果
‘X>>=1’	X>=1
‘X>>>=1’	X>=1

注：不可把>定义成助记符。

所指定的助记符必须是单字节字符，而且不能是空白符。在选项正文中仅能用一个助记符。能为多个选项定义同一个助记符。

返回字段参数是可选的。它用来定义控制是否因为选中了选项而返回应用程序。当控制返回应用程序时，这个参数指定了在含有选中选项的菜单条记录中的隐藏字段名。隐藏字段被定义成类型为 Y 的字段，它的长度为 2，没有小数点，这个字段中的选项号码表示，因为菜单条选项被选中控制返回给了应用程序。而应用程序的下一个操作就更新（需要的话）或写与选中选项相关的下拉记录，即在 MNUBARCHC 键字中规定的下拉记录。当一个选项号返回字段时，在下拉式输入被接收之后，把 0 返回到有选项号的字段。这样，当下拉记录被接收，0 返回给这个字段，应用程序记录中的菜单条字段或选择字段的选项号指明应用程序应该处理下拉的输入。

在 MNUBARCHC 键字中的菜单条字段被定义成一个类型为 Y（数字型）的输入字段，字段的长度为 2，小数位是 0。在读菜单记录时，选中的选项号被返回菜单条字段。菜单条字段总被定义成从第一行第二列开始。

当在字段上定义了 MNUBARCHC 键字时，也需在记录层上定义 MNUBAR 键字。

一个菜单条字段可定义多个 MNUBARCHC 键字。可定义的 MNUBARCHC 键字的数量

只和选项正文参数的长度以及对 MNUBAR 的 12 行限制有关。为菜单条字段定义的所有选项必须能放进一屏中（每个与选项间有 3 个空格）。

下列键字可与 MNUBARCHC 键字在字段上一起定义：

ALIAS	INDTXT
CHCAVAIL	MNUBARSEP
CHCSLT	TEXT

可选指示器对这个键字有效。

图 3-131 和图 3-132，以及图 3-133 给出如何规定 MNUBARCHC 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               DSPSIZ(*DS3 *DS4)
A       R MENUBAR               MNUBAR
A       MNUFLD      2Y 0B  1  2
A                           MNUBARCHC(1 PULLFILE +
A                               ' >File      ')
A   01                      MNUBARCHC(2 PULLEDIT +
A                               ' >>Edit      ')
A                           MNUBARCHC(3 PULLVIEW +
A                               ' >View      ')
A                           MNUBARCHC(4 PULLOPT +
A                               ' >Options    ' &RTNFLD)
A                           MNUBARCHC(5 PULLHELP +
A                               ' >Help      ')
A       RTNFLD      2Y 0H
```

图 3-131 规定 MNUBARCHC 键字（例 1）

在图 3-131 中，在菜单条中定义了五个选项（File, Edit, View, Options 和 Help）。如果指示器 1 为 ON，并且菜单条记录在系统显示它之前写入，则当系统显示菜单条时显示 Edit 选项。如果指示器 01 为 OFF，或者在系统显示菜单条记录之前记录没有写入，则不显示 Edit 选项。如果 Edit 选项没有显示，则选项列表被压缩且 View 选项将跟在 File 选项之后显示，它们之间是三个空格。

如果 File 选项被选中，则记录 PULLFILE 被作为下拉菜单在 File 选项下显示。如果 Options 选项被选中，则控制返回应用程序。应用程序能在系统把它作为下拉式菜单显示之前更新 PULLOPT 记录。

在能为单个字符作下划线的显示上，每个选项的助记符是正文第一个字符。如果读菜单条记录，则菜单条字段 MNUFLD 包含选中选项的号，如没有选中选项时则为 0。

每个选项的正文都被定义成字符串，正文可用 15 个空格。但系统计算一行中有几个选项时要去掉结尾部空格，这样，菜单条所要求的最大空间是 87。（每个字符占 28 个位置，每个选项再加 3 个空格）。菜单条选项占一行，分隔行占一行。这样整个菜单条占 2 行。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
DSPSIZ(*DS3 *DS4)
```

```

A      R MENUBAR          MNUBAR
A      MNUFLD           2Y 0B  1  2
A                  MNUBARCHC(1 PULLFILE +
A                  &FILETXT)
A  01            MNUBARCHC(2 PULLEDIT +
A                  &EDITTXT)
A                  MNUBARCHC(3 PULLVIEW +
A                  &VIEWTXT)
A                  MNUBARCHC(4 PULLOPT +
A                  &OPTTXT &RTNFLD)
A                  MNUBARCHC(5 PULLHELP +
A                  &HELPTXT)

A      FILETXT          15A  P
A      EDITTXT          15A  P
A      VIEWTXT          15A  P
A      OPTTXT           15A  P
A      HELPTXT          15A  P
A      RTNFLD            2Y 0H
A

```

图 3-132 规定 MNUBARCHC 键字（例 2）

图 3-132 和图 3-131 基本上一样，唯一不同的是选项正文用系统到程序字段定义。

运行时，选项正文在显示时从系统到程序字段接收。正文中的助记符在运行时由应用程序提供。当读菜单条记录时，象图 3-131 一样，菜单条记录 MNUFLD 包含选中选项的号码，没有选中时为 0。

像图 3-131 一样，每个选项正文可用的长度为 15，菜单条需要 78 个位置。（5 个选项，每个选项占 15 个位置，外加与选项之间的 3 个空格）。因为最小显示尺寸是 24×80(\*DS3)，菜单条选项占两行。分隔行又占一行，所以整个菜单条占 3 行。实际上用于显示选项的行数依赖于包含在系统到程序字段的正文。当显示菜单条时，P 字段的尾部空格被去掉，在与选项之间加三个空白符。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A      R MENUBAR          MNUBAR
A      MNUFLD           2Y 0B  1  2
A                  MNUBARCHC(1 PULLFILE +
A                  ' >File      ')
A                  MNUBARCHC(2 PULLEDIT +
A                  ' >Edit      ' &RTNFLD)
A      RTNFLD            2Y 0H

```

图 3-133 规定 MNUBARCHC 键字（例 3）

在图 3-133 中，如果选中了菜单条的选项 2，则控制返给应用程序，字段 RTNFLD 包含

2. 字段包含 0 时，说明此时没有接收到下拉输入。应用程序必须读记录 MENU BAR，以得到字段 RTNFLD 的内容。然后应用程序写记录 PULLEDIT。系统恢复对菜单条交互功能的控制。如果输入进到记录 PULLEDIT 中，控制返给应用程序，字段 MNUFLD 包括 2。字段 RTNFLD 包括号码 0 时表示控制由于收到下拉输入而返回。

如果选中了选项 1，则系统显示下拉记录 PULLFILE，如果输入进到 FULLFILE，则控制返给应用程序，并且字段 MNUFLD 包含 0。如果字段 RTNFLD 包含 0，则说明下拉输入已收到控制返回。

### 3.3.89 MNUBAR DSP (菜单条显示)

用这个记录层键字显示菜单条。MNUBAR DSP 键字有两种格式：一种是用于包含 MNUBAR 键字的记录，另一种用于不包含 MNUBAR 键字的记录。

用于不是菜单条记录的 MNUBAR DS 的格式为：

MNUBAR DSP (菜单条记录 & 选项字段[&下拉输入])

用于菜单条记录的 MNUBAR DSP 的格式为：

MNUBAR DSP[ (&下拉输入) ]

菜单条记录参数指出写此记录时要显示的菜单条记录。菜单条记录和定义它的记录必须在同一文件中。

参数&选项字段指出隐藏字段名，此字段在包含选中选项号的输入中。此字段必须存在定义它的记录中，并且要定义为数字型 Y，用法为 H，长度 2，小数位为 0。

参数&下拉输入是可选的，用于定义特别的隐藏字段名，在下拉菜单只包含单选项字段时，该隐藏字段从下拉菜单得到输入。该字段必须存在于定义它的记录中，且长度为 2，小数位为 0，用法为 H 的区位字段（位置 35 为 S）。在输出中，该字段包含如下值之一：

值	含义
0	没有作出选择
n	下拉菜单的第 n 号选项被选中
-1	下拉记录包含除了一个单选字段之外的内容。必须读下拉记录来得到它的内容。

可选指示器对 MNUBAR DSP 键字有效。在一个记录上可以定义多个 MNUBAR DSP 键字，条件是所有这些键字都是可选的。当写一个记录时多个 MNUBAR DSP 键字同时生效，则只使用第一个。

图 3-134 给出如何在不是菜单条的记录上规定 MNUBAR DSP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R RECORD1
A 01          MNUBAR DSP (MENURCD &MNUCHOICE &INPUT)
A      FIELD1    10A  B 10 12
A      FIELD2    5S 0B 14 12
A      MNUCHOICE 2Y 0H
A      INPUT      2S 0H
A      R MENURCD          MNUBAR
A      F1        2Y 0B 1 2
A                      MNUBARHC(1 PULLFILE 'File')
A                      :
```

A :  
A

图 3-134 规定 MNUBAR DSP 键字（例 1）

在这个例子中，在记录 RECORD1 显示时如果指示器 01 为 ON，则系统显示在记录 MENURCD 中的菜单条。当菜单条被激活且下拉菜单被选择时，该菜单条的选项号返给字段 MNUCHOICE。如果所选的下拉菜单包含一个单选字段，对该字段所作的选择被返给字段 INPUT。否则，字段 INPUT 包含-1，指出应用程序必须读下拉记录以接收下拉输入。

图 3-135 给出如何在一个菜单记录上定义 MNUBAR DSP。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A           R MENURCD          MNUBAR  
A   01           MNUBAR DSP  
A           F1           2Y 0B  1  2  
A           MNUBARCHC(1 PULLFILE 'File')  
A           :  
A           :  
A           :
```

图 3-135 规定 MNUBAR DSP 键字

当显示记录 MENURCD 时，如果指示器 01 为 ON，则系统显示在 MENURCD 中定义的菜单条。如果用户从菜单条中选了下拉菜单，选中的菜单条的选项号返给字段 F1。

### 3.3.90 MNUBAR SEP (菜单条分隔符)

在菜单条字段上使用这个字段层键字，定义用于组成菜单条分隔符行的颜色，显示属性或字符。

这个键字的格式为：

MNUBAR SEP ([颜色][显示属性][字符])

必须至少定义一个参数。

颜色参数规定彩色工作站分隔符的颜色。这个参数以 (\*COLOR 值) 的表达式形式定义。

颜色参数的有效值为：

值	含义
BLU	蓝色
GRN	绿色
PNK	粉红色
RED	红色
TRQ	蓝绿色
YLW	黄色
WHT	白色

如果没有规定颜色参数，缺省为 BLU (蓝色)。如果这个参数是为单显中的菜单定义的，则被忽略。

显示属性参数规定分隔符的显示属性。该参数以如下格式指定：

(\*DSPATR 值 1 <值 2 <值 3…>>)。

显示属性的有效值为：

值	含义
BL	闪烁
CS	列分隔
HI	高亮度
ND	不显示
RI	反象显示
UL	下划线

缺省的显示属性是正常亮度（或低亮度）。

注：显示属性 CS、HI 和 BL 能使特定字段表现为彩色。这样的字段是在 5292、3179、3197 - C1、C2 以 3487HC、3488 (6) 工作站上的字段。当使用显示属性 HI、RI 和 UL 时不显示分隔符行线。如果进一步了解这方面的知识，请参看 3.3.30 中的“COLOR”。

字符参数指出用于组成分隔符行的字符。这个参数的定义格式为 (\*CHAR ‘分隔符’）。分隔符的值是一个字符。如果没定义这个参数，缺省的分隔符是破折号（—）或者在图形显示器上显示一条直线。尽管任何字符都可以定义成分隔符，但建议使用恒定字符。

图 3-136 列举了恒定字符

图 3-136

图 3-136 系统数据的字符集

注：另外选用下面的任何字符：

大写的英文字母：A 至 Z

数字字符：0 至 9

当在一个字段上定义 MNUBARSEP 时，必须在相关的记录上定义 MNUBAR 键字。如果定义了 MNUBARSEP，可以不在 MNUBAR 键字上使用参数\*NOSEPARATOR。

可选指示器对这个键字有效。

如果定义了多个 COLOR 键字，则颜色、分隔符和显示属性参数都取第一个 COLOR 键字规定的。

图 3-137 给出如何规定 MNUBARSEP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R MENU BAR          MNU BAR
A      MNUFLD        2Y 0B  1  2
A                  MNUBARSEP((*COLOR PNK) +
A                  (*DSPATR RI) (*CHAR   , ))
A                  MNUBARHC(1 PULLFILE +
A                  'File      ')
A                  MNUBARHC(2 PULLEDIT +
A                  'Edit      ')
A
```

图 3-137 规定 MNUBARSEP 键字

在此例中，菜单条分隔符由粉红色反相显示的空白符组成。

### 3.3.91 MNUBARSW (菜单条开关键)

用这个文件层或记录层键字分配一个 CA 键用作转向菜单条键。如果菜单条开关键是活动的且菜单条被显示，则按下 CA 键将完成以下功能：

如果光标位于应用记录上，按下开关键可把光标移到菜单条的第一个字段上，再按下开关键可把光标移到先前按下开关键把光标移进菜单条的位置。

如果光标已被用手动移到菜单条（例如使用光标移动键），则按下开关键可把光标移到应用记录的第一个输入字段。

这个键字的格式为：

MNUBARSW[ (CAnn) ]

参数 CAnn 是可选的。若没指定这个参数，缺省值是 CA10。CAnn 有效的参数值为 CA01 至 CA24。

在一个记录中，用 MNUBARSW 定义的 CAnn 不能用其它键字（始 MNUCNL）再次定义。因为文件层上的 MNUBARSW 对文件中的全部记录都有效，这在分配 CAnn 键时必须注意。

如果在记录层定义了 MNUBARSW 键字，则 CAnn 键或 CA10 键在其它记录上只能用

作 CA 键，而不能用作 CF 键。

MNUBARSW 键字只允许在有菜单条记录的文件中使用。

可选指示器对这个键字有效。

图 3-138 给出如何规定 MNUBARSW 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                                     MNUBARSW(CA10)
A     R MENUBAR                  MNUBAR
A     MNUFLD       2Y 0B  1  2
A                                     MNUBARCHC(1 PULLFILE +
A                               ' >File      ')
A   01          MNUBARCHC(2 PULLEDIT +
A                               ' >Edit      ')
A
A     R PULLEDIT                PULLDOWN
A     F1           1D 0B  1  2RANGE(1 3)
A                           1  5' 1. Undo      ,
A                           2  4' 2. Mark      ,
A                           3  4' 3. Copy      ,
A   :
A   :
A     R APPSCR                 MNUBARDSP(MENUBAR &MNUCHOICE)
A     FIELD1      10A  B 10 12
A     FIELD2      5S 0B 14 12
A                           24  1' F12=Cancel ,
A     MNUCHOICE    2S 0H
A
```

图 3-138 规定 MNUBARSW 键字

在这个例子中，CA10 定义成文件中所有记录的菜单开关键。当光标在菜单条外的任何位置上，按下 CA10 键时，光标都移到菜单条的 File 选项。这时，如果再次按下 CA10，无论此时光标在菜单条的任何位置，光标都要移回它先前在 APPSCR 记录中的位置上。

### 3.3.92 MNUCNL (菜单取消键)

用这个文件层或记录层级键字把一个 CA 键分配做菜单条或下拉菜单的取消键。当 MNUCNL 键字活动且下拉菜单显示时，按下 CA 键会取消下拉菜单并把光标返回到菜单条中的选项。如果没有下拉菜单显示且光标位于菜单条中时，按下 CA 键取消菜单条并把光标返回到应用屏幕。如果没有下拉菜单显示而且光标在应用屏幕时，按下 CA 键可把控制返给到应用程序。

这个键字的格式是：

MNUCNL[ (CAnn[应答指示器]) ]

参数 CAnn 是可选的。如果没定义这个参数，缺省值是 CA12。这个参数的有效值是 CA01 到 CA24。

应答指示器参数是可选的。应答指示器为 ON, MNUCNL 键字是在记录上活动而不是在菜单条或下拉菜单上活动，并且控制返给应用程序。

在一个记录中，由 MNUCNL 键字规定的 CAnn 键不能用其它键字（如 MNUBARSW）再次定义。因为文件层上的 MNULCNL 对文件中所有记录都有效，当分配 CAnn 键时必须注意这一点。

如果在记录中定义了 MNUCNL 键字，CAnn 键或 CA12 键在其它记录上只能用作 CA 键，而不能用作 CF 键。

MNUCNL 只允许在有菜单条记录的文件中使用。

可选指示器对这个键字有效。

图 3-139 给出如何规定 MNUCNL 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                                     MNUCNL(CA12 12)
A       R MENUBAR                  MNUBAR
A       MNUFLD        2Y 0B 1 2
A                                     MNUBARCHC(1 PULLFILE +
A                               ,>File      ')
A 01                           MNUBARCHC(2 PULLEDIT +
A                               ,>Edit      ')
A
A       R PULLEDIT                PULLDOWN
A       F1          1D 0B 1 2RANGE(1 3)
A           1 5' 1. Undo      ,
A           2 4' 2. Mark      ,
A           3 4' 3. Copy      ,
A       :
A       :
A       R APPSCR                 MNUBOARDSP(MENUBAR &MNUCHOICE)
A       FIELD1      10A B 10 12
A       FIELD2      5S 0B 14 12
A           24 1' F12=Cancel ,
A       MNUCHOICE    2S 0H
```

图 3-139 规定 MNUCNL 键字

在此例中，CA12 规定为文件所有记录的取消键。当下拉菜单 FULLEDIT 显示时，如果按下 CA12 键，则下拉菜单被取消。当光标位于菜单条时（此时没有下拉菜单显示）按下 CA12 键，则菜单条被取消，而且光标移回到应用记录。如果光标不在菜单条且没有下拉菜单显示时按下 CA12 键，则应答指示器 12 为 ON 且控制返回给应用程序。

### 3.3.93 MOUBTN (鼠标键)

用这个文件层或记录层键字把一个命令键或 EVENT-ID (事件指示符) 和一个或两个指针设备事件连结起来。

当一个指定的指针设备指出某一事件执行且没有其它更高优先级功能时，锁住键盘，光

标移到指针设备光标位置，指定的命令键或 EVENT-ID 被返回应用程序。如果命令键或 EVENT-ID 引起入口字段数据确认，则首先执行数据确认。如果指定的命令键 EVENT-ID 正常返回入口字段数据，则包括入站入口字段数据。

对指针设备双事件，直到也发生尾沿事件，入站数据才返回。当查到前沿事件时，输入一个可编程双事件状态，拉出一个标记框，放在指针设备光标周围，(字符周围的 4 条蓝线)，指针设备光标在彩色不可编程工作站 (NWS) 上被改成白色并且查找尾沿事件。击键和主数据流将会取消可编程双事件状态。当等待尾沿事件时，忽视一些指针设备事件。当收到尾沿事件时，删除标记框，指针设备光标的颜色变成输入禁止的，然后锁住键盘。正文光标被移到指针设备光标位置，并且入站数据被返回主机。

注：尽管允许，但仍不提倡对同一鼠标键和转换状态的指针设备事件的组合进行编程。例如，如果对鼠标右键进行了编程，则不该再对双击右键进行编程，因为可能由于单击右键事件而锁住键盘以至寻查不到双击事件。使用\*QUEUE 参数使应用程序能够处理这种情况。

这个关键字的格式是：

MOUBTN (事件[尾沿事件]{命令键 | EVENT-ID}{\*QUEUE | \*NOQUEUE})

事件参数是必需的，用来指明与命令键或 EVENT-ID 参数有关的指针设备事件。事件参数的有效值是：

值	含义
*ULP	按下未换档左键
*ULR	松开未换档左键
*ULD	双击未换档左键
*UMP	按下未换档中键
*UMR	松开未换档中键
*UMD	双击未换档中键
*URP	按下未换档右键
*URR	松开未换档右键
*URD	双击未换档右键
*SLP	按下换档左键
*SLR	松开换档左键
*SLD	双击换档左键
*SMP	按下换档中键
*SMR	松开换档中键
*SMD	双击换档中键
*SRP	按下换档右键
*SRR	松开换档右键
*SRD	双击换档右键

尾沿事件参数是可选的。如规定了这个参数可为一个双事件指针设备定义规定的尾沿事件。这个参数的有效值和事件参数的一样。一个尾沿事件可是多个前沿事件的尾沿事件，对每个都有不同的命令键或 EVENT-ID 自己特定的动作。一个事件既可为尾沿事件，也可为单一事件（与不同命令键或 EVENT-ID 有关）。

注：对事件定义有如下限制：

一个事件不可既是单一事件又是双事件序列的前沿事件。

与一个前沿事件有关的尾沿事件只能有一个。

如果与多鼠标键定义使用相同的事件作为单一事件或前沿事件，只使用第一个定义。命令键或 EVENT-ID 参数只能指定其中一个，并分配一个命令键或事件标准值与第一个参数

(如有第 2 个,也要考虑)指定的指针设备事件连系起来。命令键的有效值是 CA01 到 CA24, CF01 到 CF24, ENTER, ROLLUP, ROLLDOWN, HELP, HOME, PRINT 和 CLEAR。有效的事件标记是 E00 到 E15。EVENT-ID 和 CAxx 键类似,没有从设备传来的输入数据。

\*QUEUE 参数是可选的,用于定义在键盘锁住时收到的单一事件是否由控制器排队。这个特性主要用于允许定义双击的鼠标键,如果不能给双击排队,则应用程序将不能适时的知道是否发生双击,由于处理鼠标键的按下和松开键盘被锁住。这个参数的缺省值为 \*NOQUEUE。

当在 MOUBTN 键字上使用了下列命令键时不能定义下面的键字:

命令键	互斥的键字
CFxx	ALTHELP(CAyy),CAxx 这里 xx=yy
CAxx	ALTPAGEDWN(CFyy),ALTPAGEUP(CFyy),CFxx 这里 xx=yy
CF01	不带参数的 ALTHELP
CF07	不带参数的 ALTPAGEUP
CF08	不带参数的 ALTPAGEDWN

相关的功能键被定义与单击或双击鼠标事件有关的命令键,定义 CA01-CA24, CF01-CF24, ROLLUP, ROLLDOWN, PAGEUP, PAGEDOWN, CLEAR 和 HLPRTN 键字不是必须的,但却是有效的。把一个命令键的一个鼠标事件联系起来能自动使键盘上的相应命令键能够使用。如果想把一个应答指示器和功能键连系起来,则必须使用一个列出中的键字来完成。在这种情况下,不管命令键是从键盘还是从鼠标产生的,应答指示器都设为 ON。

可选指示器对这个键字有效。

图 3-140 给出如何规定 MOUBTN 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
      A          MOUBTN(*URP CF03)
      A          MOUBTN(*SRP CF12)
      A          CF12(12 'CANCEL')
      A          R RECORD1
      A          MOUBTN(*ULP *UMP ROLLUP)
      A          MOUBTN(*UMP *ULP ROLLDOWN)
      A          1 10' ONE--:'
      A          FIELD1    10A I 1 17TEXT(' ONE')
      A          2 10' TWO--:'
      A          FIELD2    10A I 2 17TEXT(' TWO')
      A
```

图 3-140 规定 MOUBTN 键字

在此例中,定义了两个可编程鼠标事件,它对文件中所有记录都有效(不包括在记录级以上越权的)。这些定义把未换档鼠标右键按下事件与 CF03 相关,把换档鼠标右键按下事件与 CF12 相关。CF03 没有应答指示器,CF12 有应答指示器。

在 RECORD1 中,定义了两个鼠标双击事件。第一个与把 ROLLUP 键与先按未换档左键再按未换档中键相连。第二个事件把 ROLLDOWN 键与先按未换档中键,后接未换档左

键相连。这些定义只有在 RECORD1 是写到显示的最后一个记录时才有效。

### 3.3.94 MSGALARM (信息报警)

这是一个文件层或记录层关键字。它可使系统在如下情况下发出警报声：当定义的记录和活动的 ERRMSG、EREMSGID、SFLMSG 或 SFLMSGID 键字一起显示时，或者发现有效性检查错误时。警报持续时间很短。

这个关键字没有参数。

如果对同一个记录定义了 MSGALARM 和 ALARM，而且两者同时起作用，警报只响一次。

可选指示器对这个关键字有效。

图 3-141 给出如何在记录层规定 MSGALARM 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RCD1  
00020A          MSGALARM  
00030A      FLD01      8A    12 10  
00040A  12          ERRMSGID(XYZ0123 MSGFILE)  
A
```

图 3-141 在记录层规定 MSGALARM 键字

当显示记录格式 RCD1 时，想要再次显示 RCD1 而且选择指示器 12 为 ON，则信息文件 MSGFILE 中的信息 XYZ0123 在信息行上显示，同时工作站有报警声。

图 3-142 表示如何在文件层定义 MSGALARM 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A  01          MSGALARM  
00020A      R RCD1  
00030A      FLD01      8A    12 10  
00040A  12          ERRMSGID(XYZ0123 MSGFILE)  
00050A  
00060A      R RCD2  
00070A      FLD02      8A    12 10  
00080A  10          ERRMSG('Message text')  
A
```

图 3-142 在文件层上规定 MSGALARM 键字

当记录格式 RCD1 已经显示，想要再次显示并且指示器 01 和 12 都为 ON，信息文件 MSGFILE 中的信息 XYZ0123 显示在信息行上，同时工作站有警报声。

当记录格式 RCD 已经显示，想要再次显示并且指示器 10 为 ON，而 01 为 OFF，则在信息行上显示正文，工作站不发出警报声。

### 3.3.95 MSGCON (信息常量)

用这个字段层键字指出常数字段的正文包含在信息描述之中。如果在 DDS 编译时间信息正文不存在，则文件不能生成。如果修改信息描述，想要使显示文件包含更新后的信息，则必须重新生成文件。

这个键字的格式是：

**MSGCON** (长度 信息标识[库名/]信息文件名)

长度参数定义信息描述的最大长度，可为 1 到 132 字节。如果信息描述的长度小于所定义的长度，则余下的字节由空白符填充。如果信息描述的长度大于所定义的长度，则多去的部分被截去，同时产生警告信息。

信息标识参数定义包含用作常数字段值的正文的信息描述。

信息文件名参数定义包含信息描述的信息文件。库名参数是可选的。

**MSGCON** 键字必须明确的为字段定义，此键字不能用来初始化一个已命名的字段。

**DFT** 和 **MSGCON** 具有相同的功能。如果对同一字段定义了 **DFT** 和 **MSGCON**，则 **MSGCON** 键字被忽略而且文件不能生成。

**MSGCON** 键字不能和以下键字一起定义：

**DATE**

**DFT**

**EDTCDE**

**EDTWRD**

**TIME**

可选指示器对改变信息行的值无效，但是可以调整信息在显示器上的存在情况。

图 3-143 表示如何规定 **MSGCON** 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A          R RECORD1  
00020A          2 1MSGCON(10 MSG0001 MESSAGE/MSGF)  
A
```

图 3-143 规定 **MSGCON** 键字

库 MESSAGES 中的信息文件 MSGF 中的 MSG0001 包含了信息正文。

### 3.3.96 **MSGID** (信息标识)

可用这个字段层键字在运行时间允许应用程序识别包含命名字段的文本的信息描述。这个键字的参数可定义为包含信息标识符、信息文件和库的字段。在程序把字段置成期望值后，输出操作引起信息从信息文件中取出并显示。如果取出的信息比 **MSGID** 字段长则被截短，如果比 **MSGID** 字段短则以空白符补充。

如想了解如何在 System/36 环境所用文件中定义 **MSGID** 键字，请参看附录 F。

这个键字的格式为：

**MSGID** (信息标识[库名/]信息文件)

或 **MSGID(\*NONE)**

信息标识参数定义包含命名字段的文本的信息描述。有两种方法定义这个参数：

[信息前缀]&字段名

字段名必须与字段 **MSGID** 存在于同一个记录格式中。如果定义前缀，则前缀的长度必须是 3，字段名为字符型，长度为 4，用法为 H、P、B 或 O。如果不定义前缀，则必须把字

段名定义成字符型字段，长度为 7，用法为 H、P、B 或 O。

[信息标识前缀]信息标识

可以把信息标识符定义成一个值或值的结合。如果定义前缀，则前缀的长度必须是 3，而信息标识的长度必须是 4。如果没定义前缀，则信息标识的长度必须是 7。

信息标识是必选的参数。

信息文件和库参数表示包含信息描述的信息文件。可以用以下格式定义信息文件和库参数：

[库名/]信息文件

或

[&字段 1]&字段 2

此处字段 1 和字段 2 的长度都是 10。字段名必须存在于和 MSGID 字段同一记录格式中，而且必须是字符型，长度为 10，用法为 H、P、B 或 O。

字段名和常数的组合：

—库名/&字段 1（字段 1 的长度为 10）

—&字段 2/文件名（字段 2 的长度为 10）

信息文件是必选的参数。如果没有定义库名，则用\*LIBL 来查找程序运行时的信息文件。库名是可选的参数。

\*NONE 参数指明没有信息正文显示。

用户和程序可以用 OVRMSGF 键字来覆盖信息文件名。

在一个字段上可以定义多个 MSGID 键字。当定义多个 MSGID 时，除了字段做最后一个 MSGID 键字，其余的都要定义可选指示器。不允许为最后一个（或唯一的）一个 MSGID 键字定义可选指示器。如果一个字段中不只一个 MSGID 键字有效，起作用的是第一个 MSGID。

如果想更多了解 MSGID 键字，参看 F.2.3 的“MSGID”。

可以在一个记录格式中定义多个 MSGID 键字。可在多个 MSIGD 键字上定义字段名作为参数。

在程序运行时间必须对信息文件有使用权限。

MSIGD 字段必须是一个输出字段（用法 B 或 O）。字段的长度是要显示的信息正文的长度。它应该是要显示的最长信息的长度。

MSG 本身不出现在输出缓冲区中，但如果 MSGID 被定义成输出/输入字段（用法 B），可出现在输入缓冲区中。定义成仅输出字段的 MSGID 字段不能用在高级语言程序中。

如果有 MSGID 参数字段，则它们出现在输出缓冲区中，但只有当它们被定义成隐藏字段（用法 H）或输出/输入字段（用法 B）时，才会出现在输入缓冲区中。

一个系统到程序字段可以被定义成信息标识、文件或库名。

不能在子文件记录格式（SFL 键字）中定义 MSGID。

定义 MSGID 的字段不能再定义以下键字：

DFT

DFTVAL

FLTFIXDEC

FLTPCN

MSGCON

图 3-144 给出如何规定 MSGID 键字。

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

```

A      R RECORD1
A      MSGFIELD1    40A B 02 10MSGID(CPD0001 QGPL/USRMSG)
A 01   MSGFIELD2    10A 0 02 60
A 25           MSGID(&MSGIDNUM &MSGFILENM)
A           MSGID(CPD1234 QGPL/USRMSG)
A      MSGFIELD3    80A B 02 60
A           MSGID(USR &MSGNBR +
A           QGPL/&MSGGILENM)
A      MSGIDNUM     7A P
A      MSGFILENM    10A H
A      MSGNBR       4A B 07 01
A

```

图 3-144 规定 MSGID 键字

当显示 RECORD1 时：

字段 MSGFIELD1 包含信息 CPD00001 的前 40 个字符，信息 CPD0001 在库 QGPL 中的信息文件 USRMSG 中。因为字段 MSGFIELD1 是输出/输入字段，所以字段的值可被用户改变。

如果可选指示器 01 为 OFF，字段 MSGFIELD2 不显示。当可选指示器 01 和 25 都为 ON 时，MSGFIELD 包含被字段 MSGIDNUM 和 MSGFILENM 标识信息的前 10 个字符。MSGIDNUM（信息标识符）和 MSGFILENM（信息文件）在程序中的值必须放在 RECORD1 显示之前。

当可选择指示器 01 为 ON 而可选指示器 25 为 OFF，字段 MSGFIELD2 包含信息 CPD1234 的前 10 个字符，信息 CPD1234 来自库 QGPL 的信息文件 USRMSG。因为 MSGFIELD2 是一个只输出字段（用法 O），所以它不能用在程序中。

字段 SMGFIELD3 包含有前缀 USR 信息的前 80 个字符，信息号码放在字段 MSGNBR 中，而且信息文件放在字段 MSGFILENM 中。

### 3.3.97 MSGLOC (信息位置)

用这个文件层键字把信息行移到规定的行。

MSGLOC 的格式为：

MSGLOC (行号)

参数是必需的，范围是 1 – 28 之间的正整数。可用 1 – 28 中的任何值，不管 DSPSIZ 定义的显示尺寸有多大。对 24×20 显示尺寸，当文件打开时如果信息在 26 行于 28 行，则会给出诊断信息。

如果没定义 MSGLOC，则信息行是显示的最后一行。信息行在显示位置给出如下信息：

有效性检查错误

无效功能键信息

非法功能键信息

为 ERRMSGID 和 SFLMSGID 键字标识的信息（整个显示用于信息帮助）

定义为 ERRMSG 和 SFLMSG 参数值的信息

信息字段

操作错误码及相关信息。

如果辅显示尺寸的信息行不同于缺省信息行，则必须定义显示尺寸条件名。

如果没定义 MSGLOC 键字，则用以下的缺省值：

24×80 显示尺寸：第 25 行

24×132 显示尺寸：第 28 行

24×80 的显示尺寸的缺省第 25 行将会产生如下后果：

如果显示送往 5250 或 5251–12 的设备，则第 24 行用作信息行。

如果显示被送往 3180–2 设备或 3197 的 D1, D2, W1 或 W2 (和本地 6040 或 6041 控制器相连，或和远程 5294 或 5943 相连)，则第 25 行被用作信息行。

如果在文件中定义了 ERRSFL 键字，不能显示尺寸 24×80 定义第 25 行作信息行，同样不为显示尺寸 27×132 定义第 28 行作为信息行。在文件中有 ERRSFL 键字而没有定义 MSGLOC，则使用下列缺省值：

24×80 显示尺寸：第 24 行

27×132 显示尺寸：第 27 行

从文件打开到文件关闭，MSGLOC 键字定义一直保持有效。当文件被挂起而另一个文件在同一工作站设备上打开时，挂起文件的 MSGLOC 可暂停权处理。在挂起文件恢复之前使用其它文件的信息行位置。

在信息显示之前存在信息行的数据都被保留，在按下 Reset 键之后恢复。

可选指示器对这个键字无效。

图 3-145 和图 3-146 给出如何规定 MSGLOC 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A                         MSGLOC(1)  
A
```

图 3-145 规定 MSGLOC 键字（例 1）

在图 3-145 中，主显示尺寸的信息行被移到第一行。（不带 DSPSIZ 键字，主显示尺寸是 24×80。）

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A                         DSPSIZ(*DS3 *DS4)  
A                         MSGLOC(1)  
A   *DS4                  MSGLOC(1)  
A
```

图 3-146 规定 MSGLOC 键字（例 2）

此例中，主显示尺寸 1 和辅显示尺寸 2 的信息都被移到第一行。

### 3.3.98 NOCCSID (无编码字符集标识)

用这个字段层键字定义不对字段作 CCSID 转换。

这个键字没有参数。

字符数据表示体系 (CRDA) 规定把'3F'X 字符定义成替换字符。这个字符也是 5250 数据流定义中的字段属性定义。无论\*JOBCCSID 转换是否活动，都要为所有字段的输出把'3F'X 转换成'1F'X。

使用 NOCCSID 键字可以避免在字段层上进行这种转换。

如果没定义 NOCCSID 键字，字段的转换正常进行。

图 3-147 给出如何规定 NOCCDIS 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A  
A          R RECORD  
A          FIELD1      5A  B  2 10NOCCSID  
A
```

图 3-147 规定 NOCCSID 键字

### 3.3.99 OPENPRT (打开打印文件)

用这个文件层键字定义一旦打开打印机文件（用户第一次按下 Print 键后），文件将一直保持打开状态，直到相关显示文件关闭为止。如果没有定义 OPENPRT(而且没有定义 PRINT 键字)，每次打印显示屏幕时，都要打开并关闭打印机文件。

如果多个作业使用同一个打印机文件和设备，则打印机文件要用假脱机处理。当打印机文件在非假脱机方式下打开时，相关的打印机利用这个功能分配给程序或进程。

这个键字没有参数。

只有在定义了文件层的 PRINT 键字并带有打印机文件参数时，OPENPRT 键字才有效。在记录层定义 PRINT 时，OPENPRT 无效。

只有在 PRINT 键字上定义了打印文件时，OPENPRT 键字才有效。

可选指示器对这个键字无效。

图 3-148 给出如何规定 OPENPRT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00031A                      PRINT(PRTFILE)  
00032A                      OPENPRT  
A
```

图 3-148 规定 OPENPRT 键字

### 3.3.100 OVERLAY (复盖)

用这个记录层键字定义：当定义的记录格式出现在屏幕上时，不必先删除整个显示。

这个键字没有参数。

通常每个输出操作都删除整个显示。有全部或部分重叠字段的所有显示记录都要在这个记录显示之前删除。所有留在显示上的其它部分都不变。即使记录中的某些字段位置没被选做显示，已经在显示上的记录也要被删除。举例来说，假设下列记录正在显示：

REC1 (第 1、2 行)

REC2 (第 3、4 行)

REC3 (第 5 行)

REC4 (第 9 行)

有 OVERLAY 规定的 REC5 (第 4、5 行) 的输出会使屏幕显示如下：

REC1 (第 1、2 行)

REC5 (第 4、5 行)

REC4 (第 9 行)

如果规定了有效 OVERLAY 键字的记录已经显示，并且没有定义 PUTOVR，PUTRETAIN，或者 CLRL 键字，它也被删除而且做为一个新记录来重写。

如果一个记录的开始属性字符与一个已经显示记录的结尾属性字符重叠，则属性字符在一行的第一列互相覆盖。（显示的第一个记录的最后一个字段在前一行的最右显示位置结束）。

在上面例子中，如果只是在第 4 行的 REC2 部分是 REC2 最后一个字段的结尾属性字符（REC2 的最后字段的最后两个显示字符是在第 3 行的最后一个位置时就是这种情况），在有 OVERLAY 的 REC5 显示之后 REC2 仍然保持显示，显示器上将是以下记录：

REC1 (第 1、2 行)

REC2 (第 3 行)

REC5 (第 4、5 行)

REC9 (第 9 行)

显示总是在文件打开之后用第一个输出操作删除，除非同时定义了 ASSUME 和 OVERLAY。

OS/400 程序对 ERRMSG, ERRMSGID, PUTOVR 和 CLRL 功能假定使用 OVERLAY。

如果 OVERLAY 有条件限制没被选择，则 ERASE, ERASEINP, MDTOFF, PROTECT 和 PUTERTAIN 这些键字即使被选中也不起作用，除非选中了 PUTOVR，在这种情况下，ERASE、ERASEINP 和 MDTOFF 键字才能起作用。

如果定义了 OVERLAY，应同时在 CRTDSPF 或 CHGDSPF 命令由定义 RSTDSP(\*YES)。如果文件挂起，显示上的数据可能丢失。

要删除显示上的任何记录，可使用 ERASE 键字指定要删除的记录格式的名字。

如果定义了 CLRL 键字，处理将根据 CLRL 定义进行，而不是根据 OVERLAY 规定进行。

如果 OVERLAY 键字在记录上和 DSPMOD 键字一起定义，则在文件建立时会发出警告信息。

可选指示器对这个键字有效。

图 3-149 给出如何规定 OVERLAY 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00021A      R RECL          OVERLAY  
A
```

图 3-149 规定 OVERLAY 键字

### 3.3.101 OVRATR (复盖属性)

用这个字段层或记录层键字和 PUTOVR 键字一起来复盖已经显示的字段或记录的显示属性。

OVRATR 键字可以和 OVRDTA 键字在同一个字段或记录上使用。

当在记录层和字段层上都定义了 OVRATR 时，字段层定义用于本字段。

如果想知道在 System/36 环境的文件中如何定义 OVRATR，请参看应用显示编程手册。这个键字没有参数。

可被 OVRATR 键字复盖的显示属性为：

CHECK(ER)  
CHECK(ME)  
DSPATR(除了 OID 和 SP 外的全部)  
DUP

当 OVRDTA 键字有效时，显示属性可在同一个输出操作中被复盖（如同 OVRATR 键字也生效一样）。

当在字段层上定义 OVRATR 时，只对下列类型的字段有效：

只输入字段  
只输出字段  
输出/输入字段  
常数字段

当在记录层上定义 OVRATR 时，对以下每种类型的字段都有效：

只输入字段  
只输出字段  
输出/输入字段  
常数字段

可选指示器对这个键字有效。

使用 OVRATR 键字的例子，参见 3.3.110 的“PUTOVR”。

### 3.3.102 OVRDTA (复盖数据)

用这个字段层或记录层键字与 PUTOVR 键字一起，复盖已经显示的字段或记录的数据内容。

OVRDAT 键字和 OVRATR 键字可用于同一字段或记录。

当在记录层和字段层都定义了 OVRDTA 时，字段定义将应用于本字段。

这个键字没有参数。

如果为只输出字段或输出/输入字段定义了 DFT 键字，则必须定义 OVRDTA 键字。

在字段层定义 OVRDTA 键字，只对以下类型字段有效：

只输出字段  
输出/输入字段  
信息字段

当在记录层上定义 OVRDTA 键字时，此键字可用于以下类型字段：

只输出字段  
输出/输入字段  
信息字段

可选指示器对这个键字有效。

使用 OVRDTA 键字的例子，参见 3.3.110 的“PUTOVR”。

### 3.3.103 PAGEDOWN/PAGEUP (下页/上页键)

通过这个文件层或记录层键字可使应用程序处理以下情况：工作站用户按下 Page Down 或 Page Up 键时，OS/400 程序却不能在显示器上翻页。如果这种情况发生了，你却没定义 Page Down 或 Page Up 键，则 OS/400 程序会给出出错信息指出所按的键无效。

如了解在 System/36 环境所用的文件中如何定义 PAGEDOWN/PAGEUP，可参见附录 F。  
这两个键字的格式分别是：

PAGEDOWN[ (应答指示器[ ‘正文’ ]) ]

### PAGEUP[ (应答指示器[ ‘正文’ ]) ]

可以为这些键字定义一个应答指示器。如果这样做了，当按下适当的 Page 键时，OS/400 程序打开输入记录中的应答指示器。并在处理完输入数据后把控制返给应用程序。如果没定义应答指示器时按下 Page 键，则 OS/400 程序执行正常的输入记录处理。

可选的正文出现在程序编译时产生的打印输出中，用以解释指示器的用途。这个正文在文件或程序中只起注释的作用，必须用撇号括起来。如果两个撇号之间定义的字符多于 50 个，则在打印输出中截去尾部而只输出 50 个字符。

这两个键可使数据从显示设备传到应用程序（和命令功能键 CF 以及 Enter 键类似）。

不能和 PAGEDOWN 一起定义 ROLLUP，也不能和 PAGEUP 一起定义 ROLLDOWN。

注：PAGEDOWN 和 ROLLUP 相同；PAGEUP 和 ROLLDOWN 相同。

如果 OS/400 程序为子文件（SFLSIZ 值不等于 SFLPAG 值）执行翻页功能，则无需使用 PAGEDOWN 和 PAGEUP 键字。如想了解在子文件中定义了这两个键字会发生什么情况，请参见 3.3.149 “SFLROLVAL”。

可选指示器对这两个键字有效。

图 3-150 给出如何规定 PAGEDOWN 和 PAGEUP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A N64          PAGEUP(52 'Page Up')  
A                   PAGEDOWN(61)  
A
```

图 3-150 规定 PAGEUP 和 PAGEDOWN 键字

### 3.3.104 PASSRCD (传送记录)

用这个文件层键字定义：从其它程序往你的程序传送未格式化数据时，OS/400 所用的记录格式。只有在程序在文件打开后的第一个请求是没有规定记录格式名的输入操作时，才处理被传送的数据。数据必须根据该记录格式处理。

这个键字的格式是：

#### PASSRCD (记录格式名)

记录格式名是这个键字的必选参数且必须存在于文件中。在记录格式中不能定义以下键字：

```
ALWROL  
CLRL  
SLNO
```

可选指示器对这个键字无效。

图 3-151 给出如何规定 PASSRCD 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A          PASSRCD(RECKEEP)  
00020A          R RECORD  
A
```

图 3-151 规定 PASSRCD 键字

### 3.3.105 PRINT (打印键)

用这个文件层或记录层键字定义工作站用户可以按下 Print 键来打印当前显示。

若想了解在 System/36 环境下的文件中如何定义 PRINT, 请参见附录 F。

这个记录的格式是:

PRINT[(应答指示器['正文'] | (\*PGM) | ([库名/]打印文件名))]

以下四个例子说明了定义 PRINT 的四种方法。

PRINT	OS/400 程序把输出送往打印假脱机文件 QSYSPRT, 除非在 CRTDEVDS 指令或 CHGDEVDS 指令中的 PRTFILE 为工作站用户定义另外的打印文件
PRINT(01'USER PRESS PRINT KEY')	程序得到控制并决定做什么 (例如产生格式化打印输出), 应答指示器设为 ON, 没有数据从设备中送出
PRINT(*PGM)	当按下 PRINT 键字时控制返给程序
PRINT(LIBI/PRINTFILE1)	OS/400 程序把打印输出送往规定的打印假脱机文件 (可用 DDS 或 CRTDEVDS 指令或 CHGDEVDS 指令的 PRTFILE 参数规定)。如果在打印文件打开之前 (这时按 PRINT 键) 有一个 OVRPRTF 命令起作用, 也能改变打印设备的名字。

在下面部分将进一步讨论 PRINT 键字的用法。

如果以任何格式定义了 PRINT 键字, 工作站用户都可打印包含信息帮助的显示屏。在这种情况下, 打印操作执行如同定义了不带参数的 PRINT 键字。

#### 3.3.105.1 不带参数的 PRINT 键字

对本地工作站: OS/400 程序把输出送到规定的打印假脱机文件 (可通过 DDS 定义或通过 CRTDEVDS、CHGDEVDS 指令的 PRTFILE 参数定义)。

非显示字段以空白符出现。通过按复制键复制的字符显示为星号 (\*)。显示属性显示为空白符。如果不能成功执行打印功能, OS/400 程序会试图使用一个打印文件来完成打印功能, 这个打印文件是在为系统描述显示设备的 CHGDEVDS 和 CRTDEVDS 命令的 PRTFILE 参数中定义的。

对通过工作站控制连接的工作站打印机, 表示没有问题的信息被发往请求打印的用户。工作站用户要想办法使打印机准备好或按下 Reset 键。如要在打印完成之前取消打印请求, 工作站用户可以先按住 Shift 键 (不松手), 然后按下 Print 键。

注: 在当前显示打印完之后, 打印纸前进两倍于当前显示尺寸的行数 (24×80 显示尺寸是 48 行, 27×132 显示尺寸的 54 行)。

对远程工作站: OS/400 程序试图在不通过系统传送数据的相关工作站打印机上打印显示屏。相关工作站打印机是用在描述本地显示设备的 CRTDEVDS 或 CHGDEVDS 命令的 PRINTER 参数中规定的打印机设备。

如果在按下 Print 键时打印机还没准备好, 则没有信息传给用户。请求打印功能的工作站要保持不可操作状态, 直到打印机准备好或打印请求被取消 (使用换档打印键) 才能进行操作。

注: 在打印完当前显示后, 打印纸前进的行数和当前显示尺寸的行数相同。(对 24×80 显示尺寸是 24 行, 对 27×132 显示尺寸是 27 行。)

可选指示器对此键字有效。

图 3-152 给出如何规定不带参数的 PRINT 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                      PRINT
00020A          R RECORD1
A

```

图 3-152 规定 PRINT 键字

### 3.3.105.2 带有应答指示器或特殊值\*PGM 的 PRINT 键字

如果定义了带应答指示器的 PRINT 键字, OS/400 把指示器设为 ON, 把控制返回应用程序。没有从设备来的数据, 键盘被锁住, 直到程序发出另一个输出操作给显示文件时为止。在本地和远程工作站之间的打印功能没有区别。如果定义\*PGM, OS/400 程序会把控制返给应用程序。这两种格式的唯一不同在于应答指示器; 其他过程都是相同的。

应答指示器格式中的可选正文包含在程序编译时产生的打印输出中, 用以解释指示器的用途。这个正文只能起到注释的作用。正文必须使用撇号 (')。如果在撇号之间定义的字符多于 50 个, 则多出的字符在打印输出中被去掉。

图 3-153 给出如何规定带有应答指示器的 PRINT 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                      PRINT(01 'User presses Print key')
00020A          R RECORD1
A

```

图 3-153 规定有应答指示器的 PRINT 键字

图 3-154 给出如何规定带有特殊值\*PGM 的 PRINT 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                      PRINT(*PGM)
00020A          R RECORD1
A

```

图 3-154 规定有\*PGM 的 PRINT 键字

### 3.3.105.3 与特定打印机文件有关的 PRINT 键字

OS/400 程序读显示缓冲区, 并根据指定的打印文件打印显示屏, 被指定作为这个键字参数值的打印文件可以是外部描述也可以是程序描述的, 可以是或不是假脱机文件, 如果是外部描述的, 那它必须包括一个与文件同名的记录格式名。

当按 print 键时, 打印文件必须存在并且用户有权使用这个显示, 如果使用库名的话, 库名也须如此。如果 OS/400 不能完成指定打印文件功能, 就要使用在 CRTDEVDSF 或 CHGDEVDSF 命令的 PRTFILE 参数中规定的打印文件, 为了避免键盘锁定, 可在 CRTPRTF 或 CHGPRTF 中指定 SPOOL(\*YES)。

如果设指定库名, 就使用程序运行时的当前库列表。

要了解打印文件何时打开与关闭的细节, 请看 3.3.99 “OPENPRT”。

下图说明如何把显示直接送到打印文件 LIB1/PRINTFILE1。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A          PRINT(LIB1/PRINTFILE1)  
00020A          R RECORD1  
A
```

图 3-155 规定与特定打印文件有关的 PRNT 键字

### 3.3.106 PROTECT (保护)

使用这个记录层键字与 OVERLAY 键字一起规定，当定义的记录格式被显示时，已在显示屏上所有输入字段变为仅输出字段，这可以保护这些字段被键入。此键字对指定它的记录格式不起作用。所影响到的字段的数据内容不会改变，除非首先用显示指定可输入字段的记录格式，程序无法读取这些数据。

为保护某一个字段不被键入，请看 DSPATR(PR)键字。

这个键字没有参数。

在指定了 PROTECT 的记录格式中，必须指定 OVERLAY 键字。此外，对有效的 PROTECT、OVERLAY 或 CLRL 键字之一也必须有效。

仅在已选择了 PUTOVR 的第一个输出操作上，能用 PROTECT 键字来保护在其它记录中的输入字段，在以后的输出操作上，仅当 PUTOVR 键字不起作用时，PROTECT 才有效。

如果在某个输出操作中 ERASEINP 和 PROTECT 都起作用，OS/400 首先清除在 ERASEINP 参数中指定的可输入字段，然后对显示屏上的所有可输入字段进行键入保护。

如果在规定了 DSPMOD 键字的记录上指定了 PROTECT 键字，则在生成文件时出现警告信息。在运行时，当显示方式改变了，则忽略 PROTECT 键字。

可选指示器对此键字有效。

图 3-156 给出了如何规定 PROTECT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A          R RECORD1  
00020A          FLD1      5   I   5   3  
00030A  
00040A          R RECORD2          OVERLAY  
00050A  32          PROTECT  
00060A          FLDA      10   I   6   3  
A
```

图 3-156 规定 PROTECT 键字

在图 3-156 中，RECORD1 有一个已经显示和读的可输入字段，当显示和读 RECORD2 时，这些字段仍保留在显示屏上。为了避免对 RECORD1 中的 FLD1 再次输入，应对具有有

效 PROTECT 键字的 RECORD2 发送一个输出操作，这样做 FLD1 不被保护，而 FLD1 被保护。

### 3.3.107 PSHBTNCHC (按钮字段选项)

使用这个字段层关键字，用来选择一个按钮字段。

格式如下：

PSHBTNCHC (选项号 选项说明 [命令键] [\*SPACEB])

选项号参数定义了选项标号，是个必需的参数，它返回给应用程序，指出选择的按钮字段的选项。选项号的有效值为 1-99 间的整数。按钮字段不允许有重复值。

选项说明参数定义按钮字段的注释。这个参数是必需的，它可以用下列格式之一指定：

字符串：‘选项的说明’

程序到系统字段：&字段名

规定的字段必须与选择字段在同一记录内，并且必须定义成用途为 P 的字符字段。

选项说明必须适合为文件规定的最小尺寸的一行。选项说明的最大长度依赖于以下几点：

1. 选项字段的位置
2. 选项说明的长度
3. 选项的列数
4. 选项之间的间隔
5. 最小显示尺寸
6. 如果在窗口内显示的话，窗口的宽度

在选项说明中，可以用大于号 (>) 为注释指定一个助记符，大于号右边的字符就是助记符，它只用于连在支持不可编程工作站增强接口控制器的基于字符的图形显示站上。助记符的例子如下：

选项说明	显示
‘F2=>File’	F2=File
‘F3=F>inish’	F3=Finish
‘>Enter’	Enter

如果在注释中有>号，必须指定两次；正如必须指定两次单引号才能在注释中出现一个单引号一样。例如：

选项说明	显示
‘X>>=1’	X>=1
‘X>>>=1’	X>=1

注：不能指定>作为助记符。

助记符只能是不含空格的单字节字符，在选项说明中只能有一个助记符，并且不应当为各个选项指定同一个助记符。如果多次用同一个助记符，那么使用首次定义的助记符。

命令键参数是可选的，用来指明当选择此选项时哪一个功能键是有效的。下面是可用的参数值：CA01-CA24, CF01-CF24, PRINT, HELP, CLEAR, ENTER, HOME, ROLLUP,

RODOWN。如果此记录没在文件层定义指定的命令键，那么会把这个键加到记录格式上。如果没定义任何键，那么用 ENTER。

\*SPACEB 参数是可选的，指出在选项定位前应插入一个空位。这个参数用来指定选项的逻辑组。

在一个字段中指定键字 PSHBTNCHC 时，也必须指定键字 PSHBTNFLD。

可以为一个选项字段指定多个 PSHBTNCHC 键字，能指定的 PSHBTNCHC 键字数目取决于选择字段的位置和显示尺寸。在一行上可有多个选项，所有的选项必须适合为文件指定的最小显示尺寸，选项的最大数目是 99。

可选指示器对此键字有效，当 PSHBTNCHC 键字置 OFF 时，选项列表被压缩。

按钮总是表现得如同 AUTOENT 和 AUTOSLT 有效一样。

图 3-157 给出如何规定 PSHBTNCHC 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
```

```
A  
A      R RECORD  
A      :  
A      :  
A      F1          2Y 0B 24 02PSHBTNFLD  
A 01          PSHBTNCHC(1 '>Help' HELP)  
A          PSHBTNCHC(2 &F3 CA03)  
A          PSHBTNCHC(3 'E>nter')  
A      F3          4A  P  
A  
A
```

图 3-157 规定 PSHBTNCHC 键字

在此例中，为按钮字段 F1 定义了三个选项，选项 2 的说明放在 F3 中，且选项 2 的助记符必须包含在一应用程序所提供的正文中。如果在写记录时指示器 01 置 OFF，那么只显示选项 2 和选项 3。

### 3.3.108 PSHBTNFLD (按钮字段)

使用这个字段层键字，定义一个字段是按钮字段，按钮字段是由固定数目的选项组成，用户可以从中进行选择，字段呈现为一系列用 ‘〈〉’ 括起的命令键或是一组选项。

键字格式：

```
PSHBTNFLD[ (*NORSTCSR | *RSTCSR]  
    [ (*NUMCOL 列数) | (*NUMROW 行数) ]  
    [ (*GUTTER 间隔) ] ]
```

参数是可选的并能以任意顺序输入。如果没指定参数，选项字段按水平排列。**\*GUTTER**

参数缺省值为 3，字段会按着能显示所有选项要占用的行数显示出来，在选项之间有 3 个空格。

参数 RSTCSR 指定是否允许用箭头键把选择光标移出字段。\*RSTCSR 指定不允许。  
\*NORSTCSR 指定允许，缺省值为\*NORSTCSR。

参数\*NUMCOL 指定要多列显示字段，选项要按指定列数以如下方式排列：

〈F1〉	〈F2〉	〈F3〉
〈F4〉	〈F5〉	〈F6〉
〈F7〉	〈F8〉	〈F9〉

行数指定按钮字段应包括含多少行，行数必须是正整数，整个的单选的按钮字段，当被放到指定数目的行中时，要在显示中放得下。

参数\*GUTTER 指定按钮字段每列间空格数目，不象 SNGCHCFLD 键字，即使没规定 \*NUMCOL 或\*NUMROW 也可以指定\*GUTTER。如果没规定\*GUTTER，缺省值为 3 个空格。间隔必须是大于 1 的正整数。

要了解如何支持不同的设备配置，请看《应用显示编程手册》。

一个字段如果有 PSHBTNFLD 键字，那么它必须也有一个或多个 PSHBTNCHC 键字来为其定义选项。

有 PSHBTNFLD 键字规定的字段，必须把它定义成数据类型为 Y 的可输入字段，长度为 2，小数位为 0。为字段指定的位置是第一个选择按钮的位置。对于输入，字段是被选择的选项号，如果没有选择选项的话为 0。对于输出，字段的值被忽略。

下列键字可与 PSHBTNFLD 键在同一个字段中指定：

ALIAS	INDTXT
CHANGE	NOCSSID
CHCAVAIL	PSHBTNCHC
CHCUNAVAIL	DSPATR(PC)
CHCCTL	TEXT

可选指示器对此键字无效。

图 3-158 给出如何规定 PSHBTNFLD 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R RECORD
A      :
A      :
A          2 40' MENU'
A      F1          2Y 0B 24 02PSHBTNFLD
A                      PSHBTNCHC (1 'Cmd1' CF01)
A                      PSHBTNCHC (2 'Enter')
A 01
A
```

图 3-158 规定 PSHBTNFLD 键字

此例中，当使用一个连在支持不可编程工作站增强接口控制器的图形显示站时，选择字段看起来象这样：

pic20

### 3.3.109 PULLDOWN (下拉菜单)

使用这个记录层键字，来为菜单条定义一个下拉菜单记录。写记录时，系统将它存起来，以后会作为菜单条选项的下拉菜单被显示出来。

键字格式为：

```
PULLDOWN[ (*SLTIND | *NOSLTIND) ]  
          [ (*NORSTCSR | *RSTCSR) ]
```

这些参数是可选的。

参数 SLTIND 指定是否显示下拉菜单中选择字段的选择指示器（类似收音机按钮）。

\*SLTIND 为显示， \*NOSLTIND 为不显示，缺省是\*SLTIND。

参数 RSTCSR 指定在光标移出下拉窗口时是否限制用户使用功能键。\*NORSTCSR 指定当光标在窗口外边时，允许用户按功能键就象光标在窗口内一样。当规定\*RSTCSR 时，如果用户企图当光标在窗口外时按一个功能键，会听到嘟嘟声，而光标被放回到下拉窗口内，控制不返到应用程序，缺省值是\*NORSTCSR。

可以认为有 PULLDOWN 键字的记录是一个窗口记录，尽管没有使用 WINDOW 键字，系统计算下拉窗口的尺寸并生成边界。

下列键字不能与 PULLDOW 键字在同一记录中指定：

ALARM	FRCDTA	OVRDTA
ALTNAM	HLPCLR	PUTOVR
ALWGPH	HLPSEQ	PUTRETAIN
ALWROL	INVITE	RTNDDTA
ASSUME	INZRCD	SFL
CLEAR	MDTOFF	SLNO
CLRL	MNUBAR	USRDFN
ERASE	OVERLAY	WDWTITLE
ERASEINP	OVRATR	WINDOW

可选指示器对此键字无效。

图 3-159 给出如何规定 PULLDOWN 键字

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R MENUBAR          MNUBAR
A      MNUFLD           2Y 0B  1  2
A                      MNUBARCHC(1 PULLFILE 'File      ')
A      :
A      :
A      R PULLFILE        PULLDOWN
A      :
A      :
A

```

图 3-159 规定 PULLDOWN 键字

在这个例子中，记录 PULLFILE 被定义成一个菜单条选项的下拉菜单，写记录 PULLFILE 时，系统把它存起来并在把它从菜单条中选取时显示它。在系统显示 PULLFILE 记录时，它根据 PULLFILE 记录的内容计算下拉窗口需要的尺寸，并生成下拉窗口的边界。

### 3.3.110 PUTOVR (放置明显覆盖)

这是记录层键字 PUTOVR，对于在工作站上已显示了的记录，使用此键字，允许对记录中的字段的显示属性或数据内容（或二者）进行覆盖，通过使用 PUTOVR，可以减少用户显示设备发送的数据量。

在 System/36 环境下如何使用 PUTOVR，请查阅《应用显示编程》手册。

此键字无参数。

如果使用 PUTOVR 和子文件时，有一些限制。有关的更详细内容，请查阅《应用显示编程》手册。

要在一个已显示的记录中选择某一字段进行更改，对向有有效 PUTOVR、OVRDTA 和 OVRATR 键字的记录发送的输出或输出/输入操作，只能更改那些有有效 OVRDTA 或 OVRATR 键字的字段。OVRDTA 更改字段的数据内容，而 OVRATR 更改字段的显示属性。字段的更改方式是通过设置可选指示器来控制的。

下列条件会导致 PUTOVR 键字无效，并且不发生错误：

- 输出操作时 PUTOVR 无效
- 输出操作时 OVRDTA 和 OVRATR 都无效
- 记录格式已不在屏幕上

对一个命名的可输出字段指定 DFT 键字，必须同时指定 PUTOVR 和 OVRDTA 键字。

当 PUTOVR 和 OVRDTA 在字段中都起作用时，DFT 键字指定的缺省值仅在字段的第一次显示中出现，在其后的 PUTOVR 和 OVRDTA 键字起作用时，显示程序值。

如果记录的第一次输出操作中没有显示某个字段，那么是有些限制起了作用，如果在接下来的输出操作中仍存在限制，字段被选择显示并且 PUTOVR 键字也有效：

- 对于选择了 OVRDTA 或 OVRATR 键字的仅输出字段，OS/400 程序不发送结束属性

字符，所有显示属性（如反像）一直在显示中存在，直到下一个字段的开始属性字符出现在显示屏上。要想为后面的覆盖提供结束属性字符，应该在第一个输出操作中显示仅输出字段（也许是由 DSPATR(ND) 键字，这样就无法看到了）。

- 对于选择了 OVRDTA 或 OVRATR 键字的可输入或信息字段，OS/400 发送结束属性字符，必须在最初的输出操作中显示这个字段。

在一个记录格式中不能同时指定 PUTRETAIN 和 PUTOVR。

OVRDTA 只允许在仅输出、输出/输入、程序到系统，或信息字段（处理方式为 O、B、P、M）中使用。

OVRATR 只允许在仅输出、仅输入、输出/输入字段中使用。

如果指定了 PUTOVR，也应该在 CRTDSPF 或 CHGDSFP 命令中指定 RSTDSP(\*YES)。否则，如果文件挂起，显示屏上的数据可能丢失。

如果在有 DSPMOD 键字的记录中指定 PUTOVR 键字，在文件生成时会给出一个警告信息。在运行时，如果显示方式改变了，那么忽略 PUTOVR 键字。

OVRATR 键字只能用来覆盖下列显示属性：

CHECK(ER) CHECK(ME) DUP DSPATR (除 OID 和 SP 外的所有可选值)。

具有有效的 OVRDTA 键字的输出操作不需要用有效的 OVRATR 键字来覆盖字段的显示属性。

可选指示器对 PUTOVR、OVRATR 和 OVRDTA 有效。

图 3-160 给出如何规定 PUTOVR、OVRATR 和 OVRDTA 键字。

00010A	R INVRCD	PUTOVR
00020A	FLD1	1 26' INVENTORY REMAINING IN WAREHOUSE 1'
00030A*		
00040A		3 2' Remaining on hand:'
00050A		OVRATR
00060A 11		DSPATR(HI)
00070A*		
00080A	INVBAL	5Y 0 +2
00090A 12		OVRDTA
00100A*		
00110A		+2' Low on stock' OVRATR
00120A N70		DSPATR(ND)
00130A 70		DSPATR(HI)
00140A*		
00150A	SUPPL	20 B 5 2DFT(' INTERNAL' )
00160A 13		OVRDTA
00170A*		
00180A	ACCT	20 6 20VRDTA
00190A		DSPATR(HI)
00200A 14		DSPATR(RI)

图 3-160 规定 PUTOVR、OVRATR 和 OVRDTA 键字

初始输出操作生成一整屏信息，在第二个输出操作中，PUTOVR 起作用，程序会设置可选指示器对屏幕作下列更改：

- 如果可选指示器 11 置 on，‘Renaining on hand’；常量被置成高亮度，为了使属性恢复正常，将可选指示器 11 置 off，重新显示记录格式。
- 如果可选指示器 12 置 on，程序会改变字段 INVBAL 的显示值。
- 如果可选指示器 70 置 off，‘Low on stock’常量字段是个不可显示字段。如果 70 置 on，字段被置为高亮度。
- 如果可选指示器 13 置 on，程序会用字段 SUPPL 的值来覆盖缺省值 (INTERNAL)。第一个显示的值总是 INTERNAL，为了在把 INTERNAL 改变为别的值后再重新显示值 INTERNAL，程序必须把字段的值设为 INTERNAL，而后再显示它。
- 如果可选指示器 14 置 on，字段 ACCT 的显示属性，在给字段送新数据的同时，由高亮度变成高亮度并且反映像。如果 14 置 off，显示属性变回高亮度。在每次输出操作中新数据会被送到显示屏上。

### 3.3.110 PUTRETAIN (设置保留)

它是记录层或字段层键字，在与 OVERLAY 键字一起使用时，会使 OS/400 程序在重新显示记录时不删除已经显示的数据。PUTOVR 键字与 PUTRETAIN 键字功能相似，但更有效。

此键字无参数。

为了理解这个键字对输出操作有什么效果，请看下列内容：

1. 程序向 RECORD1 发送一个输出操作，显示 RECORD1。如果 PUTRETAIN 有效，被忽略。在 RECORD1 的记录区中的任何数据，在显示 RECORD1 之前均被删除。

2. 一段时间后，RECORD1 还在显示中，你的程序对 RECORD1 发送第二个输出操作，有两种情况可能发生：

- 如果 PUTRETAIN 键字无效，OS/400 程序首先删除 RECORD1 的记录区，然后显示 RECORD1。此时，选择用来显示的字段显示新的数据内容和新的显示属性，这些可以与以前一样。记录区包括 RECORD1 的字段的每一行或字段的一部分。
- 如果 PUTRETAIN 有效，OS/400 不删除 RECORD1 的记录区。被选择的字段的数据内容也不改变，但被选择字段的显示属性送到显示屏上，并能被改变（通过为输出操作选择有效的 DSPATR 键字），没被选择做显示的字段由选择了显示字段一个字符接一个字符地覆盖写下。（详细内容，请看 3.3.111.2 的内容。）

注：在使用字段层的 PUTRETAIN 时，如果记录中的没有字段使 PUTRETAIN 打开，那么删除整个记录区。如果至少一个字段有没被选择的字段层 PUTRETAIN 键字，这将确保记录区不被删除。

如果指定 PUTRETAIN 键字，应该也在 CRTDSPF 或 CHGDSFP 命令中指定

RSTDSP(\*YES)。否则，如果文件挂起，显示屏上的数据可能丢失。

可选指示器对此键字有效。

### 3.3.111.1 影响 PUTRETAIN 键字的条件

PUTRETAIN 只对那些指定了此键字的记录格式有用，并且仅仅在此记录格式已经被显示时才适用。如果指定了 PUTRETAIN 的记录没被显示，忽略 PUTRETAIN。

如果在记录层指定 PUTRETAIN，它对记录格式中被选择显示的所有字段都适用。

这个键字可为记录格式中多个字段指定，但每个字段只能指定一次，这个键字在记录格式层指定，也可以在同一记录格式中的字段层指定。

PUTRETAIN 不能与 PUTOVR 同时指定。如果在有 DSPMOD 键字的记录中指定了 PUTRETAIN。在文件生成时会出现一个警告信息。运行时，在显示方式改变时忽略 PUTRETAIN。

无论什么时候指定 PUTRETAIN 必须同时指定 OVERLAY，OVERLAY 无效，忽略 PUTRETAIN 并且在记录显示以前删除整个显示。

### 3.3.111.2 通过可选指示器选择字段

当 PUTRETAIN 在包括选择字段的输出操作中有效时，记录中没被选择做为显示字段不被删除，它们能部分地或全部地被新选取的字段重写。

如果 PUTRETAIN 仅对新选择的字段有效（在字段层指定），仅字段的开始属性字符送到显示屏，但不送结束属性字符。对于在同一个记录格式中没有指定 PUTRETAIN 的字段，OS/400 发送显示属性和数据。如果 PUTRETAIN 对整个记录有效（在记录层指定），只把开始和结束属性字符送到显示屏。如果一个字段紧接着的前一个字段被选择，而这个字段本身没被选择的话，它的显示属性能够被恢复到正常状态。

例如，假定 DSPATR 对具有覆盖属性字符的两个连续字段有效。如果在有效的 PUTRETAIN 的输出操作中，两个字段中的第一个被选择，而第二个没被选择，第二个字段的显示属性恢复正常，这是因为 OS/400 把显示的第一个字段与开始及结束属性字符一起发送出去，而它的结束属性字符覆盖了第二个字段的开始属性字符。

图 3-161 给出如何规定 PUTRETAIN 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00101A      R CUST  
00102A          PUTRETAIN OVERLAY  
    A
```

图 3-161 规定 PUTRETAIN 键字

### 3.3.112 RANGE (范围)

对可输入字段指定这个字段层键字，可使 OS/400 对工作站用户键入字段中的数据进行有效性检查，键入的数据必须大于等于低值，小于等于高值。注意：OS/400 只检查由工作

站用户更改的或者用 DSPATR (MDT) 设置了数据更改标志 (MDT) 的字段。

注：有关定义用户指定的错误信息的内容，请参考 CHKMSGID 键字。

此键字的格式为：

RANGE (低值 高值)

当字段是字符字段时，参数值必须用撇号 (') 括起。如果是数字型字段，不用撇号。

不能对浮点字段指定 RANGE。

可选指示器对此键字无效。

图 3-162 给出如何规定 RANGE 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RECORD1
00020A* 字符字段
00030A          FIELD1      1   I  2  2RANGE('B' 'F')
00040A          FIELD2      1   I  3  2RANGE('2' '5')
00050A* 数字字段
00070A          FIELD3      1   0I  4  2RANGE(2 5)
00080A          FIELD4      4   0B  5  2RANGE(1 1500)
00090A          FIELD5      7   2B  6  2RANGE(100 99999.99)
00100A          FIELD6      3   0B  7  2RANGE(-100 -50)
00110A          FIELD7      3   2I  8  2RANGE(.50 1.00)
00120A          FIELD8      3   2I  9  2RANGE(.5 1)
00130A          FIELD9      5Y 2I 10  2RANGE(.01 999.99)
A
```

图 3-162 规定 RANGE 键字

FIELD7 和 FIELD8 有相同的 RANGE 参数值。原因是：对于数字型字段，小数点对齐是基于在 36 至 37 位指定的小数位数的，对 FIELD7 和 FIELD8，低值是 0.50，高值是 100。

向数字型字段输入的数据要在小数位上对齐，并且对头尾的空位要填零。例如，如果对 FIELD9 键字 1.2，则送给程序的是 00120，如果键入 100 给 FIELD9，返回到程序的值为 10000。

### 3.3.113 REF (引用)

用这个文件层键字指定一个能引用字段描述的文件名，或要复制以前描述过的记录格式中的几个字段描述信息时所用的文件名。REF 只能用编码一次文件名，而不能象 REFFLD 键字那样对文件的每个字段都能用一次。为了引用多个文件，用 REFFLD 键字，(REF 只能指定一次)。

键字的格式是：

REF ([库名/]数据文件名[记录格式名])

如果在引用文件中有多个记录格式，在键字中要指定记录格式名作为参数值，这样可以告诉 OS/400 使用哪一个记录格式，在顺序检索记录格式时不必如此。

数据文件名参数是必需的，库名和记录格式名是可选的。

如果没指定库名，使用文件生成时的库列表 (\*LIBL)。如果没指定记录格式名，则按顺序检索每个记录格式（如同已指定一样），使用第一次出现的字段名，请看附录 A “指定 REF 和 REFFLD”，来了解选择 REF 和 REFFLD 所确定的检索顺序。

可以对分布式数据管理文件 (DDM) 指定此键字。

使用 DDM 文件时，数据库文件名和库名是源系统上的 DDM 文件名和库名，记录格式名是目标系统的远程文件中的记录格式名。

可选指示器对此键字无效。

图 3-163 给出如何规定 REF 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               REF(FILE1)  
00020A       R RECORD  
00030A       FLD1      R          2  2  
A
```

图 3-163 规定 REF 键字 (例 1)

在图 3-163 中，FLD1 与文件 FILE1 中的第一个（或唯一）FLD1 有相同的属性。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               REF(LIB(FILE1 RECORD2)  
00020A       R RECORD  
00030A       FLD1      R          2  2  
A
```

图 3-164 规定 REF 键字 (例 2)

在图 3-164 中，FLD1 与 LIB1 中的文件 FILE1 的记录 RECORD2 里的 FLD1 有相同的属性。

### 3.3.114 REFFLD (引用字段)

这是字段层键字，在下列情况之一时，用来引用一个字段：

- 被引用字段的名字与 19 列到 28 列上的名字不同。
- 被引用字段的名字与 19 到 28 列上的名字相同，但是它的记录格式、文件或库与在 REF 中指定的不同。
- 被引用字段与引用它的字段在同一个 DDS 源文件中。

键字格式为：

REFFLD ([记录格式名/]被引用字段名[{\*SRC | [库名/]数据文件名}])

被引用字段名参数是必需的，即使是与引用它的字段同名，也要写上字段名。当被引用

文件有多个记录格式时，使用记录格式名。引用字段与被引用字段在同一个 DDS 源文件中时使用\*SRC，（不用数据库文件名）。如果没指定数据文件名和库名和没在文件层指定 REF，那么\*SRC 是缺省值。

注：当引用在同一个 DDS 源文件的字段时，被引用的字段必须在引用字段前面。

指定数据文件名（如果需要的话，要有库名）来检索某个 DB 文件。

如果在同一 DDS 源文件中，既在文件层指定了 REF 又在字段层指定了 REFFLD，那么，具体的检索顺序依赖于 REF 和 REFFLD 二者，详细的内容请看附录 A “指定 REF 和 REFFLD”。

必须在 29 列指定 R。有些情况下，在 DB 文件中规定的字段键字不包括在显示文件中详细内容，请查阅 3.2.9 的“参考”。

可对分布式数据管理（DDM）文件指定此键字。

使用 DDM 文件时，DB 文件名和库名是源系统中的 DDM 文件名和库名，被引用字段名和记录格式名是目标系统中的远程文件中的字段名或记录格式名。

注：IDDU 文件不能用作引用文件。

可选指示器对此键字无效。

图 3-165 给出如何规定 REFFLD 键字（解释这些规定，请看附录 A 的例子）。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R FMAT1
00020A      ITEM      5      3   1
00030A      ITEM1    R      5 2REFFLD(ITEM)
00040A      ITEM2    R      5 12REFFLD(FMAT1/ITEM)
00050A      ITEM3    R      5 22REFFLD(ITEM FILEX)
00060A      ITEM4    R      5 32REFFLD(ITEM LIBY(FILEX)
00070A      ITEM5    R      5 42REFFLD(FMAT1/ITEM LIBY(FILEX)
00080A      ITEM6    R      5 52REFFLD(ITEM *SRC)
A
```

图 3-165 规定 REFFLD 键字

### 3.3.115 RETKEY（保留功能键）和 RETCMDKEY（保留命令键）

它们都是记录层键字，用来指示当显示正定义的记录时，在显示中可用的功能键或命令功能/注意键应被保留。

此键字无参数。

如何规定 RETKEY 和 RETCMDKEY 键字，请看附录 F “System/36 环境考虑”。

### 3.3.116 RETLCKSTS（保持锁状态）

这是记录层键字，用来指定在没发生下一个输入操作时系统锁住键盘。这个键字避免在一个输入操作已开始并且数据已经从键盘传输时丢失数据。

注：正常情况下，一个输入操作可直接解锁键盘，即使键盘已经解锁，任何在解锁时从

键盘传输的数据都有可能丢失。

此键字无参数。

注：仅当键盘没锁定时使用此键字，在键盘锁定时使用此键字会妨碍从键盘进行输入，因为在输入操作时，不能解锁键盘，工作站保持输入禁止状态。

可选指示器对此键字有效。

图 3-166 给出如何规定 RETLCKSTS 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A  
00020A      R REC1           INVITE  
00030A  10          RETLCKSTS  
A
```

图 3-166 规定 RETLCKSTS 键字

如果在记录 REC1 被送到显示屏上时指示器 01 为 ON，那么在访问显示设备时，系统不会直接解锁键盘。

### 3.3.117 RMVWDW (取消窗口)

这是记录层键字，用来在记录显示前取消屏幕上所有存在的窗口。

此键字无参数。

当指定 RMVWDW 键字时，必须在同一个记录格式中指定 WINDOW 键字。RMVWDW 仅仅当 WINDOW 键字定义了一个窗口时才起作用。如果 WINDOW 键字指定了一个记录格式名，那么 RMVWDW 不起作用。

可选指示器对此键字有效。

图 3-167 给出如何规定 RMVWDW 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A      R WINDOW1           WINDOW(6 15 9 30)  
A      FIELD1      5A B 2 2  
A      FIELD2      20A B 8 5  
A      R WINDOW2           WINDOW(&LINE &POS 9 30)  
A 01          RMVWDW  
A      FIELD3      5A B 2 2  
A      FIELD4      20A B 8 5  
A      LINE        2S P  
A      POS         3S P  
A
```

图 3-167 规定 RMVWDW 键字

WINDOW1 已经在屏幕上。如果指示器 01 置 on 时，WINDOW2 写到屏幕上，那么在 WINDOW2 被显示之前要取消 WINDOW1。如果指示器 01 为 off 时，WINDOW2 写到屏幕上，那么显示 WINDOW2 时，WINDOW1 仍然保留在屏幕上。

### 3.3.118 ROLLUP/ROLDDOWN (上翻/下翻)

这是文件层或记录层键字，用来指定当用户按 rollup 或 rolldown 键时，要用程序来处理这些情况，而 OS/400 不能移动屏幕上的内容。如果发生了这种情况而你没有定义这个键字（任何一个都可以），OS/400 发送错误信息，指出这个键此时无效。

要想详细了解在 System/36 环境下，如何使用 ROLLUP/ROLDDOWN 键字，请查阅附录 F。

键字格式为：

ROLLUP[ (应答指示器[ ‘说明’ ]) ]

ROLDDOWN[ (应答指示器[ ‘说明’ ]) ]

如果为键字指定了应答指示器，并且按了相应的翻页键，OS/400 把输入记录中的应答指示器置 on，并在处理完输入数据后把控制返给程序。如果不指定应答指示器，按了翻页键，OS/400 执行正常的输入记录处理。

‘说明’是可选的，它包含在用来解释指示器用途的程序编译时生成的打印输出中。它仅仅是文件或程序的注释，撇号是必须的。如果在两个撇号之间指定的字符多于 50 个，在程序输出中被截断为 50 个字符。

这两个键使数据由显示设备返给程序（与命令功能键 CF 和 ENTER 键近似）。

PAGEUP 不能与 ROLDDOWN 一起指定，PAGEDOWN 不能与 ROLLUP 一起指定。

注：ROLLUP 和 PAGEDOWN 键字相同，ROLDDOWN 和 PAGEUP 相同，ROLL 和 PAGE 相同。

如果 OS/400 执行了子文件的翻页功能 (SFLSIZ 值不等于 SFLPAG 值)，不必指定这两个键字。

当 ROLLUP 和 ROLDDOWN 在子文件中定义时会发生什么情况，请查阅 3.3.149 的“SFLROLVAL”。

可选指示器对此键字有效。

### 3.3.119 ROLLUP/ROLDDOWN 键字的例子

图 3-168 给出如何规定 ROLLUP 和 ROLDDOWN 键字

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8	
00010A N64	ROLDDOWN(52 'Roll Down')
A	ROLLUP(61)
A	

图 3-168 规定 ROLLUP 和 ROLDDOWN 键字

### 3.3.120 RTNCSRLOC (返回光标位置)

这是记录层关键字，用来向应用程序返回光标位置。

可用两种格式指定关键字，分别是：

- 返回光标当前所在处的记录名和字段名，可选地，可指定第三个参数用来保存字段内的相对光标位置。
- 返回光标相对于屏幕的行和列位置。可选地，可规定两个附加参数用来返回光标相对于“活动窗口”（如果存在的话）的行和列位置或者相对于二事件鼠标按钮定义开始的光标位置。

关键字格式为：

RTNCSRLOC ([\*RECNAME] &光标记录 &光标字段 [&光标位置])

或

RTNCSRLOC ({\*WINDOW | \*MOUSE} )  
&光标行 &光标列  
[&光标行 2 [&光标列 2]])

第一种格式的参数是：

\*RECNAME 参数指出 RTNCSTLOC 要返回光标所在处的记录名和字段名。可选地，它也把光标对字段的相对位置一起返回，这个参数是可选的。

光标记录参数指出一个隐藏字段的名字，对于输入，它保存光标所在处的记录名，这个字段必须在这个记录格式中被定义成一个处理方式为 H，长度为 10 的字符型字段。如果光标不在输入的记录区，光标记录字段中包含空格。

光标字段参数也指定一个隐藏字段的名字，在输入中，字段用来保存光标所在处的字段名。这个字段在记录格式中要定义为一个长度为 10，处理方式为 H 的字符型字段。如果光标不在输入的一个字段上，光标字段中包含空格。

光标位置参数是可选的。它指出一个隐藏字段的名字。对于输入，这个字段将保存光标所在字段内的相对位置。字段必须在记录格式中被定义成一个处理方式为 H，长度为 4，小数位为 0 的有符号数字型字段。如果光标在字段的第 1 个位置上，光标位置字段将保存值 1；如果光标在字段的第 I 个位置上，光标位置字段将保存值 I；如果光标不在字段中，光标位置字段将保存值 0；如果光标位于一个菜单条或选择字段列表中，那么光标位置参数返回光标所在处的选项号。

如果光标在有 RTNCSRLOC 键字规定的记录区外，在 RTNCSRLOC 键字中规定的三个字段都保存输入值。如果光标位于一个子文件中，这些字段也保存输入值。如果光标在子文件中的任何地方，光标记录字段将保存一个值。如果光标位于子文件中的一个字段内的话，光标字段和光标位置字段对保存一些值。

第二种格式的参数为：

参数\*WINDOW 或\*MOUSE 参数用来限定光标行 2 和光标列 2 参数。**\*WINDOW** 使得这些参数返回光标相对于活动窗口中第一个可用位置的相对位置，**\*MOUSE** 使得这些参数的返回值是一个二事件鼠标定义被处理前的光标位置。

光标行参数指定一个隐藏字段名，对于输入，它保存光标所在处的行号，这个字段必须

在记录格式中被定义成一个数据型为 S, 长度为 3, 处理方式为 H, 小数位为 0 的字段, 返回给隐藏字段的值是相对于整个显示的值, 这里显示的第一行为 1。

光标列参数指定一个隐藏字段名, 有关这个参数的详细说明请参考光标行, 把行换为列, 二者完全相同。

光标行 2 参数是可选的, 用来指出一个隐藏字段名。如果\*WINDOW 被指定为第一个参数, 这个隐藏字段保存光标相对于‘活动窗口’的第一个可用处的位置。如果没有活动窗口, 这个值与光标行相同。如果光标位于窗口的第一个可用位置上, 光标行 2 字段保存值 1。如果光标在活动窗口外, 这个值有要为负数。如果指定\*MOUSE 为第一个参数, 此隐藏字段保存调用一个二事件鼠标定义前光标所在处的行号。如果没有处理二事件鼠标定义, 此字段为零, 此字段必须在记录格式中被定义成数据类型为 S, 字段长度为 3, 处理方式为 H, 小数位为零的字段。

光标列 2 参数是可选的, 详细说明与光标行 2 相同, 把行换成列, 二者完全相同。

键字的两种格式可以对同一个记录都指定。如果隐藏字段被任何一个参数多次使用, 会发生不可预测的后果。

可选指示器对此键字无效。

图 3-169 给出如何规定 RTNCSRLOC 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R REC01                      RTNCSRLOC (&RCD &FLD &POS)
A
A          FLD      10A   H             RTNCSRLOC (*MOUSE &ROW &COL)
A          RCD      10A   H
A          POS      4S    OH
A          FLD1A    2A    I   3   2
A  10     FLD2A    6A    O   3   18
A N10    FLD3A    10A   O   3   18
A*
A          R REC02                      OVERLAY
A          FLD1A    2A    I   5   2
A          FLD2A    10A   O   5   5
A          FLD3A    6A    O   5   18
A
```

图 3-169 规定 RTNCSRLOC 键字

REC01 和 REC02 都显示在屏幕上, 可选指示器 10 置 off。

下面的表显示光标在指定位置时的返回值。

行	列	光标记录	光标字段	光标位置	光标行	光标列
3	2	REC01	FLD1A	1	3	2
3	19	REC01	FLD3A(1)	2	3	19

3	25	REC01	FLD3A	8	3	25
3	40	REC01	空 白	0	3	40
4	40	空 白	空 白	0	4	40
5	5	REC02	FLD2A	1	5	5
5	40	REC02	空 白	0	5	40

(1) 如果可选指示器 10 为 on, 当光标在 3 行 19 列时返回 FLD2A。

### 3.3.121 RTNDDA (返回数据)

这是记录层关键字，用来指定当程序向此记录格式送一个输入操作时，OS/400 返回与前一个送给这个记录格式的输入操作中返回的相同数据。如果记录格式不曾被读过，忽略 RTNDDA 键字，在 RTNDDA 有效时，程序不用 OS/400 实际从显示设备上传送数据到程序中就可以重读显示屏上的数据。

此关键字无参数。

RTNDDA 在下列情况中被忽略：

- O/I 操作的输入部分 (Put-Get 操作)
- 预先由输出操作对同一记录格式做的输入操作。

RTNDDA 键字只对发送给同一记录格式的没有插入输出操作的输入操作有效。

可以如下使用 RTNDDA：

- 用 RTNDDA 允许主程序去读一个被工作站用户修改了的记录格式。所读的数据告诉主程序哪个程序被调用，这个子程序发送一个输入操作给同一个记录格式（具有有效的 RTNDDA）来读相同的数据，这个例程可以代替向子程序传递参数。

注：必须对两个显示文件都指定 SHARE(\*YES)。

- 用 RTNDDA 允许一个 RPGIII 程序用较少的数据库记录锁定来实现对文件的维护。

例如，程序读一个数据库记录并在显示设备上显示这个记录。工作站用户查看记录，作必要的修改，而后按 Enter 键。在工作站用户进行修改时，数据库记录如果被锁定了的话，不能被其它程序使用。因此建议保持数据库处于非锁状态。然而，当程序从屏幕读记录且修改数据库记录时，数据库记录覆盖程序中显示记录的内部表示。程序重读显示文件而不必通过对显示记录及数据库记录使用不同的字段名来避免覆盖。由指定的 RTNDDA 键字，程序能再次取得显示记录，并完成对数据库的修改。

如果指定了 UNLOCK，就不能指定 RTNDDA。

可选指示器对此关键字无效。

图 3-170 给出如何规定 RTNDDA 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RECORD1          RTNDDA
00020A      FLD1           5   I  2  2
00030A      FLD2           5   B  3  2
A
```

图 3-170 规定 RTNDA 键字

### 3.3.122 SETOF (设置 OFF)

这是记录层键字，用来指定当向记录格式发送的输入操作完成时，指定的应答指示器置 off。

要了解在 System/36 环境下如何使用 SETOF，请查阅附录下“System/36 环境考虑”。

键字格式为：

SETOF (应答指示器[‘说明’])

‘说明’是可选的，它包含在程序编译时生成的打印输出中来解释指示器的用法。‘说明’只作注释，撇号是必需的。如果在两个撇号内指定了 50 个以上的字符，在程序清单中被截断成 50 个。

这个键字可以用来使一个由输出操作置 on 的可选指示器，在接下来的对此记录所作的输入操作完成时返回 off 的条件，(如果没有执行输入操作，应答指示器不变)，不必用程序来关闭指示器。

SETOF 与 SETOFF 等价。

任何指示器对这个键字都是可用的。不必预先定义可选或应答指示器。在你指定 SETOF 时，指示器变成应答指示器。

如果用于 SETOF 键字的指示器也被用于其它键字，比如 CHANGE，指示器的 on/off 状态由其它键字决定。

可选指示器对此键字无效。

图 3-171 给出如何规定 SETOF 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R CUSMST           SETOF(63 'On=display MSG2000 +  
00020A                  CONSOLEMSG')  
00030A      QRYORD       3 0I 5 3  
00040A 63          ERRMSGID(MSG2000 CONSOLEMSG)  
A
```

图 3-171 规定 SETOF 键字

当指示器 63 置 on 时，程序向 CUSMST 发送输出操作，在信息行上显示 MSG2000 信息。对 CUSMST 的下一个输入操作中，SETOF 键字使指示器 63 置 off。(指示器 63 既被用作可选指示器又被用作应答指示器)。

### 3.3.123 SETOFF

SETOFF 等价于 SETOF。

格式是：SETOFF (应答指示器[‘说明’])

建议使用 SETOF 键字，请参考 3.3.122。

### 3.3.124 **SFL** (子文件)

这是记录层关键字，用来指定记录格式是子文件记录格式。记录格式（包括有关的字段描述）后面必须紧跟子文件控制记录格式（用 **SFLCTL** 键字指定）。

此关键字无参数。

在子文件记录中必须甚少指定一个显示字段，除非子文件是信息子文件（见 3.3.143 “**SFLMSGRCD**”）。在记录格式中为字段指定的位置是显示子文件显示的任何一页的第一个子文件记录的位置。这页记录的剩余部分在第一个记录的下面显示。

在一页中的记录数由关键字 **SFLPAG** 的参数值决定。一个子文件能显示多少行，请参看 3.3.145 “**SFLPAG**”。

在子文件控制记录格式中指定的可显示字段，可在同一时间做为子文件记录来显示。然而，在子文件控制记录格式中指定的字段不能与子文件记录中指定的字段发生重迭，即使是由可选指示器指定的。

如果子文件记录或者子文件控制记录在第一个位置处开始会发生重迭，开始于位置 1 的字段有一个在前一行的开始属性字节。因此前一行也是记录格式的一种。在一个显示文件中能指定的子文件个数（每个指定 **SFL** 和 **SFLCTL** 键字）仅仅受限于在显示文件中允许的记录格式的数目（最多 1024 个记录格式，或 512 个子文件），一次可以有 12 个子文件包含活动记录或者被显示。

对显示尺寸\*DS3，如果字段超过 80 列，将与下一行重叠。

可选指示器对此关键字无效。

除 **SFL**，下列关键字也对子文件记录格式有效：

对信息子文件：

**SFLMSGRCD** (在记录层需要)

**SFLMSGKEY** (在字段层需要)

**SFLPGMQ**

对所有其它子文件 (在记录层)

CHANGE                    LOGINP

CHECK(AB)              LOGOUT

CHECK(RL)              SETOF

CHGINPDFT             SETOFF

INDTXT                 SFLNXTCHG

KEEP                    TEXT

下列关键字，在子文件记录格式指定时，在字段层是无效的：

DATE                   ERRMSGID

DFTVAL                MSGID

ERRMSG                TIME

图 3-172 给出如何规定 **SFL** 键字。

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

00010A	R SFLR	SFL
A*		
00020A*	(至少一个显示的字段)	
A*		
00030A	R SFLCTRL	SFLCTL(SFLR)
00040A		SFLPAG(17)
00050A		SFLSIZ(17)
00060A		SFLDSP
00070A		SFLDSPCTL
A		

图 3-172 规定 SFL 键字

### 3.3.125 SFLCHCCTL(子文件选项控制)

这是字段层键字，用在选择列表上，控制可用的列表选项。

键字格式是下列二者之一：

SFLCHCCTL[ (信息标识[[信息库/]信息文件]) ]

或 SFLCHCCTL[ (&信息标识[[&信息库/]&信息文件]) ]

信息标识和信息文件参数是可选的，它们指定当用户选择了无效选项时要显示的信息。如果没指定这些参数，当用户选择了无效选项时，系统发出缺省信息 CPD919B。如果一个字段被用作信息标识，此字段必须在现在定义的记录中且被定义成数据类型为 A，处理方式为 P，长度为 7 的字段。当使用信息标识参数时，信息文件参数是必须的。如果没规定库参数，用\*LIBL 来检索信息文件。如果用一个字段做信息库或信息文件，此字段必须在现定义的记录格式中，且定义为数据类型为 A，处理方式为 P，长度为 10 的字段。

当在一个字段中指定 SFLCHCCTL 时，这个字段被认为是记录的控制字段。它必须是在子文件记录中被定义的第一个字段，它的长度为 1，数据类型为 Y，小数位为 0，处理方式为 H，它必须被定义成子文件的第一个字段。

控制字段如下工作：

图 3-173 SFLCHCCTL 键字的控制字段

控制值	输出时含义	输入时含义
0	可用	不能选
1	可选	可选
2	不可用。除非对此选项有可用的帮助，否则不能把光标放在选项上。(1)	
3	不可用。允许把光标放在选项上	
4	不可用。即使有可用的帮助，也不能把光	

标放在选项上。(1)

注：(1) 仅适用于连在支持不可编程工作站增强接口控制器上的显示器。

可选指示器对此键字无效。

SFLNXTCHC 不能用在对字段规定了 SFLCHCCTL 键字的记录。

在一个子文件记录内只能使用一个 SFLCHCCTL 键字。

图 3-174 给出如何规定 SFLCHCCTL 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R SFLRCD          SFL
A      CTLFLD     1Y 0H   SFLCHCCTL
A      F1        4A 0 6 10
A      R SFLTLRCD        SFLCTL (SFLRCD)
A                      SFLMLTCHC
A                      SFLPAG (5) SFLSIZ (&SFLSIZ)
A                      SFLDSP SFLDSPCTL
A                      ROLLUP (10)
A 10                 SFLEND (*SCRBAR)
A      F3        5S 0H   SFLSCROLL
A      F2        4S 0H   SFLRCDNBR (CURSOR *TOP)
A      SFLSIZ       5S 0P
A                      1 30' Panel Title'
A                      4 5' Multiple selection list:'
```

图 3-174 规定 SFLCHCCTL 键字

### 3.3.126 SFLCLR (子文件清除)

这是记录层键字，用在子文件控制记录格式中，这样程序能清除所有记录的子文件。这个键字与 SFLDLT 不同，它不删除子文件，与 SFLINZ 也不同，它清除后子文件中不保存数据。清除子文件不会影响屏幕显示。然而，在清除后，子文件不保存活动记录。

此键字无参数。

当活动记录已经在子文件中并且都要被替换时，程序可以向选择了 SFLCLR 键字的子文件控制记录格式发送输出操作。这会清除子文件并且允许程序向子文件写新的记录（通过向子文件记录格式发送输出操作来增加相关记录号。），向已往活动的子文件记录发送输出操作，会导致错误信息返回给程序。

如果 SFLCLR 对输出操作有效，并且在子文件中没有记录，则忽略 SFLCLR。

这个可选键字只对子文件控制记录格式有效，显示尺寸条件名对这个键字无效。

为了避免 OS/400 每次对子文件控制记录格式的输出操作中都清除子文件，必须使用可选指示器。

图 3-175 给出如何规定 SFLCLR 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R SFLR          SFL
    A*
    A*          (至少一个显示的字段)
    A*
00020A      R SFLCTRL        SFLCTL(SFLR)
00030A          SFLPAG(17)
00040A          SFLSIZ(17)
00050A 01      SFLDSP
00060A 01      SFLDSPCTL
00070A N01     SFLCLR
    A

```

图 3-175 规定 SFLCLR 键字

当可选指示器 01 在对 SFLCTRL 的输出操作置 on 时，显示子文件，并且当指示器 01 在对 SFLCTRL 的输出操作置 off 时，清除子文件。一般情况下，为 SFLCLR 指定的可选指示器是与为 SFLDSP 和 SFLDSPCTL 指定的可选指示器状态是相反的。

### 3.3.127 SFLCSRPRG (子文件光标处理)

这是字段层键字，用来定义当光标离开字段时，它走到下一个子文件记录的同一个字段中，而不是走到同一记录的下一个字段中去。当工作站不是连到支持增强数据流的控制器时，忽略 SFLCSRPRC 键字。

此键字无参数。

可选指示器对此键字无效。

SFLIN 不允许在有 SFLCSRPRG 的记录中指定。

图 3-176 给出如何规定 SFLCSRPRG 键字

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
    A
    A      R SFL01          SFL
    A      S1              10A  B  5  5SFLCSRPRG
    A      S2              10A  B  5  25
    A      R CTL01         SFLCTL(SFL01)
    A                  SFLPAG(5) SFLSIZ(20)
    A                  SFLDSP   SFLDSPCTL

```

图 3-176 规定 SFLCSRPRG 键字

本例中，光标离开字段 S1 时，它走到下一个子文件记录中的 S1 中去。

### 3.3.128 SFLCSRNN (子文件光标相对记录号)

这是记录层关键字，用在子文件控制记录格式中，返回子文件中光标所在处的相对记录号。如果子文件记录出现多行，此关键字可与 SFLMODE 一起使用来确定光标位置。

关键字格式为：

**SFLCSRNN (&相对记录)**

相对记录参数是必需的，它指定一个隐藏字段的名字，对于输入，此字段保存子文件记录中的光标所在处相对记录号。此字段必需在子文件控制记录格式中定义成一个长度为 5，小数位数为 0，处理方式为 H 的有符号数字型字段。

如果光标不在与子文件控制记录相关的子文件中，或者如果光标虽在子文件中，但不是在子文件中的活动记录中，那么相对记录字段就包含 0。如果指定了 SFLMODE，在任何情况下都要返回子文件方式。

此关键字可用于有选择字段的子文件或有 SFLLIN 键字的子文件。如果光标位于两个水平子文件记录中间，返回的相对记录号为 0。

SFLCSRNN 键字的例子在 3.3.140 的“SFLMODE”中。

### 3.3.129 SFLCTL (子文件控制)

这是记录层关键字，用来指定记录格式是子文件控制记录格式。这个记录格式必须紧跟在子文件记录格式后面。

关键字格式为：

**SFLCTL (子文件记录格式名)**

必须为这个关键字指定子文件记录格式名作为参数值。子文件控制记录格式包括字段描述及子文件控制键字。

只有向子文件控制记录格式发送一个输出操作，程序才能显示子文件记录。

子文件记录格式 (SFL) 定义子文件中的记录格式，而子文件控制记录格式 (SFLCTL) 定义如何显示、清除、删除和初始化子文件。程序向子文件记录格式发送输出操作来建立子文件。程序也向子文件控制记录格式发送输出操作，为各个子文件键字设置可选指示器，来显示、清除、删除及初始化子文件。

有关信息子文件，请看 3.3.142 的“SFLMSGKEY”的说明，3.3.143 的“SFLMSGRCD”和 3.3.146 的“SFLPGMQ”。

下面是与 SFLCTL 一起使用的子文件键字的概括：(它们都是子文件控制记录中字段使用的字段层键字)。

必须的：

**SFLCTL**

**SFLDSP**

**SFLPAG**

**SFLSIZ**

可选的：

**CHCAVAIL**

**SFLLIN**

CHCSLT	SFLMCLTCHC
CHCUNAVAIL	SFLMSG
SFLCLR	SFLMSGID
SFLDLT	SFLPGMQ
SFLDROP	SFLRCDNBR
SFLDSPCTL(1)	SFLRNA
SFLEND	SFLROLVAL
SFLENTER	SFLSCROLL
SFLFOLD	SFLSNGCHC
SFLINZ	

(1) 如果程序向子文件控制记录格式发送一个输入操作, 键字 SFLDSPCTL 是必需的。如果子文件尺寸等于子文件页尺寸, 忽略下列键字。当使用多个显示尺寸时 (DSPSIZ), 仅对子文件尺寸等子文件页尺寸的显示尺寸忽略这些键字:

SFLDROP      SFLFOLD      SFLROLVAL

如果子文件记录格式中有选择字段, 下列键字对子文件控制记录格式无效:

SFLDROP      SFLFOLD      SFLINZ

SFLLIN      SFLRCDNBR      SFLRNA (因为 SFLINZ 无效)

SFLROLVAL

键字 USRDFN 对子文件控制记录格式无效。

可选指示器对此键字无效。

图 3-177 给出如何规定 SFLCTL 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R SFLR          SFL
             A*
             A*          (至少一个显示的字段)
             A*
00020A      R SFLCTRLR     SFLCTL (SFLR)
00030A          SFLPAG (17)
00040A          SFLSIZ (17)
00050A          SFLDSP
00060A          SFLDSPCTL
             A
```

图 3-177 规定 SFLCTL 键字

### 3.3.130 SFLDLT (子文件删除)

这是记录层键字, 与在子文件控制记录格式中的可选指示器一起使用可以使程序能删除子文件。当显示文件中的活动子文件已达到最大数 (24), 而另一个子文件又要求活动时, 在激活新子文件前必须先删除已活动的某个子文件。

此键字无参数。

为了激活子文件，程序向子文件记录格式发送输出操作或向有有效 SFLINZ 键字的子文件控制记录格式发送输出操作。为了删除子文件，程序应向有有效 SFLDLT 键字的子文件控制记录格式发送输出操作。(关闭显示文件可删除所有活动的子文件)。

如果程序向未活动子文件发送有 SFLDLT 的输出操作，键字 SFLDLT 被忽略。可选指示器对此键字有效，显示尺寸条件名无效。

图 3-178 给出如何规定 SFLDLT 键字。

00010A	R SFLR	SFL
A*		
A*	(至少一个显示的字段)	
A*		
00040A	R SFLCTRL	SFLCTL(SFLR)
00050A		SFLPAG(17)
00060A		SFLSIZ(17)
00070A 01		SFLDSP
00080A 01		SFLDSPCTL
00090A 04		SFLDLT
A		

图 3-178 规定 SFLDLT 键字

当可选指示器 01 为 on 时，对 SFLCTRL 的输出操作显示子文件，而当可选指示器 04 为 on 对 SFLCTRL 做输出时，子文件被删除。一般情况下，为 SFLDLT 指定的可选指示器与为 SFLDSP 及 SFLDSPCTL 指定的可选指示是不相同的。

### 3.3.131 SFLDROP (子文件撤消)

这是记录层键字，用在子文件控制记录格式上，它分配一个命令注意键 (CA) 或命令功能键 (CF)，用户可以按它来折叠或截断需要多个显示行的子文件记录。

键字格式为：

SFLDROP (CAnn | CFnn)

如果不指定 SFLDROP，OS/400 显示整个子文件记录，并且在必要时把它重叠起来。当指定 SFLDROP 时，OS/400 首先以截断方式显示子文件，子文件记录被截断成适于在一行上显示。当工作站用户按下指定键时，OS/400 再以重叠方式显示子文件记录，每条记录的续行紧接着起始行来显示。

通过按指定键，子文件的显示格式由一种状态变为另一种状态。

以截断方式显示的记录比在 SFLPAG 键字中指定的要多，以重叠方式显示的记录与在 SFLPAG 键字中指定的一样多。

OS/400 在仅输出字段的中间截断子文件记录。然而，如果在可输入字段中间作截断，

那么整个字段在显示中丢失。如果这导致整个记录从显示丢失，会显示一个错误信息且不截断记录，仍以重叠方式显示。

注：

1. 如果整个记录可在一行显示，在文件生成时会有一条警告信息。
2. 如果子文件尺寸等于子文件页尺寸，忽略 SFLDROP。当使用多个显示尺寸时（指定 DSPSIZ 键字），SFLDROP 只对那些子文件尺寸等于子文件页尺寸的显示尺寸的被忽略。如果子文件记录格式包含选择字段，SFLDROP 无效。
3. 如果子文件中有输入字段，建议你使用 CF 键而不用 CA 键字。如果在这种情况下，指定了 CA 键，按此键时修改的数据会丢失。
4. 如果一次显示多个使用 SFLDROP 键字的子文件，应在每个 SFLDROP 键字中指定相同的功能键。如果功能键不同，只有为最近显示的子文件指定的键有效。按功能键会影响子文件所用的光标，如果光标不位于子文件中，功能键影响上级子文件。
5. 在同一个子文件控制记录格式中可以指定 SFLDROP 为 SFLFOLD，如果两个键字都活动，使用 SFLFOLD 键字。显示子文件时检查指示器。两个键字必须使用同一个键（key）。可选指示器对此键字有效。

图 3-179 给出如何规定 SFLDROP 键字

00010A	R SFLR	SFL
A*		
A*	(子文件记录在屏幕的一行放不下)	
A*		
00040A	R SFLCTRLR	SFLCTL (SFLR)
00050A		SFLPAG (17)
00060A		SFLSIZ (34)
00070A		SFLDSP SFLDSPCTL
00090A		SFLDROP (CF03)
A		

图 3-179 规定 SFLDROP 键字

显示子文件时，工作站用户可以按 F3 键来把子文件从截断方式变换到折叠方式或从折叠方式变换到截断方式。

### 3.3.132 SFLDSP (子文件显示)

这是记录层键字，用在子文件控制记录格式中，这样在程序向子文件控制记录格式发送输出操作时，可以让 OS/400 显示子文件。如果没对此键字使用可选指示器，在每次对子文件控制记录格式的输出操作中显示一页子文件记录。

此键字无参数。

要想确定在显示子文件时，显示的是子文件中哪一页，请看 3.3.147 “SFLRCDNBR”。

当 SFLDROP 有效而子文件没被激活时（通过加记录或用 SFLINZ），如果程序对子文件控制记录格式发送了输出操作，那么会有错误信息送给程序。

此关键字只对子文件控制记录格式有效而且是必需的。显示尺寸条件名对此关键字无效。  
可选指示器对此关键字有效。

图 3-180 给出如何规定 SFLDSP 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R SFLR          SFL  
    A*  
    A*      (至少一个显示的字段)  
    A*  
00040A      R SFLCTRL       SFLCTL(SFLR)  
00050A          SFLPAG(17)  
00060A          SFLSIZ(17)  
00070A 01        SFLDSP  
00080A          SFLDSPCTL  
    A
```

图 3-180 规定 SFLDSP 键字

在可选指示器 01 为 on，对 SFLCTRL 进行输出操作时，显示子文件。

### 3.3.133 SFLDSPCTL (子文件显示控制)

这是记录层关键字，用在子文件控制记录格式中，当程序向子文件控制记录格式发送输出操作时，OS/400 程序显示子文件控制记录格式中的字段。如果此关键字不使用可选指示器，在每次对子文件控制记录格式所作的输出操作都要显示子文件控制记录。

此关键字无参数。

这是个只对子文件控制记录格式有效的可选关键字。显示尺寸条件名对此关键字无效。  
可选指示器对此关键字有效。

注：当子文件由于子文件控制记录的输入操作而被显示时，SFLDSPCTL 必须是有效的，即使在子文件控制记录格式中没有可能显示字段也是这样。

图 3-181 给出如何规定 SFLDSPCTL 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R SFLR          SFL  
    A*  
    A*      (至少一个显示的字段)  
    A*  
00040A      R SFLCTRL       SFLCTL(SFLR)  
00050A          SFLPAG(17)  
00060A          SFLSIZ(17)
```

00070A 01	SFLDSP
00080A	SFLDSPCTL
A	2 10' NAME'
A	2 34' ADDRESS'
A	

图 3-181 规定 SFLDSPCTL 键字

当可选指示器 01 为 on, 对 SFLCTRLR 进行输出操作时, 子文件控制记录格式中的子文件和可显示字段都被显示。

### 3.3.134 SFLEND (子文件结束)

这是记录层键字, 用在子文件控制记录格式中, 允许在子文件或翻卷条所在显示的右下方显示加号 (+) 或标记 (More…或 Bottom) 加号 (+) 或标记 More…表示工作站用户可以通过按 PageUp 键来去掉标记, 显示更多的记录。

键字格式为:

```
SFLEND[ (*PLUS | *MORE | {*SCRBAR[*SCRBAR |
    *PLUS | *MORE]}) ]
```

翻卷条指示不同类型的子文件信息:

- 用户在子文件中的哪个地方
- 子文件多大
- 用户看子文件的哪一部分

如果所用的设备配置支持指针设备 (Pointer device), 翻卷条也可以在整个子文件中翻动。关于如何支持不同的设备配置, 请看应用显示编程手册。

参数值\*PLUS、\*SCRBAR、\*MORE 是可选的。如果没有规定, \*PLUS 为缺省值, 只有指定\*SCRBAR 为第一个参数时, 才能使用第二组\*PLUS、\*MORE、\*SCRBAR。对于第二组参数, \*SCRBAR 是缺省值。

\*PLUS 告诉系统使用加号指出可用 Page Down 键来看更多的记录。

MORE 告诉系统用 More… 来提示可用 Page Down 键来看更多记录。\*MORE 也告诉系统用 Bottom 来提示显示的是最后一个子文件记录。

指定\*MORE 时, 子文件在屏幕上多占一行 (SFLPAG+1), 这一行放 More… 和 Bottom。如果在屏幕或窗口中没有地方来作附加行时, 在文件生成时会有一条信息, 且文件不能生成。

SCRBAR 告诉系统对图形显示使用图形翻卷条。当指定\*SCRBAR 时, 子文件中各行的最后 3 列留给翻卷条。在使用\*SCRBAR 时, 可以指定第二个参数, 第二个参数告诉系统对非图形显示使用哪个翻卷条指示, \*SCRBAR 是这些显示的缺省值。\*MORE 和\*PLUS 可用作第二个参数值。使用\*SCRBAR 时, 子文件必须至少占 3 行。SFLFOLD 或 SFLDROP 与翻卷条一起处理。子文件的两种方式 (折叠或截断) 都必须占 3 行。

关于怎样用 PAGEUP、PAGEDOWN 及 SFLSIZ 键字控制翻卷条, 请看《应用显示编程手册》。

必须为此键字指定可选指示器。

### 3.3.134.1 由应用程序翻页 (SFLPAG 等于 SFLSIZ)

程序通过使用 SFLEND 中的指示器来控制显示加号或 More… 及 Bottom。关闭指示器可显示加号或 More，打开指示器要取消加号或 Bottom。按 Page Up 键时，程序控制处理过程，例如读子文件，清除子文件，然后用新记录重写子文件并重新显示它。如果程序做这些动作，要显示加号或 More…。否则，从屏幕上取消加号或 Bottom。

注：当 SFLPAG 等于 SFLSIZ 时，可用 \*SCRBAR。翻卷条会和圆点、箭头及覆盖整个箭头的翻卷盒一起显示。

### 3.3.134.2 由 OS/400 翻页 (SFLPAG 不等于 SFLSIZ)

显示的子文件中有较多的记录，那么无论可选指示器在什么状态，OS/400 都显示加号。翻卷条会与放在指出用户在子文件位置的翻卷箭头上的翻卷盒一起显示。当显示子文件的最后一页时，如果指示器为 off，OS/400 显示加号，More…，或带有在翻卷按钮上有一页尺寸的翻卷盒的翻卷条，如果指示器为 on，OS/400 不显示这些内容。

在显示子文件时，应用程序必须设置指示器 on 或 off（当 OS/400 对子文件翻页时，应用程序不知道显示的是子文件哪一页）。

如果显示子文件时应用程序关闭 SFLEND 的指示器，那么或者显示加号，More… 或者与子文件的最后一页一起显示带有在翻卷按钮上有一页尺寸的翻卷盒的翻卷条。因为显示了加号，但 OS/400 不能再对子文件翻页，应用程序必须提供更多的页。在子文件控制记录格式中指定 PageDown 键字，可在按 Page Down 键时把控制传给应用程序。程序接收到控制后可在子文件增加更多的记录，并用 SFLRCDNBR 键字来显示新的一页。

注：如果与翻卷条一起指定 PageDown 键字，那么，当按了 PageDown 键控制传给程序或图形翻卷条的操作显示部分页。

### 3.3.134.3 用 \*PLUS 时加号的位置

对于  $24 \times 80$  显示尺寸，子文件所用最后一行的 78 到 80 列用做开始属性字符，加号及结束属性字符。对于  $27 \times 132$  显示尺寸，子文件最后一行的 130 到 132 列被用于开始属性字符，加号及结束属性字符。

注：如果输入字段占了加号的位置并且字段被修改，加号及其属性字符作为字段的数据返回给程序。对于选择列表，加号放在列表选项的右边。

### 3.3.134.4 用 \*MORE 时，More… 和 Bottom 的位置

对于  $24 \times 80$  显示尺寸，紧接子文件的最后一行的 67-80 列被用做开始属性字符，右对齐 More… 或 Bottom 及结束属性。对  $27 \times 132$ ，是在紧接子文件最后一行的 119-132 列用做这些内容。对选择列表，More… 和 Bottom 放在列表选项的右边。

### 3.3.134.5 用 \*SCRBAR 时，翻卷条的位置

对于  $24 \times 80$  显示尺寸，子文件每行的 77 到 80 列保留做翻卷条。子文件中任何字段不能占用这几列。这样在子文件的多行中没有字段出现。可用多行子文件。对于  $27 \times 132$  显示尺寸，129 到 132 列是保留做翻卷条，对选择列表，翻卷条会放在列表选项的右边。对于其它子文件，翻卷条会放在 79 列。

图 3-182 给出如何规定无参数的 SFLEND 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R SETSFL1          SFL  
00020A 50          SFLNXTCHG  
00030A      SETSEL      1Y 0B 6 2VALUES(1 2 9) CHECK(AB)  
00040A      SETNAME     10A 0 6 4  
00050A      R SETCTL1        SFLCTL(SETSFL1)  
00060A          SFLSIZ(34)  
00070A          SFLPAG(17)  
00080A 40          SFLDSP  
00090A 41          SFLDSPCTL  
00100A 42          SFLDLT  
00110A 43          SFLCLR  
00120A 49          SFLEND  
00130A N49         ROLLUP(26)  
00140A      SETRRN      4S 0H  SFLRCDBNR(CURSOR)  
A
```

图 3-182 规定无参数的 SFLEND 键字

由 OS/400 提供翻页功能，加号出现在显示屏的右下角，当最后一条记录写到子文件中时，指示器 49 为 on，这使 Page Down 键无效并且从显示屏上取消加号。

图 3-183 给出如何规定带\*More 参数的 SFLEND。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R SETSFL2          SFL  
00020A 50          SFLNXTCHG  
00030A      SETSEL      1Y 0B 6 2VALUES(1 2 9) CHECK(AB)  
00040A      SETNAM      10A 0 6 4  
00050A      R SETCTL2        SFLCTL(SETSFL2)  
00060A          SFLSIZ(34)  
00070A          SFLPAG(17)  
00080A 40          SFLDSP  
00090A 41          SFLDSPCTL  
00100A 42          SFLDLT  
00110A 43          SFLCLR  
00120A 49          SFLEND(*MORE)  
00130A N49         ROLLUP(26)
```

00140A	SETRRN	4S 0H	SFLRCDBR (CURSOR)
A			

图 3-183 规定带\*More 参数的 SFLEND 键字

OS/400 提供翻页功能，More…出现在屏幕的右下角的子文件的最后一行后面，当写入子文件最后一条记录时，指示器 49 为 on，这使 Page Down 键失效，并且 Bottom 取代 More…。

图 3-184 给出如何规定带有\*SCRBAR 参数的 SFLEND。

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8			
00010A	R SETSFL2	SFL	
00020A 50		SFLNXTCHG	
00030A	SETSEL	1Y 0B 6 2VALUES(1 2 9) CHECK(AB)	
00040A	SETNAM	10A 0 6 4	
00050A	R SETCTL2	SFLCTL(SETSFL2)	
00060A		SFLSIZ(34)	
00070A		SFLPAG(17)	
00080A 40		SFLDSP	
00090A 41		SFLDSPCTL	
00100A 42		SFLDLT	
00110A 43		SFLCLR	
00120A 49		SFLEND(*SCRBAR *MORE)	
00130A N49		ROLLUP(26)	
00140A	SETRRN	4S 0H	SFLRCDBR (CURSOR)
A			

图 3-184 规定带有\*SCRBAR 参数的 SFLEND

OS/400 提供翻页功能，翻卷条会在图形显示中出现。如果没有图形显示，要看多行子文件记录要看，那么 More…出现在显示屏右下角的子文件的最后一行后面，当写入子文件的最后一条记录时，指示器 49 为 on，这使 Page Down 键失效，并且 Bottom 取代 More…。

翻卷条会与翻卷按钮上的翻卷盒一起显示。

### 3.3.135 SFLENTER (子文件 enter 键)

这是记录层键字，用在子文件控制记录格式中，指定 Enter 键用作 Page Up 键。在按 Enter 键时 OS/400 程序能处理多页子文件的翻页。

键字格式为：

**SFLENTER (CAnn ! CFnn)**

这是仅对子文件控制记录格式有效的可选键字。如果子文件当前没显示或不能对显示的子文件翻页（只有一页或更多），这时按 Enter 键，那么控制会返给应用程序，用这个键字参数值，来指定可代替 Enter 键的功能键。这个键字一般在子文件完全由工作站用户输入

或用户对子文件进行修改及增加时使用。

可选指示器对此键字无效。

注：这个键字仅在下一次输出操作前有效。在下一次输出操作时，使用对那个记录的规定。

若在同一时间多个子文件用 **SFLENTER** 键字，仅在最近显示的子文件中规定做 **enter** 键的 CA 和 CF 键有效。按 **enter** 键时光标的位置决定哪个子文件有效。

图 3-185 给出如何规定 **SFLENTER** 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R SFLR           SFL  
00020A*  
00030A*      (至少一个显示的字段)  
00040A      R SFLCTRL        SFLCTL(SFLR)  
00050A          SFLPAG(17)  
00060A          SFLSIZ(51)  
00070A          SFLDSP SFLDSPCTL  
00080A          SFLENTER(CF01)  
A
```

图 3-185 规定 **SFLENTER** 键字

**enter** 键用作 **Page Up** 键。为了输入数据，工作站用户要按 **CF01**。

### 3.3.136 **SFLFOLD** (子文件折叠)

这是记录层键字，用在子文件控制记录格式中，来指定一个命令注意键 (CA) 或命令功能键 (CF)，工作站用户可以按它来折叠或截断需要多行显示的子文件记录。

键字格式为：

**SFLFOLD** (CAnn | CFnn)

当指定 **SFLFOLD** 时，OS/400 首先以折叠方式显示子文件。当工作站用户按指定键时，OS/400 以截断方式显示记录，由按规定键显示的子文件状态由一种方式变为另一种方式。在截断时，子文件记录只占一个显示行。

如没有 **SFLFOLD** 的话，OS/400 显示整个子文件，必要的话有折叠，但是工作站用户无权选择以截断方式显示子文件。

用折叠方式，显示的记录由 **SFLPAG** 决定，用截断方式，显示的记录比由 **SFLPAG** 键字中指定的要多。

OS/400 在仅输出字段中间截断子文件记录，这样做的话，整个字段在显示中被丢失。如果由此导致整个记录从显示中丢失，会显示一个错误信息，并且记录不被截断，仍以折叠方式显示。

注：

- 1.如果整个记录可在一行显示，在文件生成时会有一条警告信息（级别为 10）。

2.如果子文件尺寸等于子文件页，产生错误信息（级别为 20）并且忽略 SFLFLD。当用多个显示尺寸（指定 DSPSIZ 键字）时，SFLFOLD 对那些子文件尺寸等于文件页的显示尺寸忽略。如果子文件记录格式包括选择字段，SFLFOLD 无效。

3.如果子文件包含输入字段，建议指定 CF 键而不要用 CA 键。如果在此时指定 CA 键，按此键时修改的数据会丢失。

4.如果一次显示几个使用 SFLFOLD 的子文件，应在每个 SFLFOLD 键字中指定相同的功能键。如果功能键不同，只有为最近显示的子文件指定的键有效。按功能键会影响子文件包括的光标，如果光标不位于子文件中，功能键影响上级子文件。

5.SFLFOLD 在同一个子文件控制记录格式中可以被指定为 SFLDROP。如果两个键字都活动，使用 SFLFOLD 键字，显示子文件时检查指示器。两个键字必须使用同一个键。

可选指示器对此键字有效。

图 3-186 给出如何规定 SFLFOLD 键字。

....+....	1....+....	2....+....	3....+....	4....+....	5....+....	6....+....	7....+....	8
00010A	R SFLR	SFL						
00020A*								
00030A*	(子文件记录在屏幕的一行放不下)							
00040A	R SFLCTRL	SFLCTL(SFLR)						
00050A		SFLPAG(17)						
00060A		SFLSIZ(34)						
00070A		SFLDSP SFLDSPCTL						
00080A		SFLFOLD(CF03)						
A								

图 3-186 规定 SFLFOLD 键字

显示子文件时，工作站用户可以按 CF03 键来把子文件在折叠方式和截断方式之间转换。

### 3.3.137 SFLINZ (子文件初始化)

这是记录层键字，用在子文件控制记录格式中，在对子文件控制记录格式作输出操作时，OS/400 要将子文件中的所有记录初始化。子文件的字符字段被初始化为空格，浮点字段被初始化为空，其它类型的数字字段被初始化成 0，或者被初始化为 DFT 键字中为仅输入字段指定的常数值。

当显示子文件时（对子文件控制记录做输出操作），在子文件的所有记录用同一个值显示。任何以前写的记录都被覆盖，不再有原来的值。

此键字无参数。

当 SFLINZ 对子文件控制记录格式所作的输出操作有效时，下面情况是正确的：如果在子文件记录格式中的字段上指定某些键字（如 DSPATR(HI)），并且这些键字指定了可选指示器，那么也显示子文件，就好象所有可选指示器为 off 一样。注意，如果为可选指示器指定 N (否) 的话，也可选择一个键字。

在应用程序对带有效 SFLINZ 键字的子文件控制记录格式发送输出操作后，子文件中所有的记录被认为是活动的，但都没有修改。只有当工作站用户更改它们或者程序向有有效 SFLNXTCHG 键字的子文件控制记录格式发送输出操作时，它们才认为是被修改了。

为了初始化不带活动记录的子文件，请看 3.3.148 “SFLRNA”。

一般来说，使用 SFLINZ 为了下列目的：

- 和 SFLRNA 一起指定 SFLINZ，以便应用程序能初始化子文件，然后向没有初始化记录活动的子文件中增加记录。
- 与 SFLPGMQ 一起指定 SFLINZ，以便应用程序能建立有简单输出操作的信息子文件，详细内容请看 3.3.146 “SFLPGMQ”。

注：

1. 如果在子文件记录格式中使用了字段选择，SFLINZ 无效。应用程序只能通过对子文件记录格式的几个输出操作选择必要的字段来初始化子文件。

2. SFLINZ 不能对信息子文件控制记录格式指定。(请看 3.3.143 “SFLMSGRCD”)。除非 SFLPGMQ 键字也在同一个子文件控制记录格式的字段层指定。

可选指示器对此键字有效，显示尺寸条件名无效。

图 3-187 给出如何规定 SFLINZ 键字。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A R SFLR SFL
00020A*
00030A* (至少一个显示的字段)
00040A R SFLCTRL SFLCTL(SFLR)
00050A SFLPAG(17)
00060A SFLSIZ(17)
00070A 01 SFLDSP SFLDSPCTL
00080A 02 SFLLINZ
00090A UNLOCK(*ERASE *MDTOFF)
A

图 3-187 规定 SFLINZ 键字

### 3.3.138 SFLLIN (子文件行)

这是记录层键字，用在子文件控制记录格式中来指定子文件被显示成一个横向子文件(记录显示有多列)。

键字格式为：

SFLLIN (空格)

参数值指定记录列之间的空格数(包括属性字符)。

例如，指定 SFLLIN 键字使一个有四个记录的子文件显示为：

REC1	REC3
REC2	REC4

如果不指定 SFLLIN，显示为：

REC1  
REC2  
REC3  
REC4

如果子文件记录格式包含字段选择，此键字无效。

为了对辅显示尺寸使用 SFLLIN，为每个辅显示尺寸指定有显示尺寸条件名的 SFLLIN 键字。

因为 SFLPAG 键字指定了一次可以显示的子文件记录数，在指定 SFLPAG 键字时必须考虑 SFLLIN 键字。

SFLLIN 对信息子文件无效。

可选指示器对这个键字无效。

图 3-188 给出如何规定 SFLLIN 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R SFLR          SFL
A*
A*          (至少一个显示的字段)
A*
00040A      R SFLCTRL       SFLCTL (SFLR)
00050A          SFLPAG(17)
00060A          SFLSIZ(17)
00070A          SFLDSP SFLDSPCTL
00080A          SFLLIN(5)
A
```

图 3-188 规定 SFLLIN 键字

### 3.3.139 SFLMLTCHC (子文件多选项选择列表)

这是记录层键字，用来定义子文件是一个多选项选择列表，多选项选择列表是一个用户可以从中选择多个选项的可翻卷选项组。

键字格式为：

SFLMLTCHC[(&选择的号码)[\*NORSTCSR | \*RSTCSR][\*NOSLTIND | \*SLTIND]]

参数可选并可以以任何顺序输入。

&选择的号码参数让应用程序找到在多选项选择列表中选择的选项号，此参数必须是个隐藏字段，其长度为 4，数据类型为 Y，无小数位。

参数\*RSTCSR 指定是否允许用箭头键把选择光标移到选择列表外。\*RSTCSR 指定箭头键无法把选择光标移到选择字段外边。\*NORSTCSR 指定箭头键可以把选择光标移到字段外。如果 SFLMLTCHC 子文件控制记录被定义成下拉方式，缺省值是\*RSTCSR。否则，缺省值为\*NORSTCSR。

参数\*SLTIND 指定在图形显示屏上显示选择列表时，是否使用可选指示器。\*SLTIND 指定检测框在彩色图形显示中被用作可选指示器。\*NOSLTIND 指定在彩色图形显示中不用可选指示器，而用选择光标用来作选择，缺省为\*NOSLTIND。

有 SFLMLTCHC 键字的子文件必须：

- 仅有一个输出字段。
- 不能有可输入字段。
- 可以有隐藏字段。

这是个只对子文件控制记录格式有效的可选键字。

下列子文件控制记录键字不能与 SFLMLTCHC 键字一起指定：

SFLDROP

SFLFOLD

SFLSNGCHC

CHCAVAIL、CHCSLT 和 CHCUNAVAIL 键字指出在彩显中选择列表每项的颜色。

CHGAVAIL 指出列表中各项的颜色可用于选择。CHCSLT 指出已选项的颜色。CHCUNAVAIL 指出列表中各项的颜色不可用于选择。这些键字只有在用了 SFLSNGCHC 或 SFLMLTCHC 键字的子文件控制记录中才可以使用。

可选指示器对此键字无效。

图 3-189 给出如何规定 SFLMLTCHC 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R SFLRCD          SFL
A      CTLFLD      1Y 0H    SFLCHCCTL
A      F1          10A 0 6 10
A      R SFLTLRCD        SFLCTL (SFLRCD)
A                      SFLMLTCHC
A                      SFLPAG (5) SFLSIZ (&SFLSIZ)
A                      SFLDSP SFLDSPCTL
A                      ROLLUP (10)
A 10                  SFLEND (*SCRBAR)
A      F3          5S 0H    SFLSCROLL
A      F2          4S 0H    SFLRCDNBR (CURSOR *TOP)
A      SFLSIZ        5S 0P
A                      1 30' Panel Title'
A                      4 5' Select Multiple Items:'
```

图 3-189 规定 SFLMLTCHC 键字

在本例中，当使用一个连在支持不可编程工作站增强接口控制器上的图形显示站时，多选项选择列表看起来这样：

## PICTURE21

图 3-190 给出如何在彩显的列表中规定各项的颜色，可用的选项用红色显示，选择了的用蓝色，不可用的选项用黄色。CHCAVAIL、CHCUNAVAIL 和 CHCSLT 也可用于设置列表中各项的颜色。这些键字的详细说明请看这本书中设置显示属性的例子。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R  SFLRCD          SFL
A      CTLFLD      1Y 0H      SFLCHCCTL
A      F1          10A 0 6 10
A      R  SFLCTRLCD        SFLCTL(SFLRCD)
A                      SFLMLTCHC
A                      SFLPAG(5) SFLSIZ(&SFLSIZ)
A                      SFLDSP SFLDSPCTL
A                      ROLLUP(10)
A                      CHCAVAIL((*COLOR RED))
A                      CHCSLT((*COLOR BLU))
A                      CHCUNAVAIL((*COLOR YLW))
A  10                  SFLEND(*SCRBAR)
A      F3          5S 0H      SFLSCROLL
A      F2          4S 0H      SFLRCDBNR(CURSOR *TOP)
A      SFLSIZ        5S 0P
A                      1 30' Panel Title'
A                      4 5' Select Multiple Items:'
```

图 3-190 规定 SFLMLTCHC

### 3.3.140 SFLMODE (子文件方式)

这是记录层关键字，用在子文件控制记录格式中来指明了文件的输入是折叠方式还是截断方式。这个关键字与 SFLCSRNN 一起使用可确定子文件中光标的位置。

关键字格式为：

SFLMODE (&方式)

参数&方式是必需的，它指定一个隐藏字段名，在输入时，用它保存子文件方式，此字段必须在子文件控制记录格式中被定义成一个长度为 1，处理方式为 H 的字符型字段。如果子文件为折叠方式，字段值为 0；如果为截断方式，字段值为 1。

如果在子文件控制记录格式中没指定 SFLDROP 或 SFLFOLD 键字，返回的方式值为 0。可选指示器对这个关键字无效。

图 3-191 给出如何规定 SFLMODE 和 SFLCSRNN 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A      R SFL01          SFL
A      FLD2A           2A  I  3  2
A      FLD2B           30A 0  3  5      A       FLD2C           6A 0  4 18
A      R CTL01          SFLCTL(SFL01)
A                  SFLSIZ(25)
A                  SFLPAG(4)
A                  SFLDSP
A                  SFLEND
A                  SFLCSRNN(&RELRCD)
A                  SFLMODE(&MODE)
A 10    SFLDROP(CF03)
A 11    SFLFOLD(CF03)
A                  RTNCSRLOC(&CSRRCD &CSRFLD)
A      RELRCD          5S 0H
A      MODE            1A H
A      CSRFLD          10A H
A      CSRRCD          10A H
A
A      R SFL02          SFL
A      FLD2A           2A  I 13  2
A      FLD2B           30A 0 13  5
A      FLD2C           6A 0 14 18
A      R CTL02          SFLCTL(SFL02)
A                  SFLSIZ(25)
A                  SFLPAG(4)
A                  SFLDSP
```

A		SFLEND
A		SFLCSRRRN (&RELRCD)
A		SFLMODE (&MODE)
A 12		SFLDROP (CF03)
A 13		SFLFOLD (CF03)
A		RTNCSRLOC (&CSRRC&CSRFLD)
A	RELRCD	5S 0H
A	MODE	1A H
A	CSRFLD	10A H
A	CSRRC&	10A H
A		

图 1-191 规定 SFLMODE 和 SFLCSRRRN 键字

指示器 10 和 13 为 on。两个子文件中各自加了两条记录。两个子文件都被显示。

下面的表显示了当光标位于指定位置时，返给 CTL02 的值：

行	列	RELRCD	MODE	CSRRC&	CSRFLD
13	2	1	0	SFL02	FLD2A
14	18	1	0	SFL02	FLD2C
15	2	2	0	SFL02	FLD2A
15	62	1	0	SFL02	空 白
17	2	0	0	SFL02	空 白
24	2	0	0	空 白	空 白
3	2	0	0	SFL01	FLD2A

下面的表显示了当光标位于指定位置时，返给 CTL01 的值：

行	列	RELRCD	MODE	CSRRC&	CSRFLD
3	2	1	1	SFL01	FLD2A
4	18	2	1	SFL01	FLD2B
5	18	0	1	SFL01	空 白
13	2	0	1	SFL02	FLD2A

### 3.3.141 SFLMSG (子文件信息) 和 SFLMSGID (子文件信息标识)

这是记录层键字，用在子文件控制记录格式中，在应用程序对子文件控制记录格式作输出操作时，用它来标识要在信息行上显示的信息。程序应有能力处理字段的反映像，并能把光标放到显示子文件中的恰当位置上。

键字格式为：

SFLMSG ('信息正文' [应答指示器])

SFLMSGID (信息标识 [库名/]信息文件名 [应答指示器] [&信息数据])

键字 SFLMSG：象规定 ERRMSG 一样指定 SFLMG。参数是信息正文（必须的）和应

答指示器（可选的）。信息正文是要显示的信息。

如果指定了应答指示器，它应当与在条件 SFLMSG 中所用的可选指示器相同。在显示错误信息后的输入操作中，OS/400 关闭指示器。如果应答指示器与可选的指示器是相同的，都被关闭。例外的情况是，如果应答指示器也被其它键字（如 CHANGE、CAnn、CFnn）指定，此时，应答指示器的开/关状态取决于 CHANGE 或 CFnn 键字提供的功能结果。

指定了应答指示器时，信息正文的开头 50 个字符也作指示器说明用，各自的应答指示器说明对 SFLMSG 无效。

键字 SFLMSGID：象规定 ERRMSGID 一样指定 SFLMSGID。

SFLMSGID 的参数指定了：

- 要显示的信息标识
- 信息文件，库（可选）
- 应答指示器（可选）
- 信息数据字段名（可选）

如果指定了应答指示器，应与 SFLMSGID 所用的可选指示器相同。在其后的输出操作中，在显示错误信息后，OS/400 关闭指示器，但如果应答指示器也被其它键字（CHANGE、CAnn、CFnn）指定，那么应答指示器的开/关状态取决于 CHAGE、CAnn、CFnn 提供功能的结果。

注：不能为 SFLMSGID 指定指示器说明。

如果指定信息数据字段，它保存对规定信息的替代文本。字段必须在记录格式中存在，它定义为一个处理方式为 P 的字符型（数据类型为 A）字段。

关于替代文本如何工作，请看 CL 参考手册中的 SNDPGMMSG 命令。

#### 3.3.141.1 在信息显示期间发生的情况

用 SFLMSG 和 SFLMSGID 的信息显示类似于 OS/400 在检测到字段有错误时的信息，它与 ERRMSG 和 ERRMSGID 的重要差别在于，是程序而不是 OS/400，把光标放到子文件中适当的字段位置，并将子文件中的字段反映像，需要的话也可一次反映像多个字段。在 5250 工作站，也允许闪烁光标和高亮度信息。

注：对于要处理的 SFLMSG 和 SFLMSGID，SFLDSP 键字必须有效。

#### 3.3.141.2 反映像字段的恢复

请看 3.3.50 中的 ERRMSG 和 ERRMSGID。

#### 3.3.141.3 选择键字的优先级

可以为一个子文件控制记录格式多次指定 SFLMSG 或 SFLMSGID。在程序中，可以设置可选指示器来选择要显示的具体信息和以反映像方式显示某个字段。当再次显示子文件时，可以在子文件的不同记录中以反映像方式显示几个字段。然而，一次只能显示一条信息。

如果一次选择多个错误信息，OS/400 显示下面的第一条，即如果选择了多个 ERRMSG、ERRMSGID、SFLMSG、SFLMSGID 信息字段，那么显示被选择的第一个。

在错误子文件允许有多个子文件信息（SFLMSG 和 SFLMSGID）。

可选指示器对这两个关键字有效。

图 3-192 给出如何规定 SFLMSG 和 SFLMSGID 键字

. . . + . . . 1 . . . + . . . 2 . . . + . . . 3 . . . + . . . 4 . . . + . . . 5 . . . + . . . 6 . . . + . . . 7 . . . + . . . 8		
00010A	R SFLR	SFL
00020A*		
00030A*	(至少一个显示的字段)	
00040A*		
00050A	R SFLCTL R	SFLCTL (SFLR)
00060A		SFLPAG (17)
00070A		SFLSIZ (17)
00080A		SFLDSP SFLDSPCTL
00090A 11		SFLMSGID (USR0006 PAYROLL/UMSGF1 + 11 &RPLTXT)
00100A		
00110A 12		SFLMSGID (USR0007 PAYROLL/UMSGF1 + 12 &RPLTXT)
00120A		
00130A	RPLTXT	78A P
	A	

图 3-192 规定 SFLMSG 和 SFLMSGID 键字

### 3.3.142 SFLMSGKEY (子文件信息键)

这是字段层关键字。用在信息子文件记录格式的第一个字段中(指定 SFL 和 SFLMSGRCRD)。它对子文件控制记录格式无效。为了从程序信息队列中选择信息来显示，应用程序要在此字段中放置一个信息引用键。程序也把程序信息队列名放到子文件记录格式的第二个字段中。请看 3.3.146 的“SFLPGMQ”。

此关键字无参数。

这个字段应预先被定义成长度为 4 的字符型隐藏字段。下列规则是适用的：

- 这个字段必须是定义在子文件记录格式中的第一个字段。
- 字段名和 SFLMSGKEY 只能是为这个字段指定的 DDS。

可选指示器对此关键字及有关字段无效。

更多的建立和显示信息子文件的信息，请看“SFLPGMQ”。

图 3-193 给出如何规定 SFLMSGKEY 键字

. . . + . . . 1 . . . + . . . 2 . . . + . . . 3 . . . + . . . 4 . . . + . . . 5 . . . + . . . 6 . . . + . . . 7 . . . + . . . 8		
00010A	R RCDMSG	SFL SFLMSGRCRD (3)
00020A	FLDKEY	SFLMSGKEY
00030A	FLDPGM	SFLPGMQ
00040A	R SFLCTL	SFLCTL (RCDMSG)
00050A 01		SFLINZ
00060A		SFLPAG (17)
00070A		SFLSIZ (17)

00080A		SFLDSP SFLDSPCTL
00090A	FLDPM	SFLPGMQ
	A	

图 3-193 规定 SFLMSGKEY 键字

### 3.3.143 SFLMSGRCD (子文件信息记录)

这是记录层键字，用在子文件记录格式中来指出子文件是信息子文件，在显示子文件时显示的记录是来自程序信息队列的信息。

键字格式为：

**SFLMSGRCD** (行号)

参数值给出信息显示屏的第一行位置，指定的值必须不大于所用显示尺寸的最大行号。显示的信息数依赖于为子文件指定的 SFLPAG 值。

建立和显示信息子文件的详细信息，请看 3.3.146 的“SFLPGMQ”。

在 SFLMSGRCD 的记录层 TEXT 键字有效。

对信息子文件，在子文件记录格式中，要预先定义两个字段：

- 信息标识字段，4 位长字符型的隐藏字段。应用程序用这个字段来向 OS/400 传递信息标识。这个字段必须是在信息子文件中定义的第一个字段，对这个字段必须指定 SFLMSGKEY。
- 程序队列名，10 位长字符型隐藏字段。程序传递保存字段中信息的程序信息队列名，它必须被指定为子文件信息记录格式的第二个字段，并且紧接着第一个字段。如果它在子文件控制记录指定，它可以在记录规范的任何地方。必须对此字段指定 SFLPGMQ，详细内容请看 3.3.146 的“SFLPGMQ”。

可为 SFLMSGRCD 指定显示尺寸条件名。如果基于显示尺寸的第一条显示信息的行号改变了，那么显示尺寸条件名是必需的。

如果应用程序不对信息子文件作输入操作，数据不返回到输入缓冲区。

信息可显示如下：

- 一行显示一条信息。如果信息比一行长，要截断。
- 每条信息从第 2 列开始，对 24×80 显示尺寸最大信息长度是 76 个字符，对于 27×132 显示尺寸最大信息长度是 128 个字符。
- 每条信息用高亮度属性显示。

当信息子文件由 OS/400 翻页时，光标在按 Page 键时光标所在的位置上。

支持这些信息的帮助，工作站用户把光标放在某个信息行上，而后按 Help 键，就可以得到对这个信息的帮助。

注：

1. 指定 SFLMSGRCD 后，不能指定没有 SFLPGMQ 的 SFLINZ。
2. 信息子文件，在由 SFLMSGRCD 指定的行上开始，必须不与子文件控制记录中的任何可显示字段重叠，即使这些可显示字段指定了可选指示器。

可选指示器对此键字无效，显示尺寸条件名有效。

图 3-194 给出如何规定 SFLMSGRCD 键字

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A R RCDMSG SFL SFLMSGRCD(3)
00040A FLDKEY SFLMSGKEY
00050A FLDPGM SFLPGMQ
00060A R SFLCTL SFLCTL(RCDMSG)
00070A SFLPAG(17)
00080A SFLSIZ(17)
00090A SFLDSP SFLDSPCTL
A

图 3-194 规定 SFLMSGRCD 键字

对一个信息子文件的子文件记录格式，需要高亮度键字。必须按图示顺序指定 SFLMSGKEY 和 SFLPGMQ。

### 3.3.142 SFLNXTCHG (子文件下一个修改)

这是记录层键字，用在子文件记录格式中，强制工作站用户改正被程序读入的子文件记录中的程序检测到的键盘输入错误。由修改记录来做这些事。这样，就必须用一个取得下一次修改操作来读记录，下面各节给出详细解释。

此键字无参数。

#### 3.3.144.1 有 SFLNXTCHG 的子文件操作

一个典型的使用 SFLNEXCHG 的例子如下：

一个工作站用户修改显示子文件中的某些记录（这可能是数据输入或数据更新），修改完某些记录后，工作站用户按 Enter 键，程序只读有取得下一次修改操作的修改了的记录。（例如，RPG 中的 READC 和 COBOL 中的 READ-SUPFILE-NEXT-MODIFIED）。

如果程序在修改了的记录中检查到键入错误，它可以向有错误的子文件记录发送修改操作（RPG 的 UPDATE，COBOL 中的 REWRITE SUBFILE），设置指示器以便在修改操作期间 SFLNXTCHG 有效。这些修改操作送往子文件记录格式。

所有有错误的记录被修改完后，程序向子文件控制记录格式发送输出/输入操作来重新显示文件。

子文件重新显示后，用户可以重新键入数据，而后按 Enter 键。如果数据正确，程序不再显示子文件。

错误的记录（和其它任何被工作站用户修改的记录）在下一次的取得下一次修改操作中被返回给程序，这是因为 SFLNXTCHG 导致子文件记录被认为是修改过的，即使工作站用户并没修改它们。这允许程序禁止工作站用户忽略子文件记录中的程序检测键入错误。

#### 3.3.144.2 没有 SFLNXTCHG 的子文件操作

如果没指定 SFLNXTCHG，或者虽然指定了但在对子文件记录的修改操作中没被选择，那么工作站用户可以只按 Enter 键而不更正程序检测到的错误。程序读不到记录，因为在第二次按 Enter 键时，取得下一次修改操作找不到更改了的记录。

可选指示器对这个键字有效。

不能与 SFLMSGRCD 一起指定 SFLNXTCHG。

图 3-195 给出如何规定 SFLNXTCHG 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R SFLR          SFL
00020A  14           SFLNXTCHG
A*
A*      (至少规定一个可输入的字段)
00040A      R SFLCTRL       SFLCTL (SFLR)
00050A          SFLPAG (17)
00060A          SFLSIZ (17)
00070A          SFLDSP  SFLDSPCTL
A
```

图 3-195 规定 SFLNXTCHG 键字

### 3.3.145 **SFLPAG** (子文件页)

这是记录层键字，用在子文件控制记录格式中来指定一次显示的子文件记录数。(例外情况，请看 3.3.145.1 “字段选择”)。

键字格式为：

**SFLPAG** (要显示的记录数)

SFLPAG 参数值和每个子文件记录所需要的行数决定显示一页记录所需要的实际行数。并非子文件中所有的记录必须一次显示出来，也并非要屏幕中的所有行用来显示一页子文件记录。

这个键字对子文件控制记录格式是**必需的**。

#### 3.3.145.1 子文件页尺寸等于子文件尺寸

当 SFLPAG 和 SFLSIZ 的参数值相同时，在子文件中包含的最大记录数等于在屏幕上一次能显示的最大子文件记录数。这时，在按 Page Up 或 Page Down 时，OS/400 不自动对子文件翻页。如果指定 ROLLUP 和 ROLLDOWN 键字并且按了其中某一个，OS/400 把控制返给应用程序。如果没指定 ROLLUP 和 ROLLDOWN，会送给工作站用户一个信息，指出在显示中不支持某个键。

如果子文件尺寸等于子文件页尺寸，下列键字**不允许**：

**SFLDROP**      **SFLFOLD**      **SFLROLVAL**

当使用多个显示尺寸时（指定 DSPSIZ 键字），这些键字只对那些子文件尺寸等于子文件页尺寸的显示尺寸无效。

字段选择：当子文件页尺寸等于子文件尺寸时，可以为子文件记录格式中的字段指定可选指示器，这叫做字段选择。当在子文件记录中使用字段选择时，SFLPAG（值）指定可用于显示子文件记录的显示行数。（没有字段选择，SFLPAG（值）指定一次可被显示的子文件记录数）。当子文件记录显示多于一行时，必须考虑这点，SFLPAG 的值必须大于或等于子文件显示行数。

如果子文件记录格式中有字段选择，下列关键字对子文件控制记录格式无效：

SFLDROP        SFLFOLD        SFLINZ        SFLLIN  
SFLRCDNBR SFLRNA (因为 SFLINZ 无效) SFLROLVAL

### 3.3.145.2 子文件页尺寸不等于子文件尺寸

当 SFLPAG 和 SFLSIZ 指定不同参数值时，OS/400 认识 Page Up 和 Page Down 键并自动按在 SFLROLVAL 键字规定的字段值对子文件进行翻页，如果不指定 SFLROLVAL 键字，OS/400 按键字 SFLPAG 指定的参数值翻页，除非子文件使用 SFLDROP 键字。如果使用了 SFLDROP 键字，当记录用截断方式显示时，显示多个记录，这比 SFLPAG 值要多，对于截断的记录，OS/400 按照以截断方式显示的记录数来翻页。

可选指示器对此键字无效，显示尺寸条件名有效。如果想要改变基于显示尺寸的一次被显示的记录数，显示尺寸条件名是必需的。

图 3-196 给出如何规定 SFLPAGE 键字

00010A	R SFLR	SFL
A*		
A*	(至少一个显示的字段)	
A*		
00040A	R SFLCLTR	SFLCTL (SFLR)
00050A		SFLPAG (17)
00060A		SFLSIZ (17)
00070A		SFLDSP SFLDSPCTL
A		

图 3-196 规定 SFLPAGE 键字

因为 SFLPAG 和 SFLSIZ 指定的值相同 (17)，子文件页尺寸等子文件尺寸。

### 3.3.146 SFLPGMQ (子文件程序信息队列)

这是字段层键字，用在信息子文件的子文件记录格式的第二个（或最后）字段，字段包含由 OS/400 建立信息子文件的程序信息队列名。另外，当在子文件控制记录格式中指定 SFLINZ 键字时，SFLPGMQ 可以在子文件控制记录格中指定。

关键字格式：

SFLPGMQ ([10] | [276])

当指定 10 时, SFLPGMQ 生成一个 10 字节字段, 10 是缺省值。当指定 276 时, SFLPGMQ 生成一个 276 字节字段。

字段要预先定义成字符型的隐藏字段, 要符合下列规则:

- 字段名和 SFLPGMQ 键字及参数是为这个字段指定的唯一 DDS。
- 如果在执行时放在字段中的程序信息队列名长度小于字段长度 (10 或 276 字节), 它必须左对齐并且用空格填充。

对于使用 276 字节参数值的集成语言环境 (ILE\*) 程序, 字段数据的格式必须如下:

- 开始的 256 字节保留 ILE 调用信息队列名。调用信息队列名与 ILE 过程名相同。名字必须左对齐空格填充。
- 第 257 到 266 字节是可选的, 它保存 ILE 模块名, 当指定模块名时, 它必须左对齐空格填充。如果没提供模块名, 这些字节为空格。
- 字节 267 到 276 是可选的, 它保存 ILE 连接程序名。规定名字时, 必须左齐空格填充。如果没有连接程序名, 这些字节置空格。

注:

1.如果 SFLPGMQ 使用参数值 10, 但在处理时比 10 字节长的 ILE 过程名放入字段中, 那么过程名被截断成 10 字节, 这样结果不可预测。

2.如果 SFLPGMQ 使用参数值 276, 程序信息队列名在处理时到字段中, 257 到 276 字节必须被置为空格。如果这些字节非空格, 系统认为已经给了一个调用信息队列, 而不再寻找指定的程序信息队列。

如果 SFLPGMQ 既在子文件又在子文件控制记录中指定, SFLPGMQ 的参数值必须匹配。然而, 在同一文件中的不同子文件可以使用不同的 SFLPGMQ 参数值。

通过对子文件记录格式的多次输出操作一次一个信息地建立子文件, 这个字段对子文件记录格式 (SFL 键字指定) 是必需的。

也可以在子文件控制记录格式中指定这个字段, 通过一个对子文件控制记录格式的输出操作可一次建立起整个子文件。指定可选指示器可控制子文件建立的方式。

### 3.3.146.1 多次输出操作

如果在子文件记录上指定了字段名和 SFLPGMQ, 用对子文件记录格式的每次输出操作, 一次可以建立子文件的一条信息。对每次输出操作, 信息引用键必须在记录的第一个字段 (SFLMSGKEY 键字), 程序信息队列名必须在第二个字段, 在输出操作时, OS/400 从信息队列中取出标识信息, 而后把它作为一条记录放到子文件中。

注: CL 程序不能用于多次输出操作, 每次建立信息时所需要的相对记录号不支持 CL。

### 3.3.146.2 单次输出操作

如果在子文件控制记录格式上指定了 SFLPGMQ (带有其命名字段) 和 SFLINZ 键字, 那么直接对子文件控制记录格式进行一次输出操作就可以建立整个子文件。在输出操作中, OS/400 用在 SFLPGMQ 中指定名的程序信息队列中的所有信息来初始化子文件。如果必要的话, OS/400 扩充子文件来包含队列中的所有信息。要做到这点, 必须在子文件记录格式

(SFL 键字) 中指定 SFLMSGRCD、SFLMSGKEY 及 SFLPGMQ 等键字。此功能忽略 SFLPGMQ 和 SFLMSGKEY，应用程序不必为它们的字段设值。

### 3.3.146.3 特殊值

SFLPGMQ 字段可以包含\*（星号）代替程序队列名。如果程序把星号放到 SFLPGMQ 字段中，OS/400 使用发送输出操作的程序所用的信息队列。如果程序是 CL 程序，不可使用星号。

### 3.3.146.4 多次和单次都有的输出操作

如果既对子文件记录格式又对子文件控制记录格式都指定 SFLPGMQ 键字，可以这一次使用单次输出操作功能，而另一次使用多次输出功能。通过在发送输出操作前设置指示器就可这样做。但在为子文件的某显示作准备时，对子文件的所有操作必须是固定的（多次或单次，不许混合使用）。

可选指示器和显示尺寸条件名对此键字无效。

图 3-197 给出如何规定 SFLPGMQ 键字

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
00010A	R RCDMSGILE	SFL SFLMSGRCD(3)
00020A	FLDKEY	SFLMSGKEY
00030A	FLDPGM	SFLPGMQ(276)
00040A	R SFLCTLILE	SFLCTL(RCDMSG)
00050A 01		SFLINZ
00060A		SFLPAG(17)
00070A		SFLSIZ(17)
00080A		SFLDSP SFLDSPCTL
A	:	
A	:	
00110A	FLDPGM	SFLPGMQ(276)
A	R RCDMSGOPM	SFL SFLMSGRCD(3)
A	FLDKEY	SFLMSGKEY
A	FLDPGM	SFLPGMQ
A	R SFLCTLOPM	SFLCTL(RCDMSG)
A 02		SFLINZ
A		SFLPAG(17)
A		SFLSIZ(17)
A		SFLDSP SFLDSPCTL
A	:	
A	:	
A	FLDPGM	SFLPGMQ(10)
A		

图 1-197 规定 SFLPGMQ 键字

在图 3-197 中，程序可以用对子文件控制记录格式的单次输出操作（指示器 01 为 on），或用对子文件记录格式的多次输出操作（指示器 01 为 off）来建立子文件。

在第一个记录格式中，子文件程序队列名列是 276 字节长，而在第三个记录格式中子文件程序队列名只能是 10 字节长。

### 3.3.147 SFLRCDNBR (子文件记录号)

这是字段层关键字，用在子文件控制记录格式中，指定要显示的子文件页是由这个字段中的相对记录号的记录所在的页。如果不指定此关键字，OS/400 根据缺省值显示子文件的第一页。

关键字格式为：

SFLRCDNBR[ ([CURSOR][\*TOP]) ]

如果指定 CURSOR，光标被放在由这个字段中的相对记录号所对应的那条子文件记录上，光标位于子文件记录中第一个可输入字段处。如果没有可输入字段，光标位于第一个仅输出或常数字段。例如，如果子文件中有 9 条记录，一页只能有 3 条记录，如果 SFLRCDNBR 字段值为 8，那么，即显示有第 7、8、9 条记录的那页。如果指定 CURSOR，光标在第 8 条记录上。如果指定\*TOP，子文件记录的相对记录号是由这个字段内容确定的，它要做为子文件显示页的第一条记录来显示。

这个字段必须是 0 位小数的区位十进制字段，它必须是有符号数的键盘转换属性 (35 列为 S)，它是 4 位长，必须定义成一个仅输出、输出/输入或隐藏字段。在对子文件控制记录格式的输出操作中，此字段的值小于 1 或大于在这个字段的子文件记录号，就要有错误返回给程序。

这是个只对子文件控制记录格式有效的可选关键字。

不能对同一个字段同时规定 SFLROLVAL 和 SFLRCDNBR。

如果子文件记录格式有字段选择，不允许有此关键字。

可选指示器对此关键字无效。

图 3-198 给出如何规定 SFLRCDNBR 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R SFLR          SFL  
          A*  
          A*          (至少一个显示的字段)  
          A*  
00040A      R SFLCTRL       SFLCTL (SFLR)  
00050A          SFLPAG (17)  
00060A          SFLSIZ (17)  
00070A          SFLDSP SFLDSPCTL  
00080A      DSPREC        4S 0B 4 12SFLRCDNBR (CURSOR)  
          A
```

图 3-198 规定 SFLRCDNBR 键字

在图 3-198 中，在显示子文件前，程序或工作站用户可以为此字段指定 SFLRCDNBR 设值。

### 3.3.148 SFLRNA (子文件非活动记录)

这是记录层键字，用在子文件控制记录格式中，与 SFLINZ 键字一起使用，使应用程序可以用非活动记录初始化一个子文件。为此，程序向选择了 SFLINZ 键字的子文件控制记录格式发送输出操作。子文件本身被激活，子文件记录不认为是活动的，除非发生下列情况之一：

- 应用程序对子文件记录格式发送一个输出操作，把数据放到子文件的一个记录中。  
子文件记录成为活动的但不认为修改了，除非 SFLNXTCHG 键字也有效。
- 在应用程序显示子文件后，工作站用户把数据键入到子文件记录中，被键入的记录成为活动的并被修改。

此键字无参数。

一般的，是用 SFLRNA 使程序能在显示子文件前向其写入一些记录，然后工作站用户能向子文件中加记录。

当程序显示有有效的 SFLINZ 和 SFLRNA 键字已经初始化的子文件时，在非活动记录中字段有如下值：

- 字符型字段为空格
- 数字型字段为零
- 有常数值的仅输入字段具有常数值

程序不能对非活动子文件记录发送输入操作。对子文件的一个记录发送的取得下一次修改 (get-next-changed) 操作，只有在记录已经成为活动并被修改时，才把记录返回。

程序不能对活动记录（没指定 SFLRNA）发送输出操作。它必须发送更新操作，程序也不能对非活动记录（指定 SFLRNA）发送更新操作，必须发送输出操作。

当指定 SFLRNA 时，必须也指定 SFLINZ 键字。

SFLRNA 不能对信息子文件（在子文件记录格式中用 SFLMSGRCD 标识）指定。

如果子文件记录格式中有字段选择，SFLRNA 无效。

可选指示器对此键字无效。

### 图 3-199 给出如何规定 SFLRNA 键字

00010A	R SFLR	SFL
A*		
A*	(至少一个显示的字段)	
A*		
00040A	R SFLCTRL	SFLCTL(SFLR)
00050A		SFLPAG(17)
00060A		SFLSIZ(17)

00070A	SFLDSP SFLDSPCTL
00080A	SFLINZ
00090A	SFLRNA
A	

图 3-199 规定 SFLRNA 键字

### 3.3.149 SFLROLVAL (子文件翻卷值)

这是字段层键字，用在子文件控制记录格式中，工作站用户可以向这个字段中键入一个值来告诉 OS/400 程序，当按翻页键时翻多少条记录。

此字段必须是有符号数的键盘转换属性，4 位长，无小数位，是输出/输入或仅输入字段。此键字无参数。

工作站用户可以先键入每次要翻的记录数而后按 Page Up 或 Page Down 键来上翻或下翻显示的数据。(在下一页，SFLROLVAL 值保持不变，除非在翻页前键入新数据)。如果键入负数或零后按翻页键，工作站上显示错误。

此键字只对子文件控制记录格式有效。如果要 OS/400 支持按记录翻页功能，必须指定此键字。

如果没指定此键字，除非子文件使用 SFLDROP 键字，否则 OS/400 按照 SFLPAG 值翻页来显示子文件。如果使用 SFLDROP，在以截断方式显示记录时，要显示比 SFLPAG 值多的记录。对于截断的记录，OS/400 按照以截断格式被显示的记录数来翻页。

如果子文件尺寸等于子文件页尺寸，忽略 SFLROLVAL。当使用多个显示尺寸时(指定 DSPSIZ)，SFLROLVAL 只对那些子文件尺寸等于子文件页尺寸的显示尺寸忽略。如果子文件记录格式有字段选择，SFLROLVAL 无效。

此字段作为对子文件控制记录输入的一部分返给应用程序。

如果按 Page Up 键，页超出子文件的第一页记录，发生下列情况之一：

- 如果第一页记录现在没显示，按 Page Up 将显示它。
- 如果第一页记录现在已显示，按 Page Up 会出现一个信息。

如果指定 Page Down 键，页超出子文件的最后一条活动记录，发生下列情况之一：

- 如果最后一整页记录没显示，按 Page Down 将显示它。
- 如果最后一整页记录已显示，按 Page Down 会出现一个信息。

例外情况是，当 SFLROLVAL 值小于 SFLPAG 值时，OS/400 重新对子文件翻页，不会产生信息。

指定 SFLROLVAL 时有些键字是有帮助的：

- 在显示最后的子文件记录时，SFLEND 会提示工作站用户。详细内容请看 3.3.134 的“SFLEND”。
- 按 Page Up 或 Page Down 键，如翻页超出子文件的末尾，PAGEUP 或 PAGEDOWN 键使得控制返给程序，没有 PAGEUP (ROLLDOWN) 或 PAGEDOWN (ROLLUP)，会显示一个信息。

下面举例说明 SFLROLVAL 的用法：

上翻页：假设 SFLPAG 指定的值为 3，在子文件中有 11 个活动记录。如果当前显示记录 8 到 10，而用户键入的值大于 7，按 Page Up 键显示 1 到 3 的记录。

如果当前显示记录 1 到 3，并且 SFLROLVAL 值大于 0，按 Page Up 键时或者信息送给工作站用户（没指定 PAGEUP），或者控制返给用户程序（指定 PAGEUP，程序响应上翻页）。

下翻页：假设 SFLPAG 指定的值为 3，在子文件中有 11 条活动记录。如果记录 8 到 10 是当前显示的记录，用户键入 3 到 SFLROLVAL 字段，按 Page Down 键在屏幕的最上面的页区显示记录 11。没被记录占用的那些行为空白。如果再按 Page Down 键，显示最后一整页子文件记录（记录 9 到 11）。最后，如果第三次按 Page Down 键，或者会有一条信息送给工作站用户（没指定 PAGEDOWN），或者控制返给用户程序（指定 PAGEDOWN，程序对下翻页作出响应）。

下面列出了翻页超出子文件末尾发生的情况（当 SFLROLVAL 值大于 SFLPAG 值时）：

按 Page Down 键：

—如果尚未显示最后一整页记录，那么显示它。

—如果最后一整页记录已显示，那么显示一个信息。

第二次按 Page Down 键：

—如果指定 Page Down，控制返给应用程序。

—如果未指定 Page Down，显示一条信息。

按 Page Up 键：

—如果未显示第一页子文件，那么显示它。

—如果已显示和一页子文件，那么：

—如果指定 PAGEUP，控制返给应用程序。

—如果未指定 PAGEUP，显示一条信息。

不能对同一个字段同时指定 SFLROLVAL 和 SFLRCDNBR。

可选指示器对此键字无效。

图 3-200 给出如何规定 SFCROLVAL 键字。

00010A	R SFLR	SFL
A*		
A*	(至少一个显示的字段)	
A*		
00040A	R SFLCTL R	SFLCTL (SFLR)
00050A		SFLPAG (17)
00060A		SFLSIZ (17)
00070A		SFLDSP SFLDSPCTL
00080A	ROLVAL	4S 0B 1 47SFLROLVAL
A		

图 3-200 规定 SFCROLVAL 键字

### 3.3.150 SFLRTNSEL (子文件返回选择的选项)

这是记录层关键字，用在一个选择列表上子文件控制记录中，用来控制选项怎样用取得下次修改操作返给应用程序。

此关键字无参数。

如果指定此关键字，那么必须指定 SFLMLTCHC 或 SFLSNGCHC。

指定此关键字使取得下次修改操作返回所有已选择的选项。这包括实际上没修改的缺省选项。比如用户从没选择的选项。

如果没指定此关键字，只有被修改的记录取得下次修改操作返给应用程序。这意味着缺省选择不被返回，因为选项没被用户修改。

可选指示器对此关键字无效。

图 3-201 给出如何规定 SFLRTNSEL 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R SFLRCD          SFL
A      CTLFLD      1Y 0H    SFLCHCCTL
A      F1          4A 0 6 10
A      R SFLCTRLRCD      SFLCTL (SFLRCD)
A                      SFLMLTCHC
A                      SFLRTNSEL
A                      SFLPAG (5) SFLSIZ (&SFLSIZ)
A                      SFLDSP SFLDSPCTL
A                      ROLLUP (10)
A 10                  SFLEND (*SCRBAR)
A      F3          5S 0H    SFLSCROLL
A      F2          4S 0H    SFLRCDNBR (CURSOR *TOP)
A      SFLSIZ      5S 0P
A                      1 30' Panel Title'
A                      4 5' Single selection list:'
A
```

图 3-201 规定 SFLRTNSEL 键字

在此例中，取得下次修改操作返回所有被选取的选项。这包括实际上没修改的缺省选项。比如，用户从没指定的选项。

### 3.3.151 SFLSCROLL (子文件上翻)

这是字段层关键字。用在子文件控制记录格式中，在控制被返回到应用程序时，返回在子文件顶部的记录的相对记录号。

此关键字无参数。

此字段必须是零位小数，有符号数字的键盘转换属性，5 位长的隐藏字段。

字段作为子文件控制记录输入的一部分返回给应用程序。如果控制是由按 Enter 键返给程序的，那么返回值将是当前显示子文件页中上数第一条记录的相对记录号。

当按 Page Up 或 Page Down 键超出子文件一页的范围时，如果指定 ROLLUP 或 ROLLDOWN，控制返给程序。如果没指定 ROLLUP 或 ROLLDOWN，显示一条信息。如果由于用 ROLLUP 键字使控制返给应用程序，那么返回值将是下一页子文件最顶部记录的相对记录号。如果由于用 ROLLDOWN 使控制返给的程序，那么返给相对记录号字段的值为 1。

当 SFLSIZ 等于 SFLPAG 时，不允许用 SFLSCROLL 键字。

这个键字只对子文件控制记录格式有效。

在使用翻卷条（指定 SFLEND(\*SCRBAR)）时，此键字是很有用的，当用户使用翻卷条时，有 SFLSCROLL 的隐藏字段保存用户要显示记录的相对记录号。控制仅当用户企图翻到没写过的子文件部分或者按了 Enter 键时，才返给应用程序。

另一个有用的键字是用\*TOP 做参数的 SFLRCDNBR。在你向子文件加记录后，用有效的 SFLRCDNBR 可重新显示子文件。对这个键字使用与 SFLSCROLL 键字返回值相同的值。

不能对同一个字段指定 SFLROLVAL、SFLSCROLL 和 SFLRCDVAR。

在子文件控制记录中只允许一个 SFLSCROLL。

可选指示器对此键字无效。

图 3-202 给出如何规定 SFLROLL 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R SFLRCD          SFL
A      CTLFLD      1Y 0H    SFLCHCCTL
A      F1          4A 0 6 10
A      R SFLCTRLRCD      SFLCTL (SFLRCD)
A                      SFLSNGCHC
A                      SFLPAG (5) SFLSIZ (&SFLSIZ)
A                      SFLDSP SFLDSPCTL
A                      ROLLUP (10)
A 10                  SFLEND (*SCRBAR)
A      F3          5S 0H    SFLSCROLL
A      F2          4S 0H    SFLRCDNBR (CURSOR *TOP)
A      SFLSIZ      5S 0P
A                      1 30' Panel Title'
A                      4 5' Multiple selection list:'
```

图 3-202 规定 SFLROLL 键字

本例中，当控制返给程序时，字段 F3 保存子文件顶部记录的相对记录号。

### 3.3.152 SFLSIZ (子文件尺寸)

这是记录层键字，用在子文件控制记录格式中来指定子文件中的记录数。允许的最大记录数为 9999，此键字对子文件控制记录格式是必需的。

键字格式为：

SFLSIZ (子文件中的记录数目！&子文件字段中的记录数)

子文件记录数参数可用两种方式规定：数字或程序到系统字段。程序到系统字段必须定义为长度为 5、数据类型为 S 的字段。

当用有\*SCRBAR 参数的 SFLEND 键字时，P—字段可用作子文件的尺寸。应用程序把加到子文件中的记录数传给 OS/400。因此，翻卷条可以把子文件显示的更好。

注：P—字段的值必须大于子文件页值。如 P 的值不大于子文件页值，子文件尺寸将是页值加 1。

### 3.3.152.1 子文件尺寸等于子文件页尺寸

当为 SFLSIZ 和 SFLPAG 指定相同参数值时，可以为子文件记录格式中的字段指定可选指示器。(这叫作字段选择)。

在子文件建立时，记录长度的变化依赖于被选择的字段，并且每个输出操作把记录放到子文件中的连续位置中。在显示子文件时，每条记录需要不同数的显示行。实际填入子文件的记录数依赖于写入子文件中的每条记录所作的字段选择。

如果写入子文件中的最后一条记录恰好填入子文件，状态信息 (CPF5003) 返回给应用程序。如果写入子文件的最后一条记录溢出子文件，注意信息 (CPF5043) 返回给应用程序。

如果显示的子文件记录数没有占满一整屏的话，SFLPAG 指定的值增加到显示屏上能显示的最大记录数。

用同一值增加 SFLSIZ 的值。例如，如果规定 SFLPAG(13)和 SFLSIZ(13)，并且子文件记录格式和 SFLLIN 值被指定为可以在一个显示行上填 3 条记录，则 SFLPAG 和 SFLSIZ 增加到 15。

可选指示器对此键字无效。如果子文件中记录数的改变依赖于显示尺寸，显示尺寸条件名是有效而且必需的。当用程序到系统字段作为此键字参数时，不能对其使用显示尺寸条件名。

图 3-203 给出如何规定 SFLSIZ 键字

00010A	R SFLR	SFL
00020A 21	FIELD1	78 2 2
00030A 22	FIELD2	40 3 2
00040A*		
00050A	R SFLCTRL	SFLCTL(SFLR)
00060A		SFLPAG(5)
00070A		SFLSIZ(5)
00080A		SFLDSP
00090A		SFLDSPCTL
A		

图 3-203 规定 SFLSIZ 键字

应用程序发送下列输出操作：

输出操作	可选指示器设置	结果
SFLR	21 on	只有 FIELD1 被写入子文件
	22 off	
SFLR	21 on	FIELD1 和 FIELD2 被写入子文件
	22 on	
SFLR	21 off	仅 FIELD2 被写入子文件
	22 on	
(OS/400 向应用程序发送状态信息 CPF5003)		
SFLCTRLR	不需指示器	显示子文件

结果显示如下：

PICTURE22

图 3-24 子文件显示

在图 3-24 中，第四个记录不能写到子文件中，因为没有空间来显示它（在 DDS 中指定了 SFLPAG(5)）。

### 3.3.152.2 子文件尺寸不等于子文件页尺寸

为 SFLPAG 和 SFLSIZ 指定不同的参数值时，SFLSIZ 值指定能被放到子文件中的记录数。如果程序放一个相对记录号大于 SFLSIZ 值的记录到子文件中，子文件自动扩展到能包括它（直到最大值 9999 条记录）。你所指定的参数值应当足够大，以适合一般情况下在子文件中包括记录的最大数。

### 3.3.153 SFLSNGCHC (子文件单选项选择列表)

这是记录层键字，用来定义一个子文件是一个单选项选择列表。单选项选择列表是用户可以从中选取一项的可翻卷的选项组。

键字格式为：

```
SFLSNGCHC[(*NORSTCSR ! *RSTCSR)[*NOSLTIND ! *SLTIND]
[*NOAUTOSLT ! *AUTOSLT ! *AUTOSLTENH])]
```

参数是可选的并可用任何顺序输入。

\*RSTCSR 参数指定是否允许用箭头键把选择光标移到字段的外边， \*RSTCSR 不允许， \*NORSTCSR 允许。如果 SFLSNGCHC 子文件控制记录被定义成一个下拉记录，缺省值为 \*RSTCSR。否则，缺省值为\*NORSTCSR。

\*SLTIND 参数指定，当在一个图形屏幕上显示选择列表时，是否使用选择指示，\*SLTIND 指定在彩色图形屏上用广播按钮作为选择指示， \*NOSLTIND 指定不用选择指示，只用选择光标来作选择，缺省值为\*NOSLTIND。

\*AUTOSLT 参数指示是否用 ENTER 键来自动选择当前选择光标所指定的选项。  
\*NOAUTOSLT 指明只有设备连接到增强控制器时，自动选择才有效。如果 SFLSNGCHC 子文件控制记录被定义为下拉记录，缺省值为\*AUTOSLT，否则，缺省值为\*NOAUTOSLT。

一个有 SFLSNGCHC 规定的子文件必须：

- 至少有一个仅输出字段
- 没有可输入字段
- 可有隐藏字段

这是个仅对子文件控制记录格式有效的可选键字。

下列子文件控制记录键字不能与 SFLSNGCHC 一起对同一个记录规定：

```
SFLDROP      SFLFOLD      SFLMLTCHC
```

当选择列表在彩显工作站上显示时，能用 CHCAVAIL, CHCSLT 和 CHCUNAVAIL 键字指出选择列表中各项的颜色。CHCAVAIL 指出列表中可用选项的颜色，CHCSLT 指出选中的选项颜色，CHCUNAVAIL 指出不可用的选项。如果也使用了 SFLSNGCHC 和 SFLMLTCHC 键字，这些键字只能用在子文件控制记录中。

可选指示器对此键字无效。

图 3-205 给出如何规定 SFLSNGCHC 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R SFLRCD          SFL
A      CTLFLD      1Y 0H      SFLCHCCTL
A      F1          10A 0 6 10
A      R SFLCTRLCD      SFLCTL(SFLRCD)
A                      SFLSNGCHC
A                      SFLPAG(5) SFLSIZ(&SFLSIZ)
A                      SFLDSP SFLDSPCTL
A                      ROLLUP(10)
A 10                  SFLEND(*SCRBAR)
A      F3          5S 0H      SFLSCROLL
A      F2          4S 0H      SFLRCDNBR(CURSOR *TOP)
A      SFLSIZ      5S 0P
A                      1 30' Panel Title'
```

A

4 5' Select One Item:'

图 3-205 规定 SFLSNGCHC 键字

此例中，当使用一个连在支持不可编程工作站增强接口控制器上的图形显示站时，单选项选择列表看起来象这样：

PICTURE23

图 3-206 给出如何在彩显上规定列表中各项的颜色，可用项显示为红色，已选择的项为兰色，不可用的项是黄色。也使用 CHCAVAIL，CHCUNAVAIL，CHCSLT 键字来设置列表中各项的显示属性。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A  
A R SFLRCD SFL  
A CTLFLD 1Y 0H SFLCHCCTL  
A F1 10A 0 6 10  
A R SFLCTRLCD SFLCTL(SFLRCD)  
A SFLSNGCHC  
A SFLPAG(5) SFLSIZ(&SFLSIZ)  
A SFLDSP SFLDSPCTL  
A ROLLUP(10)  
A CHCAVAIL((*COLOR RED))  
A CHCSLT((*COLOR BLU))  
A CHCUNAVAIL((*COLOR YLW))  
A 10 SFLEND(*SCRBAR)  
A F3 5S 0H SFLSCROLL  
A F2 4S 0H SFLRCDNBR(CURSOR *TOP)  
A SFLSIZ 5S 0P  
A 1 30' Panel Title'  
A 4 5' Select One Item:'
```

图 3-206 规定 SFLSNGCHC 键字

### 3.3.154 SLNO (开始行号)

这是记录层关键字，用来指定正定义的记录格式的开始行号。如果指定了的话，SLNO 要校正记录格式中每个字段的行号。如果不指定 SLNO，记录格式中的字段用 39 – 41 列中指定的行数来显示。

关键字格式为：

SLNO (n | \*VAR)

此关键字可指定两个参数中的某一个。

指定 n，n 是一个 1 – 27 间的值。记录格式中所有字段从指定它们的位置偏移 n – 1 行，如果指定 SLNO(1)，记录格式必须没有在第 1 行第 1 列开始的字段。

指定\*VAR，使应用程序运行时，在显示该记录格式前设置开始行号，文件生成时，OS/400 设置开始行号为 1。如果记录有一个在 1 行 1 列开始的字段，那么在文件生成时出现一个警告信息。如果程序不设置开始行号或设为零，OS/400 假定其值为 1，如果程序设了一个开始行号值，但它使记录格式中的第一个字段超出显示屏，或把它设为负值，OS/400 向应用程序发送一个注意信息 (CPF5002)，而记录不能显示。如果开始行号设置为 1 而记录格式中有一个开始在 1 行 1 列的字段，OS/400 向应用程序发送一个错误信息 (CPF5398)，记录不显示。

要计算字段的实际显示的行，从 39 – 41 列上指定的行号中减 1，然后加上开始行号。记录格式在由 SLNO 指定的行上开始，除非有一个字段在 1 行 1 列上，在此情况下，起始属性字节是在前一行的最后一个位置上，并且记录格式的开始行号是比 SLNO 指定的值小 1 的数。

当指定\*VAR 时，在记录中没有字段可以占据显示的最后一个位置。

如果 CLRL(nn) 或 CLRL(\*END) 也对正在显示的记录有效，显示时起始于开始行号的那些行被清除。

如果与 OVERLAY 一起使用 SLNO(\*VAR)，但没用 CLRL 且要写这个记录几次，每次都以不同的开始行号，在新记录显示之前，删除前一个记录。

如果与下面的关键字一起使用 SLNO 键字，OS/400 检查开始行号来确定对记录的前一个输出操作是否有同一个开始行号：

ERRMSG, ERRMSGID, PUTOVR, PUTREATTIN

如果开始行号相同，那么执行由 ERRMSG, ERRMSGID, PUTOVR 或 PUTREATTIN 指定的操作。

如果开始行号不相同，则忽略 ERRMSG, ERRMSGID, PUTOVR 或 PUTREATTIN 键字。记录格式以用新值校正后的行号显示。

SLNO 不允许与下列关键字在同一个记录格式中指定：

ASSVME    KEEP    SFL    SFLCTL    USRDFN

SLNO 不能对被 PASSRCD 指定的记录格式指定。

图 3-207 给出如何规定 SLNO 键字。

00010A	R RECORD1	SLNO (*VAR)
00020A	FIELD1	5 I 2 2
00030A	FIELD2	5 B 3 2
00040A		
00050A	R RECORD2	SLNO (2)
00060A	FIELD3	5 10 2

00070A	FIELD4	5 B 10 13
	A	

图 3-207 规定 SLNO 键字

在图 3-207 中, 当开始行号是 0 或 1 时, FIELD1 显示在第 2 行, FIELD2 显示在第 3 行, 当程序中设开始行号为 2 时, FIELD1 显示在第 3 行 ( $2 - 1 + 2 = 3$ ), FIELD2 显示在第 4 行 ( $2 - 1 + 3 = 4$ )。

FIELD3 和 FIELD4 总显示在第 11 行 ( $2 - 1 + 10 = 11$ )。

### 3.3.155 SNGCHCFLD (单选项选择字段)

这是字段层键字, 用来定义一个字段是单选项选择字段, 单选项选择字段是一个有固定数目的选项, 用户可以从中选择一项的字段。这个字段出现在垂直选择列表中, 左上方有一个输入字段或一组广播按钮。

键字格式为:

```
SNGCHCFLD[([*NORSTCSR | *RSTCSR]
[*NOAUTOSLT | *AUTOSLT | *AUTOSLTENH]
[*NOSLTIND | *SLTIND]
[*NOAUTOENT | *AUTOENT | *AUTOENTNN]
[[(NUMCOL 列数) | (*NUMROW 行数)]
[(*GUTTER 间隔宽度)]])]
```

参数是可选的, 如果一个也没指定, 单选项字段选项安排在一列, 用户可用箭头键把选择光标移出字段外。

参数 RSTCSR 指定是否允许用箭头键把选择光标移出选择字段外。\*RSTCSR 为不允许, \*NORSTCSR 为允许, 缺省值为\*NORSTCSR。

注: \*RSTCSR 的一个例外是: 如果选择字段是下拉窗口中唯一的字段, 当选择光标在最左列或最右列时, 左或右箭头将关闭当前下拉窗口, 打开当前菜单选项选择的左侧或右侧的下拉窗口。

\*RSTCSR 参数在不是连到支持不可编程工作站增强接口控制器上的显示站时被忽略。

\*AUTOSLT 参数指明用 enter 键是否会自动选择光标所在的当前选项。\*NOAUTOSLT 指定用户必须选择选项。\*AUTOSLTENH 指出自动选择只在设备与增强控制器相连时才有效。缺省值为\*AUTOSLT。

\*SLTIND 参数指明是否显示选择指示 (如广播按钮)。\*NOSLTIND 为不显示, 缺省值为\*SLTIND。

\*AUTOENT 会导致选取一个选项记录立即就返回 (就象用户已经按了 ENTER 键)。AUTOENT 参数指明在什么范围内自动输入有效。\*NOAUTOENT 指明无效, \*AUTOENT 指明除非需要双数字选择号, 否则自动输入对所有的显示有效。\*AUTOENTNN 指出只有在选项不需要数字型的选择时自动选择才有效。参数的缺省值为\*NOAUTOENT。

\*NUMCOL 指定选择字段的多列方式显示, 如下:

Choice1	Choice2	Choice3
Choice4	Choice5	Choice6
Choice7	Choice8	Choice9

参数的列数部分指出选择字段有多少列, 它必须是正整数并且整个单选项选择字段必须都放进显示屏。

\*NUMROW 指定选择字段以多行方式显示，如下：

Choice1	Choice4	Choice7
Choice2	Choice5	Choice8
Choice3	Choice6	Choice9

其中行数指出选择字段包含多少行，它必须是正整数并且整个单选项选择字段要都放进显示屏。

\*GUTTER 参数是可选的，指定在单选字段的列之间的空格数。只有指定了\*NUMCOL 或\*NUMROW 时才能指定它。并且要跟着 (\*NUMxxx #) 参数，‘间隔宽度’必须是不小于 2 的正整数。如果不指\*GUTTER，缺省值为 3 (包括首尾选择正文属性)。

有 SNGCHCFLD 键字的字段必须也有一个或多个描述字段选项的 CHOICE 键字。

有 SNGCHCFLD 的字段必须定义成数据类型为 Y，长度为 2，小数位为 0 的可输入字段。字段的位置是第一个选项左面的输入字段的位置，或是最上边的广播按钮的位置。输入时，字段中是被选择的选项号，如果没选则为 0。输出时，如果字段中有一个选项号，这个选项作为缺省选择被显示。

在一个有 SNGCHCFLD 的字段上，可以指定下列键字：

CHCACCEL	DSPATR(RI UL BL CS HI ND PC)	(4)
CHCAVAIL	ERRMSG	
CHCUNAVAIL	ERRMSGID	
CHCSLT (1)	INDTXT	
CHCCTL	PUTRETAIN	
CHECK(ER) (2)	TEXT	
CHECK(FE) (3)		

注：1.只在一个不显示选择指示的下拉菜单中显示单选项选择字段，CHCSLT 功能才有效（例如，指定 PULLDOWN(\*NOSLTIND)）。

2.如果指定\*AUTOENT 或\*AUTOENTNN，CHECK(ER)不允许与 SNGCHCFLD 一起指定。

3. CHECK(FE)只适于连在不支持增强接口控制器的显示器上。

4.如果 COLOR 或 DSPATR 为有 SNGCHCFLD 的字段指定，它仅用在基于字符显示屏上选择字段的输入部分。

可选指示器对比键字无效。

图 3-208 给出如何规定 SNGCHCFLD 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R RECORD
A      :
A      :
A          3 3' Single selection field.      :
A      F1      2Y 0B 3 35SNGCHCFLD
A 01          CHOICE(1 '>Undo      ')
A          CHOICE(2 &MARKTXT)
A          CHOICE(3 '>Copy      ')
```

```
A          MARKTXT      10A  P  
A
```

图 3-208 规定 SNGCHCFLD 键字

此例中，当使用一个连在支持不可编程工作站增强接口控制器的图形显示站时，选择字段看起来象下图：

PICTURE24

### 3.3.155 SYSNAME (系统名)

这是字段层键字，用来把当前系统名显示为一个 8 字符长的常数（仅输出）字段。可以指定字段的位置。SYSNAME 键字，或可选的 COLOR, DSPATR 和 TEXT 键字，17 到 38 列必须空白。

此键字无参数。

可选指示器对此键字无效，尽管可选指示器可用来作为字段的条件。

图 3-209 给出如何规定 SYSNAME 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1  
00020A          2 62' SYSTEM:'  
00030A          2 72SYSNAME  
A
```

图 3-209 规定 SYSNAME 键字

### 3.3.156 TEXT (文档说明)

这是记录层或字段层键字，用来为程序所用的记录格式或字段提供文档说明（或注释）。

TEXT 对除 SFLMSGKEY 或 SFLPGMQ 字段外的任何字段或记录格式都有效。

键字格式为：

TEXT ('说明')

'说明' 必须被用'号括起，如果 '说明' 的长度大于 50，只有前 50 个字符被高级语言编译采用。

可选指示器对此键字无效。

图 3-210 给出如何规定 TEXT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R CUSMST          TEXT('Customer Master Record')  
00020A      FLD1           3  0    TEXT('ORDER NUMBER FIELD')  
A
```

图 3-210 规定 TEXT 键字

### 3.3.158 TIME (时间)

这是字段层键字，用来把当前系统时间显示成一个仅输出的常数字段。

此键字无参数。

只能指定此字段的位置，与 TIME 一起，也可规定 EDTCDE, EDTWRD, COLOR, DSPATR, TEXT 键字，17 – 38 列必须是空白。

假定 TIME 用编辑字 ‘0:\_:\_’ (\_代表一个空格)，你可以指定另一种编辑字或用户定义的编辑码 (5 – 9) 来改变 IBM 提供的编辑方式。

可选指示器对此键字无效，尽管可用可选指示器定义字段的条件。

图 3-211 给出如何规定 TIME 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A 20           1 56TIME
00020A 21           1 56TIME
A               EDTWRD('0 &HRS& &MINS& &SECS')
A
```

图 3-211 规定 TIME 键字

在图 3-211 中，系统时间为 110645。

如果可选指示器 20 为 on，时间显示为：11:06:45

如果 21 为 on, 20 为 off，时间显示为：11 HRS 06 MINS 45 SECS

### 3.3.159 UNLOCK (解锁)

这是记录层键字，用来指定在对你正定义的记录格式发送一个输入操作后，OS/400 立即解锁键盘。没有 UNLOCK 键字，在从屏幕上读数据后，OS/400 保持键盘锁定，在处理刚读入数据时，工作站用户不能向可输入字段键入数据。

键字格式为：

```
UNLOCK[(*ERASE) | (*MDTOFF) | [(*ERASE *MDTOFF)] | [(*MDTOFF
*ERASE)]
```

参数值\*ERASE 和\*MDTOFF 是可选的。如果没有规定任何参数，则\*ERASE 是缺省值。

当程序发送一个输入操作时，发生下列顺序的操作：

1. 键盘解锁（如果不是已解锁），允许工作站用户向屏幕中的可输入字段键入数据。
2. 工作站用户按 Enter 键（或一个有效功能键）。
3. 记录格式中可输入字段已被键入或它们用有效 DSPATR (MDT) 键字显示的话，则设置它们的更改数据标识 (MDTs) 为 on。
4. 输入操作完成时，UNLOCK 的参数值将如下所述那样影响有 MDT 为 on 的可输入字段：

UNLOCK (不带 GETRETAIN) 或 UNLOCK (\*REASE)：键盘保持解锁，清除屏幕上的可输入字段，并且它们的 MDTs 在接下来的输入操作保留为 on。

UNLOCK (\*MDTOFF) 或 UNLOCK (带 GETRETAIN)：键盘保持解锁，不清除屏幕上的可输入字段，并且它们的 MDTs 在接下的输入操作中被关闭。

如果 GETRETAIN 与 UNLOCK(任何参数)一起指定, 在文件生成时, 忽略 GETRETAIN 并且产生错误信息,

注: 此键字不阻止应用程序在输入操作后立即发送一个输出操作, 然而当输出操作改变显示时, 键盘解锁, 工作站用户能键入数据。

可选指示器对此键字无效。

图 3-212 给出如何规定 UNLOCK 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1          UNLOCK(*ERASE)  
00020A      FLD1      4   B   2   2  
00030A      FLD2      4   B   3   2  
          A  
00040A      R RECORD2          UNLOCK(*MDTOFF)  
00050A      FLD21     4   B   4   2  
00060A      FLD22     4   B   5   2  
          A  
00070A      R RECORD3          UNLOCK(*ERASE *MDTOFF)  
00080A      FLD31     4   B   6   2  
00090A      FLD32     4   B   7   2  
          A
```

图 3-212 规定 UNLOCK 键字

### 3.3.158 USER (用户)

这是字段层键字, 用来把当前工作的用户配置文件名显示成一个 10 字符长的常数(仅输出)字段。你可以指定字段位置, USER 键字也可与 COLOR, DSPATR, TEXT 键字一起定义, 17 - 38 列必须为空白。

此键字无参数。

可选指示器对此键字无效, 尽管可用可选指示器来作为字段的条件。

注: 对 System/36 环境 MRT 作业, 被显示的用户配置文件名与显示工作站的交互作业的用户配置文件名相同。

图 3-213 给出如何规定 USER 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1  
00020A  10          2 12'USER:'  
00030A  10          2 20USER  
00040A          DSPATR(HI)  
00050A N10         15 18'USER:'  
00060A N10         15 26USER  
00070A          DSPATR(HI)  
          A
```

图 3-213 规定 USER 键字

在图 3-213 中, 如果指示器 10 为 on, 用户名显示在第 2 行 20 列开始的位置上, 如指示器 10 为 off, 用户名显示在第 15 行 26 列开始的位置上。

### 3.3.161 USRDFN (用户定义)

这是记录层关键字, 用来指定此记录的数据是用户定义的数据流格式。

此关键字无参数。

由于是用数据流格式化显示屏幕, 所以此记录没有有效的字段, 除了 INVITE, KEEP, PASSWORD, HLPRTN, HELP, HLPCLR, PRINT, OPENPRT 和 TEXT 外没有记录层或字段层的键字适用于此记录, 但 HELP, HLPRTN 和 INVITE 键字在记录层指定有效, 在文件层则不行, 帮助规范对此记录有效。一旦对此记录输出操作, OS/400 不再保存记录有关设备的状态信息。在使用此键字之前应有较多的设备信息。

可选指示器对此键字无效。

图 3-214 给出如何规定 USRDFN 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00020A      R USRREC          USRDFN  
00030A  
A
```

图 3-214 规定 USRDFN 键字

### 3.3.162 USRDSPMGT (用户显示管理)

这是文件层关键字, 用来指定写到屏幕上的数据保留, 直到重写或用 CLRL 键字清除。

此关键字无参数。

参考附录 F “System/36 环境考虑”, 来了解怎样指定 USRDSPMGT 键字。

### 3.3.163 USRRSTDSP (用户重存显示)

这是记录层关键字, 在窗口记录上指定由应用程序管理显示, 窗口记录不自动取消。如果不指定此键字, 当窗口记录显示时系统保存和重存背景显示。

此关键字无参数。

WINDOW 必须与 USRRSTDSP 在同一记录中指出, 只有当 WINDOW 键字定义一个窗口时, USRRSTDSP 键字才起作用。如果 WINDOW 键字定义一个记录格式名, USRRSTDSP 不起作用。详细内容请看 3.3.169 “WINDOW”。

可选指示器对此键字有效。

图 3-215 给出如何规定 USRRSTDSP 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A      R APPRCD  
A      FIELD1      10A 0 5 40  
A      FIELD2      5S 0B 6 45  
A      R WINDOW1      WINDOW(6 15 9 30)  
A 25          USRRSTDSP  
A      FIELD1      5A B 2 2
```

```
A          FIELD2      20S  B  8  5  
A
```

图 3-215 规定 USRRSTDSP 键字

此例中，假定 APPRCD 已在显示屏上，当 WINDOW1 写到屏幕上时，如果指示器 25 为 ON，系统不保存背景显示（内有 APPRCD），当用户退出 WINDOW1 时，应用程序必须重存背景显示，可能是由重写 APPRCD 来完成。

注：使用 USRRSTDSP，窗口数目没有限制。如不使用 USRRSTDSP，窗口数目的限制是 12。

### 3.3.164 VALNUM（有效数字）

这是文件、记录或字段层键字，用来增强对仅数值型字段所作的错误检查。当在一个仅数值型字段上指定此键字时，如果用户企图在数值位之间嵌入空格，加号或减号或者把加号或减号放到数字之前的话它会返回一个错误。

此键字无参数。

有 VALNUM 键字的字段必须定义成一个数据类型为 Y 的可输入字段。

可选指示器对此键字无效。

图 3-216 给出如何规定 VALNUM 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD  
00020A      F1      5Y 0B  3  4VALNUM  
00030A      F2      5Y 0B  4  4
```

图 3-216 规定 VALNUM 键字

此例中，字段 F1 不允许用户在数值间嵌入空格、加号或减号或在数值前使用加号或减号，字段 F2 看作在 3.2.11 中说明的字段。

### 3.3.165 VALUES（值）

这是字段层键字，用来指定用户向此字段键入的一组有效值。OS/400 对你键入的数据用这里指定的某个值进行相等与否的检查。注意，只有在字段被工作站用户改变或者它的修改数据标识（MDT）被用 DSPATR（MDT）置 ON 时，OS/400 才执行这个检查。

注：有关用户定义的错误信息内容请参考 CHKMSGID。

键字格式为：

VALUES (值 - 1 [值 - 2] .....[值 - 100])

可有 1 到 100 个值，值与值之间至少有 1 个空格。

注：不能在一个 DDS 语句中有 5000 个以上字符。因此，不能指定使 VALUES 大于 5000 个字符值。如果对一个字段指定其他键字，要把它们加起不大于 5000 个字符。例如，为字段指定 DSPATR (HI) 会使留给 VALUES 的字符数减少。

值可以是字符值或数字值。它与要被检测的字段在长度上一致。字符值必须用'号括起，数字值要用 0 到 9，负值可在值前加负号，要在低位字符位处对齐。

### 3.3.165.1 定义一个数字型字段

当工作站用户键入数据时，OS/400 按照字段中的小数位数校准输入的字符。当字段传给应用程序时，头尾的空格用零代替。如果没输入小数点，OS/400 把小数点放到输入的最右边字符后。例如，对一个 5 位长(34 列)2 位小数(37 列指定)的数值型字段，1.2 为 001.20，100 为 100.00。

比较是基于被传到程序的值（例如，右对齐填充或左对齐填充）。

不能对浮点数字段指定 VALUVE。

可选指示器对此键字无效。

图 3-217 给出如何规定 VALUES 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      RESPC      11   I   8   2VALUES('A' 'B' 'C' 'D')  
00020A      RESPN      31   0I   9   2VALUES(33 -42 01)  
00030A      DECFLD     1   2I  11   2VALUES(1.2 100)  
A
```

图 3-217 规定 VALUES 键字

### 3.3.166 VLDCMDKEY (有效命令键)

这是文件层或记录层键字，当工作站用户按了除 Enter 键外的任何有效命令键时，OS/400 将相应的应答指示器置 ON，这个功能的一个用处是执行一个简单检测，来确定工作站用户是否已经请求在程序中监控这个功能。

怎样在 System/36 环境下的文件中指定 VLDCMDKEY，参照附录 F “System/36 环境考虑”。

键字格式为：

VLDCMDKEY (应答指示器 [‘说明’])

应答指示器是必需的。

‘说明’是可选的，它在程序编译时生成的计算机打印输出中解释指示器的用法，它只有注释功能，’号是必须的，如果‘说明’多于 50 个字符，在程序打印输出中被截断为 50 个字符。

为使一个命令键有效，必须使用下列某个键字来激活所用的键：

键字	解释
ALTHELP(CAnn)	带或不带应答指示器的 HELP 键字，使得指定的命令注意键是有效的命令键
ALTPAGEUP(CFnn)	带或不带应答指示器的 PAGEUP 键字，如果也指定 PAGEUP 使得指定的命令功能键被认为是有效的命令键
ALTPAGEDWN(CFnn)	带或不带应答指示器的 PAGEDOWN，如果也指定 PAGEDOWN 键字，使命令功能键为有效的命令键
CAnn	带或不带应答指示器
CFnn	带或不带应答指示器
CLEAR	带或不带应答指示器
HELP	只有在 HELP 键向下面那样返给应用程序时才有效：

	HELP 和 HLPRTN (带或不带应答指示器)
	HELP (带或不带应答指示器) 并且没有为当前显示的任何记录定义帮助区
HOME	带或不带应答指示器
PAGEDOWN	带或不带应答指示器
PAGEUP	带或不带应答指示器
PRINT	只有当 PRINT 键象下面那样返给应用程序时才有效: PRINT (带应答指示器) PRINT (*PGM)
ROLLUP	带或不带应答指示器
ROLLODOWN	带或不带应答指示器

可选指示器对此键字无效。

图 3-218 给出如何规定 VLDWORD 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R REC1          VLDCMDKEY(90 'Any valid key')
00020A          CA01(91)
00030A          CA02(92)
00040A          CA03(93)
00050A          CLEAR(94)
A
```

图 3-218 规定 VLDWORD 键字

在图 3-218 中, 如果按了 CA01, CA02, CA03, CLEAR 中任一个, 指示器 90 置 ON。

### 3.3.167 WDWBORDER (窗口边界)

这是文件层或记录层键字, 用来指定窗口边界的颜色, 显示属性和字符。

键字格式为:

WDWBORDER ([颜色] [显示属性] [字符])

必须至少指定一个参数。

颜色参数指定在一个彩色显示站上 (3179 - C1 和 C2, 5295 彩色显示站, 或 5555 - C01 和 F01) 边界字符的颜色, 此参数为\*Color 值的表达式:

颜色参数的有效值是:

值	含义
BLU	蓝色
GRN	绿色
WHT	白色
RED	红色
TRQ	蓝绿色
YLW	黄色
PNK	粉红色

缺省值为 BLU。在黑白显示屏上忽略此参数。

显示属性参数指定边界字符的显示属性, 此参数用 (\*DSPATR[值 1 [值 2...]]) 给出。

如果使用多个 DSPATR 值，它们联合起来形成一个用于整个边界的 DSPATR。

显示属性的有效值为：

值	含义
BL	闪烁
CS	列间隔
HI	高亮度
ND	不显示
RI	反映像
UL	下划线

显示属性无缺省值。

注：显示属性 CS, HI, BL 会使 5292, 3179, 3197 型 C1 和 C2 显示站上的字段看起来是彩色字段，显示属性 HI, RI, UL 会导致边界不被显示，请看 3.3.30 的 ‘COLOR’ .

字符参数指定构成边界的字符，参数以 (\*CHAR ‘边界字符’) 的格式指定，边界字符值是字符串，以如下顺序定义边界字符。

左上角  
上界  
右上角  
左界  
右界  
左下角  
下界  
右下角

如果没指定这个参数，缺省的边界字符为：左上角和右上角、顶界和底界是 (•)，左、右边界和左下角、右下角是 (:)。尽管任何可显示的字符都可做边界字符，建议你使用恒定的字符。下面的表给出这些恒定字符。

图 3-219

图 3-219 系统数据字符集

注：另外，可使用下列中的任何一个：

大写英文字母：A-Z

数值字符：0-9

如果在记录层规定 WDWBORDER，必须也在同一个记录中指定 WINDOW 或 PULLDOWN。如果一个 WINODW 键字引用另一个窗口，则发出警告信息。

可选指示器对此键字有效。

在同一个记录中可以指定多个 WDWBORDER 键字。如果在文件层或记录层多次指定了 WDWBORDER 键字，键字的有效参数组合起来在这层起作用。如果为同一参数指定了不同的值，用第一个键字的参数值。

如果在窗口或下拉定义记录上即在文件层又在记录层指定了 WDWBORDER 键字，在两层定义的通用数值结合起来用。如果同一参数指定了不同的值，记录层的参数值有效。

图 3-220 给出如何规定 WDWBORDER 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A 01                               WDWBORDER((*COLOR PNK) +
A                                     (*DSPATR BL))
A           R RECORD1             WINDOW(6 15 9 30)
A N01                               WDWBORDER((*COLOR GRN))
A 01                               WDWBORDER((*COLOR RED))
A           FIELD1      5A B 2 2
A           FIELD2      20A B 8 5
A           R RECORD2            WINDOW(8 20 9 30)
A                                     WDWBORDER((*COLOR YLW) +
A                                     (*DSPATR RI))
A           FIELD3      5A B 2 2
A           FIELD4      20A B 8 5
A           LINE        2S 0P
A           POS         2S 0P
A           R RECORD3            WINDOW(&LINE &POS 9 30)
A                                     WDWBORDER((*CHAR +
A 02                                     '++-||+-+'))
A           FIELD3      5A B 2 2
A           FIELD4      20A B 8 5
A           LINE        2S 0P
A           POS         3S 0P
A
```

图 3-220 规定 WDWBORDER 键字

如果由 RECORD1 定义的窗口在指示器 01 为 off 时写到屏幕上，它有一个绿色的边界，垂直边界由冒号构成，而水平边界由点号构成。如果指示器 01 为 on，窗口有个闪烁的红色边界。

当由 RECORD2 定义的窗口写到屏幕上时，它有一个黄色的边界，由缺省的边界符构

成，反映象显示。

当由 RECORD3 定义的窗口写到屏幕上时，显示如下：

如果指示器 02 为 on，且指示器 01 为 off，窗口有一个蓝色边界，顶部和底部边界由破折号构成，左右边界是垂直棒，四角是加号。

如果指示器 02 为 off，且 01 为 on，窗口是粉红边界。

如果指示器 01、02 都为 on，窗口有一个粉红边界，顶部和底部边界由破折号构成，左右边界是垂直棒，四角是加号。

### 3.3.166 WDWTITLE (窗口标题)

这是记录层键字，用来为嵌入到窗口顶部或底部的标题指定正文、颜色和显示属性。

键字格式：

```
WDWTITLE ([标题正文] [标题正文颜色]  
          [标题正文显示属性] [*CENTER | *LEFT | *RIGHT]  
          [*TOP | *BOTTOM])
```

必须至少指定一个参数。

标题正文是个可选参数，指定放在窗口边界的正文，正文长度应当小于等于 WINDOW 键字定义的窗口位置参数。如果正文开头是空格，标题在结尾被缩短与空格一样多的数目，如果正文太长（大于窗口位置），它被从右边截断，此参数以表达式方式 (\*TEXT 值) 给出，其中值可以用下列两种方法之一指定：

- 作为一个字符串：‘标题正文’
- 作为一个程序到系统的字段：&字段名

指定的字段名必须在窗口记录中存在，并被定义成一个处理方式为 P 的字符型字段。

注：

1.字母型图表 (GRAPHIC) 不能做标题正文参数。

2.如果标题字符是空格，将显示空格标题。

3.如果标题正文为空，不显示标题。

标题正文颜色参数指定在彩色显示屏上标题的颜色，格式为 \*COLOR 值。

标题正文颜色参数的有效值为：

值	含义
BLU	蓝色
GRN	绿色
WHT	白色
RED	红色
TRQ	蓝绿色
YLW	黄色
PNK	粉红色

如果没指定标题正文颜色参数，缺省值为边界颜色，如果在黑白显示屏上指定，忽略此参数。

标题正文显示属性参数指明标题正文的显示属性，格式为 \*DSPATR[值 1 [值 2…]]。如果使用多个 DSPATR，它们组合在一起成为一个标题正文。

标题正文显示属性参数的有效值为：

值	含义
BL	闪烁
CS	列间隔

HI	高亮度
ND	不显示
RI	反映像
UL	下划线

如果没指定标题正文显示属性，其缺省值为边界的显示属性。

如果标题正文颜色和标题正文显示属性都没指定，窗口边界将滑到窗口标题的第一个字符，并假定紧接最后一个字符。如果这两个参数都指定了，那么在窗口标题的前后都有空格。

\*CENTER/\*LEFT/\*RIGHT 参数指定窗口标题排在窗口边界的中间、左边或右边。如果没指定，则在下个参数为\*TOP 时为窗口标题在边界中间，在下个参数为\*BOTTOM 时为窗口边界左边。

注：不是所有的控制器都支持校准排列，对这些控制器标题被放在中间。

\*TOP/\*BOTTOM 参数指定标题应嵌在顶边界还是底边界，缺省值为顶边界 (\*TOP)。

注：如果\*BOTTOM 被指定在一个不支持文档在窗口底边界的增强控制器上，此键字被忽略。

注：在 CRTDSPF 或 CHGDSFP 命令中指定 ENHDSP(\*NO)，可以使\*BOTTOM, \*LEFT 和\*RIGHT 总有效；然而，要丢失所有其他的增强用户接口功能。

可选指示器对此键字有效。

WDWTITLE 只能用在有 WINDOW 键字的记录，如果 WINDOW 键字引用另一个窗口，会出现警告信息。

可以对一个记录指定多个 WDWTITLE 键字。如果在记录层多次指定 WDWTITLE 键字，这些键字的有效参数要组合起来使用。如果对同一参数指定了不同值，使用第一个键字的参数值。

图 3-221 给出如何规定 WDWTITLE 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A      R RECORD1           WINDOW(6 15 9 30)
A N01          WDWTITLE((*TEXT &TTL1) (*COLOR GRN))
A 01          WDWTITLE((*TEXT &TTL1) (*COLOR RED))
A      FIELD1        5A B 2 2
A      FIELD2        20A B 8 5
A      TTL1         10A P
A
A      R RECORD2           WINDOW(8 20 9 30)
A          WDWTITLE((*TEXT &TTL2) +
A                      (*COLOR YLW) +
A                      (*DSPATR RI))
A      FIELD3        5A B 2 2
A      FIELD4        20A B 8 5
A      TTL2         10A P
A
```

图 3-221 规定 WDWTITLE 键字

如果由 RECORD1 定义的窗口写到显示屏上，它把在 TTL1 中的正文嵌入到窗口的顶边界中。当指示器 01 为 off，正文为绿色，指示器 01 为 on，正文为红色。

当由 RECORD2 定义的窗口写到显示屏上时，在 TTL2 中的正文会嵌入到窗口的顶边界中，正文将以反映像方式显示成黄色。

### 3.3.169 WINDOW (窗口)

这是记录层关键字，用来指定定义的记录格式要用窗口显示，窗口是覆盖部分屏幕的一些信息。窗口一般比实际工作站显示屏小一些，可以放在显示屏的任何位置。

WINDOW 关键字可以使用的记录格式有两种：

指定窗口的位置和尺寸来定义一个窗口，这叫作窗口定义记录。

引用一个已经定义了的窗口尺寸和位置记录格式，这叫作窗口引用记录。

关键字格式是下列之一：

WINDOW(起始行 | &起始行字段 起始列 | &起始列字段 窗口行 窗口列

[\*MSGLIN | \*NOMSGLIN]

[\*RSTCSR | \*NORSTCSR])

或 WINDOW(\*DFT 窗口行 窗口列

[\*MSGLIN | \*NOMSGLIN]

[\*RSTCSR | \*NORSTCSR])

指定 WINDOW 关键字的格式来定义窗口。定义的记录格式用窗口显示，一次可在屏幕上显示 12 个窗口。在 DDS 中可定义多于 12 个窗口，但一次只可显示 12 个窗口。如果指定 USRRSTDSP 窗口的数目不限制，在记录中定义的所有字段必须装进窗口中。

用参数规定：

要指定窗口边界左上角的行号，或存有行号的字段名，如果指定字段的话，它必须在记录格式中存在，并被定义成数据类型为 S，处理方式为 P，长度不大于 3 的字段。

要指定窗口边界左上角的列号或存有列号的字段名，如果规定了字段，它必须在记录格式中存在，且定义为 S 型，处理方式为 P，长度不大于 3 的字段。

规定窗口中的行数，它不能大于显示尺寸的行数减 2，这是因为上下窗口边界各占一行，窗口中的最后一行用作信息行且不能有任何字段，例如规定窗口有 10 行，仅 9 行能有字段，第 10 行做信息行。

指定窗口的列数，它不能大于显示尺寸的列数减 4，这是因为在窗口中左右边界字符都需要有属性字节，在边界字符和有效列之间有一个属性字节，对于 DBCS 属性的窗口，系统在窗口的每侧增加两个字节用做 DBCS 字段的转出和转入字符。

MSGLIN 指定一个窗口是否包括信息行。如果没规定，缺省值为 \*MSGLIN。

\*NOMSGLIN 把信息行移出窗口放到屏幕底部或放到由 MSGLOC 所定义的位置，窗口的最后一个可用行用来做错误信息行，在这一行没有记录显示。如果错误信息多于一行，它被截断放在窗口中。

\*RSTCSR 指定当光标离开窗口时是否限制用户功能键的使用。当指定\*RSTCSR 且光标在窗口外时，用户按功能键可实现其功能好象光标在窗口内一样。当用户对一个支持不可编程工作站增强接口的控制器指定\*RSTCSR 时，用户可以把光标移出窗口（不可用鼠标），对其它的工作站，当光标在窗口外用户按功能键时，用户会收到“嘟嘟”声，而光标放回到窗口内，控制不返给应用程序。\*RSTCSR 为缺省值。

特殊值\*DFT 可以代替起始行和起始列参数，指出由系统确定窗口的开始行和列。窗口位置与光标位置有关，类似于应用程序可变起始位置的帮助窗口，详细信息请看“应用显示编程”一书。

此关键字的第二种格式为：

**WINDOW** (记录格式名)

用 **WINDOW** 的这种格式，来指明定义的记录格式将显示在另一个记录格式所定义的窗口。

参数指定有窗口属性的记录格式名，用这个参数的记录格式被显示在在引用记录中定义的窗口内。

在有 **WINDOW** 键字的记录格式内指定的字段位置是相对于第一个可用窗口中的左上角位置，第一个可用窗口位置是在上边界下的第一行、左边界右边第二列上（结束属性字节占据左边界右边的第一个字节）。

当显示窗口时，如果没指定 **USRSTDSP**，当前显示中的任何记录都被暂停显示，暂停的记录可以在窗口周围看见。只有在活动窗口内允许输入，要把窗口从屏幕上移出，可以向背景窗口写一条记录或把一条非窗口记录覆盖到显示上。

**WINDOW** 不允许与下列关键字在同一记录格式中使用：

ALWROL	ASSUME	MNUBAR
PULLDOWN	SFL	USRDFN

注：**WINDOW** 在有 **SFLCTL** 的记录上是允许的。这样，子文件就可以在窗口内显示。

**WINDOW** 不能对有 **PASSRCD** 键字的记录格式指定，对有 **WINDOW** 键字的记录，忽略 **ERRSFL**。

对有 **WINDOW** 键字的记录，忽略 **MSGLOC** 键字，除非指定 **NOMSGL**。

如果一个记录格式即有 **WINDOW** 又有 **WDWBORDER** 键字，在 **WINDOW** 键字中指定起始行、起始列、窗口行和窗口列参数。**WINDOW** 键字不指定记录格式名参数。

可选指示器对此键字无效。显示尺寸条件名有效。

图 3-222 给出如何用 **WINDOW** 键字来定义一个窗口。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A      R WINDOW1           WINDOW(4 20 9 30 *NORSTCSR)
A      FIELD1        8A  B  5 10
A      FIELD2        10A B  6 10
A      R WINDOW2          WINDOW(*DFT 9 30 *NOMSGLIN)
A
```

图 3-222 规定 **WINDOW** 键字 (例 1)

当显示 **WINDOW** 记录时，窗口边界的左上角在显示屏的第 4 行 20 列，右下角比上边界低 10 行，比左边界往右 33 列。

下边界行号 = 上边界行号 + 窗口行 + 1

右边界列号 = 左边界列号 + 窗口列数 + 3

字段 **FIELD1** 开始在上边界下第 2 行，左边界字符右边第 11 列（边界字符的结束属性字节已计算在内），即屏幕上的第 60 行 31 列。

实际字段行号 = 上边界行号 + 字段行号

实际字段列号 = 左边界列号 + 字段列号 + 1

**FIELD2** 在显示屏的第 10 行 31 列。

如果光标移出窗口，功能键仍保持活动。

当显示 **WINDOW2** 时，窗口边界的左上角位于运行时光标的位置上，窗口内没有信息

行，信息出现在屏幕底部。

如果光标移出窗口，功能键不活动。如果用户按了功能键，将收到嘟嘟声，而光标被放回到窗口中。

图 3-223 给出使用 WINDOW 在同一窗口中显示多条记录。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A      R WINDOW1           WINDOW(&LINE &POS 9 30)
A      USERID      8A 0 2 10
A      LINE        2S 0P
A      POS         3S 0P
A
A      R RECORD1          WINDOW(WINDOW1)
A                  OVERLAY
A      FIELD1      5A B 7 2
A      FIELD2      20A B 8 5
A
A      R RECORD2          WINDOW(WINDOW1)
A      FIELD3      10A B 2 2
A      FIELD4      8A B 8 5
A      FIELD4      8A B 8 5
A
```

图 3-223 规定 WINDOW 键字（例 2）

当显示 WINDOW1 记录时，窗口边界左上角的行号和列号由字段 LINE 和 POS 指定，右下角的位置是比上边界低 10 行，比左边界多 33 列。

USERID 字段开始在比上边界多 2 行，比左边界多 11 列的位置。

如果显示前例中的 RECORD1，它放在 WINDOW 中，它的字段的位置是相对于窗口左上角的。因为 RECORD1 用了 OVERLAY 键字，记录 WINDOW1 中已在显示上的字段仍在屏幕上保留，但有两个记录没有重叠。详细信息请看 3.3.100 OVERLAY 键字。

如果显示 RECORD2，它也被放 WINDOW1 中，它的字段的位置是相对于窗口左上角的，因为没用 OVERLAY 键字，WINDOW1 和 RECORD1 中的字段从窗口中取消。

图 3-224 给出如何对子文件指定 WINDOW 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A      R SFLDATA          SFL
A      NAME      20A B 4 5
A      RANK       10A B 4 27
A      SERIAL     8A B 4 38
A
A      R WINDOW1          SFLCTL(SFLDATA)
A                  WINDOW(8 25 10 50)
A                  SFLPAG(4)
A                  SFLSIZ(17)
```

A	SFLDSP
A	SFLDSPCTL
A	2 5' Full Name'
A	2 27' Rank'
A	2 38' Serial Nbr'
A	

图 3-224 规定 WINDOW 键字（例 3）

当显示 WINDOW1 子文件控制记录时，它和子文件被显示在窗口中，窗口边界的左上角在显示屏的第 8 行 25 列，右下角在第 19 行 78 列。

子文件记录和子文件控制记录中的字段位置是相对于窗口左上角的第一个可用窗口位置的。例如，SFLDATA 记录中的 NAME 字段开始在第 4 窗口行和第 5 窗口列，即屏幕上的第 12 行 31 列。

### 3.3.170 WRDWRAP（字重叠）

这是文件层、记录层或字段层键字，它定义命名字段，使它们溢出到下一显示行或继续输入字段。这个键字使重叠发生在数据的空白处而不是数据行的结尾。这样做使很长的说明字段容易读，缺省情况是数据在物理行的结尾或继续输入字段中重叠。

这个键字只能对处理方式为 I（仅输入）或 B（输出/输入）的字段指定。

此键字无参数。

不能对下列键盘转换指定 WRDWRAP：

有符号数 (S)	仅数值 (Y)	仅数字 (D)	仅数值字符 (M)
浮点数 (F)	仅 DBCS (J)	开放的 DBCS (O)	可选的 DBCS (E)
图形 (G)			

WRDWRAP 不能与下列键字一起指定：

AUTO (RAZ, RAB)
CHECK (MF, M10F, M11F, RB, RZ, RL, RLTB)
CHGINPDFT (MF)
DSPATR (OID, SP)
DUP
FLTFIXDEC
IGCALTTYP

可选指示器对此键字无效。

当使用 WRDWRAP 时，字段长度不能增加。因此，如果输入太多的数据，要关闭字重叠功能。

注：

- 对于不是连在支持不可编程工作站增强接口控制器上的显示站，忽略 WRDWRAP。

- WRDWRAP 可以用在只有一行的字段中，尽管不发生重叠，字段的字符插入功能还是会改变。

有关 WRDWRAP 更多的信息，请看应用显示编程手册。

图 3-225 给出如何规定 WRDWRAP 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RECORD1  
A          FIELD1      100A  0  1 17  
A          FIELD2      100A  I  4 17WRDWRAP  
A          FIELD3      100A  B  7 17WRDWRAP  
A          FIELD4      100A  B 10 17  
A          FIELD5      100A  O 13 17  
A
```

图 3-225 规定 WRDWRAP 键字

在此例中，RECORD1 是输入、输出，输出/输入字段描述。当显示站连到支持不可编程工作站增强接口控制器上时，FIELD2 和 FIELD3 具有字重叠功能，而 FIELD4 没有这功能。

# 第四章 打印文件键字

这章给出有关打印文件的下列信息：

    定义

    位置项

    键字项

位置项(1-44 列)给出打印文件 DDS 格式中第 1 列-44 列填写的规则和例子，键字项(45-80 列)给出打印文件 DDS 键字的规则和例子，这些键字是以字母顺序排列的。

对某些特殊类型的打印机，哪些 DDS 键字有效，请参考打印设备程序设计手册。

## 4.1 定义打印文件

要定义打印文件，按下列顺序指定：

- 1.文件层（可选）
- 2.记录层
- 3.字段层

每个文件至少要有一个记录格式，一个文件中最多可有 1024 个记录格式，一个记录格式中最多可有 32767 个字段，在一个记录格式中命名字段和指示器加在一起最大长度为 32767 字节。

注：文件名是由 CRTPRTF 命令确定的，不是由 DDS 命名，可在第一章 DDS 简介中找到有关文件层、记录格式层、字段层的解释，可在附录 B 中找到打印文件完整的例子，在 1.2 中找到 DDS 键字的语法规则。

图 4-1 给出打印文件编码的例子。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A* PRINTER FILE CODING EXAMPLE
00020A*
00030A      R TITLER           SKIPB(3)
00040A      FLD1             40     47SPACEA(2) UNDERLINE
00050A 30    FLD2             40     47SPACEA(2) UNDERLINE
00060A*
00070A      R AUTHROR
00080A                  66' by'
00090A      FIELD1            40     47SPACEB(1)
00100A                  50DFT(' Task Force I ')
00110A 31    SPACEA(1)
00120A 31    65' and'
00130A 31    FIELD2            40     47
00140A*
00150A      R PUBR             SKIPB(58)
```

00160A			47' Published by Department'
00170A	DEPT	3 0	+1
00180A			47DATE EDTCDE(Y)
00190A			SPACEB(1)
00200A N15			
00210AO 32 33 34			47TIME
00220A			SPACEB(1)
A			

图 4-1 打印文件编码例子

#### 4.2 在打印设备描述中有 AFP(\*YES)需要的键字

从 OS/400 版本 3.1 开始，先进的打印机功能（AFP\*）子系统是一个独立的 OS/400 特性，它叫做打印服务工具（PSF/400\*）。

在 OS/400 系统中有两个分开的子系统，原来的 OS/400 打印子系统继续支持行打和 IBM IPDS 打印机的一部分及其功能。对所有 IPDS 打印机的全方位支持是由集成 AFP 打印子系统提供的。处理应用输出的打印子系统是由打印机的设备描述决定的。仅定义为 DEVTYPE(\*IPDS) 和 AFP(\*YES) 的打印机（它们都是在打印设备描述中定义的）是由 AFP 打印子系统控制的。要打印由某些键字规定的值，需要 PSF/400。例如，由 DEVTYPE(\*APFDS) 生成的假脱机文件只能由 PSF/400 来打印。

安装 PSF/400 对下列键字有影响：

- BOX
- CDEFNT
- ENDPAGE
- FNTCHRSET
- GDF
- IGCCDEFNT
- LINE
- OVERLAY
- PAGSEG
- POSITION
- TXTRTT

有关 PSF/400 的详细信息，请看打印设备程序设计手册。

#### 4.3 位置项 (1-44 列)

这章说明打印文件 DDS 格式中前 44 列的填写规则，其余列的填写规则，请看 4.4 内容。

图 4-1 给出打印文件的一些位置项。

##### 4.3.1 顺序号 (1-5 列)

用这些列规定每行的顺序号，它是可选的，只做文档用。

#### 4.3.2 格式类型（第 6 列）

在这列上的 A 表示这是 DDS 格式，它是可选的，只做文档用。

#### 4.3.3 注释（第 7 列）

在这列的\*号表示这行是注释，注释的正文可在 8-80 列上写，一个空行（7-80 列上无字符），也看一个注释。注释行可在 DDS 的任何地方并仅出现在源文件中，注释在源打印输出中而在扩充的源打印输出中。

#### 4.3.4 条件（7-16 列）

用这些列规定可选指示器（01-99 的二位数字），然后由程序设置这些指示器的 ON 和 OFF 来选择一个字段或键字。在选择字段或键字之前这些指示器（2-9 个）必须有效（没规定 N 为 ON，规定 N 为 OFF），能生成一个 AND 条件。

注：字段或键字必须与最后一个（或唯一一个）指示器在同一行上。

每个条件最多可有九个指示器，每个字段或键字可有九个条件，也即最多可用 81 个指示器。

对字段或键字规定几个条件，可以生成 OR 关系。如果满足其中一个条件，就选择字段或键字。

注：OR 关系的条件可由一个指示器或几个指示器 OR 在一起组成，可以 AND 指示器形成一个条件，条件可 OR 在一起给应用程序提供选择字段或键字的不同方式。

键入下面的值来规定条件：

第 7 列（AND “与”）：如果需要三个以上的指示器来形成一个“与”条件，那么在下一行或下几行上指定指示器。可以在第二行或随后的几行上的第 7 列中指定一个 A 来继续“与”条件。或者第 7 位为空，因为 A 是缺省项。

第 7 列（OR “或”）：如果指定了几个“或”在一起的条件，那么每个条件都必须在新的一行开始，并且除第一个条件外，其它条件在第 7 列上为“O”。如第一个条件就指定 O 会产生一条错误信息，并且那一位被假定为空格。

##### 4.3.4.1 设置多个字段或键字的条件

若为一个字段设条件，字段名（或常量）和最后一个指示器（或唯一一个）必须在同一行上。如果这个字段有多个键字，那么条件仅限制字段而不限制键字。如果输出操作没有选择这个字段，那么这个字段的键字全不起作用，而不管键字是什么条件。例如：在图 4-1 中，如果指示器 30 为 OFF，SPACEA 和 UANDERLINE 就无效。

如果想条件限制多个键字，在同一行上写键字和最后一个指示器。如果条件要管多行上的键字，必须对此指示器用键字连续提供给所有的键字。

#### 4.3.5 名字或规范的类型（第 17 列）

这列上的值指出在 19-28 列上的名字类型，对打印文件有效的值为：

R	记录格式名
空白	字段名

图 4-1 给出名字类型编码的例子

有关信息, 请看 4.3.7 的内容。

#### 4.3.6 保留 (第 18 列)

任何文件都不用这列, 除了做注释外, 这列为空白。

#### 4.3.7 名字 (19-28 列)

用这些列规定记录格式名或字段名。

请参考 1.2 内容了解规定名字时的规则。

名字必须从 19 列开始。图 4-1 给出如何规定记录格式名和字段名。

##### 4.3.7.1 记录格式名

在 17 列写 R 时, 19-28 列就是一个记录格式名, 对打印文件可规定多个记录格式名, 但文件中的每个记录格式名必须唯一, 在记录格式中也必须有字段名或常数字段, 组成打印文件。

##### 4.3.7.2 字段名

17 列为空格, 19-28 列就是一个字段名, 字段名在记录格式中必须唯一。

##### 4.3.7.3 常数字段

常数字段是非命名字段 (19-28 列必须为空白), 下列规则适合于常数字段:

17-38 列必须是空白;

字段位置是必需的 (39-44 列);

字段用可选指示器 (7-16 列) 来限定条件。

常数字段可在有 BOX、ENDPAGE、GDF、LINE、OVERLAY 或 PAGSEG 的记录格式中定义或者在有 POSITION 规定的记录格式中定义。

常数本身在 45-80 列, 用下列方式之一来定义:

—明显的 DFT 键字 (用 DFT ('值')) 格式指定

—隐式的 DFT 键字 (用 '值' 的格式指定)

—DATE 键字 (不规定值, 见 DATE 键字说明)

—TIME 键字 (不规定值, 见 TIME 键字说明)

—PAGNBR 键字 (不规定值, 见 PAGNBR 键字说明)

—MSGCON 键字 (规定信息描述、信息文件、库名、信息描述的长度, 看 MSGCON 的说明)

仅在常数字段规定了 DATE、TIME 或 PAGNBR 键字时, 才可以规定 EDTCDE 或 EDTWRD 键字。

如果使用行号，可以用任意顺序来规定字段；如果不使用行号，那么必须按照要在打印页上出现的顺序来规定字段的顺序。

#### 4.3.8 引用（第 29 列）

在 29 列用 R 来把以前定义的命名字段（被引用字段）的属性复制到正定义的字段中来，（如果没有 R，就必须规定字段属性）。例如，可以引用一个外部定义的文件的字段，从数据库文件打印一个报表。

当用引用功能时，要规定被引用字段名，即使它与引用字段名相同也要这样。（被引用字段名是在 REF 或 REFFLD 中规定的以前生成好的数据库文件字段），被引用的字段属性是长度、数据类型和小数位以及 ALIAS、FLTPCN、TEXT 和编辑键字。

如果被引用字段名与正定义字段名相同，在 29 列写 R，在 19-28 列写正定义的字段名。如被引用字段名与正定义的字段名不同，要用 REFFLD 键字规定被引用字段名。

在文件层和记录层的 29 列必须是空格。

可把被引用字段所在的文件名做为 REF 或 REFFLD 键字的参数，有关内容请看 REF 和 REFFLD 键字的说明及附录 A。

如果不复制被引用字段的全部属性，可对字段自己规定某些属性，如下所述：

为了替换编辑属性，可重新规定 EDTCDE 或 EDTWRD。要删除这些键字，用 DLTEDT。

如果被引用文件规定了有效性检查键字（CHECK、COMP、RANGE、VALUES），在打印文件中无效。

当替换某些规定时，要影响一些事情，如：

如对定义的字段规定了数据类型，长度或小数位，则不从被引用字段复制编辑键字。

打印文件不支持压缩十进制和二进制字段，这样，当引用这些类型的字段时，要在打印文件中转换成区位十进制。

注：一旦生成打印文件，可以修改或删除被引用字段而不影响打印文件中的字段定义。若要包括被引用字段做过的修改，就要删除再重新生成打印文件。

#### 4.3.9 长度（30-34 列）

对每个命名字段要规定长度（除非从被引用字段复制），长度表示对这个字段做输出操作时，从程序传出的数据的字节数。（如果字段被编辑，有关的编辑属性决定字段的打印长度）。

图 4-1 给出字段长度的编码。

区位十进制字段的最大长度为 31，DDS 允许的最大字段长为 32767 个字符。如果字段长超出页的尺寸，出现错误信息。单精度浮点字段的最大长度是 9 个数字，双精度为 17 个数字。

如用被引用字段，要替换被引用字段的长度可规定一个新值，也可规定增加或减少的长度，+号表示增加，-号表示减少。例如，+4 表示定义的字段要比被引用字段长度多 4，字段长度的替换不考虑小数点。

长度要右对齐，前导零是可选的。

图 4-2 给出正确及错误的长度规定。

00010A	FIELD1	7
A		
00020A	FIELD2	7
A		
00030A	FIELD3	R +7

图 4-2 正确和错误的长度规定

注：FIELD1 的长度规定不正确，FIELD2 和 FIELD3 是正确的。

对浮点字段，要在 30-34 列规定的长度上加 7，这额外的 7 位用做有效数字、符号、小数点或逗号、指教字符、指数符号及指数。某些时候，如果规定了长度值，数据库文件中字段的某些键字不能包括在打印文件中，详细信息请看 4.3.8。

#### 4.3.10 数据类型（第 35 列）

用这列规定数据的类型，打印文件有效的数据类型是：

S	区位十进制
A	字符
F	浮点数

图 4-1 给出数据类型编码的例子

如果没有规定数据类型，也没有从被引用字段复制，则系统的缺省值为：

如果小数位（36-37 列）为空白，则为 A（字符）；

如果小数位（36-37 列）有 0-31 的数，则为区位十进制。

注：

1. 在 37 列填 0 表示是整数；
2. 35 列为 F 时，表示是单精度浮点数，要用 FLTPCN 键字规定双精度，或修改已经定义的浮点数为双精度。
3. 浮点值由五部分组成：(a) 有效数字符号；(b) 有效数字；(c) 指教字符；(d) 指数符号；(e) 指数。

正值的有效符号不打印，有效数字是由 30-34 列的长度决定的，小数位和逗号由 36-37 列的小数位确定。总打印指教字符和符号，指数为 3 个数字。

浮点字段的打印长度的 30-34 列规定的长度多 7，这 7 个多出的位用于有效符号、小数点或逗号、指教字符、指数符号和 3 个指数数字。

#### 4.3.11 小数位（36-37 列）

用这二列规定出现在程序中的区位十进制字段的小数位和字段的数据类型。

如果这二位是空格, OS/400 认为字段是字符型。

如果这二位有数字, OS/400 认为字段是区位十进制数。

它规定了小数的位数, 必须小于或等于字段长度, 最大为 31, 图 4-1 给出小数位编码的例子。

如果用被引用字段, 不用规定小数位, 也可以修改引用的小数位。规定新值可替换被引用字段的小数位规定, 用+号或-号可以修改小数位, 例如+4 表示比被引用字段的小数位多 4.

注: 高级语言可能有不同的长度规定, 在小数位上也可有限制, 要遵守这些语言的规则。

#### 4.3.12 用法 (第 38 列)

用这列规定命名字段是仅输出字段或是程序一系统字段, 不要在这位上用常数字段。

打印文件的有效值是:

空格或字符 O	仅输出
P	程序一系统字段 (特殊的输出字段)

仅输出字段在程序打印记录时, 把数据从程序传给打印机。程序一系统字段是一个命名的数值或字母数字的仅输出字段, 用来在程序和系统之间传递数据, 它是不打印的, 程序也能用输出操作往字段中送数据, 但这个数据也不打印。

下列规则适于程序一系统字段:

- 字段总要命名
- 不用位置项
- 长度、数据类型和小数位与其它命名字段一样规定
- 程序一系统字段必须做为 PAGSEG、OVERLAY、GDF、LINE 或 POSITION 键字的参数在同一记录格式中定义。在其它键字中它就不是一个合法的参数。如果它不是在上述某个键字中规定的, 就会出现严重错误信息。
- 程序一系统字段可在缓冲区的任何地方出现。
- 程序一系统字段的有效键字为 ALIAS、INDTXT、REFFLD 和 TEXT。

#### 4.3.13 位置 (39-44 列)

用这几列规定在打印页上出现的字段的起始位置, 可规定行 (39-41 列) 和列 (42-44 列) 位置, 要考虑下列情况:

在规定行号时, 字段可用任何顺序规定, 当放到打印文件中时, 再按行号排序。

当字段或 SPACE/SKIP 有条件限制时, 在诊断覆盖字段时, 数据描述处理器按选择了它们来对待。

对有几个字段的记录格式, 当有 SPACE/SKIP 键字的字段有条件限制或有字段层的 SPACE/DKIP 条件限制, 则出现警告信息, 指出发生了字段覆盖, 但没诊断信息。

由于字段的长度、位置或有关的 SPACE/SKIP 键字引起打印超出一页时, 出现警告信息。

最大行数为 255, 实际一页的最大行数要少, 这取决于 CRTPRTF 中 PAGESIZE 参数和每英寸多少行的规定。

最大列数是 255, 实际一页的最大列数取决于 CRTPRTF 中 PAGESIZE 参数和每英寸多

少字符，也与使用的 FONT 有关。

溢出行（一页中最后一行）取决于 CRTPRTF, CHGPRTF, OVRPRTF 中的 OVRFLW 和 PAGESIZE 参数，对外部定义的文件，RPG 不控制页溢出。

当高级语言编译用到外部定义文件时，字段用 DDS 的顺序在输出记录区中排序，要参考相应的高级语言手册。如果字段重叠，打印机要重印。字段长度和输出缓冲区位置请看 CRTPRTF 命令生成的编译输出中的扩充源语句部分。如果规定了 POSITION 键字，这些位置必须是空白。

行：(39 - 41) 这里规定字段打印的起始行，必须右对齐，前导零是可选的。行号可以是记录中的命名或非命名字段。如果要对一个字段规定行号，可以在 39 - 41 列给出行号也可以在 42 - 44 列上加一个值(+n)。如果在记录层或字段层规定了 SKIP 或 SPACE 键字，则不允许规定行号。但在文件规定了 SKIP 键字，行号是合法的（在文件层不允许 SPACE 键字）。

列：(42 - 44) 规定打印字段的起始列，你规定的位置是根据打印文件每英寸的字符数及使用的字型。如果打印文件用\*DEVD 字型，用一个编码字型，或用一个字型字符集，正文字段用空格(X'40') 来给定义的列定位，没正文的字段，例如条形码是用 10 CPI 的隐含值定位的。如果使用预留空间字型，将产生同行上的一些列。在这种情况下建议使用 POSITIN 键字。这二位的内容必须右对齐，前导零是可选的。

如果在记录中规定了一个字段且字段有效。在同一记录中这个字段的后续字段的位置可如下定位：行号处空格，在 42 - 44 列规定加上的值(+n)，加的值指出在前面字段的结尾和正定义字段开始之间的空格数。加的值在 0 - 99 之间。如果规定一个加的值，行号项必须是空格，假如加的值导致隐含的 SPACE 操作，而这个记录格式又没有用行号，则必须用 SKIP/SPACE 键字来加上空行。

在给图形字段定位时，系统使用 CRTPRTF 命令中规定的页宽度做为宽度限制。例如，用户规定页宽 132，如果用引用位置而不是自己编码位置生成记录格式，字段要在 132 列处卷回。如果格式中的字段规定了行号，重叠的字段在下一行卷回，如果没规定行号，数据将卷回复盖同一行上的起始数据。

一旦计算了列，要保存实际的值就象自己编码规定的一样对待，这样，假定字段卷回且现在位于第一行第五列，字段就保留在这个位置上，即使用 CHGPRTF 命令增加了页宽度也是如此。

图 4-3 解释了这个问题及两种可能的情况（对 132 个字符的页宽）。

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

```
A*
A*      位置上加的值引起 PRFLD1 复盖 PRFLD2
A*
A       R PRTOUT           SKIPB(1)
A       PRFLD1      130      1TEXT(' START LOC 1,1 END LOC 1,130')
A       PRFLD2      130      +2TEXT(' OVERLAPS PRFLD1')
A*
```

```

A* 避免复盖的方法之一是对 PRFLD1 规定 SPACEA 或者 SKIPA,
A* 或对 PRFLD2 规定 SPACEA 或 SKIPA
A*
A      R PRTOUT2           SKIPB(1)
A      PRFLD1A    130      1
A      PRFLD2A    130      +2SPACEB(1)
A*
A* 方法之二是不用 SKIP/SPACE
A*
A      R PRTOUT3
A      PRFLD1B    130      1  1
A      PRFLD2B    130      +2
A

```

图 4-3 规定行和列的位置

如果在 CRTPRTF, CHGPRTF, OVRPRTF 命令中规定了 FOLD(\*YES), 则超出结束行的字段继续在下一行打印。在行结束时产生中断, 但可用 BCKFOLD 键字让其保持一个空白。如果 FOLD(\*NO)有效, 则超出结束行的字段被截断。

DDS 决定哪些字段出现在同一页上。数据描述处理器在生成文件期间诊查重叠, 有可选键字的键字或字段都假定为有效。这样, 就不能检查没被选择的字段或键字引起的重叠。在图 4-4 中, 除非指示器 01 为 ON, 就没有重叠字段。当 01 为 ON 时, F1、F3 和 F4 重叠, 那么就有诊断信息送出来, 指出在运行时 SPACE/SKIP 键字选择和限制的字段能导致字段的重叠。

在某些打印机上, 在 DDS 中同一行的字段按从右往左顺序定义时打印速度要快一些。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A*
A* OVERLAPPING FIELDS ONLY IF IND 01 IS ON
A*
A      R REC1           SKIPB(1)
A      F1        1        1
A  NO1    F2        1        1SPACEB(1) SPACEA(1)
A      F3        1        1
A      F4        1        1
A  NO1                           SPACEB(1)
A

```

图 4-4 复盖字段

#### 4.4 键字项 (45 – 80 列)

这部分包括定义打印文件有效的键字项, 这些键字写在 DDS 的 45 – 80 列中 (功能),

规定键字的一般规则请看 1.2。

下列键字对打印文件有效：

ALIAS	EDTCDE	PAGNBR
BARCODE	EDTWRD	PAGRTT
BLKFOLD	ENDPAGE	PAGSEG
BOX	FLTFIXDEC	POSITION
CDEFNT	FLTPCN	PRTQLTY
CHRID	FNTCHRSET	REF
CHRSIZ	FONT	REFFLD
COLOR	GDF	SKIPA
CPI	HIGHLIGHT	SKIPB
CVTDTA	INDARA	SPACEA
DATE	INDTXT	SPACEB
DFNCHR	INVMMAP	TEXT
DFT	LINE	TIME
DLTEDT	LPI	TRNSPY
DRAWER	MSGCON	TXTRTT
DTASTMCM	OVERLAY	UNDERLINE

#### 4.4.1 ALIAS (替换名)

用这个字段层键字给字段规定一个替换名，当程序编译时替换名带入程序中代替 DDS 的字段名。所用的高级语言编译程序决定是否用替换名，支持 ALIAS 的高级语言详细情况，请参考各自的语言手册。

键字的格式为：

ALIAS (替换名)

替换名参数必须不同于同一记录格式中的所有替换名和所有的字段名，假如有相同的名字，则出现错误。

替换名不能在 DDS 中使用，也不能用于其它 AS/400 功能（例如，用它做索引字段名，做 REFFLD 键字的字段名，或做 CPYF 命令的字段名）。

当引用一个有 ALIAS 规定的字段时，除非引用它的字段明显的规定了 ALIAS 键字，要复制被引用字段的 ALIAS 信息。

可选指示器对此键字无效。

图 4-5 给出如何规定 ALIAS 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00070A      FIELD A      25A      1 2ALIAS(CUSTOMERNAME)  
A
```

图 4-5 规定 ALIAS 键字

#### 4.4.2 BARCODE (条形码)

用这个字段层键字打印一个用户定义的条形码字段，它仅对 IPDS\*打印机和设备类型为 \*IPDS 或 \*AFPDS 的打印文件有效。

键字的格式为：

```
BARCODE(条形码 ID [高度] [[*HRZ | *VRT]  
[*HRI | *HRITOP | *NOHRI] [*AST | *NOAST]  
[检测数字] [部件宽度] [宽/窄比率]])
```

条形码 ID 参数是必须的，有效值在图 4-6 列出。

高度参数是可选的，如要规定，必须是这个键字的第二个参数，有效值是 1 – 9 行，规定的高度不包括条形码下的可读解释。如果没有规定，则使用缺省的高度。

其余六个参数可用任何顺序规定，用它们可以指定：

- 水平或垂直打印条形码，缺省值为 \*HRI (水平)；
- 是否包括可读的条形码解释，缺省值是在条形码底部打印解释 (\*HRI)；
- 指出可读的解释打印在条形码的顶部 (\*HRITOP)；
- 是否包括 CODE30F9 条形码上的星号，缺省值是不包括 (\*NOAST)；
- 选择检测数字，这是一个不包括 X ‘FF’ 的一个字符的十六进制值；
- 规定窄条形/空宽度，用格式表达式的方式规定 (\*WIDTH 值)，详细内容请看 1.2，有效值为 0.007-0.208；
- 规定宽条形/空与窄条形/空的比率，用格式表达式方式规定 (\*RATIO 值)，这个参数的有效值为 2.00-3.00。

注：整个条形码宽度取决于：

- 用户 DDS 中的比率和宽度参数；
- 条形码中的实际客户数据；
- 打印机硬件的限制。例如：PEL 密度、打印针等等。

对 4234 和 4224 打印机忽略宽度和比率参数。详细内容请看 4224 打印机产品手册及程序设计说明。

如果用了不支持条形码 ID 的可选参数，则打印机忽略这个可选参数。

如果试图在不支持条形码的打印机上打印，码中的数字以说明对待，并给出诊断信息指出不能打印条形码。

对字段规定的行和列做条形码的左上角，由于 DDS 中规定的行基行（打印字符的假设行），它用作条形码的上边界，条形码从你规定的行底部往下延伸。

图 4-6 给出 BARCODE 字段的有效数据类型和字段长度。

图 4-6 有效的条形码定义		
条形码标识	数据类型	字段长度
MSI	S	1 到 31(1)
UPCA	S	11
UPCE	S	10
UPC2	S	2

UPC5	S	5
EAN8	S	7
EAN13	S	12
EAN2	S	2
EAN5	S	5
CODEABAR	A	1 到 50
CODE128	A	1 到 50
CODE30F9	A	1 到 50
INTERL20F5	S	1 到 31
INDUST20F5	S	1 到 31
MATRIX20F5	S	1 到 31
POSTNET	S	1 到 31
RM4SCC	A	1 到 31

注：4234 打印机仅支持 14 个数字

图 4-7 说明支持的条形码

图 4-7 DDS 支持的条形码					
条形码	每个码的数字	允许字符的范围	生成的缺省 检测数字	打印的缺省 检测数字	有效的检 测数字
MSI (changed Plessey)	31(1)	0 through 9	2 Modulus 10	No	01 through 09
UPC-A	11	0 through 9	1	No	00
UPC-E	10	0 through 9	1	No	00
UPC-2 digit add on (must follow a UPC A or E bar code)	2	0 through 9	No	No	00
UPC-5 digit add on (must follow a UPC A or E bar code)	5	0 through 9	No	No	00
EAN-8	7	0 through 9	1	Yes	00
EAN-13	12	0 through 9	1	Yes	00
EAN-2 digit add on (must follow an EAN 8 or 13 bar code)	2	0 through 9	No	No	00
EAN-5 digit add on (must follow an EAN 8 or 13 bar code)	5	0 through 9	No	No	00
INDUST20F5 or industrial 2 of 5	31	0 through 9	1	Yes	01 02
MATRIX20F5 or matrix 2 of 5	31	0 through 9	1	Yes	01 02
INTERL20F5 or interleaved	31	0 through 9	1	Yes	01 02

2 of 5					
CODEABAR	Up to 50 characters	0 through 9 A through D (begin/end only), -, ., \$, /, +, and :	1	Yes	01 02
CODE128	Up to 50 characters	Refer to Appendix G, "CODE128 Character Set"	1	No	01(2) 02
CODE30F9 or code 3 of 9	Up to 50 characters	0 through 9, A through Z (upper case only), -, ., \$, /, +, %, and a blank	No	No	01 02
POSTNET	Up to 31 characters	0 through 9	1	Yes	Ignored
RM4SCC	Up to 31 characters	0 through 9 A through Z	1	Yes	Ignored

图 4-7 DDS 支持的条形码

注：1. 4234 打印机只支持 14 个数字；

2. 检测数字值 01 对某些打印机无效。

CODEABAR 字段数据必须用 A、B、C 或 D 开头，以 A、B、C 或 D 结束。例如，A11224455C 或 D33447799D。

不要和 CHRSIZ, CHRID, CVTDTA, DATE, EDTCDE, EDTWRD, FONT, HIGHLIGHT, PAGNBR, TIME, 或 UNDERLINE 键字一起在同一字段指定 BARCODE。

有关 IPDS 条形码命令本身的内容，请看 CVTDTA 键字。

如果在记录层规定 CHRSIZ，它适用于记录中的所有字段。如果在其中一个字段中规定 BARCODE，则是不允许的。BARCODE 不能与 BLKFOLD、CPI 或 DFNCHR 一起在同一记录格式上指定。

在数值字段中规定 BARCODE 时，小数位必须是零。

当在常数字段指定 BARCODE 时，有效的条形码 ID 是 CODEABAR、CODE128 和 CODE30F9，也必须显式或隐式的规定 DFT 键字。

在文件中规定 BARCODE 时，在 CRTPRTF 命令中要规定 DEVTYPE(\*IPDS) 或 (\*AFPDS)。

BARCODE 只能用在数据类型为 S 和 A 上。

可选指示器对此键字无效。

用户规定的检测数字是无效检测。如果无效，则会引起条形码错误，IPDS 参考手册中有条形码及有效检测数字的详细内容。

图 4-8 给出如何规定 BARCODE 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          FIELD1      11S 0 2 4BARCODE (UPCA 6)
A          FIELD2      3A    10 6BARCODE (CODE30F9 4 *NOHRI-
```

```

A *AST X'02'
A FIELD3    10S 0 4 5BARCODE(UPCE 6 (*RATIO 2.75) *HRZ +
A           X'00' (*WIDTH .02))
A FIELD4    10A 0 5 5BARCODE(CODEABAR 1 (*RATIO 2.1) +
A           *HRITOP)
A           6 5'01234567'
A           BARCODE(CODE128 2 *HRITOP *HRZ +
A           (*WIDTH 0.1) (*RATIO 2) X'01')
A

```

图 4-8 规定 BARCODE 键字

#### 4.4.3 BLKFOLD (空白折叠)

对命名字段使用这个字段层键字，在溢出发生在后续行时，在空白处折叠而不是在行的末尾折叠。如没规定此键字，在物理打印行的末尾发生折叠。

这个键字无参数。

仅在 CRTPRTF、CHGPRTF、OVRPRTF 命令中规定 FOLD(\*YES)时，BLKFOLD 才有效。如果规定 FOLD(\*NO)，BLKFOLD 无效。

当使用 BLKFOLD 时，字段的长度不增加，这样，输出数据的一部分可能会截断。

不能对浮点字段指定 BLKFOLD 键字。

BLKFOLD 只适用于 SCS 打印机。如果在用 DEVTYPE(\*IPDS)或(\*AFPDS)生成的文件规定了 BLKFOLD 键字，那么在生成时会有警告信息。BLKFOLD 不能与 IPDS 打印机键字或支持 AFP 打印的键字一起在同一记录格式中规定。如果文件的记录格式包括混合的 SCS 和 IPDS 打印机键字或 AFP 支持的键字，则文件不能生成。

可选指示器对此键字无效，但可选指示器能用在与这个键字有关的字段上做条件。

图 4-9 给出如何规定 BLKFOLD 键字

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      FIELD1      180A      10 20BLKFOLD
A

```

图 4-9 规定 BLKFOLD 键字

#### 4.4.4 BOX (方框)

用这个记录层键字打印一个矩形。

这个键字的格式为：

```

BOX (第一个下角位 ! & 下角位字段
      第一个角横位 ! & 角横位字段
      对角下位 ! & 对角下位字段
      对角横位 ! & 对角横位字段

```

行宽 | & 行宽字段)

前四个参数是必须的，它定义方框的对顶角。

也可用常量、程序—系统字段或它们的组合做为角位置参数。例如：

BOX (1.2 0.5 5.1 6.3 0.2)

BOX (1.2 &Field2 5.1 &Field4 0.2)

BOX (&Field1 &Field2 &Field3 &Field4 0.2)

第一个角下位参数规定 BOX 相对于 CRTPRTF 中 FRONTMGN 或 BACKMGN 参数指定的边界的纵向起始点，有效值是 0-57.790cm (0-22.750 英寸)。当这个字段被定义为程序—系统字段时，字段必须和 BOX 在同一记录格式中，必须是 5 位长、3 位小数、数据类型为 S、用法为 P 的字段。

第一个角横位参数规定 BOX 相对于 CRTPRTF 中 FRONTMGN 或 BACKMGN 参数指定边界的横向起始点，有效值是 0-57.790cm (0-22.750 英寸)。当这个字段被定义为程序到系统字段时，字段必须和 BOX 在同一记录格式中，必须是 5 位长、3 位小数、数据类型为 S、用法为 P 的字段。

对角下位参数规定，BOX 相对于 CRTPRTF 中 FRONTMGN 或 BACKMGN 参数指定边界的纵向起始点，有效值是 0-57.790cm (0-22.750 英寸)。当这个字段被定义为程序到系统字段时，字段必须和 BOX 在同一记录格式中，必须是 5 位长、3 位小数、数据类型为 S、用法为 P 的字段。

对角横位参数规定，BOX 相对于 CRTPRTF 中 FRONTMGN 或 BACKMGN 参数指定边界的横向起始点，有效值是 0-57.790cm (0-22.750 英寸)。当这个字段被定义为程序到系统字段时，字段必须和 BOX 在同一记录格式中，必须是 5 位长、3 位小数、数据类型为 S、用法为 P 的字段。

行宽参数是必需的，用来定义行的宽度，有效值是 0.001 到 57.790cm (0.001-22.750 英寸)，可如下规定一些特殊值：

值	行宽
*NARROW	12/1400 英寸 (0.008in,0.022cm)
*MEDIUM	24/1400 英寸 (0.017in,0.042cm)
*WIDE	36/1400 英寸 (0.025in,0.064cm)

当这个字段被定义为程序—系统字段时，字段必须和 BOX 在同一记录格式中，必须是 5 位长、3 位小数、数据类型为 S、用法为 P 的字段。

上述特殊值不能用程序—系统字段规定。

注：

- 1.CRTPRTF 命令中的 UOM 参数决定 BOX 键字中参数的计量单位，如果参数值超出范围，当生成假脱机文件时会给出标记。
- 2.根据打印机硬件，比相应的值小 0.004 英寸 (0.010cm) 的线可能打印不出来，这时不会发出任何信息。

线的宽度要能放在框里边。

在记录格式中规定 BOX 键字时，记录中的所有字段必须用 POSITION 键字定位，详细

内容请看 4.4.36。

如果在记录格式中，规定了常数字段又规定了 BOX，则会产生错误信息。

如果 BOX 超出页边界，在应用程序运行期间会发出错误信息。

在文件中指定 BOX 时，要在 CRTPRTF 命令中规定 DEVTYPE(\*AFPDS)。如果 DEVTYPE 改成其它不是\*AFPDS 的话，则忽略键字，在打印时发出警告信息。

在一个记录中可以多次指定 BOX 键字。

BOX 不能与 SPACEA、SPACEB、SKIPA、SKIP 键字一起规定。

注：使用 BOX 键字时要有 PSF/400，如果没有安装它，就不能在打印文件中用此键字，也不能规定 DEVTYPE(\*AFPDS)。

可选指示器对此键字有效。

图 4-10 给出如何规定 BOX 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A*  
A       R BOX1           BOX(1.2 0.5 5.1 6.3 0.2)  
A*  
A       R BOX2           BOX(2 5 5.0 3.33 *WIDE)  
A                   BOX(0.5 0.1 2.1 2.0 0.09)  
A*  
A       R BOX3           BOX(0 0 8.5 11.0 0.5)  
A   01  
A*  
A  
A
```

图 4-10 规定 BOX 键字

注：C RTPRTF 命令中的 UOM 参数决定参数值的计量单位。

BOX1 在由 CPTPRTF 命令的 FRONTMGN 或 BACKMGN 参数确定的边界向下 1.2 个单位，横向 0.5 个单位打印框的第一个角，在命令确定边界向下 5.1 个单位，横向 6.3 个单位打印框的对顶角，框边界为 0.2 个单位宽。

BOX2 打印两个框，第一个框起始于命令中规定边界向下 2 个单位，横向 5 个单位打印。框的对顶角在命令中规定边界向下 5 个单位，横向 3.33 个单位打印，框边界的宽度由特殊值\*WIDE 确定。

第二个框起始于命令中规定边界向下 0.5 个单位，横向 0.1 个单位。

对顶角在命令规定边界向下 2.1 个单位，横向 2.0 个单位，框边界的宽度是 0.09 个单位宽。

#### 4.4.5 CDEFNT (编码字型名)

使用这个字段层或记录层的键字规定打印编码字型的名字、常数字段或记录中的字段。

键字的格式为：

CDEFNT ([库名/] 编码字型名)

编码字型名是必需的参数，长度最多为 8 个字符。

库名是可选的，用来限定编码字型名。如果没有指定库名，就用\*LIBL，如果用\*LIBL，在检索编码字型时要把它所在的库加到库列表中去，用库名参数能更快的找到编码字型的位置，但也用库列表来给字符集及由编码字型名定义编码页。

注：如应用程序使用私有资源（例如：字型、页段、覆盖或没在系统分配的 GDF 文件），要注意以下事情：当引用这些资源时，如果用\*LIBL 或没有指定库名，那么这些资源要通过应用程序生成假脱机文件时用的库列表来使用。

编码字符集应在打印时有效，如果无效或没有分配资源，则发出错误信息。

在文件中指定 CDEFNT 时，要在 CRTPRTF 命令中指定 DEVTYPE(\*AFPDS)。如果 DEVTPPE 改成其它不是\*AFPDS 的类型，则忽略此键字且在打印时发出警告信息。

CDEFNT 不能与 FONT 或 FNTCHRSET 在同一层上指定。

注：用这个键字时要求有 PSF/400，如果没有安装，就不能用此键字和规定 DEVTYPE(\*APPDS)。

可选指示器对此键字有效。

图 4-11 给出如何规定 CDEFNT 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A*  
A          R REC1  
A          FLD1      8A      10 13CDEFNT (QFNTCPL/XOBTRR)  
A*  
A
```

图 4-11 规定 CDEFNT 键字

FLD1 规定在库 GFNTCPL 中找到的编码字型 XOBTRR。

#### 4.4.6 CHRID (字符标识)

用这个字段层键字规定这个字段能用的非设备缺省图形字符集和编码页，当打印扩充的字母数字时（例如：ü或有变音符（'）的 C）规定此键字是很重要的。

这个键字没有参数。

如果没有规定 CHRID，字段中的数据用打印设备的字符集打印。数据如何打印是依赖于设备所用的与编码页对应的源编码的编码点。例如：由于在 u.s 字符集中 X ‘51’ 是一个空的编码点，所以 “/” 打印成一个空格。

CHRID 对常数字段或数字字段无效。

对 CRTPRTF、CHGPRTF、OVRPRTF 命令中规定 CHRID(\*JOBCCSID)的打印文件的字段忽略 CHRID 键字。这时，所有打印的数据都假定使用打印文件当前作业的 CCSID。有关

CCSID 支持的详细内容，请看各自语言支持一书。

对 SCS 打印机，不能对同一字段指定 CHRID 和 TRNSPY 键字。用 IPDS 打印机，可以这样规定，但会有信息告诉你，不能把 DEVTYPE 改为 \*SCS。

对 DEVTYPE(\*AFPDS)打印机，CHRID 仅支持使用寄存器字型 ID 的文件。如果文件使用下载编码字型或字符集/编码页，则忽略此键字且有信息发出。

如果对一个记录格式规定 DFNCHR 键字，则不能在记录中用 CHRID，如果在文件层规定了 DFNCHR，也不能在文件中用 CHRID。

可选指示器对此键字无效，但可用在使用此键字的字段中。

图 4-12 给出如何规定 CHRID 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1  
00020A      TITLE        40      1 20CHRID  
A
```

图 4-12 规定 CHRID 键字

TITLE 是命名字段，由于规定了 CHRID，打印机打印相应的字符。详细内容请看打印设备程序设计一书。

#### 4.4.7 CHRSIZ (字符尺寸)

使用此字段层或记录层键字来扩大记录或字段的宽和高。它仅对 IPDS 打印机和 DEVTYPE(\*IPDS)或(\*AFPDS)的打印文件有效。

键字的格式为：

CHRSIZ (宽 高)

宽、高的有效值为 1.0-20.0。

在扩大之前，要先做好选择的格式。例如规定字型或用 EDTCDE 或 CDTWRD 做编辑。

如果在记录层规定 CHRSIZ，它适用于记录中所有没在字段层规定 CHRSIZ 的字段，当在 CHRSIZ 中规定一个数字字型（例如：011-Courier 10），则打印机用硬件字型定标（仅用整数值）。

图型字型也能规定 CHRSIZ，当用 CHRSIZ 规定 GDDM 图型字型时（例如，ADMMVSS），系统定标为图型字型，可用十进制数定标图型字型。

建议在用 CHRSIZ 时不要使用 FONT(\*DEV0)。如果这样做了，用 CHRSIZ 规定的字段就定位在 10-行矩字型的设备描述的字型中。

为扩大一个字段，CHRSIZ 使用当前的字型和每英寸多少行。例如：如果指定 FONT(011)，10-行矩字型、打印文件为 LPI(6)，对一个 10 字符字段指定 CHRSIZ(3 3)，这样把字段扩大为 3 英寸宽（30 个字符/每英寸 10 个字符），1/2 英寸高（3 行/每英寸 6 行）。

注：如果当前字型是编码字型或字型字符集/编码页，当定位字段时假定 10 - 行矩。

如果在记录格式中规定 FONT(222)，15 - 行矩和 LPI(4)，上面的 10 个字符字段就扩大

为 2 英寸宽 (30 个字符/每英寸 15 个字符), 3/4 英寸高 (3 行/每英寸 4 行)。

在 CHRSIZ 中规定十进值时, 不能规定在 FONT 键字中规定硬件字型, 如果这两个键字在同一字段上规定 (不论是记录层还是字段层), 都不能生成文件。

注: 当规定 CHRSIZ 键字又同时规定 FONT(\*DECTOR) 时, 4234 打印机使用缺省的编码页。

如果规定字型是印刷字型或打印机是 3812 或 3816 打印机, 那么就不能在这些打印机上定标印刷字型, 这是对这些打印机的限制, 印刷字型如下:

751	1351
1051	1653
1053	2103
1056	

注: 由于打印机的限制, CHRSIZ 不支持某些打印机。例如, 3825、3827 和 3900 仅支持下载字型。

生成文件时, 对扩充高度的超出和重叠页长并不发出诊断信息, 但对超出页宽有诊断, 所用的字段长度是 DDS 的字段长乘以在 CHRSIZ 中指定的扩大宽度, 取整数值。

对这个键字有效的数据类型为 A、F 和 S。

可选指示器对此键字无效。

图 4-13 给出如何规定 CHRSIZ 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A           R RECORD1                      CHRSIZ(3 3)
A 02 03                  FONT(222)
A           FIELD1        3A    6 01
A           FIELD2        6A    16 01CHRSIZ(2.5 2)
A                   FONT (ADMMVSS)
A           FIELD3        6S 0  20 01CHRISIZ(1 1)
A
```

图 4-13 规定 CHRSIZ 键字

#### 4.4.8 COLOR (颜色)

用这个字段层键字规定打印机支持的字段颜色, 它仅用于 4224 打印机。如果没规定 COLOR 或打印机设备不支持此键字, 则用黑色 (缺省值)。

键字的格式为:

COLOR(blk | brw | grn | pnk | red | trq | ylw)

键字中仅可规定一个参数。

如果在同一记录格式中同时规定了 COLOR、BLKFOLD、CPI 或 DFNCHR 键字, 则不能生成文件。

对 IPDS 和 IPDS AFP(\*YES) 的打印机, COLOR 有效, 如果在 CRTPRTF 命令中规定 DEVTYPE(\*SCS), 发出警告信息, 但文件仍然能生成。

对此键字有效的数据类型为 A、S 和 F。

如对一个字段多次指定 COLOR，必须用可选指示器，如果同时多个 COLOR 都有效，则只用第一个有效的 COLOR，对同一字段不能多次指定同一颜色。

图 4-14 给出了对一个字段规定 COLOR 的情况。

可选指示器对此键字有效。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A 99                      1 3'PRINT RED TEXT'
A                           COLOR(RED)
A      FIELD1      3A    12 01TEXT(' PINK IF 02, +
A                           YELLOW IF 07, +
A                           BLACK IF NEITHER')
A 02                      COLOR(PNK)
A 07                      COLOR(YLW)
A
```

图 4-14 规定 COLOR 键字

图中，如果指示器 99 为 ON，常数字段 ‘PRINT RED TEXT’ 用红色打印，根据指示器 02 和 07 的设置，FIELD1 用粉红、黄或黑色打印。

#### 4.4.9 CPI (每英寸字符数)

这是记录层或字段层键字，用它来指定水平打印密度，可以做：

- 给用 DFNCHR 键字生成的打印图型或标识加黑；
- 在打印格式中放更多的数据；
- 满足对格式更多的需求。

键字的格式为：

CPI(10 + 15)

10 或 15 规定每英寸的字符数。

此键字只对 5224 或 5225 SCS 打印机有效。如果没有规定 CPI，则用 CRTPRTT、CHGPRTF 或 OVRPRTF 命令的 CPI 参数中规定的值。如果在记录层规定了 CPI，除在字段层规定了 CPI 的其它所有记录中的字段都使用同一密度。如果在字段层规定了 CPI，可对在同一行上打印的字段规定不同的密度。对每个字段指定的位置(42-44 列)是依据 CRTPRTF、CHGPRTF 或 OVRPRTF 命令中的 CPI 参数。(看图 4-15 和 4-16)。

在字段层规定 CPI 时，不诊断重叠字段。

如果在用 DEVTYPE(\*IPDS)或(\*AFPDS)的文件中规定 CPI，则在生成时会有警告信息，要修改 CPI。

必须规定 FONT 键字 (见 4.4.23 中的 ‘FONT’)。

CPI 不能与 DRAWER 键字一起对同一记录格式规定。

可选指示器对此键字有效。

图 4-15 给出如何在记录层指定 CPI 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1  
00020A 02          CPI(15)  
00030A      FLD1      20      3  1  
00040A      FLD2      5  0      +2  
00050A      R RECORD2          SPACEB(1)  
00060A      FLD3            1  
A
```

图 4-15 规定 CPI 键字（例 1）

图中，如果指示器 02 为 ON，FLD1 和 FLD2 都用每英寸 15 个字符打印，如 02 为 OFF，FLD1 和 FLD2 用 CRTPRTF、CHGPRTF 或 OVRPRTF 命令的 CPI 参数规定的密度打印。

在打印 RECORD2 之前要有一空行，记录中的 FLD3 用 CRTPRTF、CHGPRTF 或 OVRPRTF 命令的 CPI 参数规定的密度打印。

图 4-15 给出在两个用 10CPI 打印的字段中间打印一个 15CPI 的字段的情况。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RCDA          SPACEA(1)  
00020A      FLD1      10      1  
00030A      FLD2      10      11CPI(15) 1  
00040A      FLD3      10      21  
A
```

图 4-16 规定 CPI 键字（例 2）

图中，1 的所有位置都用 10CPI（命令中规定）。这样，RCDA 如下打印：

11111111122222222222 3333333333

FLD2，压缩在 15CPI，用的地方比 FLD1 和 FLD3 都少，为了避免间隔太大，可把 FLD3 往左一些，要计算 FLD3 的位置，把 FLD2 的长度加上 FLD2 的位置。要计算 FLD2 的长度用下面的公式：

$$\frac{\text{规定的长度} \times \text{文件密度}}{\text{字段密度}} = \text{打印的长度}$$

对 FLD2：

$$\frac{10 \times 10}{15} = \frac{10 \times 2}{3} = 6.67 \text{ (舍入为 7)}$$

把 7 加上 11 (FLD2 的列) 如下:

$$11+7=18$$

则例子的正确的 DDS 为:

R RCDA		SPACEA(1)
FLD1	10	1
FLD2	10	11CPI(15)
FLD3	10	18

则打印的结果为:

11111111122222222222 3333333333

图 4-17 给出在两个 CPI 为 15 的字段中间打印 10CPI 字段的情况:

...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8			
00010A	R RCDB	SPACEA(1)	
00020A	FLD4	10	1
00030A	FLD5	10	11CPI(10) 1
00040A	FLD6	10	21
A			

图 4-17 规定 CPI 键字 (例 3)

图中, 1 的位置是用 15CPI 打印的 (在命令中规定), 系统用下列公式计算两个 15CPI 中间的用 10CPI 打印字段的起始位置:

$$\frac{2 \text{ (指定的位置-1)} + 1}{3} = \text{ 打印位置 (有分数要舍入)}$$

对 FLD5:

$$\frac{2 \text{ (11-1)} + 1}{3} = 7.67 \text{ (舍入为 7)}$$

舍入的结果导致 FLD4 和 FLD5 重迭, 打印出来如下所示:

44444444455555666666666666

为了避免重叠, 把 FLD5 往右一些。(12 列)

要计算 FLD6 的位置, 要把 FLD5 的长度加到 FLD5 的位置上, 要计算 FLD5 的长度, 用下面公式:

$$\frac{\text{规定的长度} \times \text{文件密度}}{\text{字段密度}} = \text{打印的长度}$$

对 FLD5:

$$\frac{10 \times 15}{10} = 15 \text{ (如需要, 要舍入)}$$

把 15 加到 FLD5 的列数上:

$$15+12=27$$

这样例子的 DDS 为:

图

则打印的记录格式为:

444444444 5555555555 6666666666

图 4-18 给出系统在字段打印格式的右边截断或折叠, CPI 键字的影响。

这依赖于 CRTPRTF、CHGPRTF、OVRPRTF 命令中 FOLD 和 PAGESIZE 键字的值。

00010A	R RCDC	SPACEA(1)	
00020A	FLD7	10	140
00030A	FLD8	10	150CPI(10)
A			

图 4-18 规定 CPI 键字 (例 4)

图中, 如果文件用宽度为 160, CPI 为 15 的格式打印, FLD7 和 FLD8 打印如下:

FLD7 从 140 列打印, 长度为 15CPI 的 10 (16.9mm 或 0.667 英寸)

FLD8 从 150 列打印, 长度为 10CPI 的 10 (25.4mm 或 1 英寸)

在 150 列打印用 15CPI 计算的 FLD8 超出右边界, FLD8 或者截断或折叠到下一行打印。  
(依据命令中的 FOLD 参数), 要计算 FLD8 的长度, 用下面公式:

$$\frac{\text{规定的长度} \times \text{文件密度}}{\text{字段密度}} = \text{打印的长度}$$

对 FLD8:

$$\frac{10 \times 15}{10} = 15 \text{ (如需要, 舍入为低值整数)}$$

注: 以 15CPI 打印的文件包含 10CPI 字段时, 格式的右边界根据下边公式的计算结果  
调整所有的字段:

$$2 \times (\text{字段规定的长度}-1) + 1$$

#### 4.4.10 CVTDTA (转换数据)

这个字段层键字在字段传送给打印机时把字符数据转换成十六进制数据，可用它定义：

- 格式中对字母头加图案或标识；
- 替换字符集或符号（例如注册符号）；
- 出现的物理格式（在格式中或发票的两列中间加竖线或横线）；
- IPDS 条形码命令。

此键字无参数。

在 SCS 打印机上，仅在用 DFNCHR 键字时才规定 CVTDTA。另外，在为没分配的编码点定义字符时，也要用 CVTDTA，一个编码点是 256 中的一个值，用它来分配字符集中的一一个字符，一个没分配的编码点是没有字符分配给它的编码点。在 AS/400 系统中，编码点是用 2 位的十六进制数字表示的。例如，用 EBCDIC 字符集，编码点 X ‘C1’ 分配给字符 A，X’51’ 是没分配的编码点。

CVTDTA 对 5224、5225 和 IPDS 打印机有效。对 IPDS 打印机，CVTDTA 允许规定数据流中的编码点。对这个打印机不要与 TRNSPY 和 DFNCAR 键字一起使用 CVTDTA。

如对没分配的编码点定义字符，按下面要求做：

- 规定 CVTDTA
- 处理程序中的十六进制数据

仅能对命名字段规定 CVTDTA，在常数字段中用户定义的字符，使用 DFT 和 DFNCHR 键字。

对 SCS 打印机，如果要指定 CVTDTA，就必须同时指定 TRNSPY。对用 DEVTYPE(\*IPDS) 和(\*AFPDS)生成的打印文件。如果指定了 CVTDTA，就不要用 TRNSPY，但在 DEVTYPE 没改成\*SCS 时会出现警告信息。

如对字段规定 CVTDTA，字段的长度必须是偶数。字段的打印长度是规定的长度除以 2。

如对字段规定了 CVTDTA，程序传送的字段中的字符数据必须仅是有效的十六进制字符(0-9, A-Z)，嵌入或结尾的空格都是非法的。如在程序运行时字段中有无效字符，OS/400 要往程序中送 CPF5234 逃逸信息。

可选指示器对此键字无效。

使用 CVTDTA 的例子，见 4.4.46 TRNSPY 键字。

对条形码命令使用 DDS CVTDTA，要遵守以下规则：

- 仅支持\*IPDS 打印机
- 允许下列命令：
  - WBCC (写条形码控制)
  - WBC (写条形码)
  - END
- 这三个命令必须在同一字段，在这个字段上也不允许其它命令。
- 字段长度必须确切

- 每个命令的长度必须确切
- 文件中的另外一个记录必须有 DDS BARCODE 键字这个记录不能用, 它告诉 OS/400 程序使用文件时要用条形码。
- IPDS 命令中不需要相关的 ID。
- 对用户条形码数据不做有效性检查, 无效数据引起打印机报告此命令无效。

在 4-19 给出命令的例子, 在例子中这些命令增加的总长度为 69 ( $45+17+1=69$ )。这个数要乘以 2 来表示包括在 CVTDTA 字段中的字符数, 即这个例子中的 CVTDTA 字段的长度为 138 ( $69 \times 2=138$ )。

详细信息, 请看打印机数据流手册。

PIC 25

PIC 26

图 4-19 使用 CVTDTA 的条形码命令格式

#### 4.4.11 DATE (日期)

用这个字段层键字显示当前作业日期或表示为 6 个字节长常数的系统日期, 可以规定这个字段的位置, 也可规定 CDEFNT、CHRSIZ、COLOR、ENTCDE、ENTWRD、FNTCHRSET、FONT、HIGHLIGHT、UNDERLINE 或 TEXT 键字。17-38 列必须是空格。

键字的格式为:

DATE (\*JOB | \*SYS)

\*JOB 规定打印当前作业日期。如果没指定参数, 就用 \*JOB。\*SYS 规定打印当前的系统日期。

如果对 DATE 字段规定 EDTCDE(Y), 则打印 8 位, 把日期从 mmddyy 格式改为 mm/dd/yy。

“/”表示运行时作业属性 DATSEP, m.d.y 的顺序由 DATFMT 确定。

如作业的 DATFMT 为 \*JUL, 不能使用 EDTWRD 键字。

可选指示器对此键字无效, 但它可以用在使用这个键字的字段中。

图 4-20 给出如何规定 DATE 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R REC01
A          1 56
A          DATE
A 21      2 56
A          DATE(*JOB)
A 22      2 56
A          DATE EDTCDE(Y)
```

A 23	2 56
A	DATE(*JOB) EDTCDE(Y)
A 24	2 56
A	DATE(*SYS)
A 25	2 56
A	DATE(*SYS) EDTCDE(Y)
A	

图 4-20 规定 DATE 键字

在每行 56 列上打印的作业日期没有编辑属性。

如果指示器 21 为 ON，打印无编辑的作业日期；如果指示器 22 或 23 为 ON，则打印编辑的作业日期；如果指示器 24 为 ON，则打印无编辑的系统日期；若指示器 25 为 ON，则打印编辑的系统日期。

#### 4.4.12 DFNCHR (定义字符)

这是文件层或记录层键字，可用来为 5224 和 5225 打印机定义你自己设计的字符，使用这个键字，可以在文件或记录层多次指定 DFNCHR，可以每次指定 DFNCHR 50 个字符。

键字的格式是：

```
DFNCHR(X'编码点 - 1'      X'点阵样式 - 1'  
[X'编码点 - 2'      X'点阵样式 - 2'.....  
[X'编码点 - 50'      X'点阵样式 - 50']]
```

注：不能在一个 DDS 语句中指定多于 5000 个字符。如果在一起指定的多个 DFNCHR 中，指定多于 5000 个字符，则可通过对另一个 DFNCHR 键字指定一个可选指示器来开始一个新的 DDS 语句，为避免指示器置为 on，须指定 N（例如 N50），这样对不活动程序，该键字的条件是 on。

用户定义的字符可以占用一个打印位置（图 4-21）或多个打印位置（图 4-23 和图 4-27）。对每个打印位置指定一个编码点和一个点阵样式。用 EBCDIC 字符集，十六进制 C1 是字符 A，十六进制是 51 是未分配的编码点（见 4.4.12.2 节的内容）。

可以在 DDS 中指定 9 个 2 位一对的十六进制数定义一个点阵样式。能指定的仅为字符 0–9 和 A–F（见 4.4.12.4 内容）。

当应用程序向记录格式发送一个输出操作而这个记录格式中 DFNCHR 定义的编码点不同于前一输出操作定义的编码点时，OS/400 装入新的定义，这样修改原定义的字符，这个处理会减慢打印速度。

然而，如果相同的 DFNCHR 键字对在一行中的两个输出操作都起作用，则 OS/400 不对第二次输出操作重装入编码点。

仅可在 SCS 打印机使用 DFNCHR。不能在 IPDS 打印机与 COLOR, LPI 和 BARCODE 键字一起对同一记录格式指定 DFNCHR。如果在文件中的任一格式有 SCS 和 IPDS 打印机键字的组合，则不能建立文件。

如果在一个用 DEVTYPE(\*IPDS)或 DEVTYPE(\*AFPDS)建立的文件中指定 DFNCHR，则在建立文件时出现一个警告信息。

不能在有 DRAWER 键字的同一记录格式中指定 DFNCHR，如果在文件中有记录层的 DFNCHR 和 ERAWER 键字，则不能建立文件。

可选指示器对这个键字有效。

#### 4.4.12.1 DFNCHR 键字的例子

图 4-21 和图 4-27, 图 4-32 给出怎样指定 DFNCHR。

例 1:

图 4-22 使用一个单一的点阵来显示如何在记录层指定 DFNCHR, 以便用十六进制的 7C 打印成 e 而不是@。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8	1            2
00010A              R RECORD	DFNCHR(X' 7C' X' 007E813CC324817E00')
00020A	58 4DFT(X' 7C') TRNSPY
00030A	+2DFT(' 1982')
A	

图 4-21 规定 DFNCHR 键字 (例 1)

这个例子重定义了码点十六进制 7C •1, 在 EBCDIC 字符集中正常的应该是@做版本标记, 而现被打印的版本标记如下 (在打印表格的 58 行上)。

1982

十六进制数字 2 定义下面的点阵模式。

PICTURE27

图 4-22 例 1 的点阵模式

详细信息参见 4.4.12.4 “指定要打印的点阵中的点”.

#### 4.4.12.2 选择要重定义的编码点

除十六进制 00 外, 可以定义任何一个编码点。关于通常使用的字符集列表, 请看在程序设计参考概要手册中的字符编码节。在定义编码点时, 可做下面事情中的一个:

- 重定义一个已存在的字符。如果在重定义之后试图打印这个字符, 则打印的是替换的字符。例 1 (图 4-21) 就是使用了一个这样的编码点。
- 对未分配的编码点定义一个字符系统 (字符集中不存在的字符)。例 2 (图 4-23) 就是这样的编码点。

如果重定义了一个已存在的字符, 则可用 DFNCHR 键字的可选指示器来选择要打印的字符 (已有的字符或重定义的字符)。如果选择了 DFNCHR, 则打印用户定义的字符。如果没选择 DFNCHR, 则打印已存在的字符。

注：定义一个未分配的编码点时，输出操作运行的比较快。未分配的编码可防止当程序选择不同的 DFNCHR 键字时重新装入编码点。

#### 4.4.12.3 点阵

对于 5224 和 5225 打印机，点阵是一个 8 行 9 列的矩阵，不考虑 CPI 或 LPI 的设置，所有的 8 行和 9 列都将被打印。

不考虑 LPI 的设置 (CRTPRTF 命令中的 LPI 参数)，两个点间的竖向距离总是 0.352mm (0.014 英寸)。LPI 的设置决定行间距，而不是字符高度。在一个 OLPI 的设置中(每行 2.81mm 或 0.111 英寸)，如果所有点的行都被使用，则没有竖向行间距。然而正常的字符集都不使用在矩阵中点的最后一行 (第 8 行)，这样即使在 9LPI 时，在行间也可有一些空隙。如果选择了，则可以用第 8 行来定义你自己的字符。

点间的横向距离依赖于在 CPI 上的设置，在 10CPI，每列的间距是 0.262mm(0.01 英寸)，每个字符为 2.54mm(0.1 英寸)。在 15CPI，每列的间距是 0.188mm(0.0074 英寸)，每个字符为 1.69mm(0.0667 英寸)，这些标准字符集不使用第 1 和第 9 列，(为了允许字符间的空隙)。

用户可使用 1 和 9 列来定义自己的字符，但有一个限制：5224 和 5225 打印机不能打印两个相邻的平行点。要打印两个相邻的平行点 (就象粗的下划线)，这一行必须打印两次。这可以在每次传送使用不同设置的编码点来作，一个定义为奇数点，而另一个定义为偶数点。在一个输出操作期间这两个传送都发生。如果程序试图打印两个相邻的平行点则不出现错误信息，但是基点的一个不打印。(一个字符的最后一个点的位置和它右侧字符的第一位被认为是相邻点)，对于相邻的竖向点没有这个限制。

在一任一输出操作上，每个编码点表示一个  $8 \times 9$  的点阵，通常每个都由不同的编码点来定义。也可能要求重复打印。

例如：要打印一个双倍宽度的字符，则需对字符的左半部分指定一个编码点，对字符的右半部分指定另一个编码点，双倍高度的字符要求字符上半部的编码点和一个下半部的编码点。在第一行上打印所有字符的上半部，在下一行上打印所有字符的下半部。为了避免在上半部和下半部之间出现间隙，必须在 CRTPRTF, CHGPRTF 或 OVRPRTF 命令上指定 LPI(9)。使用 DDS，可以在一个记录格式中定义两个字符，一个作为上半部，另一个作为下半部。例 2 (图 4-23) 显示了两倍宽和两倍高的一个字符。

#### 4.4.12.4 指定在点阵中打印的点

为用户定义的字符定义一个点阵样式时，指定 9 个 2 位一对的十六进制数字。每一对数对应在点阵中的一列，第一对数对应第一列，第二对数对应于第二列，以此类推。指定每对数的左侧字符来控制哪些点要打印在列的上半部，右侧字符控制下半部的字符。

使用在图 4-23 中的显示来为版本标记指定点阵。

PICTURE28

图 4-23 指定版本标记的点阵

使用在图 4-24 中的表格来设计你的点阵样式，并为 3 列宽乘 3 行高那样大的字符指定需要的十六进制数字。在这个格子样式中，为 9 个打印位置标记点阵样式（3 个横向和 3 个纵向的）。

PICTURE29

图 4-24 指定点阵字符的格式样版

使用图 4-25 决定哪些十六进制数在格子样式中指定每个半列。

图 4-25 位样式的十六进制数字			
位样式	十六进制数	位样式	十六进制数
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

图 4-25 位样式的 16 进制数字

每一打印位置，完成在图 4-26 中显示的格子样式的一行。对每个框应有一对十六进制数。

PICTURE30

4-26 完整的格子样式

例 2：

图 4-27 给出用于一个大的字符的点阵，这个例子显示了怎样在文件层为两列宽乘两行高的字符指定 DFNCHR 键字。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8 00010A 00020A 00030A 00040A 00050A 00060A 00070A 00080A 00090A        R RECORD1 00100A              3 00110A 00120A 00130A	<div style="display: inline-block; width: 100px; height: 100px; border: 1px solid black; position: relative;"> <span style="position: absolute; left: 50%; top: 50%; transform: translate(-50%, -50%);">1</span> <span style="position: absolute; left: 65%; top: 50%; transform: translate(-50%, -50%);">2</span> </div>	DFNCHR(X'51' X'000000FF00FF00E700' + X'52' X'E700E700E700E10000' + X'53' X'0000009C001F000700' + X'54' X'07000700FF00FC0000' + X'55' X'00000000FF00E700E7' + X'56' X'00E700E700E300000' + X'57' X'000000001E00070007' + X'58' X'0007000700FE000000')  58 4DFT(X'5152') TRNSPY 58 4DFT(X'5556') TRNSPY 59 4DFT(X'5354') TRNSPY 59 4DFT(X'5758') TRNSPY
--	---	---

A

图 4-27 指定 DFNCHR 键字（例 2）

图 4-27 重定义了 8 个编码点（十六进制 51 – 58）1 这  $2 \times 2$  字符的每个位置打印两次，这样才能打印相邻的水平点，十六进制编码2 定义点阵样式。

标记为3 的信息给出打印时大字符 5 的样子（使用 4 个打印位置；对一个打印机格，两个在 58 行上，另两个在 59 行上）；

5.

注：为避免在大字符中的水平间隙，文件应在 9LPI 上（在 CRTPRTF，CHGPRTF 或 OVERPRTF 命令上的 LPI 参数规定）。

在图 4-28 中显示的格子样式中，标记了多达 9 个打印列的点样式（横 3，竖 3）。

PICTURE31

图 4-28 规定例 2 的格子样式（51-54 列）

用图 4-25 来决定哪个 16 进制数字用来指定在格子样式中的每半列。

每个打印位置是完整的格子的一行（每个方框内一对十六进制数），对编码点 51-54，如图 4-29 所示。

PIC32

图 4-29 例 2 的完整编码点（编码点 51-54）

在这个格子样式中，标记点样式用 9 个打印列（3 个横向，3 个纵向），如图 4-30 所示。

PIC33

图 4-30 指定例 2 的格子样式（编码点 55-58）

用图 4-25 来确定用哪个 16 进制数做格式样式中每半列。

对每个打印位置是完整的格子样式的一行（每个方框一对十六进制数），就象在图 4-31 中的编码点 55-58。

PIC34

图 4-31 例 2 完整编码点 (55-58)

图 4-32 对一个大的图形使用点阵，表示如何在文件层为一个 3 列宽，两行高的大图形指定 DFNCHR。

00010A		1	2	
00020A	DFNCHR (X' B1' X' FF00E000D000C800C4' +			
00030A	X' B2' X' 00C200C100C100C200' +			
00040A	X' B3' X' C400C800D000E000FF' +			
00050A	X' B4' X' FF0007000800130023' +			
00060A	X' B5' X' 004300830083004300' +			
00070A	X' B6' X' 230013000B000700FF' +			
00080A	X' B7' X' 00FF00F000D800CC00' +			
00090A	X' B8' X' C600C100C100C100C6' +			
00100A	X' B9' X' 00CC00D800F000FF00' +			
00110A	X' BA' X' 00FF000F001B003300' +			
00120A	X' BB' X' 630083008300830063' +			
00130A	X' BC' X' 0033001B000F00FF00' )			
00140A	R RECORD1	CPI (15)		
00150A		58	4DFT (X' B1B2B3' )	TRNSPY
00160A		58	4DFT (X' B7B8B9' )	TRNSPY
00170A		59	4DFT (X' B4B5B6' )	TRNSPY
00180A		59	4DFT (X' BABBBC' )	TRNSPY
A				

图 4-32 规定 DFNCHR 键字 (例 3)

图 4-32 重定义了 12 个编码点 (十六进制 B1 – BC) 

1
---

，

2
---

 定义点阵样式。

图 4-32 的 DDS 例子在 3 个打印列宽中打印一个× (在打印格式的 58 和 59 行)。

注：为避免在大字符中的水平间隙，文件应用 9LPI (在 CRTPRTF, CHGPRTF 或

OVERPRTF 命令上的 LPI 参数)。

标记 9 个打印列 (3 个横向, 3 个纵向) 的点样式, 如图 4-33 所示。

PIC35

图 4-33 例 3 的格子样式 (B1-B6)

用图 4-25 确定要用哪个十六进制字符来规定格式中的每半列。对每个打印位置, 完整的一行格子样式如图 4-34 所示。(每个方框为一对十六进制数)。

PIC36

图 4-34 例 3 的完整编码点 (B1-B6)

如图 4-35 所示, 在格子中标记出 9 个打印列的点样式 (横 2, 竖 3)

PIC37

图 4-35 规定例 3 的格子样式 (B7-BC)

用图 4-25 决定要为这个格式指定的每半列的十六进制数。对于每个打印位置，完整的一行格子样式如图 4-36 所示（每个方框一对十六进制数）。

PIC38

图 4-36 例 3 的完整编码点 (B7-BC)

#### 4.4.13 DFT (缺省值)

使用 DFT 键字对常数（非命名）字段指定一个常数值。

键的格式是：

```
DFT ('值')
      '值'
DFT (X '十六进制值')
      X '十六进制值'
```

常数值可以是：

- 一个字符值。用指定的值打印，字符的数目等于打印字段的长度（在值中的双引号被打印成单引号），见图 4-37。
- 一个十六进制值。两个字符标识一个字符集中的编码点。这些字符打印在这个字段中（使用 DFNCHR 定义替换字符），其打印的长度是在单引号间指定的字符数的一

半，如果对 DFT 指定了十六进制值，则必须指定 TRNSPY 键字，见图 4-38。

可以不用 DFT 和括号来隐式地指定 DFT (对字符和十六进制值都可以这样做)，只要简单地在单引号间指定值。对于十六进制值，还必须在值之前用一个 X。

不能与 DFT 一起规定 EDTCDE 和 EDTWRD 键字。

可选指示器对这个键字无效。在字段位置的同一行上指定最后可选指示器可用做这个常数字段的条件限制。

图 4-37 给出如何用字符值指定 DFT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R SUPPLIES  
00020A      PENS      20    2 1  
00030A      INK       20    3 1  
00040A      PAPER     20    4 1  
00050A          7 9DFT('ON')  
00060A          8 9'ON'  
00070A  
00080A 01          12 1'Hotel name: ''Terrace Inn''  
00100A  
00110A 02          12 1'Hotel name: ''Riverview Inn''  
A
```

图 4-37 规定 DFT 键字 (字符值)

DFT('ON')和'ON'是等效的，上图也给出显式和隐式指定 DFT 间的区别。

如果指示器 01 为 ON，则打印为：

Hotel Name:'Terrace Inn'

如果指示器 02 为 ON，并且指示器 01 为 OFF，打印为

Hotel Name:'Riverview Inn'

图 4-38 给出怎样为包含一个替换字符的常数字段指定 DFT。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
1           2  
00010A      R RECORD          DFNCHR(X'7C' X'007E813CC324817E00')  
00020A          3 58 4DFT(X'7C')  TRNSPY  4  
00030A          +2DFT('1982')  
A
```

图 4-38 指定 DFT 键字 (十六进制值)

指定 DFT 的常数字段，显示在第 58 行第 4 列上，由十六进制 7C 定义的字符打印在这个字段中。这个例子中，在记录层指定的 DFNCHR 定义十六进 7C 作为一个版权标记。

下面是与这个例子中指定值等效的方法：

DFT(X'7C')

X'7C'

DFT(’表)

’表

当对 DFT 指定了十六进制时，TRNSPY 键字是必须的。

#### 4.4.14 DLTEDT (清除编辑)

使用这个字段层键字来指定 OS/400 要忽略被引用字段的任何编辑码或编码字。如果一个字段的描述是引用一个数据库文件，DLTEDT 可防止复制某些信息。

这个键字没有参数。

仅当在 29 列指定 R，并且还指定了 REF 或 REFFLD 键字时，这个键字有效。

如果需要替换编辑信息，则要为字段指定 EDTCDE 或 EDTWRD。新的键字复盖被引用字段的编辑属性。在这种情况下，不需要指定 DLTEDT。

可选指示器对这个键字无效。

图 4-39 显示了如何指定 DLTEDT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A          AMT      R      5  20DLTEDT  
A
```

图 4-39 指定 DLTEDT 键字

#### 4.4.15 DRAWER (纸盘)

使用这个记录层键来指定将从哪个纸盘取不连续的纸。

这个键字的格式是：

DRAWER (纸盘号)

纸盘号指定提供纸张或封皮的纸盘，有效值是 1-255 和 \*E1，如下所示：

1. 纸是第一个纸盘给进

2. 纸是第二个纸盘给进

n. 纸是第 n 个纸盘给进

\*E1 封皮是从封皮纸盘给进

如果没有指定 DRAWER 键字，在 CRTPRTF, CHGPRTF 或 OVRPRTF 命令的 DRAWER 参数的值决定纸张的来源的纸盘。

如果不是在一个页边界上指定 DRAWER 时，则在运行时忽略 DRAWER，在处理一页没有命名字段或常数字段时，打印机是在页边界上。一经处理了命名或常数字段，打印机就不再在页边界上。当处理 SKIP、SPACE 或 ENDPAGE 键字时，打印机重回到页边界上。

DRAWER, SKIP 和 SPACE 键字按如下顺序处理：

SKIPB SPACEB DRAWER SPACEA SKIPA

DRAWER 仅对定义了它的记录格式起作用，一旦处理这个记录格式的记录，对于下一个记录格式，如果没指定 DRAWER 键字，它的纸张来源纸盘是由文件层指定的(CRTPRTF, CHGPRTF 或 OVRPRTF 命令)。

对于用 DEVTYPE(\*SCS)建立的文件，如果 DRAWER 键字定义的记录格式要打印几页，则它只对指定它的页起作用。

不能在有 CPI 键字或记录层 DFNCHR 键字的同一记录格式上指定 DRAWER。如果在文件中的任一格式中有 DRAWER 和 CPI 或记录层 DFNCHR 键字，则不能建立文件。

可选指示器对这个键字有效。

图 4-40 显示了如何指定 DRAWER 键字。

00010A	R RECORD1	SKIPB(3)	
00020A	FIELD1	10	1SPACEA(1)
00030A	FIELD2	5	1SPACEA(1)
00040A			
00050A	R RECORD2	DRAWER(2)	
00060A	FIELD3	5	1
00070A	FIELD4	5	6SKIPA(1)
00080A			
00090A	R RECORD3	DRAWER(2)	
00100A	FIELD5	10	1SPACEA(1)
00110A	FIELD6	10	1SKIPA(1)
00120A	FIELD7	10	1SPACEA(1)
00130A	FIELD8	10	1SPACEA(1)
00140A			
00150A	R RECORD4		
00160A	FIELD9	10	1SKIPB(30)
00170A	FIELD10	10	21
00180A	R RECORD5	SKIPB(3)	
00190A	FIELD11	10	1SPACEA(1)
00200A	FIELD12	10	1SPACEA(1)
A			

图 4-40 指定 DRAWER 键字

在处理记录格式 RECORD1 之后，打印头不在页边界上。当处理记录格式 RECORD2 时，忽略 DRAWER，并且纸张继续来自预先指定的（文件层）源纸盘。由于对 RECORD2 的 FIELD4 指定了 SPACEA(1)，所以在处理了 RECORD2 之后，打印是在一个页边界上的。作为 RECORD3 的两页纸是来自纸盘 2 的。记录格式 RECORD4 和 RECORD5 纸张来源是在文件层指定的纸盘，这个例子中为纸盘 1，但是由于 RECORD4 是在一页的中间开始的，所以它与 RECORD3 打印在同一页上（纸盘 2）。记录格式 RECORD5 打印在不同页上（SKIPB(3)），并且纸张来自纸盘 1。

#### 4.4.16 DTASTMCMD (数据流命令)

用这个记录层或字段层键字来保存数据流命令或假脱机文件的某些信息。这个命令可以用来决定如何处理假脱机文件中某页上的字段或记录。这个键字只能用在设备类型为 \*AFPDS 的打印文件中。

键字格式为：

DTASTMCMD ('说明' | &说明字段)

说明必须放在引号内，如果说明长度多于 255 个字符。在编译时有错误信息给出，规定的程序到系统字段必须在与 DTASTMCMD 键字同一记录格式中。如果字段多于 255 个字符，在编译时有错误信息给出。

用户应用程序应能知道如何处理假脱机文件的某些页来检索数据流、抽取数据流命令。这些内容应包括在 AFPDS(MODCA)NOP 命令中。NOP 命令能在包括这个关键字的记录或字段的可打印数据之前，在数据流中建立。既然这些信息是刚刚保存，这个关键字对文件就没有直接影响。有关信息请参考 MD:DCA 手册。

可选指示器对此关键字有效。

注：每个字段或每个记录只能规定一次这个关键字

图 4-41 给出如何规定 DTASTMCMD 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECORD1           DTASTMCMD(' TEXT(Record 1)')
A          FIELD1    10A      5  5
A  01          DTASTMCMD(' TEXT(Field 1)')
A          FIELD2    10A      10  5DTASTMCMD(&DATA)
A          DATA      10A  P
A
```

图 4-41 规定 DTASTMCMD 键字

记录 RECORD1 的数据流做 DTASTMCMD 键字的说明。如果指示器 01 为 ON，字段 FIELD1 有 DTASTMCMD 的说明 TEXT('Field1')。如果指示器 01 为 OFF，对 FIELD1 的 DTASTMCMD 无数据流说明生成。FIELD2 要用 DATA 的内容做 DTASTMCMD 的说明。

#### 4.4.17 EDTCDE (编辑码)

用这个关键字编辑可输出的数字字段。

关键字的格式是：

EDTCDE (编辑码[\* | 浮动货币符号])

根据指定的编辑码，可以改变打印字段的样子如下：

- (1)压缩前置零。
- (2)用逗号或句号（圆点）做小数点和千分号。
- (3)在字段右侧用减号或 CR 打印负值。
- (4)用零或空白打印零值。
- (5)在有效数字的左侧打印星号来提供星号保护。
- (6)紧接在有效数字的左侧可以立即打印一个货币符号（与系统值 QCURSYM 一致，叫做浮动货币符号），对于固定货币符号使用 EDTWRD 键字。
- (7)使用用户定义的编辑码可以进一步编辑字段。详细信息见 4.4.17.2 的“用户定义编辑码”。

EDTCDE 能满足大多数编辑要求，当 EDTCDE 不能满足要求时使用 EDTWRD 键字。

EDTCDE 仅对在 35 列（数据类型）为 S 或空白的字段有效。

对同一字段不能同时指定 EDTCDE 和 EDTWRD。如果对一个已在数据库文件中定义的字段指定了 EDTCDE，则不需对再定义 EDTCDE。可在 29 列指定 R 来引用预先定义的字段。这样，对那个字段定义的编辑被包括在打印机文件之内。如果对打印文件的字段指定长度，数据类型或小数位，则对被引用字段指定的编辑不包括在打印文件中，这时必须在打印文件中重指定编辑。

可选指示器对这个关键字无效。

DFT 键字不能与 EDTCDE 键字一起指定。

EDTCDE 键字详细的信息，见 3.3.44 “EDTCDE（编辑码）”。

可以指定两种编辑码：OS/400 编辑码和用户定义编辑码。

#### 4.4.17.1 OS/400 编辑码

OS/400 编辑码是：

1~4

A~D

J~Q

X~Z

注意：AS/400 硬件操作使用推荐符号 F，与使用编辑 X 是等价的。如果 DATE 或 TIME 键字用编辑码 X 指定，则不显示分隔符。

星号填位或浮动货币符号：用编辑码 1~4, A~D 和 J~Q 可以有选择地指定星号填位或浮动货币符号。

当指定星号填位时，对每个被抑制的 0 写一个星号。对零平衡的字段打印为全部星号。

当指定一个浮动货币符号时，这个符号出现在第一个有效数字的左边。在编辑码用于抑制零字段时，不打印货币符号。你指定的符号必须与系统的货币符号值(QCURSYM)相匹配。  
(在建立文件时，这个符号必须匹配，在使用文件时，则不必匹配)。

注：如果在生成文件之后修改了编辑码，并不能用新的编辑码，除非重新生成文件，仍是继续使用文件建立时使用的编辑码。

下面的表格总结了 OS/400 编辑码提供的功能。

图 4-42 编辑码

图 4-42 OS/400 编辑码

注：1.QDECFCMT 系统值确定小数点字符（在美国使用句号），千分号字符（美国使用逗号）和抑制 0 的打印（根据逗号和句点位置）。详细内容请见数据管理指南。

2.Y 编辑码抑制 3~6 位长数据字段的最左位 0，并且抑制 7 位长字段的最左侧的两位 0。根据下面的样子 Y 编辑码在月，日，年间插入斜线。

nn/n  
nn/nn  
nn/nn/n  
nn/nn/nn  
nnn/nn/nn

如果 DATE 键字与 EDTCDE(Y)一起指定，使用的分隔符是运行时作业属性的 DATSEP，如果在作业属性上 DATSEP 没指定分隔符，则使用系统值 QDATSEP（这里，斜线 (/) 是缺省值），如果在文件建立时 DATFMT 是 JUL，日期是 nnnnn 的格式。如果指定了 EDTCDE(Y)，日期是 nn/nnn 的格式，斜线做作业日期的分隔符。

3.Z 编辑码从一个数字字段去掉符号。在写字段之前单个的列符号变成为十六进制 F。

#### 4.4.17.2 用户定义的编辑码

编辑码 5 – 9 是用户定义的编辑码，用户定义的编辑码可以比 OS/400 编辑码作更多的编辑，例如，用户可能需要包括连字号（如电话号码）或多个小数点的数字，就可以使用用户定义的编辑码来完成这些功能。这些编辑码名为 QEDIT5、QEDIT6、QEDIT7、QEDIT8 和 QEDIT9。在 DDS 或高级语言程序设计引用时用数字 5、6、7、8、9。

用户定义编辑码是 OS/400 的目标，并且在打印文件生成之前必须存在，它用建立编辑码命令 (CRTEDTD) 来生成，当生成使用用户定义的编辑码的打印文件时，编辑信息是从预先建立的编辑描述中来提取的，除非重建打印文件，在打印文件建立之后改变用户定义的编辑码不影响打印文件。

下面的表显示了未编辑的源数据和已编辑输出的例子。0 抑制和小数点字符由系统值 QDECFCMT 决定。日期分隔符号由作业属性 DATSEP 决定。在这个图中，假定 QDECFCMT 为 X (空格)，DATSEP 为 / (斜线)。

图 4-43 编辑码例子



图 4-43 有效的编辑码、源数据和编辑输出

注：

- 1.X 代表一个空格。
  - 2.Y 编辑码抑制 3 – 6 位长数据字段的最左侧一个零。并且抑制 7 位长数据字段的最左侧的两个零。详细信息见 3.3.44.1 的图 3-80 的注释 2。
  - 3.Z 编辑码去掉符号（加号或减号）并且抑制前置零。
- 图 4-44 给出了如何规定 EDTCDE 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....  
00050A           PRICE       5   2   5  2EDTCDE(J *)  
A
```

图 4-44 规定 EDTCDE 键字

#### 4.4.18 EDTWRD (编辑字)

如果使用 EDTCDE 键字不能完成期望的编辑，则指定编辑字来代替它。编辑字指定要打印字段值的格式，说明要插入字符的数据，象小数点、逗号、浮动的和固定的货币符号以及信贷平衡指示符，还用它来抑制前置 0 和提供星号填充保护。

键字的格式是：

EDTWRD ('编辑字')

如果在已预先定义的数据库文件字段中指定了 EDTWRD，则对定义的字段不需指定 EDTWRD，只要在 29 列指定 R 来引用预先定义的字段。这样被引用字段的编辑属性被调入到打印机文件中。但是，如果对打印机文件的字段指定了长工、数据类型或小数位，则对被引用字段指定的编辑不调入打印机文件中，则必须在打印机文件中指定编辑。

##### 4.4.18.1 编辑字的组成部分

编码字包括：主体、状态和扩展。下图给出这三部分。

主体包括要从数据字段传送到输出记录的数字。编辑字从它的最左列开始，并且在最右列的可用数字替换的字符处结束。它含有的空白个数（加上一个零或星号）要大于或等于被编辑数字字段的数字个数。

状态部分显示数据字段的符号（+或-），它放在主体的右侧用一个 CF（信贷）或减号，并仅当字段是负值时才打印，没有 CR 或减号的编辑字没有状态列。

扩展部分不由编辑操作改变，扩展从状态（如果没有状态，从主体）右侧的第一位开始，在编辑字最右侧的字符结束。

#### 4.4.18.2 编辑字的主体格式

在编辑字的主体有下面的字符时，它有特殊意义：

空格：空格由来自数据字段的相应位置的字符替换，空白位称为数位。

&：一个&号在编辑字段产生一个空白。&号不打印。

0：为了停止零抑制，在它要停止的最右位放一个零。这个零会被数据字段相应位置的字符代替。除非字符是零数据中停止零抑制字符的右边的零都要打印，停止零抑制字符认为是一个数位，至少极一个前置零被抑制。每个被抑制的零由空白替代。

星号：在要停止零抑制的最右位放置一个星号，停止零抑制并且用星号来代替零（星号保护）。零前面的星号被解释为星号保护，在这种情况下，零作为常数打印，停止零抑制字符右侧的零或星号是常数。

货币符号：紧挨在零抑制编码左侧的货币符号导致在每位有效数字左侧插入货币符号。以这种方式使用时称为浮动货币符号。编辑字最左列的货币符号，每次都在相同的位置固定打印，当以这种方式使用时称为固定货币符号。货币符号不作为数字替换位，这个符号必须与系统值 QCURSYM 一致。

小数点和逗号：除了小数点和逗号在第一位有效数字左侧外，将在与编辑字中编码的相同相对位置上打印小数点和逗号。否则，它们将输出空白或用星号代替。

数据字段中有效数字右边的所有字符都被打印，如果它们在数据中高位有效数字的左边，在使用星号保护时它们输出空白或用星号代替。

#### 4.4.18.3 编辑字的状态格式

当在编辑字的状态中使用下面的字符时，它们有特殊意义：

&号：在编辑输出字段产生一个空格，&不能放在编辑输出字段中。

CR 或减号符号：如果编辑输出字段的符号是（+）号，这些位置空白输出。如果编辑输出字段中的符号是减号（-）这些位置保持原样。

#### 4.4.18.4 编辑字的扩展格式

在一个编辑字的扩展部分中，总要写字符，扩展部分不允许有空白。如果编辑输出字段需要空格，则在编辑字的主体中指定一个&。

在指定有效编辑字时，遵循下面这些原则：

1.不能对同一字段同时指定 EDTWRD 和 EDTCDE。

2.编辑字必须放在单引号中。

3.EDTWRD 键字仅对数字字段（35 列指定 S）有效。

4.编辑字中空格和停止零抑制字符（数位）的总和必须等于字段的长度。

5.如果停止零抑制字符是编辑字的第一个字符，则空格的总数等于字段的长度或字段长度减 1。

6.如果使用了浮动货币符号，它不作为数字统计数。例如，为长度是 7 小数位是 2 的字

段指定浮动货币符号编辑字是：

EDTWRD(' \_ \_ \_ \$0 . \_ \_ ')

这里\_代表空格。

7.如果要显示一个负数的负号，则在编辑字包括一个符号。在最后数字替换字符的右侧使用减号（-）或字母 CR，仅当数字是负数时才打印它们。

可选指示器对这个键字无效。

DFT 键字不能与 EDTWRD 键字同时指定。

有关 EDTWRD 键字的详细信息，见 3.3.46 的“EDTCDE（编辑字）”。

图 4-45 显示了如何规定 EDTCDE 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A          CRYCST      7 2 5 2EDTWRD(' $0. ')  
A
```

图 4-45 规定 EDTCDE 键字

图 4-46 显示了带有字段程序值和字段打印值的编辑字。

PIC40

图 4-46 编辑字实例

#### 4.4.19 ENDPAGE (结束页)

使用这个记录层关键字来排除记录打印之后的当前页。

这个关键字没有参数。

在文件中指定 ENDPAGE 时，在 CRTPRTF 命令中指定 DEVTYPE(\*AFPDS)。如果 DEVTYPE 被改为其他值而不是\*AFPDS，则忽略关键字并且在打印时发出警告信息。

如果在指定了常数字段的记录格式中还指定了 ENDPAGE 关键字则将发出错误信息。

不能与下面的关键字一起指定 ENDPAGE：

SPACEA

SPACEB

SKIPIA

SKIPB

注：使用这个关键字时 PSF/400 是必需的，如果没安装 PSF/400，就不能使用这个关键字和指定 DEVTYPE(\*AFPDS) 来打印文件。

可选指示器对这个关键字有效。

图 4-47 显示如何规定 ENDPAGE 关键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A*  
A      R REC1           ENDPAGE  
A      FLD1      5A    66 10  
A*  
A      R REC2  
A 01           ENDPAGE  
A      FLD1      5A    POSITION(8.5 10.2)  
A
```

图 4-47 规定 ENDPAGE 关键字

在 REC1 打印完之后发生页的排除。

如果应用程序写 REC2 时指示器 01 为 ON，页排除发生在打印 REC2 之后。如果应用程序写 REC2 时，指示器 01 为 OFF，则不发生页排除。

#### 4.4.20 FLTFIXDEC (浮点到定点十进制)

使用这个字段层关键字来以定点十进制记数法打印浮点字段中的数。

这个关键字没有参数。

当使用 FLTFIXDEC 时，浮点数首先转换成一个指数为 0 的等价数字。如果结果数（数字或指数）能适合长度和小数值定义，则这个数用抑制指数并且按小数点对齐来打印。如果这个数不适合这个字段，则数字按标准浮点格式 n.nnnnnnE+nnn 来打印，当指定 FLTFIXDEC 关键字时，字段长度是 DDS 长度加 2（符号和小数点），字段的最小长度是 6。

当数字对于定点格式来说太大或太小时，用有总位数和小数位数的 FLTFIXDEC 关键字指定这个字段。这时，浮点格式如下那样打印有效数（有效数是一个在指数符号 E 左侧有小数点的数字串）。

- 总有效十进制数字：DDS 总位数减 5

- 小数有效数字： DDS 总位数减 6  
可选指示器对这个键字无效。
- 图 4-48 显示了怎样规定 FLTFIXDEC 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RECFMT1
A          FIELD1      10F 3   1  2FLTFIXDEC
A
A          FLTPCN(*DOUBLE)
A
```

图 4-48 规定 FLTFIXDEC 键字

这些输出数据将转换成如下的样子。

输出数字	被打印为
-4.99994321000000E-004	' -4.0000E-004'
-5.00010000000000E-004	' -0.001'
-2.69123400000000E-002	' -0.027'
-0.00000000000000E+000	' 0.000'
0.00000000000000E+000	' 0.000'
2.71828182845900E+003	' 2718.282'
3.14159000000000E-052	' 3.14163-052'
9.87654321012345E+006	' 9876543.210'
9.9999999960000E+006	' 1.0000E+007'

#### 4.4.21 FLTPCN (浮点精度)

使用这个键字来指定浮点字段的精度。

键字的格式是：

FLTPCN(\*SINGLE | \*DOUBLE)

\*SINGLE 指定单精度。\*DOUBLE 指定双精度。

这个键字只对浮点字段有效。

单精度字段可以是 1~9 位；双精度字段可以是 1~17 位，如果指定一个长度大于 9（单精度）或是 17（双精度）的字段，将出现错误信息，文件不能生成。

可选指示器对这个键字无效。

图 4-49 显示了怎样规定 FLTPCN 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00090A      FILEDA      17F 4   2  3FLTPCN(*DOUBLE)
A
```

图 4-49 规定 FLTPCN 键字

FILEDA 是一个双精度浮点字段。

#### 4.4.22 FNTCHRSET (字型字符集)

使用这个字段层或记录层关键字指定打印记录中命名的或常数字段的字形。

关键字的格式是：

FNTCHRSET([库名]/[字型字符集 [库名/]码页])

字形字符集和码页参数是必须的，它们都可以是 1~8 个字符长。

使用可选的库名参数进一步限定字形字符集或码页。如果没指定库名，用\*LIBL 查找字形字符集或码页。如果使用\*LIBL，当查找需要的字形时，系统支持的字形库加到库列表中。

注：如果一个应用程序使用了私有资源（例如：字形、页段，覆盖或不受系统限制的 GDF 文件），当引用这些资源时，如果指定了\*LIBL 或没有指定库名，对于建立假脱机文件的应用程序使用的库列表必须是可用的。

字形字符集和码页值在打印时应是合法的。如果它们不合法将发出错误信息。

注：当生成打印文件且在 FNTCHRSET 中规定码页时，用打印文件层的参数来确定列间隔，忽略在 FNTCHRSET 规定的字形或码页，而用打印文件参数规定的字形和码页。

当在文件中指定 FNTCHRSET 时应在 CRTPRTF 命令上指定 DEVTYPE(\*AFPDS)。如果 DEVTYPE 变为其它值而不是\*AFPDS，忽略关键字并在打印时发出警告信息。

不能在与 FONT 和 CDEFNT 键字相同的层上指定 FNTCHRSET。

注：使用这个关键字需要 PSF/400，如果没安装 PSF/400，就不能用这个关键字和 DEVTYPE(\*AFPDS) 来打印文件。

可选指示器对这个关键字有效。

图 4-50 显示了如何规定 FNTCHRSET 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A*  
A      R REC1  
A      FLD1A      14A      3  8FNTCHRSET(COSOCE12 T1LOPCHN)  
A*  
A
```

图 4-50 规定 FNTCHRSET 键字

FLD1A 指定字形字符集 COSOCE12 和码页 T1LOPCHN，\*LIBL 用于查找字形字符集和码页。

#### 4.4.23 FONT (字形)

使用这个记录层或字段层关键字来指定将要打印的命名或常数字段或记录中字段的字形标识。

关键字的格式是：

FONT (字形标识 [点尺寸])

字形标识是必须的参数，且是关键字的第一个参数，指定一个数字字形标识或图形字形名或\*VECTOR。

用可选的点尺寸参数来进一步定义一个数字字形。点尺寸用格式 (\*POINTSIZEx 值) 的表达式指定。这个参数的有效值是 0.1 – 999.9。

如果在带有图形字形名或\*VECTOR 的 FONT 键字上指定点尺寸参数，那在生成文件时将出现警告信息。在这种情况下，点尺寸参数被忽略。

如果没指定这个键字，则字形标识和点的尺寸是由 CRTPRTF, CHGPRTF 或 OVRPRTF 命令的 FONT 参数设置。如果在记录层指定了这个键字，则记录格式中除了那些在字段层指定 FONT 键字的字段外，所有字段使用同一个字形标识和点尺寸。

可以指定图形字形（字母数字字符）或硬件字形（数字字形标识）。对于图形字形，使用对 AS/400 系统、GDDM、PGR 和 BGU 有效的图形符号集（GSS），仅支持向量符号（在这里每个字符是用一组直线或曲线建立的）；支持由 AS/400 系统和 GDDM 提供的大多数向量符号集。不支持映象符号，在检索图形字符集时，用\*LIBL 作限定的库名。

图形字形名可由 1 – 10 个字母数字字符构成。

硬件字形可以由 1 – 10 位数字组成，并且必须是一个已经注册的字形号，关于完整的数字字形列表，请看 CL 参考手册中 CRTPRTF 命令的 FONT 参数。

可以在 FONT 键字上指定\*VECTOR 来使用 4234 IPDS 打印机上向量字形的优点，向量字形打印扩展字符比用 PRTQLTY(\*DRAFT) 键字打印它们要快。用 CHRSIZ 键字指定扩展字符。

注：当和 CHRSIZ 键字一起指定 FONT(\*VECTOR) 时，4234 打印机使用缺省的码页。

向量字形仅对下面的字符有效：

A~Z

0~9

特殊字符 (+、\$、\*、-、/、% 和空格)

如果要打印的数据包括除这些字符以外的其他字符，在打印机上所有的字符用缺省的字形打印。

FONT(\*VECTOR) 对无扩展字符没有作用，如果在一个记录层或字段层上指定 FONT(\*VECTOR) 使用 CHRSIZ (11) 或没有使用 CHRSIZ 键字，会出现警告信息。

注：如果在 4224 或 3812 打印机上使用 FONT(\*VECTOR)，打印机使用缺省字形和码页。

字形名或字形号和点尺寸值在文件生成时并不检查。如果指定的字形标识和点尺寸无效，则在记录打印时出现诊断信息并且不使用此键字。

当在字段层指定 FONT 时，不检查重叠字段。

当 CRTPRTF, CHGPRTH 或 OVRPRTF 命令中使用图形字形时，字形标识有一个与它相关的隐含页码，要得到期望的编码，必须使用适当的字形标识，不使用在 CHRID 参数指定的编码页。

如果用 CHRID 键字指定 OCR - A 和 OCR - B 字形，字形分别地需要码页 892 和 893。

如果在生成文件时，指定 FONT DDS 键字为 DEVTYPE(\*IPDS) 和 FONT(\*DEVD)，将发出警告信息。对 SCS 打印文件，当打印记录或字段时忽略 FONT 键字。对 IPDS 打印机，FONT 键字可在记录层或字段层改变。

当在不支持注册字形的 IPDS AFP(\*YES) 的打印机上打印一个使用 FONT 键字的文件时，将产生字形代换。

对使用字形支持的设备不论用 CRTPRTF, CHGPRTF 还是 OVRPRTF 命令指定 FONT(\*CPI) 时，主系统根据当前打印文件的 CPI 的行距挑选字形。

FONT(图形字型名) 和 CHRID 不能用于同一字段。如果出现下面的情况则忽略 CHRID：

- 在同一字段指定 FONT(图形字型名) 和 CHRID
- 在记录层指定了 FONT(图形字型名) 并且记录中的字段指定了 CHRID 而不是数字 FONT

不能在 CDEFNT 和 FNTCHRSET 键字的同一层指定 FONT。

对每个记录只能指定一次 FONT 键字且每个字段也能指定一次。

这个键字对数据类型 A, S 和 F 有效。

可选指示器对这个键字有效。

图 4-51 显示了如何规定 FONT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD1  
00020A 02 03          FONT(222)  
00030A      FLD1      6A    16 01  
00040A 01          FONT(ADMMVSS)  
00050A      FLD2      6S    20 01  
00060A      R RECORD2  
00070A 04          FONT(16951 (*POINTSIZ 10))  
00080A      FLD3      6A    16 01  
00090A 05          FONT(16951 (*POINTSIZ 12))  
00100A      FLD4      6S    20 01FONT(4919)  
A
```

图 4-51 规定 FONT 键字

如果指示器 01 为 ON, FLD1 使用多国向量字符集 (FONT(ADMMVSS)), 如指示器 01 为 OFF, 并且指示器 02 和 03 为 ON 时, FLD1 使用 Gothic 15 (FONT(222)), 否则使用 CRTPRTF 命令指定的字形。

如果指示器 02 和 03 是 ON, FLD2 使用 Gothic 15, 否则, 使用 CRTPRTF 命令指定的字形。

如果指示器 05 为 ON, FLD3 使用点尺寸为 12 (FONT(16951 (\*POINTSIZ 12))) 的字形, 如果指示器 05 为 OFF 而且指示器 04 为 ON, FLD3 使用点尺寸为 10 的字形。否则, 使用 CRTPRTF 命令指定的字形。

FLD4 使用(FONT(4919))。

#### 4.4.24 GDF (图形数据文件)

使用这个记录层键字来打印一个图形数据文件。

键字的格式是:

GDF(库名 | &库名字段  
图形文件 | &图形文件字段  
图形成员 | &图形成员字段  
列下拉 | &列下拉字段  
列跨越 | &列跨越字段  
图形深度 | &图形深度字段  
图形宽度 | &图形宽度字段  
图形旋转 | &图形旋转字段

图形文件和图形成员参数指出要打印的图形, 它们都是必须的参数。

使用可选的库名参数来进一步限定图形数据文件和成员。如果没有指定库名参数, 在打印时, 用\*LUBL 来检索图形数据文件。

可以指定所有这些参数为一个常数、程序到系统字段或者二者的组合。象下面显示的那样:

[库名/]图形文件 图形成员……  
[库名/]&字段 1 图形成员……  
[&字段 2]图形文件 &字段 3……  
[&字段 4]&字段 5 &字段 6……

当指定库名、图形文件或图形成员为程序到系统字段时，它必须存在于与 GDF 键字同一记录格式中。必须定义成长度为 10，数据类型为 A (字符)，用法为 P (程序到系统) 的字段。

当指定列下拉、列跨越、图形深度或图形宽度参数为程序到系统字段时，字段必须定义成长度为 5, 3 个小数位，数据类型 S，和用法为 P 的字段。当指定图形旋转参数为程序到系统字段时，字段必须被定义成 3 位长，小数位为 0 的字段。

列下拉参数是必须的，它定义图形相对于边界的列起始点。边界是由 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的。参数的有效值是 0-57.790cm, (0-22.750in)。

列跨越参数是必须的，它定义图形相对于边界的行起始点，边界是由 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的。P 参数有效值是 0 – 57.79cm(0 – 22.750in)。

图形深度参数是必需的，定义图形的深度，图形被绘制在适合于图形深度指定的范围内，有效值是 0.001-57.790cm (0.001-22.750in)。

图形宽度参数是必需的，定义图形的宽度，图形被绘制在适合于图形宽度参数指定的范围内，有效值是 0.001 – 57.790cm(0.001 – 22.750in)。

注：CRTPRTF 命令的 UOM 参数决定列下拉、列跨越、图形深度和图形宽度参数值的计量单位。如果一个参数指定的值超出了有效范围，当生成假脱机文件时，有标志给出。

图形旋转参数是必须的，用来定义图形的方向与页上正文的关系，有效值是 0、90、180、270。

如果图形在页上没定位，打印时将发出错误信息。

注：图形数据文件必须符合 IBM 图形目标目录结构 (GOCA) DR2 的子集，版本 0 (DR/2V0)。GOCA DR/2V0 的详细信息，参见 GOCA 参考手册。

在文件中规定 GDF 时要在 CRTPRTF 命令中指定 DEVTYPE(\*AFPDS)。如果 DEVTYPE 改为其它值而不是\*AFPDS，则忽略键字并在打印时发出警告信息。

当在记录式中指定 GDF 时，该记录格式内的所有字段都必须有 POSITION 键字定位，详细信息见 4.4.36 “POSITION (位置)”。

如果在指定了 GDF 键字的记录格式中指定常数字段，将产生错误信息。

在一个记录中可以多次指定这个键字。

不能和下面的键字一起指定 GDF：

SPACEA  
SPACEB  
SKIPA  
SKIPB

注：使用这个键字时，PSF/400 是必须的，如果没有安装 PSF/400，不能使用这个键字和指定 DEVTYPE(\*AFPDS) 来打印文件。

可选指示器对这个键字有效。

图 4-52 给出如何指定 GDF 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A*
A      R REC1          GDF(GRAPHLIB/GFILE MYGRAPH 1.557 +
A                      2.831 7.0 4.5 90)
A*
A      R REC2          GDF(&GLIB/&GFILE &GRAF &POSD +
A                      &POSA &GDEP &GWID &GROT)
A      GLIB            10A P
A      GFILE            10A P
A      GRAF             10A P
A      POSD             5S 3P
A      POSA             5S 3P
A      GDEP             5S 3P
A      GWID             5S 3P
A      GROT             3S 0P
A*
A      R REC3          GDF(GFILE MYGRAF 2.0 7.0 4.5 11.25 +
A                      180)
A*
A      R REC4          GDF(YOURFILE THATGRAF 2.5 7.3 3.0 +
A                      5.25 0)
A*

```

图 4-52 指定 GDF 键字

注：CRTPRTF 命令的 UOM 参数决定参数值的计量单位。

REC1 从库 GRAPHLIB 的文件 GFILE 中打印成员 MYGRAPH。图形从 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界纵向打印 1.557 个单位，横向打印 2.831 个单位。图形 7.0 个单位深，4.5 单位宽，旋转 90 度。

REC2 允许应用程序通过设置字段 GLIB、GFILE 和 GRAF 来分别地指定库、文件和图形名。图形从 CRTPRFF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界纵向打印 1.3 个单位，横向打印 5.1 个单位。图形 5.25 个单位深，6.75 单位宽，旋转 180 度。

REC3 打印两个图形。MYGRAF 打印 CRTPRTF 命令的 FRONTMGN 参数或 BACKMGN 参数指定的边界纵向打印 2.0 个单位，横向打印 7.0 个单位。图形 4.5 个单位深，11.25 个单位宽，旋转是 180 度。YOURGRAF 从 CRTPRTF 指定的边界纵向打印 0.1 个单位，横向打印 0.5 个单位。图形 3.67 个单位深。6.5 个单位宽，旋转 90 度，两个图形都用\*LIBL 和文件 GFILE 定位。

在指示器 01 为 on 时，REC4 打印 THATGRAF。

#### 4.4.25 HIGHLIGHT (高亮度)

使用这个记录层或字段层键字来指明一个字段要以醒目的字母打印。

这个键字没有参数。

这个键字对 IPDS 和 SCS 打印机都有效。

对于用 DEVTYPE(\*AFPDS)生成的文件，这个键字只适用于注册字形标识。如果 HIGHLIGHT 和编码字形或字符集和编码页一起使用，将发出信息。

如果在记录层指定 HIGHLIGHT，则键字适用于这个记录中的所有字段。这样，如果记录层和字段层的 HIGHLIGHT 键都被选择并且有一个指示器的条件满足，则使用这个 HIGHLIGHT 键字。

由于打印期间使用 FONT，HIGHLIGHT 键字可能不适用。如果指定不支持高亮度字形的数字字形或指定图形字形，则不要使用 HIGHLIGHT 键字。

HIGHLIGHT 键字对命名字段和常数字段都有效。

这个键字对数据类型为 A、S 和 F 有效。对每个记录和每个字段仅可指定一次 HIGHLIGHT。

可选指示器对这个键字有效。

图 4-53 显示了如何指定 HIGHLIGHT 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RECORD1  
A 01           HIGHLIGHT  
A           4 01' HIGHLIGHT IF 01'  
A           FLD1      3A   11 01TEXT(' HIGHLIGHT IF 02N90 +  
A           OR 01')  
A 02N90       HIGHLIGHT  
A
```

图 4-53 指定 HIGHLIGHT 键字

#### 4.4.26 INDARA (指示器区域)

使用这个文件层键字从缓冲区（也称为记录区）移出可选指示器并把它们放在另外一个 99 个字节指示器区域。

这个键字没有参数。

如果指定 INDARA 键字，一些高级语言要求在程序中指定使用的分离的指示器区域。请看相应的高级语言手册。

如果已在文件上指定了 INDARA 键字，可以在 DDS 中增加、修改、删除可选指示器及重新生成文件，而不必重新生成高级语言程序。可以这样做是因为字段在缓冲区中的位置没有变，并且级别检验数据没有变。如果程序要使用新的指示器，则需修改和重新生成程序。

可选指示器对这个键字无效。

图 4-54 显示了如何指定 INDARA 键字

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               INDARA
00020A       R RCD
00030A   41                         SPACEB(1)
00040A       ACTNBR      10          2
                                A

```

图 4-54 指定 INDARA 键字

如果指定了 INDARA 键字，可选指示器 41 从记录格式 RCD 的缓冲区移出，放在另外的指示器区中。只有命名字段 ACTNBR 仍留在 RCD 的缓冲区内。

#### 4.4.27 INDTXT (指示器正文)

使用这个文件层、记录层或字段层键字来联接正文和指定的指示器。可以为每个指示器指定一次 INDTXT。

键字的格式是：

INDTXT (指示器 ‘正文’)

如果指定 INDTXT 键字，‘正文’是一个必需的参数，指示器使用的正文必须是一个字符常数而且必须括在单引号中。如果正文的长度多于 50 个字符，高级语言编译器只使用前 50 个字符。

可选指示器对这个键字无效。

注：这个规范本身不会引起指定的指示器出现在输出记录区，这个规范仅提供与这个指示器相关的正文。如果在别处没有指定这个指示器，则正文丢失且不做检查，一旦给指示器一个正文，（通过这个键字或响应指示器正文）不再分配其它的正文。

图 4-55 显示了如何指定 INDTXT 键字

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               INDTXT(02 'Alternate month')
00020A       R MASTER
00030A       MTH           2 10
00040A   02       ALTMTH        2 10
00050A
                                A

```

图 4-55 指定 INDTXT 键字

INDTXT 键字说明了可选指示器 02 的用途。在高级语言的编译清单中 ‘Alternate month’ 做为指示器 02 的说明注释被打印。

#### 4.4.28 INVMMAP (激活中间映象)

用这个记录层键字来激活一个新的中间映象 (IMM)。IMM 规定格式定义中的中间映象名，它允许用户修改或选择象输入纸盘、页旋转或覆盖的打印参数。

键字格式为：

INVMMAP (中间映象名 | &中间映象名字段)

中间映象名参数是必须的，它是格式定义中的中间映象。它为 8 个字符，可定义为常数或程序到系统字段。

当定义中间映象名参数为程序到系统字段时，字段必须与 INVMMAP 在同一记录格式中，字段长度为 8，类型为 A，用法为 P。

这个键字与 DEVTYPE(\*IPDS)一起用有效，且在打印文件中必须有一个格式定义。如果 DEVTYPE 改为除\*IPDS 以外的类型，则忽略此键字且在打印时发出错误信息。

不能与下列键定一起定义 INVMMAP 键字：

DRAWER

PAGRTT

SPACEA

SPACEB

SKIPA

SKIPB

注：这个键字需要 PSF/400。如果没安装，就不能用这个键字和规定 DEVTYPE(\*AFPDS)，中间映象对文件的其余部分都保持有效，除非用其它的 INVMMAP 修改。

可选指示器对这个键字有效。

图 4-56 给出如何规定 INVMMAP 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A  
A      R RECORD1  
A 02          INVMMAP (MAP1)  
A      R RECORD2          INVMMAP (&MAP)  
A      MAP          8A  P  
A
```

图 4-56 规定 INVMMAP 键字

在图 4-56 中，RECORD1 有一个新的中间映象，RECORD2 允许应用程序设置程序变量 MAP 来规定中间映象的名字。

#### 4.4.29 LINE (线)

使这个记录层键字来打印一条横线或竖线。

键字的格式是：

LINE (竖位 | &竖位字段)

横位 | & 横位字段  
线长度 | & 线长度字段  
线方向 | & 线方向字段  
线宽度 [线填充])

竖位参数是必需的，用来完成相对于 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界线的纵向开始点，有效值是 0-57.790cm(0-22.750in)。

横位参数是必需的，用来定义相对于 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界线的横向开始点，有效值是 0-57.790cm(0-22.750in)。

可以指定竖位和横位参数作为常数，程序到系统字段或两者的组合，象下面显示的那样：

LINE(0.57.1…)  
LINE(&字段 1.3…)  
LINE(2.75 & 字段 2…)  
LINE(&字段 3 & 字段 4…)

字段 1、2、3、4 是程序到系统字段的名字。字段必须与 LINE 键字在同一记录格式中，定义为长度 5，3 位小数，数据类型为 S（非压缩十进制），用法为 P（程序到系统）的字段。

线长度参数是必需的，来定义线的长度，有效值是 0.001-57.790cm(0-22.750in)。

线宽度参数是必需，用来定义线的宽度，有效值是 0.001-57.790cm(0.001-22.750in)，还可以指定下面的特殊值：

值	线宽
*NARROW	12/1440in. (0.008in., 0.022cm)
*MEDIUM	24/1440in. (0.017in., 0.042cm)
*WIDE	36/1440in. (0.025in., 0.064cm)

注： 1.CRTPRTF 命令的 UOM 参数决定这些参数值的计量单位。如果对一个参数指定

的值超出有效的范围，在生成假脱机文件时将被设置标记。

2.根据打印机硬件，由于打印机的分辨能力，小于 0.004 英寸 (0.010cm) 的线不能打印。这种情况发生时，不发出信息。

线方向参数是必需的，并且可以有一个横向(\*HRZ)或竖向(\*VRT)值。

线填充参数是可选的，用它指定线宽度放在相对于实际线坐标的相对位置。例如：如果线宽度 5 并且线填充是\*TOP，则线扩展到横位和竖位参数指定的点的上边。有效值是，对 \*HRZ 是\*TOP 和\*BOT，对\*VRT 是\*LEFT 和\*RIGHT，缺省值分别是\*BOT 和\*RIGHT。

当在记录格式中指定 LINE 键字时，记录格式中的所有字段必须用 POSITION 键字来定位。见 4.4.36 的“POSITION (位置)”。

如在一个指定了 LINE 键字的记录格式中指定常数字段将出现错误信息。

如果线在一页中打印不下，那么在打印时将出现错误信息。

当在文件中指定 LINE 时要在 CRTPRTF 命令上指定 DEVTYPE(\*AFPDS)。如果 DEVTYPE 被改成其它值而不是\*AFPDS，忽略键字并且在打印时要出现警告信息。

在一个记录中最多可指定 40 次这个键字。

不能和下面的键字一起指定 LINE:

SPACEA  
SPACEB  
SKIPA  
SKIPB

注: 使用这个键字必需有 PSF/400, 如果没有安装 PSF/400 就不能用这个键字和指定 DEVTYPE(\*AFPDS)来打印文件。

可选指示器对这个键字有效。

图 4-57 显示了怎样指定 LINE 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A*
A      R REC1          LINE(1.5 3.0 4.25 *HRZ 0.2 *TOP)
A*
A      R REC2          LINE(2.1 1.5 7.5 *HRZ 0.05 *BOT)
A          LINE(&FLD1 &FLD2 4.25 *VRT 0.01 +
A          *LEFT)
A      FLD1      5S 3P
A      FLD2      5S 3P
A*
A      R REC3
A 02          LINE(1.0 1.1 6 *HRZ 1)
A*
A
A
```

图 4-57 指定 LINE 键字

注: CRTPRTF 命令上的 UOM 参数决定参数值的计量单位。

REC1 打印一个 4.25 个单位长的横线, 线从 CRTPRTF 命令中的 FRONTMGN 或 BACKMGN 指定的边界纵向 1.5 个单位, 横向 3.0 个单位开始, 直线 0.2 个单位宽, 超出的宽度被加在线的顶部。

REC2 打印两条线, 第一条线横向打印, 从 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 指定的边界纵向 2.1 个单位和横向 1.5 个单位开始, 直线 7.5 个单位长, 0.05 个单位宽, 超出的宽度加在线的下边。

第二条线的位置由分配给程序到系统字段 FLD1 和 FLD2 值决定。直线竖向打印, 4.25 个单位长 0.01 个单位宽, 超出的宽度加在线的左边。

仅当指示器 02 为 on 时 REC3 打印一条直线, 超出的宽度加在线的底部。

#### 4.4.30 LPI (每英寸行数)

用这个记录层键字来改变一个文件中每英寸行数。如果对一个记录没指定 LPI, 用由

CRTPRTF、CHGPRTF 或 OVRPRTF 的命令的 LPI 值。

键字的格式是：

LPI(4 + 6 + 8 + 9 + 12)

4、6、8、9、12 是有效的参数值。

当在一页中使用多个 LPI 时，所有的 SKIP-TO 中的行数（在 SKIPB、SKIPA 上）变成绝对位置，（每页上的固定位置）。例如，如果页长是 66 行并且文件的 LPI 值是 6，则这个格式 11.0 英寸长。

如果指定跳到行号 48，那么将向下跳 8 英寸打印。在这个例子中，如果以 6LPI 打印 24 行（4 英寸），然后以 8LPI 打印 24 行（3 英寸），则第 48 行是在这页上向下 7 英寸。

在这两个例子中，处理了 48 行。如果使用 SKIPB(55) 键字，则第一个例子跳到 55 行，是根据 6LPI（页的下端 55/6 英寸）。在第二个例子中，前进一页根据 8LPI 在 55 行上开始打印。由于我们要在这一页向下 7 英寸处打印，所以在第二个例子中向下一頁。根据 8LPI，跳到 55 行要小于 7 英寸，因此，要打印在 55 行上，必须跳过当前的页。

数据是根据位置按顺序处理的，而不是根据行号。如果使用 SKIP-TO 到达当前位置的上面位置，（甚至行号大于当前行号），则跳到新页，打印在下一页继续进行。

当在每页使用多个 LPI 时，所有的空行（SPACEA、SPACEB）是相对于当前的位置来做的。例如，要以 6LPI 打印 24 行（4 英寸），然后以 8LPI 打印 24 行（3 英寸），则第 48 行是在这页 7 英寸之下。如果在这之后做 SPACEA(4)(LPI 仍然是 8LPI)，则将从最后一行向下空 1/2 英寸，并且在这页上向下的总量是 7.5 英寸。使用 LPI 键字时，建议你使用 SPACEA 和 SPACEB 键字。

LPI、SKIP 和 SPACE 键字按下面的顺序处理：

LPI

SKIPB

SPACEB

SPACEA

SKIPA

这样，SPACE 和 SKIP 使用新的 LPI 值。作为下一行和后面行的 LPI，以在 LPI 参数上指定 LPI 的值打印，直到处理下一个记录格式为止，这个参数值保持有效。在这个记录格式结束时，LPI 的值改回到文件层的值。

LPI 仅在行边界上起作用。如果在一行内修改 LPI，则新值在行结束时起作用，且不出现诊断信息。

不能在有 BLKFOLD、CPI 和 DFNCHR 规定的同一记录格式上指定这个键字。如果与 LPI 一起使用了这些键字中任一个，则不能建立文件。

这个键字对 IPDS 打印机有效，且打印机总有先进功能打印支持。在一个用 DEVTYPE(\*SCS) 建立的文件中指定 LPI 键字时，出现一个警告信息。

溢出行号（在 CRTPRTF 命令上的 OVRFLOW 参数），不依据文件层 LPI 值转换成绝对位置（英寸）。当到达溢出位（英寸）时，发出溢出条件信号。

例如：页长=66，LPI=6，OVRFLOW=60（10 英寸）：

- 以 6LPI 打印 36 行 (6 英寸)
- 以 4LPI 打印 16 行 (4 英寸)

在 52 行处理之后发生溢出条件。

可选指示器对这个键字无效。

图 4-58 显示如何指定 LPI 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A           R RECORD1          LPI(6)
A           SPACEB(6)
A
```

图 4-58 指定 LPI 键字

不考虑在 CRTPRTF、CHGPRTF 或 OVRPRTF 命令上指定的 LPI，在打印下一行之前打印机设备向下空一英寸。

#### 4.4.31 MSGCON (信息常量)

使用这个字段层键字来指明在信息描述中的常数据段的正文。如果信息描述在 DDS 编译时不存在，则文件不能生成。如果改变信息描述必须重新生成文件。

键字的格式是：

**MSGCON (长度 信息标识 [库名/]信息文件名)**

长度参数指定信息描述的最大长度。长度可以是从 1 到 132 个字节。如果信息描述比指定的长度短，剩余的字节用空白填充（十六进制 40）。如果信息描述比指定的长度长，则按指定的长度截取信息描述并出现警告信息。

信息标识参数指定包含作为常数据段值的正文的信息描述。

信息文件名参数指定包含信息描述的信息文件。库名参数是可选的。

必须明显地为字段指定 MSGCON 键字，不能用 MSGCON 键字来初始化命名字段。

DFT 和 MSGCON 键字的功能等效，如果对一个字段同时指定 DFT 和 MSGCON 键字，则忽略 MSGCON 键字而且文件不能建立。

不能与 DATE、DFT、EDTCDE、EDTWRD 和 TIME 键字一起指定 MSGCON 键字。

可选指示器对这个键字无效，但是，它们可以为指定了这个键字的字段设置条件。

图 4-59 显示了怎样指定 MSGCON 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A       R RECORD1
00020A           2 1MSGCON(10 MSG0001 MESSAGES/MSGF)
A
```

图 4-59 指定 MSGCON 键字

在库 MESSAGES 的信息文件 MSGF 中的 MSG0001 存有信息正文。

#### 4.4.32 OVERLAY (覆盖)

使用这个记录层关键字来打印一个覆盖。

关键字的格式是：

```
OVERLAY (库名 | &库名字段  
          覆盖名 | &覆盖名字段  
          竖位 | &竖位字段  
          横位 | &横位字段)
```

覆盖名、竖位和横位参数是必需的。

使用可选的库名参数来进一步限定覆盖。如果没有指定库名参数，打印覆盖时使用 \*LIBL 查找。

注：如果应用程序使用了私有资源（例如，字形、页段、覆盖或系统没分配的 GDF 文件），当引用这些资源时，如果指定了\*LIBL 或者没指定库名，那么应用程序建立假脱机文件使用的库列表必须是可用的。

可以指定库名、覆盖名、竖位、横位参数为常数、程序到系统字段或二者的组合、象下面显示的那样：

```
[库名/]覆盖名…  
[库名/]&字段 1…  
[&字段 2/]覆盖名…  
[&字段 3/]&字段 4…
```

当指定一个库名作为一个程序到系统字段，字段必须与 OVERLAY 键字要在同一记录格式中。必须定义为长度为 10，数据类型为 A (字符型) 和用法为 P (程序到系统) 的字段。

当指定覆盖名作为程序到系统字段时，它必须与 OVERLAY 键字在同一记录格式中，且被定义成长度为 8，数据类型为 A (字符) 和用法为 P (程序到系统) 的字段。

当指定竖位或横位参数作为程序到系统字段时，必须定义它为长度为 5，有 3 位小数，数据类型 S 和用法为 P 的字段。

竖位参数定义相对于 CRTPRTF 命令中的 FRONTMGN 或 BACKMGN 参数指定边界的纵向开始点，有效值是 0 到 57.790cm(0-22.750in)。

横位参数定义覆盖相对于 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定边界的横向开始点，有效值是 0 到 57.790cm(0-22.750in)。

注：C RTPRTF 命令的 UOM 参数决定竖位和横位参数值的计量单位。如果指定的参数值超出了有效范围，在生成假脱机文件时有标志设置。

如果覆盖在一页中打不下，那么在打印时将发出错误信息。

当在文件中指定 OVERLAY 时须在 CRTPRTF 命令上指定 DEVTYPE(\*AFPDS)。如果 DEVTYPE 改为其它值而不是\*AFPDS，忽略关键字并在打印时发出警告信息。

当在一个记录格式中指定 OVERLAY 键字时，记录格式中的所有字段必须用 POSITION

键字定位，详细信息见 4.4.36 “POSITION (位置)”。

如果在指定了 OVERLAY 键字的记录格式中指定了常数字段将发出错误信息。

可以在一个记录中多次指定这个键字。

在一个单页上最多可以使用 10 次覆盖。

当在打印文件上使用 PAGRTT 键字或 PAGRTT 参数时，OVERLAY 不能自动旋转。对覆盖和旋转的详细信息见打印机设备程序设计资料。

不能与下面的键字一起指定 OVERLAY：

SPACEA、SPACEB、SKIPA、SKIPB

注：使用这个键字必须有 PSF/400。如果没有安装 PSF/400，就不能使用这个键字和指定 DEVTYPE(\*AFPDS) 打印文件。

可选指示器对这个键字有效。

图 4-60 显示了如何指定 OVERLAY 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A*
A      R REC1          OVERLAY (MYLIB/OVL04 1.234 14.62)
A*
A      R REC2          OVERLAY (&LIB/&OVLS &POSD &POSA)
A      LIB             10A P
A      OVLS            8A P
A      POSD            5S 3P
A      POSA            5S 3P
A*
A      R REC3          OVERLAY (MYOVL 11.219 0.2)
A                  OVERLAY (YOUROVL 7.3 9.27)
A*
A      R REC4
A      01              OVERLAY (MYLOGO 0.0 3.01)
A
```

图 4-60 指定 OVERLAY 键字

注：CRTPRTF 命令的 UOM 参数决定参数值的计量单位。

REC1 打印在库 MYLIB 中的覆盖 OVL04，覆盖从 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界纵向 1.234 个单位和横向 14.62 个单位打印。

REC2 允许应用程序通过设置程序变量 LIB 和 OVLS 来分别指定库和覆盖名。覆盖从 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界行向下 0.15 个单位和列向右 1.92 个单位打印。

REC3 打印两个覆盖，MYOVL 从 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界纵向 11.219 个单位和横向 0.2 个单位打印。YOUROVL 从 CRTPRTF 命令指定的边

界纵向 7.3 个单位和横向 9.27 个单位打印，两个覆盖都使用\*LIBL 定位。

只有指示器 01 为 on 时 REC4 打印 MYLOGO。

#### 4.4.33 PAGNBR (页号)

使用这个字段层键字来指定未命名的,长 4 位,有页号的区位十进制字段的位置,只需指定 PAGNBR 键字和字段的位置(字段的位置可以仅是列号,也可以是行号和列号),并且还可以选择 CHRSIZ、COLOR、FONT、HIGHLIGHT、UNDERLINE 或 TEXT 键字。

这个键字没有参数。

在打开打印机文件时,OS/400 设置页计数为 0,并且在每打印新的一页之前加 1,即使你没有指定 PAGNBR 键字也是这么做,每次打印指定了 PAGNBR 键字的字段时都打印页号,页号不能大于 9999。到了 9999 要重设置它。要重设置页计数,用可选指示器为 PAGNBR 设置条件。当程序选择 PAGNBR 时,OS/400 重设页号(请参看图 4-61)。

和 PAGNBR 键字一起可指定 EDTCDE 或 EDTWRD。

可选指示器对这个键字有效。

可以在没有指示器的一行上规定字段和键字,或在不同行上指定不同指示器来规定字段和键字。

下面的例子给出它们之间的区别:

在没有指示器的一行上指定: 总是打印页号,不能重置页号到 1,如图 4-61 所示:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00100A      R RECORD  
00110A          1 60' PAGE:  
00120A          +1PAGNBR  
A
```

图 4-61 在没有指示器的一行上规定 PAGNBR

注: 110 和 120 行指定两个常数字段: ‘PAGE:’ 和页号本身(+1 指定位置)

在不同行上规定指示器: 如果字段指示器(例中 01)为 off,不打印字段,即使键字指示器(例中的 02)为 on 也这样。当字段指示器为 ON 时,打印字段。当键字指示器为 OFF 时,页计数增加;当键字指示器为 ON 时,页计数被重新置为 1,不管键字指示器是 on 或 off,总打印页号。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00100A      R RECORD  
00110A  01          1 60' PAGE:  
00120A  01          +1  
00130A  02          PAGNBR  
A
```

图 4-62 在不同行用不同指示器规定 PAGNBR

#### 4.4.34 PAGRTT (页旋转)

使用这个记录层键字来指定装入打印页正文的旋转度数。PAGRTT 键字只对 3812、3816、3820、3825、3827 和 4028 打印机有效。如果对一个记录没指定 PAGRTT 键字，页旋转由 CRTPRTF、CHGPRTF、OVRPRTF 命令指定的值设置。

键字的格式是：

PAGRTT(0 | 90 | 180 | 270)

键字的有效参数值是 0、90、180 和 270，0 表示没有旋转。其它的值指定从 0 度顺时针旋转的度数。

注：对于使用图画纸的 3835 打印机，在 DDS 文件中使用逆时针旋转。

PAGRTT 键字产生隐含的换页操作，如果纸不在页边界上，则不能使用这个键字，并出现警告信息。

PAGRTT、SKIP 和 SPACE 键字按下列顺序处理：

SKIPB SPACEB PAGRTT SPACEA SKIPA

PAGRTT 键字对记录格式持续期间保持有效。如果在下一记录格式上不使用 PAGRTT 键字，则它恢复到命令层指定的 PAGRTT 值。

注：

1.PAGRTT 键字对整个页保持有效，在指定了 PAGRTT 的记录格式的末尾，在处理下一页文件才变回到文件层的旋转。例如：如果文件层参数指定 PAGRTT(0)并且你写下面情况之一时，则直到写记录之前，文件不返回到旋转为 0：

记录格式 A (PAGRTT(90))

记录格式 B (与记录 A 在同一页)

记录格式 C (与记录 A 在同一页)

记录格式 D (与记录 A 在同一页)

记录格式 E (下一页)

2.对于用 DEVTYPE(\*SCS)的生成文件，如果 PAGRTT 键字是在跨几页的记录格式上指定的，则它仅在指定它的页上有效。

当页旋转时，页长和页宽交换以使长变成宽并且宽变成长，在某种条件下是不能这样交换的，这些情况包括：

(1) 在 CRTPRTF、CHGPRTF 或 OVRPRTF 命令中的 PAGRTT 是\*DEVD 或\*COR。

(2) CRT/CHG/OVRPRTF 命令上的 FONT 是\*DEVD。

当长度和宽度不能交换时，将出现诊断信息。

在页旋转时，不支持折叠和截断，也就是说，不能使用 BLKFOLD 键字和 CRTPRTF、CHGPRTF 或 OVRPRTF 命令中的 FOLD 参数。

可选指示器对这个键字有效。

图 4-63 显示了如何指定 PAGRTT 键字

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

A

R RECORD1

A 02			SKIPB(1)
A 02			PAGRTT(270)
A	FIELD1	3A	2 01
A	R RECORD2		
A	FIELD1	3A	6 01
A	R RECORD3		
A	FIELD1	3A	5 01
A			

图 4-63 指定 PAGRTT 键字

如果指示器 02 为 on, RECORD1 旋转 270 度, 由于指定 SKIPB(1), 使记录在新页上开始。

当打印 RECORD2 时, 由于它和 RECORD1 在同一页上, 它也被旋转 270 度。

因为 RECORD3 在一个新页上, 所以使用 CRTPRTF、CHGPRTF 或 OVRPRTF 指定的值。

#### 4.4.35 PAGSEG (页段)

使用这个记录层键字来打印一个页段。

键字的格式是:

```
PAGSEG(库名 | &库名字段
        页段名 | &页段名字段
        竖位 | &竖位字段
        横位 | &横位字段)
```

页段名、竖位和横位参数是必需的。

用可选的库名参数来进一步限定页段, 如果没指定库名, 则页段在打印时用\*LIBL 查找。

注: 如果应用程序使用私有资源 (例如, 字形、页表、覆盖或不受系统限制的 GDF 文件), 当引用这些资源时, 如果用\*LIBL 或没指定库名, 对于应用程序生成的假脱机文件使用的库列表必须是可用的。

可以指定程序名、页段名、竖位, 横位参数为常数, 程序到系统字段或者二者的组合, 如下面所示:

```
[库名/]页段名...
[库名/]&字段 1...
[&字段 2/]页段名...
[&字段 3/]&字段 4...
```

当指定库名为程序到系统字段时, 此字段要与 PAGSEG 键字在同一记录格式中, 且定义为长度为 10, 数据类型为 A (字符) 和用法为 P (程序到系统) 的字段。

当指定页段名为程序到系统字段时, 字段要在与 PAGSEG 键字相同的记录格式中, 且定义为长度为 8, 数据类型为 A (字符) 和用法为 P (程序到系统) 的字段。

当指定竖位或横位作为程序到系统字段时，字段必须定义成 5 位长，带 3 位小数，数据类型 S 和用法为 P 的字段。

竖位参数定义相对于 CRTPRTF 命令的 FRONTMGN 或 BACQMGN 参数指定边界的纵向开始点，有效值是 0-57.790cm(0-22.750in.)。

横位参数定义相对于 CRTPRTF 命令中 FRONTMGN 或 BACKMGN 参数指定边界的横向开始点，有效值是 0-57.790cm(0-22.750in.)。

注：CRTPRTF 命令的 UOM 参数决定竖位和横位参数值的计量单位，如果对参数指定的值超过了有效范围，在生成假脱机文件时设置标记。

如果页段在一页上放不下，在打印时给出错误信息。

当在文件中指定 PAGSEG 中，需在 CRTPRTF 命令中指定 DEVTYPE(\*AFPDS)，如果 DEVTYPE 被改变成其它值而不是\*AFPDS，忽略键字并且打印时给出警告信息。

当在记录格式中指定 DEVTYPE 时，记录格式中的所有字段必须用 POSITION 键字定位。详细信息见 4.4.36 的“POSITION（位置）”。

在指定了 PAGSEG 键字的记录格式中如果指定了常数字段，将发出错误信息。

在一个记录中可多次指定 PAGSEG 键字。

每页最多可以使用 10 个页段。

在打印文件上使用 PAGRRTT 键字或 PAGRRTT 参数时，页段不能自动旋转。关于页段和旋转的信息见打印设备程序设计资料。

不能和下列键字一起在同一层指定：

SPACEA

SPACEB

SKIPA

SKIPB

注：使用这个键字是需要 PSF/400 的，如果没安装 PSF/400，不能使用这个键字和指定 DEVTYPE(\*AFPDS) 来打印文件。

可选指示器对这个键字有效。

图 4-64 显示了如何指定 PAGSEG 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A*
A      R REC1          PAGSEG (MYLIB/PAGSEG5 3.527 4.162)
A*
A      R REC2          PAGSEG (&LIB/&PSEG &POSD &POSA)
A      LIB             10A  P
A      PSEG            8A  P
A      POSD            5S  3P
A      POSA            5S  3P
A*
A      R REC3          PAGSEG (MYSEG 0.0 3.759)
A                  PAGSEG (YOURSEG 0.0 5.233)
```

```

A*
A          R REC4
A 01
A*          PAGSEG(MYSEG 0.0 3.01)
A

```

图 4-64 指定 PAGSEG 键字

注：CRTPRTF 命令的 UOM 参数决定参数值的计量单位。

REC1 打印在 MYLIB 中的 PAGSEG5，页段从 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 指定的边界纵向 3.527 个单位和横向 4.162 个单位打印。

REC2 允许应用程序设置字段 LIB 和 PSEG 来分别指定库和页段名。页段从 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界纵向 8.5 个单位和横向 6.72 个单位打印。

REC3 打印两个页段，MYSEG 从 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 指定的边界纵向 0 个单位、横向 3.759 个单位打印。YOURSEG 从 CRTPRTF 命令指定的边界纵向 0 个单位和横向 5.233 单位。打印两个页段都用\*LIBL 查找。

只有当 01 指示器为 on 时，REC4 打印 MYSEG。

#### 4.4.36 POSITION (位置)

使用这字段层键字来定义命名字段在页上的位置。

键字的格式是：

POSITION (竖位！&竖位字段 横位！&横位字段)

竖位参数是必需的，用来定义相对于 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定的边界的纵向开始点，有效值是 0-57.790cm(0-22.750in.)。

横位参数是必需的，用来定义相对于 CRTPRTF 命令的 FRONTMGN 或 BACKMGN 指定的边界的横向开始点，有效值是 0-57.790cm(0-22.750in.)。

可以指定竖位和横位参数为常数，程序到系统字段或两者的组合，如下面所示：

```

POSITION(3.56  6.24)
POSITION(&字段 1  9.625)
POSITION(0.5  &字段 2)
POSITION(&字段 3  &字段 4)

```

字段 1、2、3、4 是程序到系统字段的名字，它必须与 POSITION 键字在同一记录格式中且定义为长度为 5，有 3 位小数，数据类型 S（区位十进制）和用法为 P（程序到系统）的字段。

注：CRTPRTF 的 UOM 参数决定竖位和横位计量单位。如果指定的参数值超过了有效范围，则在生成打印文件时，有标志设置。

如果字段在一页上打不下，则在打印时将发出错误信息。

如果还指定了行和列值（39-44 列）则发出错误信息。

由于 POSITION 键字允许字段放在页上的任意位置，所以使用 POSITION 键字不会产

生新页。将用 ENDPAGE 键字来结束当前页进入到下一页。

如果对一个字段指定了 POSITION 键字，则记录格式中的所有字段也必须指定 POSITION 键字。不允许 39-44 列的位置项。

如果在指定了 POSITION 键字的记录格式中指定常数字段将出现错误信息。

当在文件中指定 POSITION 时要在 CRTPRTF 命令中指定 DEVTYPE(\*AFPDS)。如果 DEVTYPE 被改成其它值而不是\*AFPDS，则忽略键字并在打印时出现警告信息。

不能和下面的键字一起指定 POSITION:

SPACEA

SPACEB

SKIPA

SKIPB

注：使用这个键字需要 PSF/400，如果没有安装 PSF/400，则不能用这个键字和指定 DEVTYPE(\*AFPDS) 来打印文件。

可选指示器对这个键字有效。

图 4-65 显示了怎样指定 POSITION 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A*
A      R REC1
A      FLD1      6S 2      POSITION(2.0 1.983)
A*
A      FLD2      42A      POSITION(&FLD2A &FLD2B)
A      FLD2A     5S 3P
A      FLD2B     5S 3P
A*
```

图 4-65 指定 POSITION 键字

注：CRTLPRTF 命令的 UOM 参数决定参数值的计量单位。

在 REC1 中，FLD1 从 CRTLPRTF 命令的 FRONTMGN 或 BACKMGN 参数指定边界纵向 2.0 个单位和横向 1.983 个单位打印。

应用程序通过对程序到系统字段 FLD2A 和 FLD2B 赋值来决定 FLD2 的位置。

#### 4.4.37 PRTQLTY (打印特性)

使用这个记录层或字段层键字来改变文件中的打印特性。

键字的格式是：

PRTQLTY (打印特性)

参数是必需的而且必须是下面值中的一个：

\*STD (标准特性)

\*DRAFT (草图特性)

\*NLQ (近字母特性)

\*FASTDRAFT (快速草图特性)

PRTQLTY 键字只允许在指定 CHRSIZ 或 BARCODE 键字的记录或字段上使用。

如果没指定这个键字, 打印特性由 CRTPRTF、CHGPRTF 或 OVRPRTF 命令的 PRTQLTY 参数设置。

如果在记录层指定 PRTQLTY, 它适用于记录中在字段层没指定 PRTQLTY 所有字段。

PRTQLTY 仅对 IPDS 打印机有效。如果在有 DEVTYPE(\*SCS)生成的文件中指定这个键字, 则在文件生成时出现警告信息。

如果在带有 DEVTYPE(\*AFPDS)生成的文件中指定这个键字, 则打印特性只能在页的边界改变。如果 PRTQLTY 是在数据放在页上之前接收的, 页的特性改变了。否则, 忽略此键字并且往应用程序送一个诊断信息。

如果和 BLKFOLD、CPI 或 DFNCHR 键字在同一记录格式中使用 PRTQLTY, 文件不能建立。

这个键字允许可选指示器。

图 4-66 显示了如何指定 PRTQLTY 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RECORD  
A          FIELD3      10S 0 4 65BARCODE (UPCE 6)  
A          PRTQLTY (*DRAFT)  
A
```

图 4-66 指定 PRTQLTY 键字

FIELD3 中 UPCE 条形码用草图特性打印。

#### 4.4.38 REF (引用)

使用这个文件层键字来指定从哪个文件中得到字段描述。

键字的格式是:

REF([库名/]数据文件名 [记录格式名])

当要复制来自预先定义的记录格式中一个或多个字段的描述信息时, 使用 REF。在 REF 键字上可以一次给出文件名, 而不象在 REFFLD 键字上对每个字段描述给出引用的文件名。

如果在被引用的文件中有多个记录格式, 要指定一个记录格式名作为这个键字的参数值, 告诉 OS/400 程序要使用哪一个记录格式, 除非这个格式是顺序检索的。

数据文件名对这个键字是必需的, 记录格式名和库名是可选的。

如果没有指定库名, 使用在建立文件时的当前库列表。如果没指定记录格式, 则每个格式按顺序查找(按它们被指定的顺序), 使用第一个出现的字段名。详细的信息见附录 A “什么时候指定 REF 和 REFFLD”。

在这个键字上可以指定一个分布式数据管理 (DDM) 文件。当使用一个 DDM 文件时,

数据文件名和库名是源系统上的 DDM 文件和库名，记录格式名是在目标系统上远程文件中的记录格式名。

可选指示器对这个关键字无效。

图 4-67 和图 4-68 显示了怎样指定 REF 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               REF(FILE1)  
00020A      R RECORD  
00030A      FLD1      R      2  2  
A
```

图 4-67 指定 REF 键字（例 1）

FLD1 与在 FILE1 中的第一个（或唯一的）FLD1 有相同的属性。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               REF(LIB1/FILE1 RECORD2)  
00020A      R RECORD  
00030A      FLD1      R      2  2  
A
```

图 4-68 规定 REF 键字（例 2）

FLD1 与在 LIB1 中 FIL1 中的 RECORD2 中的 FLD1 有相同的属性。

#### 4.4.39 REFFLD (引用字段)

下面三种情况之一时使用这个字段层关键字：

1. 被引用字段的名字与在 19-28 列的名字不同。
2. 被引用字段名与在 19-28 列的名相同，但被引用字段的记录格式、文件或库名与 REF 键字指定的不同。
3. 被引用字段与引用字段出现在同一 DDS 源文件中。

这个关键字的格式是：

```
REFFLD([记录格式名/]被引用字段名 [{*SRC | [库名/]数据文件名}])
```

被引用字段名是必须的，即使被引用字段名与引用字段相同也一样。当被引用文件含有多个记录格式时使用记录格式名。当被引用字段名与引用字段名在同一 DDS 源文件中时使用\*SRC (而不是数据库文件名)。当没有指定数据文件名和库名时\*SRC 是缺省值。

注：当在同一 DDS 源文件中引用字段时，被引用的字段必须在正定义的字段的前面。

当需要查找某个特定的数据文件时，须指定数据文件名。(如果需要用它的库名限定)。

在同一 DDS 源文件中，如果在文件层指定了 REF 并在字段层指定 REFFLD，则查找顺序依赖于 REF 和 REFFLD 键字。详细信息见本章“引用(29 列)”。

可以在这个键字中指定一个分布式数据管理文件（DDM）。

当使用 DDM 文件时，数据文件名和库名是在源系统上的 DDM 文件名和库名，被引用字段名和记录格式名是在目标系统上远程文件中的字段名和记录格式名。

可选指示器对这个键字无效。

图 4-69 显示如何指定 REFFLD 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R FMAT1  
00020A      ITEM      5      1  
00030A      ITEM1    R      2REFFLD(ITEM)  
00040A      ITEM2    R      12REFFLD(FMAT1/ITEM)  
00050A      ITEM3    R      22REFFLD(ITEM FILEX)  
00060A      ITEM4    R      32REFFLD(ITEM LIBY(FILEX)  
00070A      ITEM5    R      42REFFLD(FMAT1/ITEM LIBY(FILEX)  
00080A      ITEM6    R      52REFFLD(ITEM *SRC)  
A
```

图 4-69 指定 REFFLD 键字

由于没指定 REF 键字，所以作为 00030 行和 00040 行的缺省值是要查找规定它们的 DDS 源文件。在第 00080 行，参数值\*SRC 明显地指定了源文件，关于这个规范的解释，见附录 A 中的例子“什么时候指定 REF 和 REFFLD”。

#### 4.4.40 SKIPA (打印后跳)

使用这个文件层、记录层或字段层键字指定打印设备在打印一行或几行之后跳指定的行数。

键字的格式是：

SKIPA (跳行数)

参数值是必需的，必须在 1-255 的范围内。

如果在文件层指定了这个键字，必须用一个或多个指示器选择它。如果在记录层或字段层指定这个键字，指示器是可选的。在文件中的每个记录打印之后执行跳转。

如果在记录层指定这个键字，在记录相关的所有行打印之后，文件层 SKIPA 键字使用之前做跳转。

如果在字段层指定这个键字，在字段打印之后执行跳转。

注：如果没有使用行号也没指定 SKIP 或 SPACE 键字则可能导致重叠打印。

可以指定这个键字在文件层一次，记录层一次和每个字段一次。

在文件层的这个键字对所有记录有效。但对在记录层或字段层指定了行号（39-41 列）的记录无效，（行号项被作为错误标志）。

如果在记录格式中指定了 BOX, ENDPAGE, GDF, LINE, OVERLAY, PAGSEG 或 POSITION 键字，则 SKIPA 键字在字段层或记录层都无效。

对在 CRTPRTF 命令有 DECTYPE(\*AFPDS) 定义的文件，不能在文件层定义这个键字。

图 4-70 显示了如何指定 SKIPA 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00011A          FIELDA      132A      1SKIPA(12)  
A
```

图 4-70 指定 SKIPA 键字

#### 4.4.41 SKIPB (打印前跳)

使用这个文件层、记录层或字段层键字来指定打印机设备在打印下一行之前要跳到指定的行数。

键字的格式是：

**SKIPB (跳行数)**

参数值是必需的，范围在 1-255。

如果在文件层指定这个键字，必须用一个或多个指示器来选择它，否则，指示器是可选的。跳转在文件中每个记录打印之前做或使用了文件层 SKIPB 操作之后执行。

如果在记录层指定这个键字，在那些相关的行打印之前执行跳转。

如果在字段层指定这个键字，在字段打印之前，执行跳转。

可以在文件层指定这个键字一次，在记录级指定一次，在每个字段指定一次。

文件层的这个键字对所有记录有效，但对在记录层或字段层指定了行号（39-41 列）的记录无效。（行号做错误标志）。

注：如果没使用行号，没指定跳转或空白键字可能引起重叠打印。

如果在记录格式上指定了 BOX、ENDPAGE、GDF、LINE、OVERLAY、PAGSEG 或 POSITION 键字，SKIPB 键字无论在字段层还是在记录层都无效。

对于在 CRTPRTF 命令中 DEVTYPE(\*AFPDS) 定义的文件，不允许这个键字。

可选指示器对这个键字有效。

图 4-71 显示了怎样指定 SKIPB 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00022A          R RFMTPR           SKIPB(5)  
A
```

图 4-71 规定 SKIPB 键字

#### 4.4.42 SPACEA (打印后空行)

使用这个记录层或字段层键字来指定打印机设备在打印一行或几行之后空出的一些行的数量。

键字的格式是：

**SPACEA(空行数)**

参数值是必需的，并且一定在 0 到 255 的范围内。

如果在记录层指定了这个键字，空行发生在和这个记录相关的所有行打印完之后。可以在记录层指定一次这个键字同时对每个字段指定一次。

如果在字段层指定 SPACEA，在字段打印之后执行空白行。

这个键字对指定了行号（39-41 列）的记录无效，（行号为错误标志）。

注：如果没使用行号，没有指定 SPACE 或 SKIP 键字，可能引起重叠打印。

如果记录格式中指定了 Box、ENDPAGE、GDF、LINE、OVERLAY、PASSAGE 或 POSITION 键字，则 SPACEA 键字无论是对字段层还是对记录层都无效。

可选指示器对这个键字有效。

图 4-72 显示了怎样指定 SAPCEA 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00100A      R RFMTPR          SPACEA(1)  
00101A      FIELDA        132      1  
A
```

图 4-72 指定 SAPCEA 键字

#### 4.4.43 SPACEB（打印前空行）

使用这个记录层或字段层键字来指定打印设备在打印下一行或几行之前空出一些行的数量。

键字的格式是：

**SPACEB**（空行数）

参数是必需的，并且一定在 0 到 255 的范围内。

如果在记录层指定这个键字，空行发生在与记录相关的所有行被打印之前。可以在记录层指定一次这个键字，或者对每个字段指定一次。

如果在字段层指定这个键字，在所有字段打印之前执行空白。

键字对指定了行号（39-41 列）的记录无效（行号为错误标记）。

注： 1. 如果没使用行号，没指定 SPACE 或 SKIP 键字，可能引起重叠打印。

如果记录格式中指定了 BOX、ENDPAGE、GDF、LINE、OVERLAY、PAGSEG 或 POSITION 键字，那么 SPACEB 键字不论在字段层或记录层都无效。

可选指示器对这个键字有效。

图 4-73 显示了如何指定 SPACEB 键字

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      FIELDA        25A      55SPACEB(3)  
00011A      FIELDB        30       100  
A
```

图 4-73 指定 SPACEB 键字

#### 4.4.44 TEXT (正文)

使用这个记录层或字段层键字来为用于程序文档的记录格式或字段提供一个描述正文。

键字的格式是:

TEXT(‘正文说明’)

正文必须用单引号括起来,如果正文的长度多于 50 个字符,则只有前 50 个字符被高级编译器使用。

可选指示器对这个键字无效。

图 4-74 显示了如何在记录层或字段层指定 TEXT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R CUSMST          TEXT('Customer Master Record'  
00020A      FLD1           3 0       TEXT('ORDER NUMBER FIELD')  
A
```

图 4-74 规定 TEXT 键字

#### 4.4.45 TIME (时间)

这个字段层键字以 6 个字节长的常数字段打印当前系统时间,可以指定字段的位置。

TIME 键字并且可选地指定 EDTCDE、EDTWRD、COLOR、HIGHLIGHT、CHRSIZ、FONT、UNDERLINE 或 TEXT 键字,17-38 列必须为空白。

这个键字没有参数。

编辑字 ‘0-:-:-’ (-代表空格) 是对 TIME 假定的编辑字。可以指定另外的编辑字或用户定义的编辑码 (5-9) 来改变 IBM 提供的编辑方式。

可选指示器对这个键字无效,尽管可以用可选指示器来为指定的字段设条件。

图 4-75 给出了如何指定了 TIME 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A 20                  1 56TIME  
00020A 21                  1 56TIME  
A                           EDTWRD('0 &HRS& &MINS& &SECS')  
A
```

图 4-75 指定了 TIME 键字

如果系统时间是 110645, 打印的时间如下所示:

如果可选指示器 20 为 on, 时间打印成

11:06:45

如果可选指示器 21 为 on, (并且可选指示器 20 为 off) 时间打印成:

11 HRS 06 MINS 45 SECS

#### 4.4.46 TRNSPY (透明度)

这个字段层关键字同来当应用程序送一个输出操作打印定义的字段时，防止把重定义的编码点（使用 DFNCHR）解释成 SCS 打印机的控制命令。

这个关键字没有参数。

如果没在你的程序中对将十六进制数据传送到打印机的字段指定 TRNSPY 键字，则编码点可能被解释成影响打印机操作的 SCS 命令。一个编码点是你能在字符集中分配给字符的 256 个值中的一个。在 AS/400 系统上，编码点由两位十六进制数来标识。

在看下列情况下，必须指定 TRNSPY 键字：

- 在用 OEVTYPE(\*SCS)建立的打印文件中指定 CVTDTA 键字。
  - DFT 指定一个十六进制值（显式或隐含的 DFT 键字）。

在用 DEVTYPE(\*IPDS)建立的打印文件中，不必和 CVTDTA 键字一起指定 TRNSPY 键字。然而，会出现一个警告信息说明 DEVTYPE 不能变成\*SCS。

当用 DEVTYPE(\*AFPDS)建立的文件中指定 TRNSPY 时在建立时，将出现警告信息。

TRNSPY 键字仅当数据类型是字符时有效。

当与 CVTDTA 键字一起指定 TRNSPY 键字时，程序可以在字段中放置字符数据，并且 OS/400 在字段送往打印机时将其转换成十六进制数。每对十六进制数对应于 OS/400 系统字符集中一个编码点。使用 DFNCHR 键字，可以为 5224 打印机和 5225 打印机设计自己定义的字符，请参考 DFNCHR 键字的描述。由于字段的打印长度是指定的长度的一半。因此，这个字段长度必须是偶数。

如果没和 CVTDTA 键字一起指定 TRNSPY 键字，指定的字段长度是打印的长度。

只有 5224 打印机和 5225 打印机支持这个键字。

可选指示器对这个键字无效。

下面的例子给出了怎样指定 TRNSPY 键字。

图 4-76 显示了如何与 CVDATA 键字一起指定 TRNSPY 键字。当你的程序在字段中传送字符数据时，OS/400 程序把它转换成十六进制数据。

00010A	R RECORD	SPACEB(1)	
00020A	FLD1	10	1TRNSPY CVTDTA
00030A	FLD2	20	6TRNSPY CVTDTA

图 4-76 规定 TRNSPY 键字 (例 1)

程序可以传送在 FLD1 和 FLD2 中的字符数据。OS/400 程序把它们转换成十六进制数据打印。只有字符 0 到 9 和 A 到 F 有效，空格无效。

FLD1 和 FLD2 的打印长度是指定长度（FLD1 是 5 位长，FLD2 是 10 位长）的一半。

为了打印用户定义字符，还必须对这个 DDS 指定 DFNCHR 键字。

当 FLD1 的内容是 ‘C1C1C1C1C1’，FLD2 的内容是 ‘C2C2C2C2C2C2C2C2C2C2’ 时，下面显示怎样打印 RECORD：

AAAAABBBBBBBBBB

图 4-77 显示了在没有 CVTDTA 键字时如何指定 TRNSPY 键字。在这个例子中，程序必须在字段中传送十六进制数。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RECORD2          SPACEB(1)  
00020A      FLD3             5      1TRNSPY  
00030A      FLD4             10     6TRNSPY  
A
```

图 4-77 规定 TRNSPY 键字（例 2）

程序必须往字段 FLD3 和 FLD4 中传送十六进制数据。只有十六进制字符 0 到 9 和 A 到 F 有效，空格无效，没有 CVTDTA 键字，所有字段的打印长度是指定的长度。

#### 4.4.47 TXTRTT (正文旋转)

使用这个字段层字来旋转包含在字段中的正文。

键字的格式是：

**TXTRTT(字段旋转度数)**

字段旋转参数是必需的，由它来控制字段的旋转，有效值是 0、90、180 和 270 度。

要在文件中指定 TXTRTT 时，须在 CRTPRTF 命令上指定 DEVTYPE(\*AFPDS)。如果 DEVTYPE 被改成其它值而不是\*AFPDS，那么忽略键字，在打印时要发出警告信息。

注：使用这个键字，PSF/400 是必需的，如果没有安装 PSF/400，就不能使用这个键字和指定 DEVTYPE(\*AFPDS) 来打印文件。

可选指示器对这个键字有效。

图 4-78 给出了如何指定 TXTRTT 键字

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A*  
A      R REC1  
A      FLD05        10      35 15TXTRTT(90)  
A*  
A      R REC2  
A      FLD06        7S 2    TXTRTT(270)  
A                  POSITION(6.5 13.8)  
A
```

图 4-78 指定 TXTRTT 键字

REC1 中的 FLD05 旋转 90 度。

REC2 的 FLD06 旋转 270 度。

#### 4.4.48 UNDERLINE (下划线)

使用这个字段层关键字来指定在字段打印时，OS/400 为其加下划线。仅当打印机支持下划线时，才能指定 UNDERLINE。

这个关键字没有参数。

可选指示器对这个关键字有效。

图 4-79 显示了怎样指定 UNDERLINE 关键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00100A          ALLOC      R        17 11  
00101A 03 04                      UNDERLINE  
A
```

图 4-79 指定 UNDERLINE 关键字

# 第五章 ICF 文件的键字

这章给出以下内容的说明：定义

位置项

键字项

位置项（1~44 列）：给出写在 DDS1~44 列的数据格式和实例。

键字项（45~80 列）：给出规定 DDS 键字的规则和实例。键字是按字母顺序给出的。

有关 ICF 文件键字的详细信息，请参看 ICF 程序设计手册。

## 5.1 用 DDS 定义 ICF 文件

按下列顺序定义 ICF 文件：

1.文件层项（任选）

2.记录层项

3.字段层项（任选）

对文件中的每个记录格式都要重复记录层项和字段层项。

在文件中至少要指定一个记录格式。

在 ICF 文件中最多有 1024 个记录格式，在一个记录格式中最多有 32767 个字段。

注：文件名是通过生成 ICF 文件命令（CRTICFF）来指定的，而不是通过 DDS。

在第一章，可以找到对文件层、记录层、字段层的解释和 DDS 键字的语法规则。在附录 B “实例” 中，也可以找到一个完整的 ICF 的例子。

图 5-1 是一个 ICF 文件编码的例子。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A* ICF FILE CODING EXAMPLE  
00020A*  
00030A      R RCD1          RCVENDGRP(14)  
00040A      FLDA           5  
00050A      FLDB           5  0  
00060A      FLDC           10  2  
00070A  
00080A      R RCD2  
00090A  72 73  
00110AON74          FAIL  
00120A      FLDD           12  
A          FLDC           R          REFFLD(A LIB1/FILEA)  
A
```

图 5-1 ICF 文件编码实例

## 5.2 位置项（1~44 列）

这一部分描述了如何规定 ICF 文件的 DDS 格式的前 44 列。格式的另部分编码可参看键字项（45~80 列）。

### 5.2.1 顺序号 (1~5 列)

利用这几列来为格式的每一行指定一个编号。顺号是可选的并且只用于文本编制。

### 5.2.2 格式类型 (第 6 列)

在这一列上指定一个 A, 表示这是 DDS 格式。格式类型是可选的并且仅用于文本编制。

### 5.2.3 注释 (第 7 列)

在这一列中指定一个星号 (\*) 来表示这一行是注释行。注释行可以出现在 DDS 中的任何地方并且它只保存在源文件中。他们出现在源计算机的打印输出结果上而不是出现在扩展的源计算机打印输出上。

可利用 8~80 列作为注释的正文，一个空行（即在 7~80 列上没有字符）也认为是注释。

### 5.2.4 条件 (7~16 列)

7~16 列是指定可选指示器的多字段域，可选指示器是从 01 到 99 的 2 个数字，程序通过调协可选指示器的开 (十六进制 F1) 或关 (十六进制 F2) 来选择输出操作的键字。在 ICF 文件中，可选指示器仅记录层和文件层键字有效。

一个条件是 2~9 个指示器的“与”组合，选择键字前，这些可选指示器必须全部有效（字段 N 为“关”；没有指定 N 则为“开”）。对每个条件最多可以指定 9 个指示器，而每个键字最多可以指定 9 个条件。也就是说，每个键字可以有 81 个指示器。当一个条件需要多个指示器时，它们之间是 AND 条件。在满足条件之前，指示器要设为 ON 或 OFF。第一个要 AND 第二个，再 AND 第三个，……以此类推，所有指示器必须在条件满足和选择键字之前全部有效。同一行规定一组指示器，可以为一个键字指定几个条件，这样当满足其中的任意一条时，键字就被选择。这就称为 OR (或) 关系。在“或”关系中，如果满足第一个、OR 第二个，OR 第三个条件，……等等，键字就能被选择。在“或”关系中的条件可以仅由一个指示器构成，或者由几个指示器的“与”构成。指示器能够通过“与”运算形成一个条件。条件可“或”在一起给程序提供几种选择键字的方式。输入下列的值来指定条件：

第 7 列 (AND “与”): 如果需要三个以上的指示器来形成一个“与”条件，那么在下一行或下几行上指定指示器。可以在第二行或随后的几行上的第 7 列中指定一个 A 来继续“与”条件。或者第 7 位为空，因为 A 是缺省项。

第 7 列 (OR “或”): 如果指定了几个“或”在一起的条件，那么每个条件都必须在新的一行开始，并且除第一个条件外，其它条件在第 7 列上一律为“O”。如第一个条件指定 O 会产生一条错误信息，并且那一位被假定为空格。

第 8、11、14 位 (NOT “非”): 如果要一个指示器为 OFF 满足某个条件的话，就在指示器所在的前位上指定一个 N。

#### 5.2.4.1 多个键字的条件设立

如果要为一个或多个键字设立条件，那么最后一个（或唯一）指示器必须和键字在同一行上。如果键字的条件在多行上，则必须用指示器为键字设条件用于所有键字。详细的 DDS 语法规则，请参见第一章“简介”。

### 5.2.5 名字或规范的类型 (第 17 列)

在这个位置中输入一个值来标识在 19~28 列中名字的类型。ICF 文件的有效项是：

项	意义
R	记录格式名
空格	字段名

图 5-1 给出了如何为编码名字类型。详细信息，请参看 5.2.7 “名字” (19~28 列)。

### 5.2.6 保留 (第 18 列)

任何文件类型都不用这一列。把这一列留作空格，除非用作注释。

### 5.2.7 名字 (19~29 列)

用这几列来指定记录格式名和字段名。

当在 DDS 中指定记录或字段名时，参照 1.2 “语法规则”。

名字必须从 19 列开始。

#### 5.2.7.1 记录格式名

在第 17 列中指定一个 R 时，在 19~28 列中指定的名就是一个记录格式名。可以为一个 ICF 文件指定多个记录格式，但每一个记录格式名在该文件中都必须是唯一的。

#### 5.2.7.2 字段名

在第 17 列为一个空格时，在 19~28 中指定的名是一个字段名。字段名在记录格式内必须是唯一的。对 ICF 文件来说，在 DDS 中指定的字段名的顺序是字段在输入和输出缓冲器中的顺序。

当关键字 CANCEL、EOS、FAIL 和 RQSWRT 用在有字段的记录中时，它们必须有可选指示器。当上述关键字中的任何一个发生作用时，字段在运行时被忽略。在生成时，如果上述关键字没有指示器且用于有字段的记录时，则会产生一个严重的错误，使文件无法生成。

### 5.2.8 引用 (第 29 列)

在该列上指定 R 以使用 OS/400 程序的引用功能来复制一个以前定义的命名字段（叫做被引用字段）到正在定义的字段上。被引用字段可以在正定义的 ICF 文件定义，也可以是以前生成的数据库中定义（被引用的数据库文件要用 REF 或 REFFLD 键字来指定）。引用的字段属性包括长度、数据类型和小数位，以及 ALIAS、FLTPCN 和 TEXT 键字。

如果没有指定 R，那么就不能使用字段的引用功能，就必须为该字段定义属性。

在文件层和记录层上第 29 列必须是空格。

被引用字段的名字可与你正定义的相同，也可不同。如果被引用字段的名与正定义的字段相同，那么只需在第 29 列上指定 R（要在 19~28 列指定要定义的字段名）。如果被引用字段的名字与正定义的不同，那么必须用 REFFLD（引用字段）来指定被引用字段的名字。

可以指定被引用字段的文件名作为 REF 或 REFFLD 键字的参数值。参看本章的后半部及附录 A “何时指定 REF 和 REFFLD” 中有关 REF 和 REFFLD 键字的描述。

也不必把以前描述的字段属性全都复制到正在定义的字段当中。为替换被引用字段的某些属性，须在正定义的字段中指定这些属性。比如指定了正在定义的字段长度，则被引用字段的长度就不会被拷贝过来。

当指定替换字符型数据类型时（在第 35 列为 A），被引用字段的小数位值就不会被复制。

注：一旦建立了 ICF 文件，可以在不影响字段描述的情况下删除或更改被引用文件。

删除和重建 ICF 文件来体现在被引用文件中的修改。

### 5.2.9 长度 (30~34 列)

在这里为每个字段指定字段长度（除非你从一个被引用字段中拷贝字段的属性）。为一个数值型字段指定数字的个数或为一个字符型字段指定字符的个数。长度必须是右对齐的；前导 0 是任选的，ICF 文件的有效长度规定如下：

数据类型	有效长度
字符型	1~32767
二进制	1~9
区位十进制	1~31
压缩十进制	1~31
单精度浮点数	1~9
双精度浮点数	1~17

可以为单精度浮点数指定的最大值为 9 位数字，双精度为 17 位数字。但 OS/400 程序为单精度提供了 7 位数字的浮点精度，为双精度提供了 15 位数字长的浮点精度。在 ICF 文件中一个记录中的所有字段所占用的字节数的总和一定不能超过 32767。系统确定的实际占用的字节数如下：

数据类型	在存储器中占用的字节数
字符型	字符个数
二进制 1~4 位数字	2 字节
二进制 5~9 位数字	4 字节
区位十进制数	数字个数
压缩十进制数	(数字个数/2)+1 (不计小数)
浮点型 (单精度)	4 字节
浮点型 (双精度)	8 字节

如果使用一个被引用字段，那么可以通过指定一个新值或指定长度的增量或减量来替换长度，要增加长度，可指定 +n，其中 n 为增量；要减少长度，可指定-n，其中 n 为减量。例如，数值型字段，+4 表明了它将比被引用字段长出 4 个数字。

注：高级语言对长度有特定的限制，要遵守这些限制。

### 5.2.10 数据类型 (第 35 列)

用这列来指定文件中字段的数据类型。ICF 文件中有效的数据类型如下：

项	意义
P	压缩十进制数
S	区位十进制数
B	二进制数
F	浮点型
A	字符型

如果没有为字段指定一种数据类型，也没有从被引用字段中复制，那么 DDS 就依据小数位中 (36, 37 列) 的值来确定一个默认值。如果小数位是空的，那么就为字符类型 (A)。如果小数位包含一个从 0~31 范围内的数，那么就为区位十进制数 (S)。

注：在第 35 列指定 F 则产生一个单精度浮点字段。使用 FLTPCN 键字来指定双精度或者改变浮点字段的精度。

### 5.2.11 小数位 (36, 37 列)

用这两列为压缩十进制、区位十进制、浮点或二进制字段指定小数位。指定一个从 0~31 的十进制数来表明有几位小数。(这个数不能大于字段的长度)。

如果使用被引用字段，那么可以替换或修改这几位，可以用一个新值来替换这几位，也可以指定要字段增长或减少的数量，在它前面用“+”或“-”来修改。例如+4 表示比被引用字段小数点位多 4 位。如果小数位的结果值超过了允许的最大值，那么就会得到一条错误信息。

注：高级语言中有特定的长度限制。要遵守那些语言的规则。

### 5.2.12 用法（第 38 列）

对这一位的有效值为：

项	意义
B 或空格	输入/输出字段
P	程序到系统字段

程序到系统字段用来进行应用程序和发送系统间的通讯（应用程序的本地系统）。它不能作为数据记录的一部分跨线发送到接收系统。

下列规则适用于程序到系统字段：

字段必须是一个命名的，数值的或字母数字的仅输出字段。

在记录格式中，程序到系统字段必须在所有的数据字段（用法为 B 或空格的字段）之后定义。

一个字段不能既作数据字段又作程序到系统字段来定义，字段名必须是唯一的。

程序到系统字段可以在 EVOKE、SECURITY、TIMER 或 VARLEN 键字上命名。

程序到系统字段有效的键字是 ALIAS、FLTPCN、REFFLD 和 TEXT。

### 5.2.13 位置（39~44 列）

这几列 ICF 文件不用，如果不把这几列作为注释，空出这几列。

## 5.3 键字项（45~80 位）

这一部分包括定义 ICF 文件有效的键字项，它们位于 45~80 列（功能）。参看 1.2 “语法规则”。

下列键字对 ICF 文件是有效的：

ALIAS	FRCDTA	RCVTKCMT
ALWWRT	INDARA	RCVTRNRND
CANCEL	INDEXT	RECID
CNLINVITE	INVITE	REF
CONFIRM	NEGRSP	REFFLD
CTLDTA	PRPCMT	RQSWRT
DETACH	RCVCANCEL	RSPCONFIRM
DFREVOKE	RCVCONFIRM	SECURITY
ENDGRP	RCVCTLDTA	SUBDEV
EOS	RCVDETACH	SYNLVL
EVOKE	RCVENDGRP	TEXT
FAIL	RCVFAIL	TIMER
FLTPCN	RCVFMH	TNSSYNLVL

FMH	RCVNEGRSP	VARBUFMGT
FMTNAME	RCVROLLB	VARLEN

### 5.3.1 ALIAS (替换名)

这是字段层键来为字段指定替换名字。当程序编译时，替换名代替 DDS 字段名送入程序中。正在使用中的高级语言编译器决定是否使用该替换名称。参阅相应的高级语言手册，了解有关该语言使用 ALIAS 的情况。

关键字格式为：

ALIAS (替换名)

参看 1.2 中“语法规则”可得到有关 ALIAS 命名的约定。

替换名必须不同于所有其它的替换名和记录格式中的 DDS 字段名。如果发现重复名称，那么在字段名或替换名上就会发生错误。替换名不能用在 DDS 内或任何其它的 OS/400 功能中（例如，作一个关键字字段名，做 REFFLD 键字指定的字段名，或用在拷贝文件（CPYF）命令中的字段名）。当引用一个有 ALIAS 键字的字段时，除非 ALIAS 键字在引用字段上明显地指定，那么就拷贝 ALIAS 键字。

可选指示器对此关键字无效。

图 5-2 显示了如何指定 ALIAS 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00070A      FIELD A      25A      ALIAS(CUSTOMERNAME)
A
```

图 5-2 指定 ALIAS 键字

### 5.3.2 ALWWRT (允许写)

使用此文件层或记录层关键字以便于程序能够指明何时完成的发送。

该关键字没有参数。

当 DETACH、EOS、RSPCONFIRM 或 RQSWRT 起作用时，在运行期间，忽略 ALWWRT。如果在记录中有 ALWWRT 时，那么这些关键字必须要有可选指示器。如果没有可选指示器的关键字 DETACH、EOS、RSPCONFIRM 或 RQSWRT 用在有关键字 ALWWRT 的记录中，则会出现错误信息，且在建立期间忽略 ALWWRT 键字，不能与 TIMER 键字一起指定 ALWWRT。

关键字 ALWWRT 可在文件层上被指定一次或在记录格式层指定一次。

可选指示器对此关键字有效，当在文件层上指定该关键字时，应指定一个可选指示器。

图 5-3 显示了如何指定 ALWWRT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
01000A 21
02000A      R CUSMST
A
```

图 5-3 指定 ALWWRT 键字

### 5.3.3 CANCEL (取消)

用该文件层或记录层关键字来取消当前送往远程程序的数据链（记录组）。

该键字没有参数。

当 CANCEL 用于下列键字所在的记录时，它必须有一个可选指示器：

CNLINVITE

EVOKE

RQSWRT

RSPCONFIRM

VARBUFMGT

VARLEN

当 CANCEL 有效时，在运行期间内可忽略数据字段和这些键字。如果一个没有可选指示器的 CANCEL 键字用于这些键字中的任意一个所在的记录中，那么就会出现一条错误信息，并且在生成期间忽略这些键字。如果一个没有可选指示器的 CANCEL 键字用在一个有数据字段的记录时，那么就出现一个严重的错误且文件也不能生成。

运行期间为 EOS、FAIL 或者 NEGRSP 起作用时，CANCEL 键字会被忽略。EOS、FAIL 及 NEGRSP 等键字用于有 CANCEL 键字的记录时，必须要有可选指示器。如果没有可选指示器的键字 EOS、FAIL 及 NEGRSP 用在有键字 CANCEL 的记录时，则会出现一条错误信息，在生成期间 CANCEL 将被忽略。

不能与 TIMER 键字一起来指定 CANCEL。

可选指示器对此键字有效。

图 5-4 显示了如何指定 CNACEL 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A 02          CANCEL  
A      R RCD1  
A
```

图 5-4 指定 CANCEL 键字

#### 5.3.4 CNLINVITE (取消请求)

用此文件层或记录层键字来取消那些没有接收到输入的有效请求操作。

该键字没有参数。

CNLINVITE 键字用在有 RQSWRT、RSPCONFIRM 及 EVOKE 键字的记录时，它必须有一个可选指示器。在运行期间，如果 CNLINVITE 有效，则这些键字被忽略。如果没有可选指示器的 CNLINVITE 键字用于有上述键字的记录，则会出现错误信息，这些键字将在建立时被忽略。

运行期间如果 CANCEL、EOS、FAIL 或 NEGRSP 有效，那么忽略 CNLINVITE 键字。这些键字用在有 CNLINVITE 键字的记录时必须要有可选指示器。无可选指示器的 CANCEL、EOS、FAIL 或 NEGRSP 键字应用在有 CNLINVITE 键字的记录上时，将会出现错误信息，在生成期间忽略 CNLINVITE 键字。

不能与 TIMER 键字一起指定 CNLINVITE。

可选指示器对此键字有效。

图 5-5 显示了如何指定 CNLINVITE 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RCD1          CNLINVITE
```

A

图 5-5 指定 CNLINVITE 键字

### 5.3.5 CONFIRM (确认)

用此文件层或记录层键字来请求远程的程序确认收到数据。

该键字无参数。

CONFIRM 键字只当传送建立在确认的同步水平 (SYNLVL(\*CONFIRM)键字) 时才有效。如果传送建立在无同步水平 (SYNLVL(\*NONE)键字) 上，则 CONFIRM 键字将被拒绝，并出现 OS/400 错误信息。

运行时，如果 EOS、RSPCONFIRM、RQSWRT 键字有效，忽略 CONFIRM 键字，这些键字用于有 CONFIRM 键字的记录时必须要有可选指示器。如果 EOS、RSPCONFIRM、RQSWRT 键字用于有 CONFIRM 键字的记录而不带可选指示器，那么将出现错误信息。在建立时忽略 CONFIRM 键字。

不能与 TIMER 键字一起指定 CONFIRM。

CONFIRM 键字可以在文件层指定一次或者在记录格式层指定一次。

可选指示器用此键字有效。

图 5-6 显示如何指定 CONFIRM 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R RCD  
00020A 01          CONFIRM  
A
```

图 5-6 指定 CONFIRM 键字

如果可选指示器 01 为 ON，那么远程程序就通过发一个或正或负的应答来确认收到了数据。

### 5.3.6 CTLDTA (控制数据)

利用此文件层或记录层键字来通知远程的程序控制数据已经发出。

此键字没有参数。

在运行期间，如果 EOS、RSPCONFIRM、RQTWRT 有效，则忽略 CTLDTA 键字。这些键字在用于有 CTLDTA 键字的记录时必须有可选指示器。如果 EOS、RSPCONFIRM、RQTWRT 用于有 CTLDTA 键字的记录而不带可选指示器，则会出现错误信息且在生成时忽略 CTLDTA 键字。

不能与 TIMER 键字一起来指定 CTLDTA。

可选指示器对此键字有效。

图 5-7 显示了如何在记录层指定 CTLDTA 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A      R SNDCTL  
A          CTLDTA  
A      USRSCLD    100A
```

图 5-7 指定 CTLDTA 键字

### 5.3.7 DETACH (脱离)

使用此文件层或记录层键字来明确地通知远程程序，本地程序已发送数据并想结束传送。

该键字没有参数。

当 DETACH 键字用到有下列键字的记录时，它必须要有可选指示器：

ALWWRT

ENDGRP

FMH

FRCDTA

INVITE

SUBDEV

在运行期间，当 DETACH 键字有效时，忽略上述这些键字。一个没有可选指示器的 DETACH 键字用于有这些键字的任一个所在的记录中，就会出现一条错误信息并且在生成时忽略这些键字。

在运行期间，当 EOS、RSPCONFIRM 或 RQSWRT 键字有效时，忽略 DETACH 键字。当这些键字用于一个有 DETACH 键字的记录时，它们必须要有可选指示器。一个没有可选指示器的 EOS、RSPCONFIRM 或 RQSWRT 键字用于有 DETACH 的记录中，则会出现一条错误信息且在生成期间忽略 DETACH 键字。

不能与 TIMER 键字一起来指定 DETACH。

DETACH 键字最多能在文件层上被指定一次或在每个记录格式中指定一次。

可选指示器对此键字有效。

图 5-8 显示了如何指定 DETACH 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          R RCD
00020A 01           DETACH
A
```

图 5-8 指定 DETACH 键字。

如果可选指示器 01 为 ON，那么结束你的程序与远程程序间的传送。

### 5.3.8 DFREVOKE (延迟激活)

用这个文件层或记录层键字与 EVOKE 键字一起来延迟一次激活请求，直到数据装满发送缓冲区为止或收到 FRCDTA 键字为止。DFREVOKE 键字只对那些与 EVOKE 同时发送数据的特殊应用有用处。

此键字无参数。

不能与 TIMER 一起指定 DFREVOKE 键字。

可选指示器对此键字有效。

图 5-9 显示了如何在记录层上指定 DFREVOKE 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R PGMSTART
A
A          EVOKE (&LIB/&PGMID)
A          DFREVOKE
A
A

```

图 5-9 指定 DFREVOKE 键字

### 5.3.9 ENDGRP (分组结束)

使用此文件层或记录层关键字使你的程序能够指明一个用户定义的记录分组结束。

此关键字无参数。

在运行期间，当 DETACH、EOS、RSPCONFIRM 或 RQSURT 有效时，忽略 ENDGRP 键字。当这些关键字用于 ENDGRP 键字所在的记录中时，它们必须有可选指示器。如果没可选指示器的 DETACH、EOS、RSPCONFIRM 或 RQSWRT 键字用于有 ENDGRP 键字的记录时，则会出现一条错误信息且在生成时忽略 ENDGRP 键字。

不能与 TIMER 键字一起指定 ENDGRP。

可选指示器对此关键字有效。(当在文件层上指定该关键字时，应该指定一个可选指示器)。

图 5-10 显示了如何指定 ENDGRP 键字

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00030A      R RECORD1      ENDGRP
A

```

图 5-10 指定 ENDGRP 键字

### 5.3.10 EOS (结束会话)

用此文件层或记录层关键字指定一个会话功能的结束。为结束一次会话，你的程序要发出一条 EOS 有效的写操作。

此关键字无参数。

当 EOS 键字用于有下列关键字的记录中时必须要有可选指示器：

ALWWRT	EOKE	NEGRSP
CANCEL	FAIL	RQSWRT
CNLINVITE	FMH	RSPCONFIRM
CONFIRM	FMTNAME	SUBDEV
DETACH	FRCDTA	VARBUFMGT
ENDGRP	INVITE	VARLEN

在运行期间，当 EOS 键字有效时忽略数据字段和上述这些关键字。如果没有可选指示器的 EOS 键字用于有这些关键字的记录中，则会出现错误信息且在生成期间忽略这些关键字。如果没有可选指示器的 EOS 键字应用到有数据字段的记录中，那么会出现一条严重的错误并且不能生成文件。

不能与 TIMER 键字一起指定 EOS。

可选指示器对此关键字是有效的。当在文件层指定此关键字时，应该指定一个可选指示器。

图 5-11 显示了如何指定 EOS 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A 01           EOS  
A      R RCD  
A
```

图 5-11 指定 EOS 键字

如果指示器 01 为 ON 并且程序没有进行输出操作，那么就结束会话。

### 5.3.11 EVOKE (激活)

用此文件层或记录层键字来启动一个远程系统上的程序。

键字的格式为：

EVOKE([库–ele/]程序名[参数-1…[参数-255]])

程序名可以是下列之一：

程序名：

这是要启动的远程系统上的程序名。该名是在生成时检测为有效的 AS/400 目标名。

‘字符串 1’

这是要启动的远程系统上的程序名。因为不能对字符串进行语法检查，所以指定的名称必须是远程系统可接受的格式。

&字段名 1

指定的字段包括将在远程系统中启动的程序名。字段名必须是一个已经在记录格式中定义的字段而且必须是字符字段（数据类型为 A）。指定的名称必须是远程系统可接受的格式。

可选的库名可以是下列之一：

库名/

这是在远程系统中要启动的程序所在的库名。该名称在生成时是一个有效的 AS/400 目标名。对该键字来说，\*CURLIB 和\*LIBL 是无效名称。如果指定了二者之一，那么要用引号括起。

‘字符串 2’/

这是要在远程系统上启动的程序所在的库名。指定的名称必须是远程系统可接受的格式。这是因为对字符串不做语法检查。

&字段名 2/

指定的字段中包含了将在远程系统中启动的程序所在的库名。字段名必须是一个已在记录格式中定义了的有效字段，而且必须是一个字符型字段（数据类型为 A）。必须是远程系统可接受的格式。

注：如果远程系统是一个 AS/400 系统且没有指定库名，那么用库列表来查找程序。

参数-1……参数-255 可以是下列之一：

‘字符串 3’

这是一个要传送到远程系统程序的字符串。因为字符串不能进行语法检查，所以它必须是远程系统可接受的格式。

[&]字段名 3

这个字段中包含了要传送到远程系统程序数据的字段名。字段名必须是在记录格式中已定义了的有效字段。

### 数值 3

这是一个要传送到远程系统的程序中去的数值。该数值可以是正，可以是负（有符号或无符号）。小数点可以是“.”，也可以是“,”，可任选。不做小数点对齐，不压缩数字的前导零。数据将作为一个区位十进制数值发送出去。下面都是有效的数值：

999.6  
-999.6  
01587

#### 5.3.11.1 特殊考虑

下面是有关使用 EVOKE 键字时的一些特殊的考虑：

当在文件层指定 EVOKE 键字时，不能指定字段名作为参数值。

组合程序名和库名允许的最大长度为 64。程序名和库名间的斜杠包括在内。斜杠要放在字母中间，否则 APPC 就不发送斜杠（如，LIBRARY/PROGRAM），参数 1~参数 255 的总长度不能超过 32767 个字节。

注：为 APPC 计算 PIP 数据的最大长度时，必须考虑下面的问题：

每个参数的长度都必须多加 4 个字节。即指定一个参数，就要另外再加上 4 个字节。这些字节是系统所要求的。

可以使用下面的公式来计算参数的总长度：

4+(第一个参数的长度+4)+(第二个参数的长度+4)+…(第 n 个参数的长度+4)

下面是使用该公式的例子：

```
EVOKE(LIBRARY1/PROGRAM1 ' THIS IS AN EXAMPLE OF A  
CHARACTER STRING' &FIELD1 35)
```

假设&FIELD1 的长度为 10，

4+(40+4)+(10+4)+(2+4)=68

每个参数（参数 1~参数 255）的长度应该与在远程程序中的对应参数的长度一致。

如果一个用法为 P 的字段名定义为 EVOKE 键字的一个参数，那么这个字段就不能作为数据记录的一部分而被发送出去。

一个在 AS/400 系统上被激活的程序将会接收到由远程程序发送的参数，就象用 CL CALL 命令传送参数一样。

注：如果 AS/400 系统上的作业是一个预启动作业，那么程序就必须使用 RTVDTAARA 命令来接收参数。

当指定 SECURITY 或 SYNLVL 时，要求使用该键字。在运行期间，只有当 EVOKE 有效时才可以使用 SECURITY 和 SYNLVL 键字。

在运行期间，当 CANCEL、CNLINVITE、EOS、FAIL、NEGRSP、RSPCONFIRM 或 RQSWRT 有效时，忽略 EVOKE 键字。当这些键字用于有 EVOKE 键字的记录中时，它们要有可选指示器。如果没有可选指示器的 CANCEL、CNLINVITE、EOS、FAIL、NEGRSP 或 RQSWRT 键字用于有 EVOKE 键字的记录时，就会出现一条错误信息且在生成期间忽略 EVOKE 键字。

不能与 TIMER 键字一起来指定 EVOKE。

如果对记录格式或文件多次指定该键字，那么可选指示器有效而且是必需的。

图 5-12 显示了如何指定 EVOKE 键字。

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

00010A            R RCD

00020A	01	:	EVOKE (LIBRARY1/PROGRAM1)	1
00030A	02	:	EVOKE (LIBRARY2/PROGRAM2)	1
	A	:	:	
	A	:	:	
00090A	R RCD2		EVOKE (&FIELD2/&FIELD1 'ABC' 10.1 +	
00100A			FIELD3)	2
00110A	FIELD1	10A P		
00120A	FIELD2	10A P		
00130A	FIELD3	5B P		
	A			

图 5-12 指定 EVOKE 键字

1.如果指示器 01 为 ON, 启动 LIBRARY1 中的 PROGRAM1。如果指示器 02 为 ON, 那么启动 LIBRARY2 中的 PROGRAM2。

2.&FIELD1 中有被启动的程序名。&FIELD2 中有库名。字符串 ABC、数字 10.1 和在 FIELD3 中的值都被传送给远程系统上的程序中。

### 5.3.12 FAIL (失败)

用此文件层或记录层键字来通知远程程序发送或接收的数据是无效的。

此键字没有参数。

当 FAIL 键字用于有数据字段(用途为 B 或空格)的记录或用下列键字所在的记录中时, 它必须要有可选指示器:

CANCEL	RQSWRT
CNLINVITE	RSPCONFIRM
EVOKE	VARBUFMGT
NEGRSP	VARLEN

在运行期间, 当 FAIL 键字有效时忽略数据字段和上述这些键字。如果一个没有可选指示器的 FAIL 键字用于有这些键字之一的记录中时, 将会出现错误信息并且在生成期间忽略这些键字。如果没有可选指示器的 FAIL 键字用于一个有数据字段的记录中时, 就会出现一条严重的错误而且文件也不能生成。

在运行期间, 当 EOS 键字有效时忽略 FAIL 键字。当 EOS 用于有 FAIL 键字的记录中时, 它必须要有可选指示器。如果没有可选指示器的 EOS 键字用于有 FAIL 键字的记录中, 那么就会出现一条错误信息且在生成期间忽略 FAIL 键字。

FAIL 不能与 TIMER 键字一起来指定。

可选指示器对该键字是有效的。在文件层上指定该键字时, 应该指定一个可选指示器。

图 5-13 显示了如何指定 FAIL 键字。

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
00010A	R INQ	
00020A	99	FAIL
	A	

图 5-13 指定 FAIL 键字

### 5.3.13 FLTPCN (浮点精度)

用此字段层键字来指定一个浮点字段的精度。

键字格式为：

FLTPCN(\*SINGLE | \*DOUBLE)

此处的\*SINGLE 参数是指单精度，\*DOUBLE 是指双精度。该键字仅对浮点字段（数据类型 F）有效。

单精度字段最多可为 9 位数字，双精度字段为 17 位数字。如果指定的字段长度超位 9 位（单精度）或 17 位（双精度），那么就会出现一条错误信息而且文件也不能生成。ICF 为单精度提供了 7 位数字的浮点精度，为双精度提供了 15 位数字的浮点精度。

可选指示器对此键字无效。

图 5-14 显示了如何指定 FLTPCN 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00090A          FIELD A           17F 4          FLTPCN (*DOUBLE)  
A
```

图 5-14 指定 FLTPCN 键字

### 5.3.14 FMH (功能管理首标)

用此文件层或记录层键字通知远程程序正在发送功能管理首标 (FMH)。

这个键字没有参数。

在运行期间，当 EOS、DETACH、RSPCONFIRM、RQSWRT 有效时，忽略 FMH，当这些键字用于有 FMH 的记录中时，必须有可选指示器。如果不是，则有错误信息，且在生成时忽略 FMH。

不能用与 TIMER 键字一起指定 FMH。

可选指示器对此键字有效。

图 5-15 显示了如何指定 FMH 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RCD          FMH  
A          FLD1          10A  B  
A
```

图 5-15 指定 FMH 键字

### 5.3.15 FMTNAME (格式名)

用此文件层或记录层键字来表明当你的程序发出一个输出操作时，发送至远程程序上的记录格式名。

此键字没有参数。

在运行期间，当 EOS、RSPCONFIRM 或 RQSWRT 有效时忽略 FMTNAME 键字。当这些键字用于有 FMTNAME 键字的记录中时它们必须要有可选指示器。如果没有可选指示器的 EOS、RSPCONFIRM 或 RQSWRT 用于 FMTNAME 所在的记录中，那么就会出现一条错误信息并且在建立期间忽略 FMTNAME 键字。

不能与 TIMER 键字一起指定 FMTNAME。

可选指示器对此键字有效。

图 5-16 显示了如何指定 FMTNAME 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RCD1  
A 01                      FMTNAME  
A          FIELD1      10A  B  
A
```

图 5-16 指定 FMTNAME 键字

如果指示器 01 为 ON 且程序做写操作，记录格式名作为 APPC 映象名发送到远程系统上。

### 5.3.16 FRCDTA (强制数据)

当不再发送数据时，用此记录层键字来清除缓冲区而不用等到缓冲区满。

注：如果在每条写语句后都指定此键字，则可能出现性能问题。

不用等待确认。CONFIRM 键字提供了类似的功能，但它另外还提供数据发送的确认。在继续下一条程序语句之前，程序必须等待其它结束的响应)。

此键字没有参数。

在运行期间，当下列键字之一有效时忽略 FRCDTA 键字：

```
DETACH  
EOS  
RQSWRT  
RSPCONFIRM
```

当上述这些键字用到指定了 FRCDTA 的记录中时，它们必须要有可选指示器。如果这些键字没有可选指示器且用到有 FRCDTA 的记录中时，就会出现一条错误信息而且在生成期间忽略 FRCDTA 键字。

不能与 TIMER 键字一起指定 FRCDTA。

FRCDTA 至多在每个记录格式上指定一次。

可选指示器对此键字有效。

图 5-17 显示了如何指定 FRCDTA 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R REC1  
00020A  10                      FRCDTA  
00030A      FLD1      10  
00040A      FLD2      5  
A
```

图 5-17 指定 FRCDTA 键字

当可选指示器 10 为 ON 并且程序执行写操作时，键字 FRCDTA 发送当前缓冲区中的通

信数据。

### 5.3.17 INDARA (指示器区)

用此文件层键字从缓冲区或记录区中清除可选和应答指示器，然后把它们放在一个 99 个字节的另一指示器区内。指定 INDARA 键字提供如下的优点：

当可选和应答指示器都被使用时，可简化 COBOL/400\*的编码过程。如果同一个指示器既作可选指示器又作应答指示器，那么不管它们在 DDS 中的顺序如何，它们总是有相同的值。

用程序描述的的 WORKSTN 文件帮助 RPG/400 程序员。

该键字无参数。

如果指定 INDARA 键字，那么某些高级语言要求在程序中使用指定的指示器区。请查看相应的高级语言手册。

如果指定 INDARA 键字，那么可以在 DDS 中进行加、修改或删除可选和应答指示器的操作，而不必重新编译高级语言程序，只重新编译文件即可，这是因为在缓冲区中的字段位置没有改变并且级别检验数据也没改变。但如果程序中使用了新的指示器，那么就需要修改程序，并重新编译。

对该键字来说可选指示器是无效的。

图 5-18 显示了如何指定 INDARA 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                         INDARA  
00020A 41                        FAIL  
00030A                           RCVTRNRND(14 'Turn around')  
00040A      R RCD  
00050A      ACTNBR      10  
A
```

图 5-18 指定 INDARA 键字

由于指定了 INDARA 键字，可选指示器 41 和应答指示器 14 从 RCD 的缓冲区中清除并放在另一个指示器区中，只有命名字段 ACTNBR 仍保留在记录格式 RCD 的缓冲区中。

### 5.3.18 INDTXT (指示器说明)

用此文件层或记录层键字来描述（指示意图或用途）一个应答或可选指示器。

键字的格式为：

INDTXT (应答或可选指示器'指示器说明')

可以为每个应答和可选指示器指定一次该键字。

指示器说明是一个必需的参数值，并且是一个由撇号括起来的字符串。如果字符串长度超过 50，那么只有前 50 个字符能被高级语言编译程序使用。在编译过程中使用说明做程序文本的说明。

INDTXT 键字不会使指定的指示器出现在输入或输出记录区内。它仅提供与指示器有关的说明。一旦给了一个指示器说明（用该键字或用应答指示器说明），就不再给其它描述。这与其它键字的区别在于其它键字能用指示器作为参数值，对其它键字来说，只有说明可以被忽略。

可选指示器对此键字无效。

图 5-19 显示了如何指定 INDTXT 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A                               INDTXT(02 'Alternate month')  
00020A      R MASTER  
00030A      MTH          2 10  
00040A 02      ALTMTH        2 10  
A
```

图 5-19 指定 INDTXT 键字

INDTXT 键字描述可选指示器 02 的用途，如果在一个高级语言的编译打印输出，那么要打印'Alternate month'作为指示器 02 的注释。

#### 5.3.19 INVITE (请求)

利用此文件层或记录层键字为其后的读操作申请程序设备。为发送一条申请给程序设备，你的程序要发出一条写操作给 INVITE 键字有效的程序设备。如果应用程序正与程序设备进行交互式处理，那么 INVITE 键字就能改进某睦性能。通常，程序发出一条输入操作时，就发送读请求给一个设备。然而，INVITE 键字允许你发出输出操作时来申请读操作。在输出操作完成后，当申请到的程序设备发送数据，OS/400 程序处理接收到的数据时，你的程序可做其它的处理。这将提高程序的性能。当应用程序准备好处理数据时，它就发一个输入操作。

此键字无参数。

在运行期间为 EOS、RSPCONFIRM 或 DETACH 有效时，忽略 INVITE 键字。当这些键字用于有 INVITE 键字的记录中时它们必须要有可选指示器。

如果没有可选指示器的 EOS、RSPCONFIRM 或 DETACH 键字用到有 INVITE 键字的记录中，那么就会发出一条错误信息而且在生成期间忽略 INVITE 键字。

不能与 TIMER 键字一起指定 INVITE。

可选指示器对此键字无效。

既可在文件层上又可在记录层上指定 INVITE 键字。

图 5-20 显示了如何指定 INVITE 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A 01                               INVITE  
00020A      R RCD1  
00030A      FLD1          10  
00040A      FLD2          5  
A
```

图 5-20 指定 INVITE 键字

仅在可选指示器 01 为 ON 时，INVITE 键字才有效。

### 5.3.20 NEGRSP (负应答)

此文件层或记录层键字发送一个负应答给远程程序来表明你的程序在接收的数据中发现了一个错误。

键字格式为：

NEGRSP[(&字段名)]

&字段名是可选参数，它指出包含检测数据的字段名，该检测数据有负应答要发送给远程程序上。指定的字段名必须存在于记录格式中，并且该字段必须是一个长度至少为 8，数据类型为 A，用法为 B 或空格的字符型字段。当 NEGRSP 键字用于下列任意一个键字所在的记录中时，它必须要有一个可选指示器：

CANCEL	RSPCONFIRM
CNLINVITE	VARBUFMGT
EVOKE	VARLEN
RQSWRT	

在运行期间，当 NEGRSP 键字有效时忽略上述这些键字，如果没有可选指示器的 NEGRSP 键字用到这些键字的记录中，那么就会出现错误信息而且在生成期间忽略这些键字。

在运行期间，当 EOS 或 FAIL 有效时忽略 NEGRSP。如果这些键字用于有 NEGRSP 的记录中它们必须要有可选指示器。如果没有可选指示器的 EOS 或 FAIL 键字用到有 NEGRSP 的记录中，那么就会出现一条错误信息且在生成期间忽略 NEGRSP 键字。

在文件层上指定 NEGRSP 时，不能指定字段名参数。

图 5-21 显示了如何指定 NEGRSP 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RCD1  
A 01                      NEGRSP (&FIELDDB)  
A          FIELD A      25A B  
A          FIELD B      80A B  
A
```

图 5-21 指定 NEGRSP 键字

如果指示器 01 为 ON，则对 RCD1 的写操作就发送一个负应答并把 FIELDDB 的前 8 个字节发送到远程程序上。注意 RCD1 只有负应答检测数据，没有其它的数据。

### 5.3.21 PRPCMT (准备落实)

用此记录层键字来要求远程程序准备一个同步指针。一个有 PRPCMT 键字的输出操作，强行发送输出缓冲区中的数据。

此键字无参数。

当此操作尚未完成，程序没有接收到应答时，不能继续下去。远程程序必须执行一个落实或返回操作，或发送一个 FAIL 或 EOS 来表明它是否准备落实它保护的内容。

PRPCMT 只对在 EVOKE 上规定\*COMMIT 的同步层上有效。

能与 PRPCMT 一起指定的键字仅有 VARBUFMGT 和 VARLEN。

可选指示器对此键字有效。

图 5-22 显示了如何指定 PRPCMT 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A           R RCD1
A
A           PRPCMT
A

```

图 5-22 指定 PRPCMT 键字

### 5.3.22 RCVCANCEL (接收取消)

用此文件层或记录层键字把应答指示器设为 ON 来通知你的程序, 表示远程程序已经发出了一个 CANCEL。

键字的格式为:

RCVCANCEL (应答指示器[‘说明’])

应答指示器参数是必需的。

可选的‘说明’放在计算机编译期间产生的输出结果中, 用来解释指示器的用途。在文件或程序中它没有其它功能只是作为一种注释。要求用撇号括起, 如果在两个撇号间指定的字符超过 50 个, 那么在程序输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVCANCEL。

可选指示器对此键字无效。

图 5-23 显示了如何指定 RCVCANCEL 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A           R RCD1
A           RCVCANCEL(34 'Received - +
A           Cancel')
A

```

图 5-23 指定 RCVCANCEL 键字

当在对 RCD1 的一个输入操作上接收到 CANCEL 时, 指示器 34 被设置为 ON。

### 5.3.23 RCVCONFIRM (接收确认)

如果你的程序接收的数据包含从远程程序中发出的确认申请, 那么就用此文件层或记录层键字把一个应答指示器设为 ON。

键字的格式为:

RCVCONFIRM (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的, 它出现在计算机编译期间产生的输出结果中, 用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号, 如果在两个撇号间字符超过 50 个, 那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVCONFIRM。

可选指示器对此键字无效。

图 5-24 显示了如何指定 RCVCONFIRM 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

```

```

00010A          RCVCONFIRM(44 'Waiting for a +
00020A                      response')
00030A      R RCD
A

```

图 5-24 指定 RCVCONFIRM 键字

应答指示器 44 被设置为 ON 来指出接收到从远程程序的确认请求。

### 5.3.24 RCVCTLDTA (接收控制数据)

用此文件层或记录层键字来设置应答指示器，通知你的程序已接收控制数据。

键字的格式为：

RCVCTLDTA (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的，它出现在计算机编译期间产生的输出结果中，用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个，那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVCTLDTA。

可选指示器对此键字无效。

图 5-25 显示了如何在记录层上指定 RCVCTLDTA 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A          R RCVCTLDTA
A
A          RCVCTLDTA(66 'received control +
A
A          data')
A          USRRCTLTD    100A
A

```

图 5-25 指定 RCVCTLDTA 键字

### 5.3.25 RCVDETACH (接收脱离)

如果远程程序结束了传送，那么就用此文件层或记录层键字把应答指示器设为 ON。

键字的格式为：

RCVDETACH (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的，它出现在计算机编译期间产生的输出结果中，用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个，那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVDETACH。

可选指示器对此键字无效。

图 5-26 显示了如何指定 RCVDETACH 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A          RCVDETACH(44 'Transaction is +

```

```

00020A                         finished')
00030A      R RCD
    A

```

图 5-26 指定 RCVDETACH 键字

当远程程序结束传送时应答指示器 44 为 ON。

### 5.3.26 RCVENDGRP (接收结束分组)

用此文件层或记录层键字来打开应答指示器,通知你的程序结束一个用户定义的记录分组。

键字的格式为:

RCVENDGRP (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的,它出现在计算机编译期间产生的输出结果中,用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个,那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVENDGRP。

可选指示器对此键字无效。

图 5-27 显示了如何指定 RCVENDGRP 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00100A      R CUSMST
00200A
    A
        RCVENDGRP(68 'End of group received-
    ') 
    A

```

图 5-27 指定 RCVENDGRP 键字

当远程程序指出一个用户定义的记录分组结束时,应答指示器 66 被设为 ON。

### 5.3.27 RCVFAIL (接收失败)

当本地程序确定远程程序已发送了一个 FAIL 时,用此文件层或记录层键字打开一个应答指示器。当没指定 RCVFAIL 键字时,如果发生了这种情况,那么 OS/400 信息就会通知本地程序,告之远程程序已发送了一个 FAIL。

键字的格式为:

RCVFAIL (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的,它出现在计算机编译期间产生的输出结果中,用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个,那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVFAIL。

可选指示器对此键字无效。

图 5-28 显示了如何指定 RCVFAIL 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A                               RCVFAIL(10 'Fail received')
00020A
00030A          R RCD
A

```

图 5-28 指定 RCVFAIL 键字

当远程程序发送一个失败标志时指示器 10 为 ON。

### 5.3.28 RCVFMH (接收功能管理首标)

用此文件层或记录层键字启动一个应答指示器来通知你的程序已接收了功能管理首标。键字的格式为：

RCVFMH (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的，它出现在计算机编译期间产生的输出结果中，用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个，那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVFMH。

可选指示器对此键字无效。

图 5-29 显示了如何指定 RCVFMH 键字。

```

|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               RCVFMH(24 'Received FMH')
A          R RCD1
A

```

图 5-29 指定 RCVFMH 键字

当接收到一个功能管理首标时指示器 24 为 ON。

### 5.3.29 RCVNEGRSP (接收负应答)

用此文件层或记录层键字启动一个应答指示器，通知你的程序远程程序已发送了一个负应答。

键字的格式为：

RCVNEGRSP (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的，它出现在计算机编译期间产生的输出结果中，用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个，那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVNEGRSP。

可选指示器对此键字无效。

图 5-30 显示了如何指定 RCVNEGRSP 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               RCVNEGRSP(67 'Negative re-
A                               sponse')
A           R RCD1
A

```

图 5-30 指定 RCVNEGRSP 键字

当接收一个负应答时指示器 67 为 ON。

### 5.3.30 RCVROLLB (接收返回应答指示器)

用此文件层或记录层键字来表明是否已收到了一次返回操作。

键字的格式为:

RCVROLLB (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的，它出现在计算机编译期间产生的输出结果中，用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个，那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVROLLB。

可选指示器对此键字无效。

图 5-31 显示了如何指定 RCVROLLB 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A                               RCVROLLB(67 'Receive RB')
A           R REC1
A

```

图 5-31 指定 RCVROLLB 键字

如果已接收到一个接收返回，那么指示器 67 为 ON。

### 5.3.31 RCVTKCMT (接收使用落实应答指示器)

用此文件层或记录层键字来表明是否已收到了使用落实的请求。

键字的格式为:

RCVTKCMT (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的，它出现在计算机编译期间产生的输出结果中，用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个，那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVTKCMT。

可选指示器对此键字无效。

图 5-32 显示了如何指定 RCVTKCMT 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
A
A
A R REC1
A

```

图 5-32 指定 RCVTKCMT 键字

如果已接收了一个使用落实的请求，指示器 67 为 ON。

### 5.3.32 RCVTRNRND (接收转向)

用此文件层或记录层键字打开一个应答指示器，来通知你的程序，发送程序已停止发送并且将发送权赋予本地程序。

键字的格式为：

**RCVTRNRND** (应答指示器[‘说明’])

应答指示器参数是必需的。

‘说明’是可选的，它出现在计算机编译期间产生的输出结果中，用来解释指示器的用途。它没有其它的功能只是作为一种注释。要求使用撇号。如果在两个撇号间字符超过 50 个，那么在程序的输出结果中就被截断至 50 个字符。

不能与 TIMER 键字一起指定 RCVTRNRND。

可选指示器对此键字无效。

图 5-33 显示了如何指定 RCVTRNRND 键字。

```

|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A RCVTRNRND(44 'Host has stopped +
00020A           sending')
00030A       R CUSMST
A

```

图 5-33 指定 RCVTRNRND 键字

### 5.3.33 RECID (记录标识)

用此记录层键字允许你的程序在用文件名发出一个从申请设备上读的操作时，来标识一个记录到记录格式。当使用一个输入操作时，OS/400 程序把在接收记录中的数据与在参数值中指定的选择值进行比较。在指定位置开始的选择值必须等于指定的比较值。然后程序可确定刚读数据的记录格式。

键字的格式为：

**RECID** (起始位置 比较值)

起始位置参数指定了一个相对于缓冲区中（不处理指示器）数据起始的位置，以此来测试记录的 ID。如果使用了 INDARA 键字，那么数据的起始位和缓冲区位置是相同的。对缓冲区的描述，请参看图 5-36。位置参数可以是：

nnnnn

或 \*POSnnnnn

此处的 nnnnn 是一个 1~5 位长的数值，例如下面是一些等效对：

1 和 \*POS1  
34 和 \*POS34  
12025 和 \*POS12025

比较值参数可以是下列之一：

值	意义
*ZERO	被检测的值是零（十六进制 F0）等效于 0。
*BLANK	被检测的值是空格（十六进制 40）。等效于空格。
‘字符串’	被检测的值是指定的字符串，串长度限制在从指定的

#### RECID

位置参数到文件内最短的非零记录格式结束的长度范围内（不包括记录格式的指示器或程序字段）。

指定了 RECID 键字的记录格式必须至少包括一个数据字段（用法为 B）。

在一个记录格式中可以多次指定 RECID 键字。如果这么做了，那么要把记录中的数据按规定顺序与每个 RECID 相比较，直到找到一个与之相匹配的为止。选择值与数据相匹配的第一个记录格式就是所选的记录格式。如果没找到匹配的或没有接收到用户数据，那么就要使用缺省 RECID 的记录格式。缺省的 RECID 记录格式是在文件中没有为其指定 RECID 键字的第一个记录格式。然而，如果文件中的每一个记录格式都有 RECID 键字，那么缺省的记录格式就是文件中的第一个记录格式。

当接收到数据但还没有找到匹配，而且 RECID 的缺省记录格式有 RECID 键字时，就会产生一条信息。

当把接收的数据同 RECID 键字进行比较时，如果比较的位置超过了接收数据的最后一个字，那么数据就会被置成空格。

在运行时，如果在 ADDICFDEVE、CHGICFDEVE 或 OVRICFDEVE 命令上没有指定 FMTSLT(\*RECID)参数，那么忽略该键字。

不能在 VARBUFMGT 键字相同的记录格式中指定 RECID。

可选指示器对此键字无效。

图 5-34~图 5-37 显示了如何指定 RECID 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R DFTFMT
00020A      ID      3A
00030A      FLD1    20A
00040A      FLD2    5B 0
A
00050A      R RCD1          RECID('ABC')
00060A      ID      3A
00070A      FLD1    10S 0
00080A      FLD2    5B 0
A
00090A      R RCD2          RECID('DEF')
00100A      ID      3A
00110A      FLD1    10S 0
00120A      FLD2    5A
00130A      FLD3    2B 0
```

A

图 5-34 指定 RECID 键字（缺省记录格式）

记录格式 DFTFMT 将是 RECID 缺省的记录格式。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RCD1                      RECID(1 'H')
00020A      ID             1A
00030A      FLD1           10A
00040A      FLD2           10A
00050A      FLD3           6S 2
A
00060A      R RCD2                      RECID(1 'D')
00070A      ID             1A
00080A      FLD1           8S 2
00090A      FLD2           10A
00100A      FLD3           5B 0
A
00110A      R RCD3                      RECID(1 'L')
00120A      ID             1A
00130A      FLD1           50A
A
```

图 5-35 指定 RECID 键字（缺省记录格式）

记录格式 RCD1 将是 RECID 缺省记录格式。如果没找到匹配，那么就会对程序发出逃逸信息。因为 RECID 缺省记录格式有为其指定的 RECID 键字，所以如果没有接收到数据，就将使用记录格式 RCD1。

一个应用程序从 ICF 文件中读标题记录和详细记录。程序向文件名（不是单个的记录名）发出一条输入操作，并按发送它们的顺序接收记录（标题记录和详细记录）。在此例中，发送和接收的应用程序都提供了显式代码（H 代表标题记录，D 代表详细记录），以此来标识记录发送、接收的类型。RECID 键字标识 H 和 D 出现在输入缓冲区的什么位置并指定标识记录类型的值（在指定的位置上起始）。

在图 5-36 中，ICF 文件中定义了三个记录格式。应用程序发出用文件名的输入操作，例如，RPTFILE。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R HEADER                     RECID(1 'H')
00020A                          RCVTRNRND(10 'Host stopped sending')
00030A      CODE            1
00040A      TITLE           30
00050A      ACTNBR          6 0
00060A      R DETAIL                     RECID(1 'D')
```

00070A	CODE	1
00080A	ITMNBR	8 0
00090A	DESCRP	20
00100A	R CATCH	
00110A	FIELD	37
A		

图 5-36 指定 RECID 键字 (RECID 键字的目的)

假设在 9 个成功的输入操作中接收的记录是一个标题记录，而后是三个详细记录，接下来是一个标题记录，四个详细记录发送的应用程序必须在 CODE 字段中放置一个 H 来标识它是标题记录，在 CODE 字段中放置 D 来标识是详细记录。对每一次输入操作，OS/400 程序都要把在缓冲区中第 1 位的值同在 RECID 键字上指定的值相比较（第 1 位是 CODE 字段在缓冲区中的位置）。如果在记录中的值是 H，那么 OS/400 程序就会选择记录格式名 HEADER；如果记录中的值为 D，那么 OS/400 程序就会选择记录格式名 DETAIL。

如果接收到记录在数据部分缓区的第一位不包含 H 或 D，那么记录格式 CATCH(RECID 缺省记录格式) 就是被选择的记录格式名。

下面是记录格式 HEADER 的缓冲区：

```

应答指示器 10 (1 字节)
CODE (1 字节)
TITLE (30 字节)
ACTNBR (6 字节)

```

....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8		
A	R RCD1	RECID(1 'ABC')
A	FLD1	10
A	R RCD2	RECID(1 'AB')
A	FLD1	10
A	R RCD3	RECID(1 'A')
A	FLD1	10
A	R CATCH	
A	FIELD	10
A		

图 5-37 指定 RECID 键字 (当值参数有类似的值时，RECID 键字间的关系)

三个记录格式需要彼此区别；值参数中的第一个字符是相同的，DDS 中最先指定最长的参数值使 OS/400 程序把第一个记录格式与其它的区别开来。原因是如果缓冲区的前 10 位包括 ABCDEFGHIJ，且 RCD3 是最先被指定的，那么即使得到了 RCD1，RCD3 也能被标识出来。RCD1 和 RCD2 则不能被识别出来，这是因为 OS/400 程序在成功地找到一个匹配对后就不再检测了。

### 5.3.34 REF (引用)

用此文件层键字来指定一个能从中检索字段描述的文件名。可以用它要从以前定义过的记录格式中的几个字段复制描述信息。该键字允许对文件名使用一次而不是用在每个字段上。对字段的引用要用 REFFLD 键字。想访问多个文件，可以使用 REFFLD 键字。REF 键字只能被指定一次。

键字的格式为：

REF ([库名/]数据文件名 [记录格式名])

如果在引用文件中有多个记录格式，那么就为该键字指定记录格式名作为参数值来通知 OS/400 程序，如果不是顺序地查找记录格式时使用哪一个记录格式。

数据文件名是必需的参数，库名和记录格式名是可选的。

如果没指定库名，那么就用文件生成时使用的当前库列表 (\*LIBL)。如果没指定记录格式名，那么就按它们被定义的顺序查找，使用第一次出现的字段。详细信息，请参见附录 A “何时指定 REF 和 REFFLD”。

可以在该键字上指定一个分布式数据管理文件 (DDM)。当使用 DDM 文件时，数据库文件名和库名是在源系统上的 DDM 文件名和库名，记录格式名是在目标系统上远程文件中的记录格式名。

可选指示器对此键字无效。

图 5-38 和图 5-39 显示了如何指定 REF 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A                               REF(FILE1)  
A           R RECORD  
A           FLD1      R  
A
```

图 5-38 指定 REF 键字（例 1）

FLD1 与文件 FILE1 中的第一个（或唯一的）FLD1 的属性相同。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A                               REF(LIB1/FILE1 RECORD2)  
A           R RECORD  
A           FLD1      R  
A
```

图 5-39 指定 REF 键字（例 2）

FLD1 与在 LIB1 中的 FILE1 中的 RECORD2 中的 FLD1 的属性相同。

### 5.3.35 REFFLD (引用字段)

在以下三种情况之下引用一个字段时，要使用此字段层键字：

- 被引用字段名与 19~28 列中的名不同。
- 被引用字段名与 19~28 列中的名相同，但记录格式、文件或被引用字段的库名与 REF 键字中指定的不同。
- 被引用字段出现在与引用字段相同的 DDS 源文件中。

键字格式为：

REFFLD ([记录格式名/]被引用字段名 [{\*SRC : [库名/]数据文件名}])

即使被引用字段与引用字段相同，也需要有被引用字段名。当被引用文件有多个记录格式时，要用记录格式名，当被引用字段与引用字段在相同的 DDS 源文件中时，使用\*SRC（而不是数据文件名）。当没有指定数据文件名、库名和 REF 键字时，\*SRC 是缺省值。

注：当引用在同一个 DDS 源文件中的字段时，被引用的字段必须在正定义字段的前面。

当要查找某一个数据文件的时候指定数据文件名（如果必要也可有库名）。如果在同一个 DDS 源文件中，在文件层指定了 REF，在字段层上指定了 REFFLD，那么有一个特定的查找序列。详细信息，请参见附录 A “何时指定 REF 和 REFFLD”。

必须在第 29 位中指定字母 R。在某些情况下，如指定了一个长度值，那么在数据文件中为字段指定的一些键字就不包括在 ICF 文件中。详细信息，请参见 5.2.8“引用(第 29 位)”。

可以在此键字上指定分布式数据管理文件 (DDM)。当使用一个 DDM 文件时，数据文件名和库名分别是在源系统中的 DDM 文件名和库名。被引用字段名和记录格式名是在目标系统上的远程文件中的字段名和记录格式名。

注：功能文件不能用做引用文件。

可选指示器对此键字无效。

图 5-40 显示了如何指定 REFFLD 键字。

00010A	R	FMAT1	1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00020A	ITEM	5	
00030A	ITEM1	R	REFFLD(ITEM)
00040A	ITEM2	R	REFFLD(FMAT1/ITEM)
00050A	ITEM3	R	REFFLD(ITEM FILEX)
00060A	ITEM4	R	REFFLD(ITEM LIBY(FILEX))
00070A	ITEM5	R	REFFLD(FMAT1/ITEM LIBY(FILEX))
00080A	ITEM6	R	REFFLD(ITEM *SCR)

A

图 5-40 指定 REFFLD 键字

因为没有指定 REF，第 00030 和 00040 行的缺省值是检索定义它们的 DDS 源文件，在 00080 行中，\*SRC 参数值明确地指定了源文件。

### 5.3.36 RQSWRT (申请写)

用此文件层或记录层键字为你的程序发送数据申请许可。

此键字无参数。

当 RQSWRT 键字用于有下列键字之一的记录中时，它必须要有可选指示器：

ALWWRT	FMTNAME
CONFIRM	FRCDTA
DETACH	SUBDEV
ENDGRP	VARBUFMGT
EVOKE	VARLEN
FMH	

在运行期间，当 RQSWRT 键字有效时忽略上述这些键字。如果没有可选指示器的 RQSWRT 键字用于有这些键字之一的记录中时，就会出现错误信息，在生成期间忽略这些键字。

在运行期间，当 CANCEL、CNLINVITE、EOS、FAIL、RSPCONFIRM 或 NEGRSP 有效时，忽略 RQSWRT 键字。当这些键字用于有 RQSWRT 键字的记录中时，它们必须要有可选指示器。如果没有可选指示器的 CANCEL、CNLINVITE、EOS、FAIL 或 NEGRSP 键字用于有 RQSWRT 键字的记录中时，则出现一条错误信息，在生成期间忽略 RQSWRT。

不能与 TIMER 一起指定 RQSWRT。

可选指示器对此键字有效。在文件层上指定此键字时，应该指定一个可选指示器。

图 5-41 显示了如何指定 RQSWRT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
00010A      R CUSMST  
00020A  14          RQSWRT  
A
```

图 5-41 指定 RQSWRT 键字

### 5.3.37 RSPCONFIRM (应答确认)

用此文件层或记录层键字给接收的确认申请发送一个正响应。

此键字无参数。

当 RSPCONFIRM 键字用于下列键字之一所在的记录中时，它必须要有可选指示器：

ALWWRT	FMH	RQSWRT
CONFIRM	FMTNAME	SUBDEV
DETACH	FRCDTA	VARBUFMGT
ENDGRP	INVITE	VARLEN
EVOKE		

在运行期间，当 RSPCONFIRM 键字有效时，忽略上述这些键字和数据字段。如果一个无选择的 RSPCONFIRM 键字用到有这些键字的记录中时就会出现错误信息且在创建期间忽略这些键字。如果一个没被选择的 RSPCONFIRM 键字用于有数据字段的记录中时，就会出现严重的错误而且文件也不能生成。

在运行期间，当 EOS、NEGRSP、CANCEL 或 CNLINVITE 有效时，忽略 RSPCONFIRM 键字。当这些键字用于有 RSPCONFIRM 的记录中时必须有可选指示器。如果无选择的 EOS、FAIL、NEGRSP、CANCEL 或 CNLINVITE 用于有 RSPCONFIRM 键字的记录中，则会出现一条错误信息且在生成期间忽略 RSPCONFIRM 键字。

可选指示器对此键字有效。在文件层上指定此键字时，应该指定一个可选指示器。

不能与 TIMER 键字一起指定 RSPCONFIRM。

图 5-42 显示了如何指定 RSPCONFIRM 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A      R RCD  
A  
A  20          RSPCONFIRM  
A
```

图 5-42 指定 RSPCONFIRM 键字

如果可选指示器 20 为 ON，那么对 RCD 的输出操作将发送一个正响应来确认从远程程序中接收的请求。

### 5.3.38 SECURITY (安全)

当你的程序启动了一个远程系统上的程序（参看 EVOKE 键字）时，用此文件层和记录层键字来包括安全信息。任何指定了 SECURITY 键字或者在文件层隐含指定了安全的记录格式，必须在记录格式上指定 EVOKE 键字，或者在文件层对这个记录格式隐含指定 EVOKE 键字。如果没指定 EVOKE 键字，则会产生严重的错误且文件不能生成。

键字的格式为：

SECURITY (安全子字段 子字段定义[.3.])

安全子字段参数用来标识被定义的子字段，此参数是必需的，其指定值必须是下列之一：

值	意义
1	(配置文件 ID)
2	(口令)
3	(用户 ID)

子字段定义参数必须是下列之一：

#### \*USER

AS/400 用户的用户配置文件名作为安全子字段的值。例如，如果为口令子字段指定 \*USER，那么用户配置文件名就作为口令来使用。

#### \*NONE

指出使用一个空安全值。

‘字符串’

可以指定 10 个字符的安全信息。

#### 字段名

指定的字段中包括了安全信息。字段的长度不能超过 10 个字符。如果在文件层指定了 SECURITY 键字，那么此参数无效。

#### &字段名

指定的字段中包含了安全信息。字段的长度不能超过 10 个字符。如果在文件层指定了 SECURITY 键字，那么此参数无效。

不能与 TIMER 键字一起指定 SECURITY。

可选指示器对此键字有效。

图 5-43 显示了如何指定 SECURITY 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A 01                      SECURITY(2 'JONES' 3 'WHITE')
00020A
00030A
00040A
00050A
00060A      R RCD1
00070A 03                      SECURITY(2 'JONES' 3 *USER)
```

00080A		EVOKE (LIB2/PGM2)
00090A		
00100A	R RCD2	
00110A		EVOKE (LIB3/PGM3)
00120A		
00130A	FIELD1	5A
00140A		
00150A	R RCD3	
00160A 60		SECURITY(2 &CLVAR1 3 &CLVAR2)
00170A		EVOKE (LIB4/PGM4)
00180A	CLVAR1	10A
00190A	CLVAR2	10A
A		

图 5-43 指定 SECURITY 键字

在文件层上指定的 SECURITY 适用于所有的记录格式，如果它被选择（指示器 01 为 ON），JONES 的口令和 WHITE 的用户 ID 就被发送到远程系统上。对 RCD1，如果指示器 03 为 ON，则当前用户的用户配置文件名作为用户 ID 与口令 JONES 一起作为安全信息送到远程系统上。

对 RCD2 没有安全信息发送到远程系统上。

对 RCD3，如果指示器 60 为 ON，在 CLVAR1 中的值作为口令，CLVAR2 中的值做为用户 ID，二者一起做为安全信息发送到远程系统上。

### 5.3.39 SUBDEV (子设备)

此文件层或记录层的键字允许你的程序请求一个特定的子设备（例如打印机），并把数据传到该子设备上。

键字的格式为：

SUBDEV(\*DC1 | \*DC2 | \*DC3 | \*DC4)

在运行期间，当 EOS、DETACH、RSPCONFIRM 或 RQSWRT 有效时，忽略 SUBDEV 键字。当这些键字用到有 SUBDEV 键字的记录中时必须要有可选指示器。如果没有可选指示器的 EOS、DETACH、RSPCONFIRM 或 RQSWRT 用到有 SUBDEV 键字的记录中，就会出现一条错误信息且在生成期间忽略 SUBDEV 键字。

只能为每个键字指定一个参数值。

在文件中，可以多次指定此键字。但不能在文件层和记录层上指定相同的参数值，即使每次都指定可选指示器也不行。例如，在文件层上指定了 SUBDEV(\*DC1)，就不能在文件中的任何地方再指定 SUBDEV(\*DC1)。

如果在文件层和记录层上都指定了 SUBDEV 键字，而且你的程序选择了文件层上的那个，那么即使记录层上的键字也被选择了，也是无效的。

最多可以为每个记录格式指定四次 SUBDEV 键字，如果多次指定 SUBDEV 键字，那么必须每次都指定可选指示器，也可以对每个键字值指定一次。

OS/400 程序发送如下的设备选择字符，设备选择字符的意义是由远程系统为设备设置的。

参数值	发送字符
-----	------

*DC1	十六进制 11
*DC2	十六进制 12
*DC3	十六进制 13
*DC4	十六进制 5D

不能与 TIMER 键字一起指定 SUBDEV。

可选指示器对此键字有效。

图 5-44 显示了如何指定 SUBDEV 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A 01                      SUBDEV(*DC1)
00020A 02                      SUBDEV(*DC4)
      A          R RECORD
      A
```

图 5-44 指定 SUBDEV 键字

如果指示器 01 为 ON，那么 OS/400 程序就发送部件选择字符十六进制 11 到输出操作中（不管指示器 02 如何设置）。

如果指示器 02 为 ON 且 01 为 OFF，则 OS/400 程序发送部件选择字符十六进制 5D。

#### 5.3.40 SYNLVL (同步级)

此文件层或记录层键字指出程序要求的同步级。SYNLVL 键字只有在 EVOKE 键字有效时它才有效。

键字的格式为：

SYNLVL[(\*NONE | \*CONFIRM | \*COMMIT)]

当你的程序和远程程序都不使用 CONFIRM 键字时指定\*NONE。如果你的程序或远程程序中有一个使用了 CONFIRM 键字时，就指定\*CONFIRM。

指定\*COMMIT 来标识本地程序可以用 PRPCMT 键字或落实退回操作使用本地系统的落实控制支持，对\*COMMIT 级对话允许使用 CONFIRM 键字。

当程序被激活时指定了 SYNLVL(\*NONE)，那么就不能指定 CONFIRM 键字。用此键字的记录必须有有效的 EVOKE 键字。

不能与 TIMER 键字一起指定 SYNLVL。

如果此键字用在文件中的多个记录格式上，那么可选指示器对此键字是有效且必需的。

图 5-45 显示了如何指定 SYNLVL 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R RCD
00020A              EVOKE(LIBRARY1/PROGRAM1)
00030A              SYNLVL(*CONFIRM)
      A
```

图 5-45 指定 SYNLVL 键字

EVOKE 键字启动在远程系统的 LIBRARY1 中的 PROGRAM1，SYNLVL 键字将设置一

个同步级，用它来确认是否接收了数据。当申请一个确认时（指定 CONFIRM 键字），远程程序必须通过一个正应答或负应答来辨别它是否收到了数据。

#### 5.3.41 TEXT (说明)

用此记录层或文件层键字为程序文档中记录格式或字段提供注释。

键字的格式为：

TEXT ('说明')

'说明' 必须括在撇号中，如果 '说明' 长度超过了 50 位，那么只有前 50 个字符可以被高级语言编译程序使用。

可选指示器对此键字是无效的。

图 5-46 显示了如何在记录层和字段层上指定 TEXT 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
00010A      R CUSMST          TEXT('Customer Master Record')
00020A      FLD1            3 0    TEXT('ORDER NUMBER FIELD')
A
```

图 5-46 指定 TEXT 键字

#### 5.3.42 TIMER (计时器)

用此记录层键字来为你的程序指定一个时间间隔，来等待特定功能完成，为设置计时器，程序要发出计时器有效的输出操作。

键字的格式为：

TIMER(HHMMSS | &字段名)

HHMMSS:

时间间隔是一个 6 位的数值，HH 表示小时(00 – 99)，MM 表示分(00 – 59)，SS 表示秒(00 – 59)。

&字段名

时间间隔参数是包含以 HHMMSS 形式表示的时间值的字段名。字段名必须存在于记录格式中，而且字段必须是一个长度为 6，数据格式为 S，用法为 P，小数位为 0 的区位字段。

下列键字不能与 TIMER 一起规定：

ALWWRT	ENDGRP	RCVCONFIRM	RCVTRNRND
CANCEL	EVOKE	RCVCANCEL	RECID
CNLINVITE	FAIL	RCVCTLDTA	RQSWRT
CONFIRM	FMH	RCVDETACH	SECURITY
CTLDTA	FMTNAME	RCVENDGRP	SUBDEV
DETACH	FRCDTA	RCVFAIL	SYNQLVL
DFREVOKE	INVITE	RCVFMH	VARBUFMGT
EOS	NEGRSP	RCVNEGRSP	VARLEN

TIMER 会代替在 CRTICFF、CHGICFF 及 OVRICFF 命令中的 WAITRCD 参数，在计时器功能有效的时间间隔内，忽略 WAITRCD 参数。

可选指示器对此键字无效。

图 5-47 显示了如何指定 TIMER 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RCD1                  TIMER(002512)  
A          R RCD2                  TIMER(&FIELD1)  
A          FIELD1      6S P  
A  
A
```

图 5-47 指定 TIMER 键字

对 RCD1 输出操作时，计时器设置为 0 小时 25 分 12 秒。在对 RCD2 的输出操作上，计时器设置成 FIELD1 中的值。

### 5.3.43 TNSSYNLVL (传输同步级)

用此文件层或记录层键字来定义传输同步级（在 SYNLVL 键字上指定）。它是在指定了 DETACH 和 ALWWRT 键字且做写操作时执行。

此键字没有参数。

必须在文件上指定 DETACH 和 ALWWRT，或者和 TNSSYNLVL 在同一个记录上。

TIMER 键字不允许和 TNSSYNLVL 一起用。

可选指示器对此键字无效。

图 5-48 和图 5-49 显示了如何指定 TNSSYNLVL 键字。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RCD1                  EVOKE(LIBRARY1/PROGRAM1)  
A  
A          R RCD2                  SYNLVL(*CONFIRM)  
A          DETACH  
A          TNSSYNLVL  
A
```

图 5-48 指定 TNSSYNLVL 键字

它给出一个对 RCD2 的写操作，你的程序和远程程序之间的传输将持续下去，直到远程程序确认收到了脱离指示。

```
|....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A          R RCD1                  EVOKE(LIBRARY1/PROGRAM1)  
A  
A          R RCD2                  SYNLVL(*CONFIRM)  
A          ALWWRT  
A          TNSSYNLVL
```

图 5-49 指定 TNSSYNLVL 键字

它给出一个对 RCD2 的写操作，你的程序和远程程序之间的对话被置为延迟接收状态。当一个 CONFIRM 或 COMMIT 操作完成以后，对话进入接收状态。

#### 5.3.44 VARBUFMGT (可变缓冲区管理)

用此记录层键字来发送或接收每个输出操作使用一个记录格式的多个或部分记录。在一个发送操作中，必须用 VARLEN 键字来指定发送的数据长度，否则，就使用记录格式的长度。在一个接收操作中，接收数据的长度就是记录格式的长度。

此键字无参数。

在运行期间，如果 CANCEL、EOS、FAIL、NEGRSP、RSPCONFIRM 或 RQSWRT 键字有效，则忽略 VARBUFMGT 键字。当这些键字用到有 VARBUFMGT 键字记录中时，必须要有可选指示器。如果 CANCEL、EOS、FAIL、NEGRSP 或 RQSWRT 用到有 VARBUFMGT 键字的记录上而没有可选指示器，则会出现错误信息，且生成期间忽略 VARBUFMGT 键字。至少应为记录格式中的用户数据指定一个数据字段（用途 B 或空格）。

下列情况下不能指定 VARBUFMGT 键字：

- 使用 TIMER 键字
- 与 RECID 在同一个记录格式中
- RECID 或 INVITE 缺省记录格式

可选指示器对此键字无效。

图 5-50 显示了如何指定 VARBUFMGT 键字。

00010A	R MULTFMT1	
00020A		VARBUFMGT
00030A	DATAFLD	32A
00040A	R MULTFMT2	
00050A		VARLEN(&LENFLD)
00060A		VARBUFMGT
00070A	DATAFLD	32A
00080A	LENFLD	5S P

图 5-50 指定 VARBUFMGT 键字

假设 42THIS RECORD WILL NOT FIT INTO ONE BUFFER 是要发送或接收的数据。第一个记录格式上的 VARBUFMGT 键字发送或接收数据的前 32 个字节。第二个记录格发送 10 个字节的数据。数据的长度 (10) 是在 LENFLD 中设定的。

#### 5.3.45 VARLEN (变长)

用此记录层键字来指示发送的记录长度是可变的。长度在运行期间由字段参数指定。

键字的格式为：

VARLEN(&字段名)

&字段名参数是必需的，它指定了包含发送的用户数据长度的字段名。字段名必须存在于记录格式中，且定义为长度为 5，数据类型为 S，用法为 P，小数位为 0 的区位字段。

在参数字段中设置的长度值即用户数据的长度，不包括指示器。长度值用十进制，并在运行期间检验。长度值不可超过 DDS 记录格式的长度。其最大值取决于使用的通讯类型。VARLEN 只对输出操作有效。

在运行期间，当 CANCEL、EOS、FAIL、NEGRSP、RSPCONFIRM 或者 RQSWRT 键字有效时，忽略 VARLEN 键字。当这些键字用到有 VARLEN 键字的记录上时，必须要有可选指示器。如果 CANCEL、EOS、FAIL、NEGRSP 或 RQSWRT 用到有 VARLEN 键字的记录上而不带可选指示器，则会出现错误信息，在生成期间忽略 VARLEN 键字。

应在记录格式中至少为用户数据指定一个数据字段（用途为 B 或空格）。

不能与 TIMER 键字一起指定 VARLEN。

可选指示器对此键字无效。

图 5-51 显示了如何指定 VARLEN 键字。

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8  
A           R RCD           VARLEN(&LEN  
FLD)  
A           DATAFLD      32760A  
A           LENFLD       5S   P  
A
```

图 5-51 指定 VARLEN 键字

在对 RCD 的输出操作中，发送的 DATAFLD 中的数据长度是由 LENFLD 中的长度确定的。

# 系统数据字符集

Hexadecimal	Character	Description
40		Blank
4B	.	Period
4C	<	Less than sign
4D	(	Left parenthesis
4E	+	Plus sign
50	&	Ampersand
5C	*	Asterisk
5D	)	Right parenthesis
5E	;	Semicolon
60	-	Minus sign
61	/	Slash
6B	,	Comma
6C	%	Percent sign
6D	—	Underline
6E	>	Greater than sign
6F	?	Question mark
7A	:	Colon
7D	'	Single quotation mark
7E	=	Equal sign