

# 字符管理程序 (CMGR) 用户手册

产品号：BA322-90021

2005 年 5 月

本手册描述字符管理程序 (CMGR) 公用程序的概念、特性及命令。

**修订 / 更新资料：** 本手册取代 OpenVMS/Hanzi V1.5 AXP 的 CMGR 用户手册

**软件版本：**  
CMGR 2.0 版本  
OpenVMS/Hanzi I64 8.2 版本  
OpenVMS/Hanzi Alpha 7.3-2 版本

Hewlett-Packard Company

Palo Alto, California

---

(c) Copyright 2005 Hewlett-Packard Development Company, L.P.

机密计算机软件。必须有 HP 授予的有效许可证，方可拥有、使用或复制本软件。

根据供应商的标准商业许可证的规定，美国政府应遵守 FAR 12.211 和 12.212 中有关“商业计算机软件”、“计算机软件文档”与“商业货物技术数据”条款的规定。

本文档中的信息如有更改，恕不另行通知。随 HP 产品及服务提供的明示性担保声明中列出了适用于此 HP 产品及服务的专用担保条款。本文中的任何内容均不构成额外的担保。HP 对本文档中的技术或编辑错误以及缺漏不负任何责任。

Intel 和 Itanium 是 Intel Corporation 及其子公司在美国及其他国家或地区的商标或注册商标。

Printed in Singapore.

# 目录

序言 .....	vii
<b>第 1 章 引言</b>	
1.1 语言转换 .....	1-1
1.2 如何调用 CMGR .....	1-1
1.2.1 调用 CMGR 字形编辑程序 .....	1-2
1.3 字符属性 .....	1-3
1.4 系统属性数据库和用户属性文件 .....	1-3
1.4.1 系统属性数据库 .....	1-3
1.4.2 用户属性文件 .....	1-4
1.5 字符管理程序的主要功能 .....	1-4
<b>第 2 章 DEC 汉字字符集</b>	
<b>第 3 章 从字形系统迁移到 CMGR</b>	
3.1 转换命令 .....	3-1
3.1.1 转换数据库 .....	3-1
3.1.2 转换字形文件 .....	3-2
3.2 废弃的字形系统公用程序 .....	3-2
3.2.1 FDESIGN 公用程序 .....	3-3
3.2.2 FUPDATE 公用程序 .....	3-3
3.2.3 CTABLE 公用程序 .....	3-3
3.2.4 PRELOAD 公用程序 .....	3-3
<b>第 4 章 使用 CMGR</b>	
4.1 CMGR 中的字符代码和区说明 .....	4-1
4.2 选择当前数据库 .....	4-1
4.3 使用 CMGR 字形编辑程序 .....	4-2
4.4 CMGR UPDATE 命令 .....	4-29
4.5 CMGR SHOW TABLE 和 CMGR SHOW BITMAP 命令 .....	4-30
4.5.1 展示用户属性文件的属性资料 .....	4-30
4.5.2 展示系统属性数据库的属性资料 .....	4-31
4.6 CMGR REMOVE 和 COPY 命令 .....	4-32
4.7 CMGR EXTRACT CODE 和 CMGR EXTRACT REFERENCE 命令 .....	4-33
4.8 CMGR CONVERT FONT_SIZE .....	4-34
4.9 CMGR SHOW CHARACTER_SET 和 CMGR SHOW VERSION 命令 .....	4-34
4.10 CMGR HELP 和 EXIT 命令 .....	4-35
<b>第 5 章 CMGR 命令参考</b>	
5.1 CONVERT DATABASE .....	5-1
5.2 CONVERT FONT_FILE .....	5-3
5.3 CONVERT FONT_SIZE .....	5-4
5.4 COPY .....	5-5

# 目录

5.5 EDIT .....	5-7
5.6 EXIT .....	5-9
5.7 EXTRACT CODE .....	5-10
5.8 EXTRACT REFERENCE .....	5-13
5.9 HELP .....	5-14
5.10 REMOVE .....	5-15
5.11 SET DATABASE .....	5-17
5.12 SHOW BITMAP .....	5-18
5.13 SHOW CHARACTER_SET .....	5-21
5.14 SHOW DATABASE .....	5-22
5.15 SHOW TABLE .....	5-22
5.16 SHOW VERSION .....	5-25
5.17 UPDATE .....	5-26

## 第 6 章 字形编辑程序参考

6.1 终端支持 .....	6-1
6.2 调用字形编辑程序 .....	6-1
6.2.1 字符属性 .....	6-1
6.2.2 装入字块 .....	6-2
6.3 内部操作 .....	6-2
6.4 屏幕编排设计 .....	6-2
6.5 缓冲区 .....	6-5
6.6 编辑功能 .....	6-5
6.6.1 启动 .....	6-5
6.6.2 Gold 状态 .....	6-6
6.6.3 移动光标 .....	6-6
6.6.3.1 步进 .....	6-6
6.6.3.2 仅移动 .....	6-6
6.6.3.3 跳跃 .....	6-6
6.6.4 编辑态 .....	6-6
6.6.4.1 光标态 .....	6-6
6.6.4.2 贴补态 .....	6-6
6.6.4.3 类型态 .....	6-7
6.6.4.4 绕回态 .....	6-7
6.6.5 DISPLAY 与 REFER 窗口 .....	6-7
6.6.6 USE 缓冲区 .....	6-7
6.6.7 选择区域 .....	6-8
6.6.8 剪切与贴补 .....	6-8
6.6.9 标记 .....	6-8
6.6.10 几何图形 .....	6-8
6.6.11 特殊功能 .....	6-9
6.6.12 编辑不同大小的字形 .....	6-10
6.6.13 转换不同大小的字形 .....	6-10
6.6.14 输入整理值 .....	6-10
6.6.15 输出、删除、取消 .....	6-10
6.6.16 保存 .....	6-10
6.6.17 取得求助 .....	6-11

6.6.18 终止字形编辑程序 .....	6-11
6.7 快速参考 .....	6-11
6.7.1 小键盘图 .....	6-11
6.7.2 字形编辑程序功能列表 .....	6-13

## 附录 A 整理值列表

A.1 拼音整理值 .....	A-1
A.2 部首整理值 .....	A-3
A.3 笔画数整理值 .....	A-3

## CMGR 词汇表

### 图

1-1 编辑程序编排设计 (英文) .....	1-2
1-2 编辑程序编排设计 (中文) .....	1-3
2-1 DEC 汉字字符平面 .....	2-2
2-2 DEC 汉字字符集 .....	2-2
2-3 DEC GB-2312 字符集的编排设计 .....	2-3
2-4 扩充 GB 字符集的编排设计 .....	2-4
4-1 按 <Enter> 输入代码 .....	4-4
4-2 按 <KP0> 调用 USE 功能 .....	4-6
4-3 从 HEI 数据库装入字形模式 .....	4-7
4-4 按 <SELECT> + <KP3> X 6 + <KP6> X 11 选择区域 .....	4-8
4-5 按 <REMOVE> 把选定部分剪切到 Paste 缓冲区 .....	4-9
4-6 按 <GOLD> + <KP0> 返回到原来的位映象 .....	4-10
4-7 按 <GOLD> + <KP7> 前去适当的位置并按 <INSERT HERE> 贴补该 Paste 缓冲区 .....	4-11
4-8 键入关键字 "MEMORY" 使用内存记录 .....	4-12
4-9 内存字形装入 USE 缓冲区 .....	4-13
4-10 按 <KP.> 以实际大小显示当前字形 .....	4-14
4-11 按 <Gold> + <KP.> 并键入 "MEMORY 10102" 以在其中一个 REFER 窗口显示内存字形 .....	4-15
4-12 内存记录的 10102 字形显示在最左面的 REFER 窗口中 .....	4-16
4-13 按 <O> 之前首先选择区域 .....	4-17
4-14 按 <O> 画椭圆 .....	4-18
4-15 按 <Gold> + <M> 标记当前光标位置 .....	4-19
4-16 使用 <KP6> 和 <KP8> 画三角形的两边 .....	4-20
4-17 按 <Select> 开始选择 .....	4-21
4-18 通过按 <Gold> + <G> 完成选择, 光标会直接跳到标记位置 .....	4-22
4-19 按 <L> 完成三角形 .....	4-23
4-20 用于编辑 32x32 大小字形的屏幕编排设计 .....	4-24
4-21 按 <F18> 并键入 24 .....	4-25
4-22 把 24x24 大小的字形转换成 32x32 .....	4-26
4-23 按 <F17> 输入整理特征 .....	4-27
4-24 显示的整理特征 .....	4-28
4-25 LIST1.LIS 的输出 .....	4-30
4-26 LIST2.LIS 的输出列表 .....	4-31
6-1 编辑程序编排设计 .....	6-3
6-2 编辑 40x40 大小字形时的编辑程序编排设计 .....	6-5

# 目录

A-1	部首列表 .....	A-3
-----	------------	-----

## 表

6-1	支持的字符属性 .....	6-2
6-2	编辑态与画几何图形 .....	6-8
6-3	应用小键盘 .....	6-13
6-5	控制键 .....	6-15
6-6	主键盘键 .....	6-15
6-4	编辑小键盘 .....	6-15
A-1	常规拼音 .....	A-1
A-2	特殊拼音 .....	A-2

# 序言

## 对象

本手册的对象为所有 OpenVMS/Hanzi 系统的用户。

## 手册结构

本手册包含六章、一个附录及一份 CMGR 词汇表。

- 第 1 章是字符管理程序这公用程序的综述。
- 第 2 章简要描述 DEC 汉字字符集。
- 第 3 章提供如何从旧字形系统迁移到 CMGR 的资料。
- 第 4 章提供能助您认识软件的资料。会运用例子来指导您怎样使用 CMGR 命令。
- 第 5 章提供所有 CMGR 命令的参考。
- 第 6 章提供 CMGR 字形编辑程序的完整参考资料。
- 附录 A 提供关于拼音、部首和笔画数的有效整理值。
- 词汇表是本手册中经常出现的一些特定术语及其定义的列表。

## 手册中所用的常规

<b>“Hanzi”</b>	世界不同的地方正使用着不同的中文字符集：中华人民共和国及新加坡使用简体汉字、香港及台湾使用繁体汉字、日本使用 Kanji, 而朝鲜则使用 Hanja。为针对这复杂的情况并避免混淆，本手册采用了“Hanzi”字来说明所指的简体汉字是由中华人民共和国国家标准信息交换用汉字编码字符基本集（GB-2312-80）所定义的简体汉字。
<b>&lt;Return&gt;</b>	回车 在例子或格式中，除非另有声明，否则用户输入的每一行的末尾都出现一个隐含的回车。用户必须在输入行的末尾按 Return 键。
<b>键符号</b>	在例子中，键和键序列都会以符号表示，如 <PF2> 和 <CTRL/Z> 或 <PF2> 和 <CTRL/Z>。
<b>CTRL</b>	CTRL/x 表示用户必须在按住标号为 CTRL 键的同时按另一个键。例如，<CTRL/C>、<CTRL/Y> 及 <CTRL/O>。
.	纵向省略号表示不展示系统在应答某一命令时要显示的所有数据，或者不展示用户输入的所有数据。非直接与讨论主题有关的资料会被省略。
...	横向省略号表示可输入附加参数、值或资料。

## 序言

### 相关手册

- *OpenVMS/Hanzi 用户手册*

# 第 1 章

## 引言

字符管理程序 (CMGR) 是一个软件部件, 用以管理中文字符的属性。这些中文字符主要是指 DEC 汉字字符集中的**用户定义字符**。用户可以操纵的属性包括字形和整理特征。

这软件取代 OpenVMS/Hanzi 5.4 及之前版本的字形系统。第 3 章会详细描述如何从旧的字形系统迁移到 CMGR 的系统属性数据库和用户属性文件。

CMGR 让用户建立用户定义字形和用户定义整理值、操纵系统属性数据库及建立用以参考的属性列表。按照默认, CMGR 命令仅在用户定义字符上操作, 但用户可以使用一些命令的 /RESERVED 限定词来操纵保留用户定义字符。

### 1.1 语言转换

与 OpenVMS/Hanzi 的其他软件部件 / 公用程序一样, CMGR 能提供中英文求助文本和信息。用户可以使用 HANZIGEN 公用程序来转换中英文, 如下所示:

转换成英文:

```
HANZIGEN> set/output=ascii_msg
```

转换成中文:

```
HANZIGEN>set/output=hanzi_msg
```

注册后的默认语言是中文。

### 1.2 如何调用 CMGR

以下列 DCL 命令调用 CMGR:

```
$ CHARACTER_MANAGER  
CMGR>
```

或

```
$ CHARACTER_MANAGER COPY 10101 10102  
(这个命令调用 CMGR 并执行 COPY 命令。)
```

如有需要, 用户可随时在 CMGR 提示符处键入 HELP 以取得求助资料。

```
CMGR> help
```

Information available:

```
Character_attributes CONVERT COPY EDIT EXIT EXTRACT  
HELP REMOVE SET SHOW System_Attribute_Database  
UPDATE User_Attribute_Files  
Topic?
```

# 引言

## 1.2 如何调用 CMGR

建议用户定义外部命令，如下所示：

```
$ CMGR == "$HSY$SYSTEM:CMGR"
```

从而可以下述方式调用 CMGR：

```
$ CMGR
```

```
CMGR>
```

### 1.2.1 调用 CMGR 字形编辑程序

在 CMGR 提示符处键入 EDIT 命令，便可调用 CMGR 字形编辑程序。图 1-1 展示英文接口的编辑程序编排设计，图 1-2 展示中文接口的编辑程序编排设计。在 CMGR 2.0 版本中，字形编辑程序仅支持 VT382 终端。

图 1-1 编辑程序编排设计（英文）

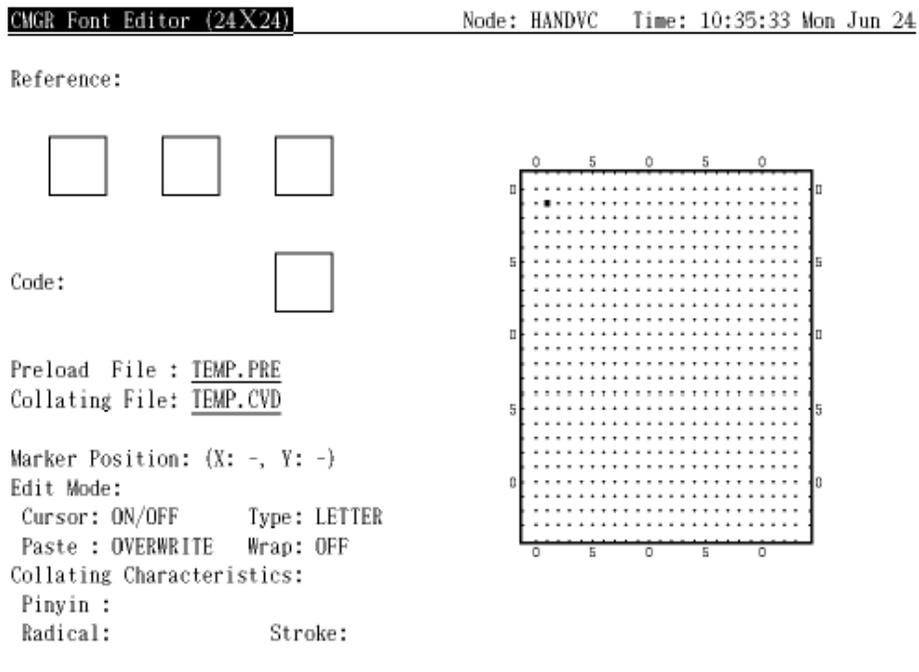
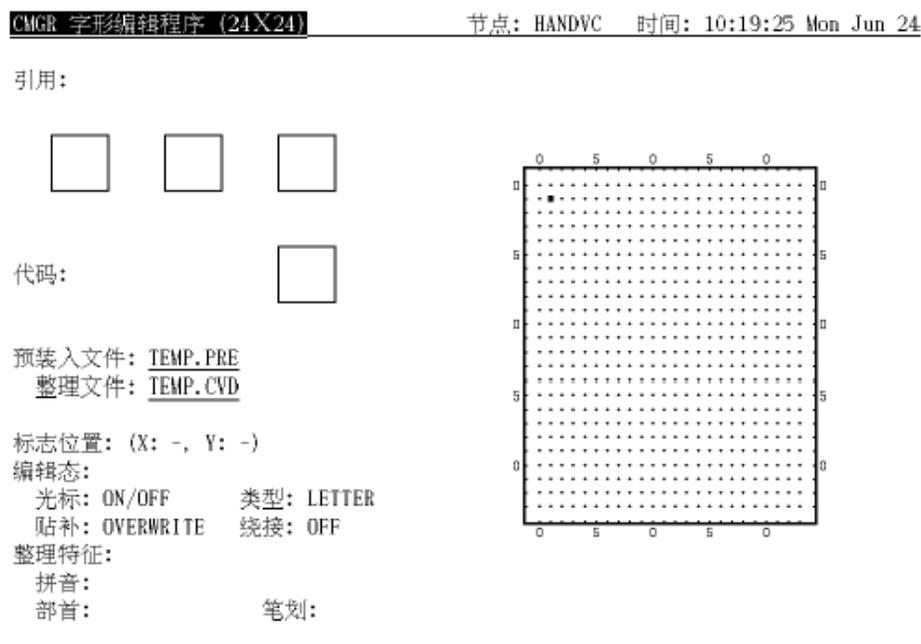


图 1-2 编辑程序编排设计（中文）



## 1.3 字符属性

字符代码可与几个字符属性有关联。用户可以自行定义 DEC 汉字字符集的用户定义区域和保留用户定义区域内代码的属性。CMGR 2.0 版本支持两个用户操纵属性，即位映象字形模式和整理特征。字形模式的大小可以是 24x24、32x32 或 40x40。支持的整理特征包括拼音、部首和笔画数。24x24 字形可用于在终端 / 行式打印机上执行按需装入和预装入。通过 DECprint Printing Services，便可在激光打印机上打印各种可支持其大小的字形模式。如果定义了整理特征，便可用 HSORT/HMERGE 公用程序来给包含用户定义字符的文件执行排序 / 合并。此后，“字符属性”这术语即指这两个受支持的属性。

## 1.4 系统属性数据库和用户属性文件

### 1.4.1 系统属性数据库

CMGR 2.0 版本中有三个系统属性数据库，可存储用户定义字符和保留用户定义字符的系统范围字符属性。这些属性会用以执行按需装入、预装入、打印及 HSORT/HMERGE。这三个数据库的名称分别为 CMGR\_DEFAULT、SONG 和 HEI。每一个数据库都可以存储所有用户定义字符和保留用户定义字符的全部属性。具备三个数据库可让用户建立和使用不同字体的字形。

CMGR 命令在当前数据库上操作。除非系统经理另行设置<sup>1</sup>，否则注册后的当前数据库应是 CMGR\_DEFAULT。要存取其他数据库，可使用 SET DATABASE 命令。

注意，虽然系统的一般用户均可存取（只读）数据库，但只有具有 SYSPRV 特权的帐户才可以执行更新。

1. 欲知详情，请参阅第 5 章中的 SET DATABASE 命令的描述。

## 引言

### 1.5 字符管理程序的主要功能

#### 1.4.2 用户属性文件

对应于每个字符属性，有两类用户属性文件，即预装入文件和整理数据文件。预装入文件用以存储 CMGR 操作所用的位映象字形模式，也可用以把用户定义的字形模式预装入到终端。CMGR 给预装入文件提供一个默认扩充名 .PRE。整理数据文件是另一类的用户属性文件，用以存储用户定义字符的整理特征。这文件只用于 CMGR 操作。同样地，CMGR 也给整理数据文件提供一个默认扩充名 .CVD。

对 CMGR 来说，用户属性文件跟任何一个系统属性数据库都没有关系。因此，同一个用户属性文件可以用于任何一个数据库。然而，不同的数据库通常存储不同字体的字形。所以，具有一种字体字形的用户属性文件可能不适用于另一种字体的数据库。

### 1.5 字符管理程序的主要功能

下面列出字符管理程序的主要功能 / 命令及其简单描述。

- CONVERT DATABASE : 把旧字形系统中的字形转换到 CMGR 的系统属性数据库。
- CONVERT FONT\_FILE : 把旧字形文件中的字形转换到 CMGR 的预装入文件。
- CONVERT FONT\_SIZE : 把预装入文件中的字形模式从一个大小转换成另一大小。
- COPY : 把字符属性从当前系统属性数据库的一个字符复制到另一个字符。
- EDIT : 编辑字符属性。
- EXTRACT CODE, EXTRACT REFERENCE : 从当前系统属性数据库抽取用户属性到用户属性文件。
- SET DATABASE, SHOW DATABASE : 设置当前数据库并展示数据库的资料。
- SHOW BITMAP, SHOW TABLE : 显示有关字符属性的资料。
- SHOW CHARACTER\_SET : 显示字符集的资料。
- SHOW VERSION : 展示 CMGR 的版本号。
- REMOVE : 从当前系统属性数据库除去字符属性。
- UPDATE : 以用户属性文件的用户属性来更新当前系统属性数据库。
- HELP : 请求求助。
- EXIT : 退出 CMGR。

参考第 5 章以了解命令的详细描述。

## 第 2 章

### DEC 汉字字符集

DEC 汉字字符集是由遵循 GB-2312-80 标准<sup>1</sup> 的汉字和非汉字字符组成的一套字符集。

DEC 汉字字符集占了 DEC 汉字字符平面的两个象限。右下面的象限包含 DEC GB-2312 字符集，左下面的象限包含扩充 GB 字符集。图 2-1 展示 DEC 汉字字符平面。

这两个象限共提供了 7,445 个字符和 8,836 个字符位。图 2-2 展示 DEC 汉字字符集中字符和字符位的分布。

扩充 GB 字符集中的字符位划分为两个区域 -- 用户定义区域和保留用户定义区域，用户可在其中定义他们自己的字符属性，以便按自己的需要来扩充 GB-2312 字符集。CMGR 便是 OpenVMS/Hanzi 提供用来定义和操纵这些属性的工具。按照默认，CMGR 命令仅允许用户操纵用户定义区域内的属性。然而，用户也可以通过在 CMGR 命令中指定 /RESERVED 限定词来操纵保留用户定义区域内的属性。

图 2-3 和 图 2-4 说明 DEC 汉字字符集的两个象限中的代码分布。用户必须注意，每一个象限可再分成 94 区，每一区对应于象限中的一行。同时，每一区又分为 94 个位置或“位”。有关 DEC 汉字字符集的详情，请参阅 OpenVMS/Hanzi 用户手册第 2 章。

要联机观看字符集的资料，可使用 CMGR 命令 SHOW CHARACTER\_SET。

---

1. GB-2312-80 标准是汉字信息交换用编码字符基本集的国家标准。由中华人民共和国国家标准局于 1981 年 5 月公布。

# DEC 汉字字符集

图 2-1 DEC 汉字字符平面

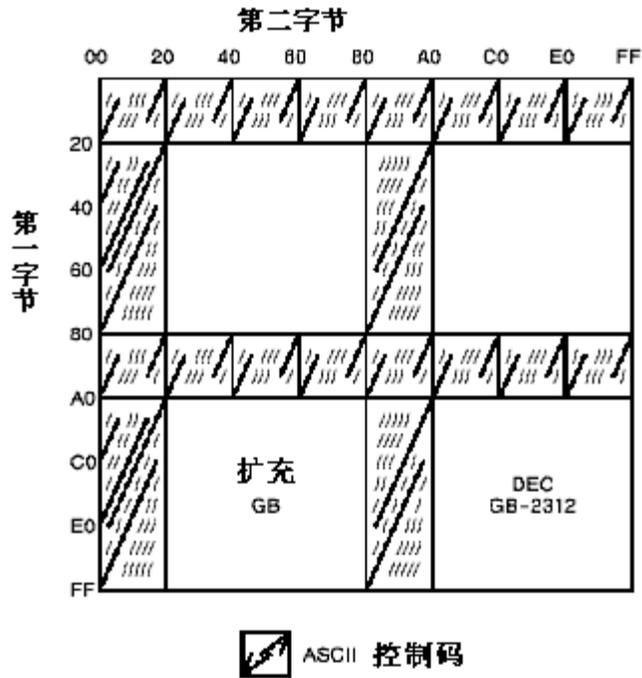


图 2-2 DEC 汉字字符集

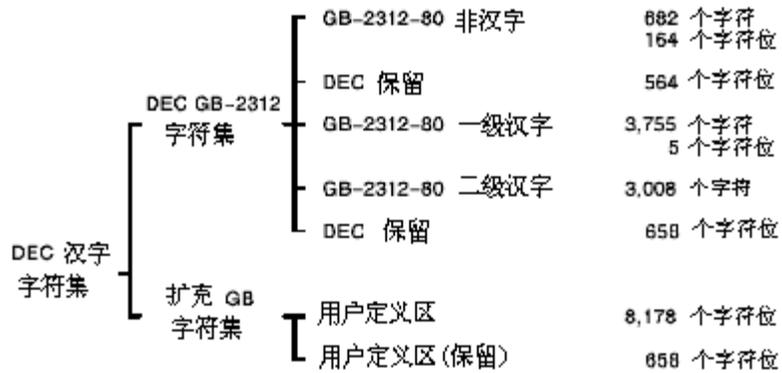
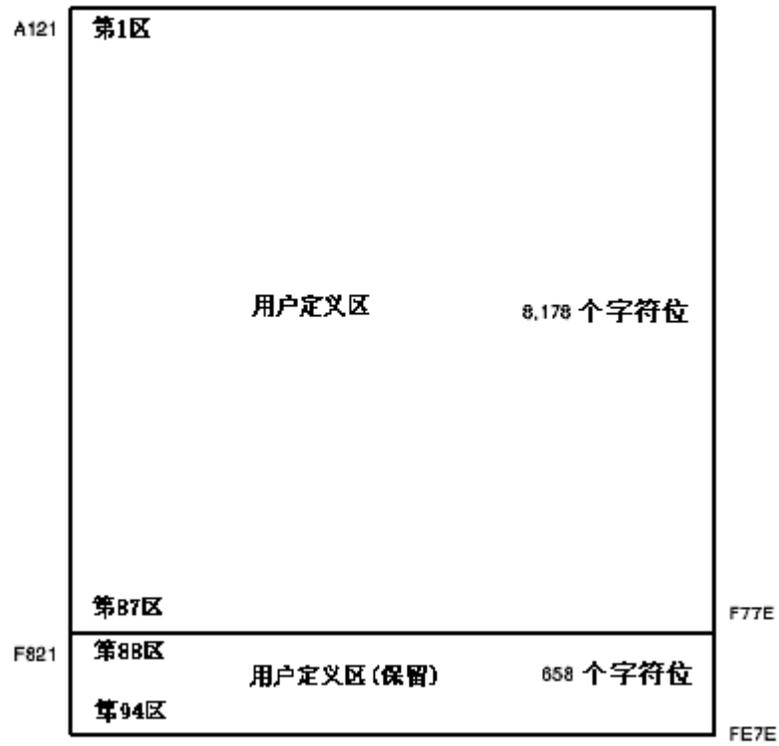


图 2-3 DEC GB-2312 字符集的编排设计

A1A1	第1区	GB-2312 非汉字 和保留字符位	682 个字符	A9FE
	第9区		164 个字符位	
AAA1	第10区	DEC 保留	584 个字符位	AFFE
	第15区			
B0A1	第15区	GB-2312 一级汉字 和保留字符位	3,755 个字符	D7FE
	第55区		5 个字符位	
D8A1	第56区	GB-2312 二级汉字	3,008 个字符位	F7FE
	第87区			
F8A1	第88区	DEC 保留	858 个字符位	FEFE
	第94区			

# DEC 汉字字符集

图 2-4 扩充 GB 字符集的编排设计



## 第 3 章

# 从字形系统迁移到 CMGR

本章适用于已在使用字形系统而现在需要迁移到 CMGR 环境的 OpenVMS/Hanzi 5.5 版本以前的所有用户。

在 OpenVMS/Hanzi 5.5 版本之前，用以存储、操纵和使用用户定义字形模式的环境称为字形系统。用户定义的字形模式存储在字形数据库内。OpenVMS/Hanzi 提供若干公用程序以帮助用户定义他们的字形、在字形数据库中更新这些字形，并以表的格式列出它们。在字形数据库以外，字形模式会存储在字形文件中。例如，FDESIGN 是建立字形模式的公用程序。FDESIGN 的输出便是字形文件。

字形数据库内定义的字形可在按需装入时供字形处理程序使用，或用户可用 PRELOAD 公用程序（自 OpenVMS/Hanzi 5.0 版本以来）来建立预装入文件并把字形预装入到他们的终端。

自 OpenVMS/Hanzi 5.5 版本推出 CMGR 后，便正式取代字形系统。CMGR 的系统属性数据库将取代旧字形系统的字形数据库。在属性数据库以外，字形模式会存储在预装入文件中，而不再会有字形文件。此外，所有随字形系统而来的公用程序都将废弃，取而代之的是 CMGR 的扩充命令集。

### 3.1 转换命令

要协助用户从旧字形系统迁移到 CMGR，CMGR 提供了两个命令。CONVERT DATABASE 命令把旧字形数据库中的字形转换成 CMGR 使用的格式，并把它们存储在新的系统字形数据库中。从字形系统升级到 CMGR（OpenVMS/Hanzi 升级）以后，系统经理应执行一次这命令。CONVERT FONT\_FILE 命令把字形文件转换成预装入文件。用户应使用这命令来把他们的字形文件转换成 CMGR 可操纵的预装入文件。

#### 3.1.1 转换数据库

在旧的字形系统中，字形数据库在 HSY\$SYSTEM 系统目录中有一个称为 FONT1.DAT 的文件。该文件存储着所有用户定义的字形模式。在 CMGR 中，字形模式都存储在系统属性数据库内，部件（如字形处理程序）会使用 CMGR 数据库中的字形，而不再使用 FONT1.DAT 中的字形。

安装 OpenVMS/Hanzi 期间，并不会删除 HSY\$SYSTEM 中的 FONT1.DAT。如果用户要使用存储在字形数据库中的字形，系统经理便必须使用 CONVERT DATABASE 命令来把那些字形转换至 CMGR 的系统字形数据库中。

CONVERT DATABASE 命令读取 FONT1.DAT 中的字形模式，并把已转换的字形写入当前的系统字形数据库。CMGR 有三个系统字形数据库 — CMGR\_DEFAULT、SONG 和 HEI。通过使用 SET DATABASE 命令，可把当前数据库设置为上述的其中一个数据库<sup>1</sup>。由于字形处理程序会使用 CMGR\_DEFAULT 数据库中的字形来执行按需装入，大多数用户很可能想把旧字形数据库中的字形转换至 CMGR\_DEFAULT 数据库。要做到这点，可使用以下的命令：

```
CMGR> SET DATABASE CMGR_DEFAULT
%CMGR-I-DBSET, 当前数据库设定为 "CMGR_DEFAULT"
```

---

1. 欲知详情，请参阅 4.2 节。

## 从字形系统迁移到 CMGR

### 3.2 废弃的字形系统公用程序

```
CMGR> CONVERT DATABASE
%CMGR-I-FDBCONVNG, 正在把旧数据库转换成 CMGR_DEFAULT 数据库 ...
%CMGR-I-READFIL, 正在读文件 HSY$SYSTEM:FONT1.DAT
%CMGR-S-FONTCONVTD, 已转换 10101 的字形数据
%CMGR-I-FDBCONVEND, 总共把 1 个模式转换至 CMGR_DEFAULT 数据库
```

这例子中，旧字形数据库中只找到代码为 10101<sup>2</sup> 的一个字形模式。于是，便将该字形模式转换并存储到 CMGR\_DEFAULT 数据库。

旧的字形系统只支持 24x24 大小的字形。但 CMGR 已扩充至可支持 32x32 和 40x40 大小的字形。因此，系统字形数据库可以存储所有这般大小的字形。在执行 CONVERT DATABASE 操作期间，旧字形数据库中的 24x24 字形会给转换成各种可支持其大小的字形，并存储到系统字形数据库中。

CMGR 也支持旧字形系统不支持的用户定义整理特征。CONVERT DATABASE 命令仅把旧数据库的字形转换至 CMGR 的数据库，但不会变换系统整理数据库中的整理值。因此，转换以后，用户仍要自行定义整理特征，否则，便不能正确地给包含用户定义字符的文件排序。

CONVERT DATABASE 命令的默认是 /LOG/CONFIRM。这表示在转换过程中会显示通知信息，并且，如果系统字形数据库已定义了字形模式，便会请求用户确认。

#### 3.1.2 转换字形文件

在旧的字形系统中，FDESIGN 公用程序建立字形文件以存储用户定义的字形模式。虽然自 OpenVMS/Hanzi 5.0 版本以来，便有建立预装入文件的 PRELOAD 公用程序，但用户大都把他们的用户定义字形模式只存储在字形文件中。然而，CMGR 会直接操纵预装入文件，因此不再使用字形文件。

预装入文件将会是系统属性数据库以外唯一用以存储字形模式的方法。为协助用户使用字形文件中的字形模式，CMGR 提供 CONVERT FONT\_FILE 命令来把字形文件的字形转换成预装入文件中的预装入序列。

把旧的字形系统升级到 CMGR 之后，如果用户要使用字形文件中的一些字形模式，他必须执行类似以下的 CONVERT FONT\_FILE 命令：

```
CMGR> CONVERT FONT_FILE OLD1.FON, OLD2.FON NEW.PRE
```

在这例子中，字形文件 OLD1.FON 和 OLD2.FON 中的字形会转换成预装入序列，并存储到预装入文件 NEW.PRE 中。执行转换后，用户便可使用 CMGR 的命令来操纵这包含原来存储在字形文件中的所有字形模式的预装入文件。

注意，字形文件的文件名没有默认的扩充名，因此，用户必须给他的字形文件指定正确的文件扩充名。预装入文件的默认扩充名则已定义为 ".PRE"。

命令的默认是 /LOG，因此会显示通知信息以告诉用户正在转换字符代码。

### 3.2 废弃的字形系统公用程序

所有随旧字形系统而来的公用程序都会废弃，取而代之的是 CMGR 的命令。这些公用程序包括 FDESIGN 公用程序、FUPDATE 公用程序、CTABLE 公用程序和 PRELOAD 公用程序。本节将描述哪个 CMGR 命令会取代哪个废弃的公用程序，以便熟悉旧字形系统的用户可以此作为起点，渐渐迁移到使用 CMGR。

---

2. 10101 指定扩充 GB 字符集中的第一区的第一位。4.1 节全面描述如何在 CMGR 中指定字符代码。

### 3.2.1 FDESIGN 公用程序

FDESIGN 公用程序是一个让用户建立用户定义字形模式的字形编辑程序，现由以 EDIT 命令调用的 CMGR 字形编辑程序取代。这字形编辑程序除了支持 FDESIGN 的所有特性外，还提供许多其他的特性。同时，它也支持编辑整理特征。但用户必须注意，他们需要在 VT382 终端上运行字形编辑程序。有关详情，请参阅第 6 章。

### 3.2.2 FUPDATE 公用程序

FUPDATE 公用程序更新字形数据库中的字形模式，并建立字形模式的位映象列表。在 CMGR 中，字形模式存储在预装入文件中。用户应该使用 CMGR 的 UPDATE 命令在系统数据库中更新其字形。如果用户想用字形文件中的一些字形来更新系统字形数据库，他可用 CONVERT FONT\_FILE 命令来把字形文件转换为预装入文件，然后使用 UPDATE 命令。

要显示字形模式的位映象列表，用户可使用 CMGR 的 SHOW BITMAP 命令。该命令显示预装入文件和系统属性数据库的位映象。

### 3.2.3 CTABLE 公用程序

CTABLE 公用程序可用以生成字符代码表，展示在字形数据库中定义的字符。这公用程序现由 SHOW TABLE 命令取代，可显示所有属性表。用户也可以使用 /FONT 限定词来显示只包含字形的表。这限定词所生成的输出跟 CTABLE 公用程序的输出相类似。此外，SHOW TABLE 也显示系统属性数据库和用户属性文件的属性表。

### 3.2.4 PRELOAD 公用程序

PRELOAD 公用程序生成预装入文件以便预装入到终端。在 CMGR 中，EXTRACT CODE 和 EXTRACT REFERENCE 命令提供这方面的功能。它们从系统属性数据库抽取字形模式到预装入文件。EXTRACT CODE 命令从命令行指定的字符代码中抽取字符属性；EXTRACT REFERENCE 命令则为指定的一列表文本文件中的用户定义字符抽取字符属性。



## 第 4 章 使用 CMGR

本章会示范如何选择一数据库作为当前数据库、如何建立一些用户定义字符、如何把对应的属性文件更新到当前的系统属性数据库中，并如何建立字符属性列表文件以供参考。然后，会简要描述其他 CMGR 命令，如 REMOVE、EXTRACT 和 CONVERT FONT\_SIZE。本章不会谈及诸如 CONVERT DATABASE 和 CONVERT FONT\_SIZE 的命令。这些命令在第 3 章中已有描述。用户应参阅第 5 章的命令参考以了解每个 CMGR 命令的详细描述。

### 4.1 CMGR 中的字符代码和区说明

字符代码由区位码或字符的内码指定。在 CMGR 中，内码以十六进制代码表示，首个字符为“H”。例如，如果具有两个八位字节的用户定义字符的第一个字节等于 161，而使它的第二个字节等于 33（两者均为十进制），它的内码便应为“HA121”。此外，这代码是在 DEC 汉字字符平面内左下象限的第一区中。在图 2-4 中，用户可看到用户定义区域是从“HA121”至“HF77E”，而保留用户定义区域是从“HF821”至“HFE7E”。

另一个普遍地用以指定字符代码的方法是使用它的“区位”。DEC 汉字字符平面的每个象限可以再分成 94 区，每区有 94 个位置（位）。图 2-3 和图 2-4 所示，每一区对应于象限中的一行。因此，第一行便是第 1 区，而最后一行是第 94 区。要区分在两个象限的代码，右下象限中的区位码以“0”开头，左下象限中的区位码则以“1”开头。如果不是以“0”或“1”开头，默认为“0”（右下象限）。

因此，区位码“00101”或“0101”等于内码“HA1A1”。区位码“10101”等于“HA121”。注意，如果区号或位号是单个数字，便会添加一个零。

如果使用区位，用户定义区域的范围是从 10101 至 18794，而保留用户定义区域则从 18801 至 19494。

用户必须注意，区位码输入方法与 VT382 终端所提供的输入方法相同。

区号是以 /QUADRANT 限定词来指定的。限定词可取 LOWER\_RIGHT、LR<sup>1</sup> 或 0 作为值来指定右下象限。也可取 LOWER\_LEFT、LL<sup>2</sup> 或 1 作为值来指定左下象限。/QUADRANT 的默认值是 /QUADRANT=LOWER\_LEFT。

有关象限和平面的详情，请参阅本手册的第 2 章或 Open VMS/Hanzi 用户手册的第 2 章。

### 4.2 选择当前数据库

CMGR 共有三个系统属性数据库，即 CMGR\_DEFAULT、SONG 和 HEI 数据库。在任何时候，其中一个数据库会成为当前数据库，CMGR 命令便在该当前数据库上操作。除非系统经理另行设置<sup>3</sup>，否则注册后的当前数据库应为默认数据库 CMGR\_DEFAULT。用户可以使用下述命令来转换到另一个数据库：

```
CMGR> SET DATABASE HEI
```

1. LR 是 LOWER\_RIGHT 的缩写。

2. LL 是 LOWER\_LEFT 的缩写。

3. 欲知详情，请参阅第 5 章中有关 SET DATABASE 命令的描述。

## 使用 CMGR

### 4.3 使用 CMGR 字形编辑程序

这命令把当前数据库设置为 HEI。

要转换回默认数据库，用户可以使用：

```
CMGR> SET DATABASE/DEFAULT
```

要展示当前数据库是哪一个，用户可以使用：

```
SHOW DATABASE
```

或者，如果要知道系统中所有数据库的名称，可使用下列命令：

```
CMGR> SHOW DATABASE/ALL
```

```
数据库名
```

```
-----
```

```
CMGR_DEFAULT
```

```
SONG
```

```
-> HEI
```

注意，在数据库名左边的箭头指明该数据库是当前数据库。

在选择一个数据库作为当前数据库后，用户便可开始使用其他 CMGR 命令来操纵该数据库。

### 4.3 使用 CMGR 字形编辑程序

#### 第 1 步：

在 CMGR 中输入以下命令以调用编辑程序。由于 EDIT 命令的默认限定词是 /FULL/SIZE=ALL/LOG/LOAD\_GLYPH，因此允许您编辑用户定义字符的所有字符属性。请注意，由于 CMGR 会自动添加 .PRE 和 .CVD 两个默认扩充名到两个输出文件，因此不得在 /FULL 限定词中使用文件扩充名。

```
CMGR> EDIT TEMP
```

然而，用这命令来调用的字形编辑程序不允许用户编辑保留用户定义区域的属性。如果需要编辑或使用保留用户定义字符，应该发出以下命令：

```
CMGR> EDIT/RESERVED TEMP
```

如果不需要编辑整理特征，输入以下命令：

```
CMGR> EDIT/FONT TEMP
```

这样，便撤销所有与整理特征有关的特性，只建立一个输出文件 -- TEMP.PRE。

由于默认是 /SIZE=ALL，因此用户可编辑任何一种可支持其大小的字形，如 24x24、32x32 和 40x40。如果不需要编辑 32x32 和 40x40 大小的字形，便应使用下列命令：

```
CMGR> EDIT/FONT/SIZE=24 TEMP
```

在启动编辑程序时，编辑程序会装入一些字块模式以作显示。除非曾重置终端或关掉电源，否则只需装入一次。为节省时间，用户可在第二次（或以后）启动编辑程序时指定 /NOLOAD\_GLYPH。

假定没有 TEMP.PRE 和 TEMP.CVD，编辑程序便会在 EXIT 编辑程序时建立这两个文件。如果用户选择放弃编辑程序，便不会建立输出文件。如果目录已包含其中一个或这两个文件，编辑程序便把文件的内容读入内存（称为内存记录）以供编辑。在 EXIT 时，便给这些文件建立新版本。

调用编辑程序以后，便会在屏幕上显示象图 1-1（或图 1-2，取决于语言设置）所示的屏幕显示。用户可在任何时间按 <PF2> 以取得关于小键盘的求助资料。

小键盘

```

<===== 编辑小键盘 =====> <===== 应用小键盘 =====>
+-----+-----+-----+ +-----+-----+-----+
|      |      |      | | Gold | Help | Output | Delete |
|      | Paste | Cut  | |      |Key Help|Curs Mo.|Past Mo. |
+-----+-----+-----+ +-----+-----+-----+
|Sel. All|      |      | |ULcorner| Top |URcorner| Cancel |
| Select |Prv Size|Nxt Size| | U&L  | Up  | U&R  | Type Mo. |
+-----+-----+-----+ +-----+-----+-----+
|      |      |      | | L side | Center | R side | Quit  |
|      | Up  |      | | Left  | Toggle | Right | Wrap Mo. |
+-----+-----+-----+ +-----+-----+-----+
|      |      |      | |LLcorner| Bottom |LRcorner|      |
| Left  | Down | Right | | D&L  | Down  | D&R  | Exit  |
+-----+-----+-----+ +-----+-----+-----+
|      |      |      | | Return from USE | REFER | Enter |
|      |      |      | |      USE      | DISPLAY | Code  |
+-----+-----+-----+

```

Additional information available:  
Application-keypad    Editing-keypad        Control-keys  
Main-keypad  
KEYPAD Subtopic?

**第 2 步：**

在编辑字形之前，用户必须输入字符代码以作编辑。字符代码以区位码或内码格式指定（第 4.1 节）。按 <ENTER> 来输入字符代码（见图 4-1）。如果命令行中没用指定 /RESERVED，用户只可输入用户定义区域内的代码，否则会有出错信息显示出来。如果指定了 /RESERVED，便可同时使用用户定义和保留用户定义字符代码。

有关用户定义字符的有效代码范围的详情，请参阅第 2 章或在 CMGR 输入以下的命令：

```

CMGR> show character_set
DEC GB-2312 汉字字符集

```

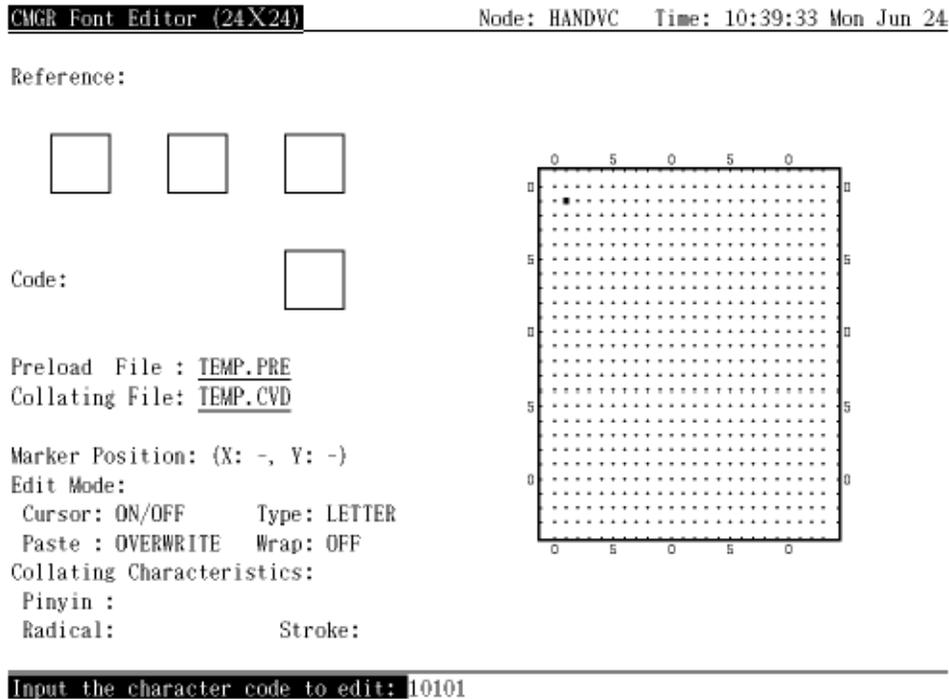
代码范围	象限	区范围	字符种类
00101 - 00994	LR	1 - 9	DEC GB-2312 字符
01001 - 01594	LR	10 - 15	DEC GB-2312 保留字符
01601 - 08794	LR	16 - 87	DEC GB-2312 字符
08801 - 09494	LR	88 - 94	DEC GB-2312 保留字符
10101 - 18794	LL	1 - 87	用户定义字符

## 使用 CMGR

### 4.3 使用 CMGR 字形编辑程序

18801 - 19494 LL 88 - 94 保留用户定义字符

图 4-1 按 <Enter> 输入代码



#### 第 3 步：

用户现在可以使用小键盘、功能键、或控制键的各种功能来执行编辑。用户应注意所有 EDIT 态的默认设置：光标态的 ON/OFF、贴补态的 OVERWRITE、字符类型的 LETTER 和绕回态的 OFF。这设置会影响编辑功能的大部分操作。用户可按小键盘功能键来更改任何一个 EDIT 态，如按 <PF3> 来更改光标态。

#### EDIT 态：

##### 1 光标态 -

有四种光标态：ON/OFF、ON、OFF 和 MOVE。ON/OFF 态互换光标移过的象素。用户可利用应用小键盘来移动光标。ON 和 OFF 态分别用以开关光标移过的象素。MOVE 态仅允许光标移动，对象素没有任何影响。

光标态也影响画几何图形。详情请参阅 6.6.10 节。

## 2 贴补态 -

贴补态可以是 OVERWRITE 或 OVERLAY。OVERWRITE 贴补态把贴补缓冲区重写到现有的像素上，而 OVERLAY 贴补态则贴补到现有的像素上而不清除原来的像素。

贴补态也影响画几何图形。详情请参阅 6.6.10 节。

## 3 字符类型 -

在 CMGR 中定义了两种字符类型，即 LETTER 类型和 BODY 类型。LETTER 类型不允许用户在位映象的最外边界（24x24 大小的是一个像素，而 32x32 及 40x40 大小的是两个像素）编辑。BODY 类型则允许用户编辑整个位映象而没有任何限制。

## 4 绕回态 -

绕回态可以是 ON 或 OFF。它决定在光标移动超过边界时是否绕回。

用户可从当前的系统属性数据库中取出位映象字形模式来执行 CUT 功能，然后使用 PASTE 功能把贴补缓冲区（里面包含从字形剪切出来的部分）插入正编辑的字形模式。USE 功能有助于执行这操作。

如果用户要使用的字形存储于默认数据库以外的另一个数据库内，在启动编辑程序之前必须执行以下命令：

```
CMGR> SET DATABASE HEI
```

此命令把当前数据库设置为 HEI，随后的 CMGR 命令都将在 HEI 数据库上操作（在这情形下，编辑程序的 USE 功能将从 HEI 数据库读取字形）。

要调用 USE，用户应按 <KP0> 键并输入关键字（任选的）和用户定义字符代码（或如果在命令行中指定了 /RESERVED，亦可输入保留用户字符代码）。关键字可以是“DATABASE”或“MEMORY”<sup>1</sup>，默认是“DATABASE”。它指定字形模式的源。如果关键字是“DATABASE”或没有指定，CMGR 会从当前系统属性数据库中检索指定的用户定义字形模式。如果关键字是“MEMORY”，CMGR 便会从内存记录<sup>2</sup>取出字形。

然后，用户便可执行 SELECT 和 CUT 操作，把合适的字形部分放进贴补缓冲区。最后，按 <GOLD + <KP0> 返回原来的位映象，并把该部分贴补（按 <INSERT HERE>）到当前编辑的位映象。在执行 PASTE 操作之前，用户必须按光标移动功能键（如 <GOLD> + <KP7>），把光标移到适当的位置。

图 4-2、图 4-3、图 4-4、图 4-5、图 4-6 和 图 4-7 展示每一操作的屏幕显示。字形是从 HEI 系统属性数据库取出的。

图 4-8 和图 4-9 展示字形从内存记录取出时的情形。

用户可以执行类似的操作来组合最终的字形模式，或使用编辑程序功能来建立新的字形模式。完成字形模式后或在编辑期间，用户可以按 <KP.> 来在 DISPLAY 窗口以实际大小显示字形（见图 4-10）。

此外，用户可能希望看到其他字符代码的实际大小字形。REFER 可提供这方面功能，它最多可显示当前系统属性数据库或内存记录中的三个字形。按 <GOLD> + <KP.>，便有提示要求提供字符代码，用类似 USE 功能那样的关键字和字符代码来回答，字形便在其中一个 REFER 窗口内显示出来（见图 4-11 和图 4-12）。

---

1. 关键字可以缩写并可以用小写。

2. 在启动编辑程序时从预装入文件装入的字形和在编辑对话期间建立的字形都会存储到内存。

## 使用 CMGR

### 4.3 使用 CMGR 字形编辑程序

字形编辑程序也提供画几何图形或重新编排字形模式的功能。例如，用户可以首先 SELECT 一个区域并按 <O> 来画一个椭圆。图 4-13 和 图 4-14 展示此例子。

CMGR 编辑程序提供 MARK 功能以协助编辑。用户按 <GOLD> + <M> 便可在当前光标位置作标记。标记座标会展示出来。然后，用户可以把光标移到任何位置。如有需要，便可在任何时候按 <GOLD> + <G> 来跳回到标记的位置。图 4-15、图 4-16、图 4-17、图 4-18 和图 4-19 展示使用标记功能来画三角形的例子。

有关详细的编辑程序功能列表和描述，请参阅第 6 章。

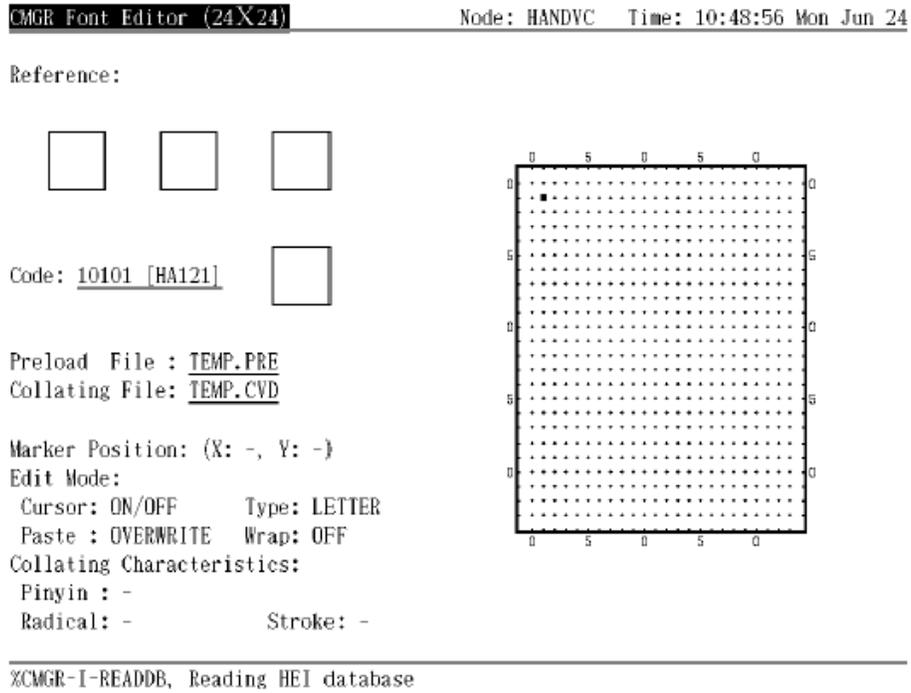
图 4-2 按 <KP0> 调用 USE 功能

```
CMGR Font Editor (24X24) Node: HANDVC Time: 10:47:56 Mon Jun 24
Reference:
  [ ] [ ] [ ]
Code: 10101 [HA121] [ ]
Preload File : TEMP.PRE
Collating File: TEMP.CVD
Marker Position: (X: -, Y: -)
Edit Mode:
  Cursor: ON/OFF      Type: LETTER
  Paste : OVERWRITE  Wrap: OFF
Collating Characteristics:
  Pinyin : -
  Radical: -          Stroke: -
Input the character code to use: 12101
```

1

1. 假定 HEI 系统属性数据库包含用户定义字符 12101 的字形模式。

图 4-3 从 HEI 数据库装入字形模式



# 使用 CMGR

## 4.3 使用 CMGR 字形编辑程序

图 4-4 按 <SELECT> + <KP3> X 6 + <KP6> X 11 选择区域

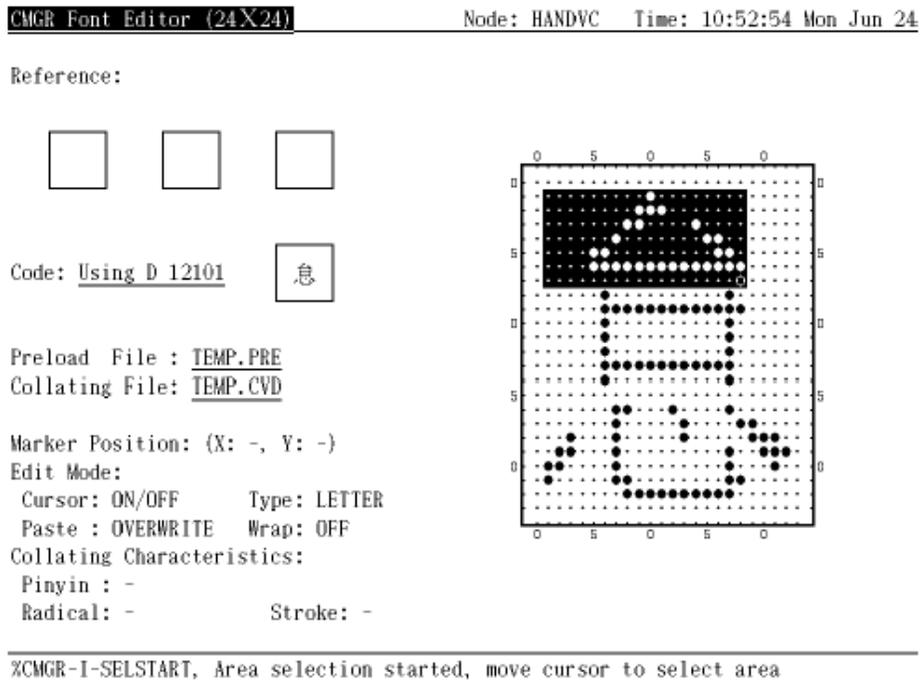
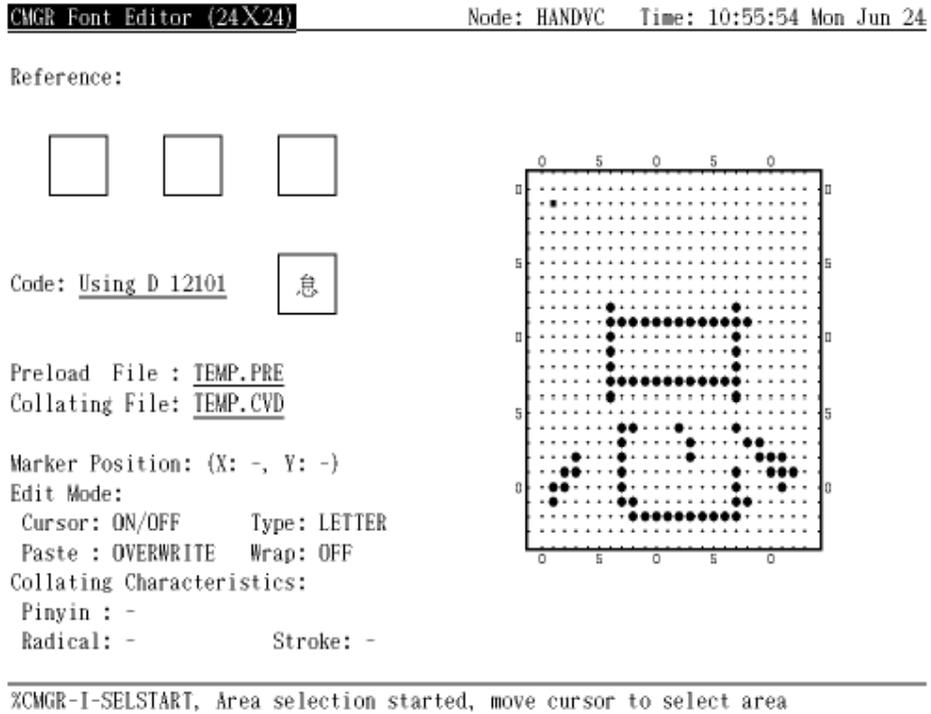


图 4-5 按 <REMOVE> 把选定部分剪切到 Paste 缓冲区



# 使用 CMGR

## 4.3 使用 CMGR 字形编辑程序

图 4-6 按 <GOLD> + <KP0> 返回到原来的位映象

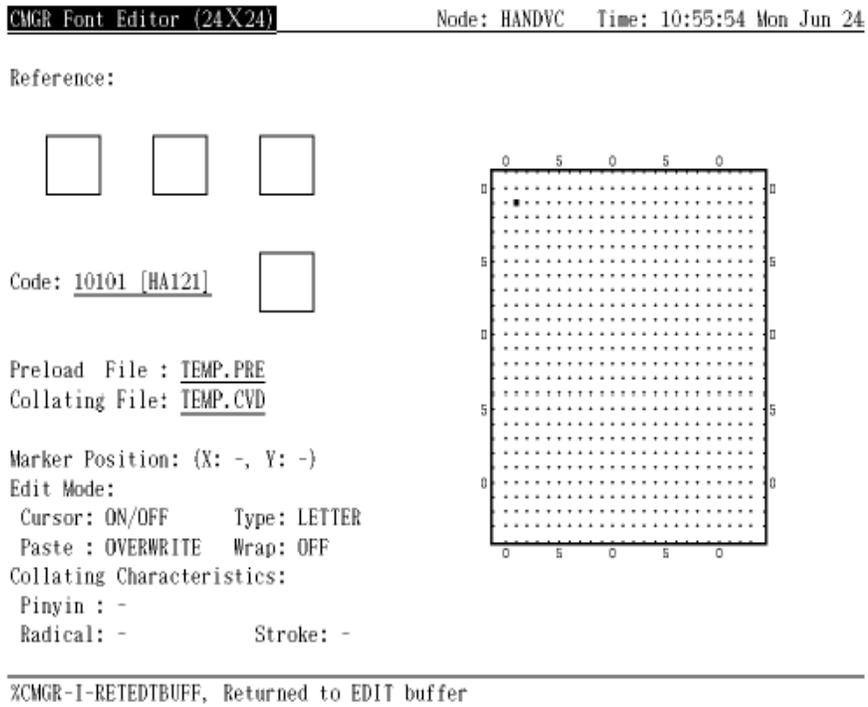
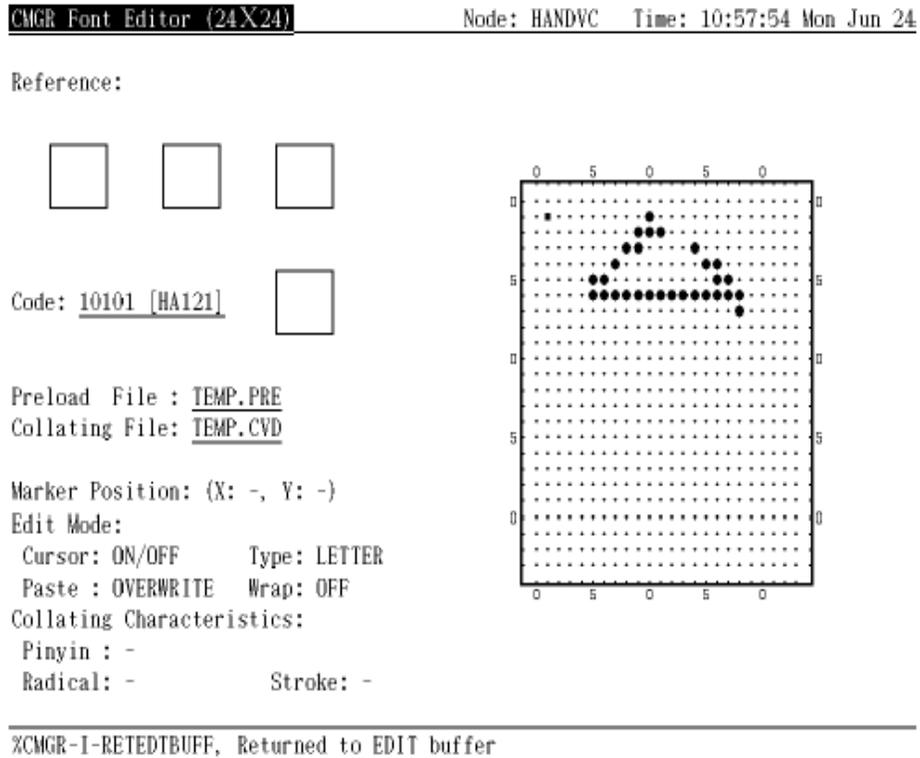


图 4-7 按 <GOLD> + <KP7> 前去适当的位置并按 <INSERT HERE> 贴补该 Paste 缓冲区



# 使用 CMGR

## 4.3 使用 CMGR 字形编辑程序

图 4-8 键入关键字“MEMORY”使用内存记录

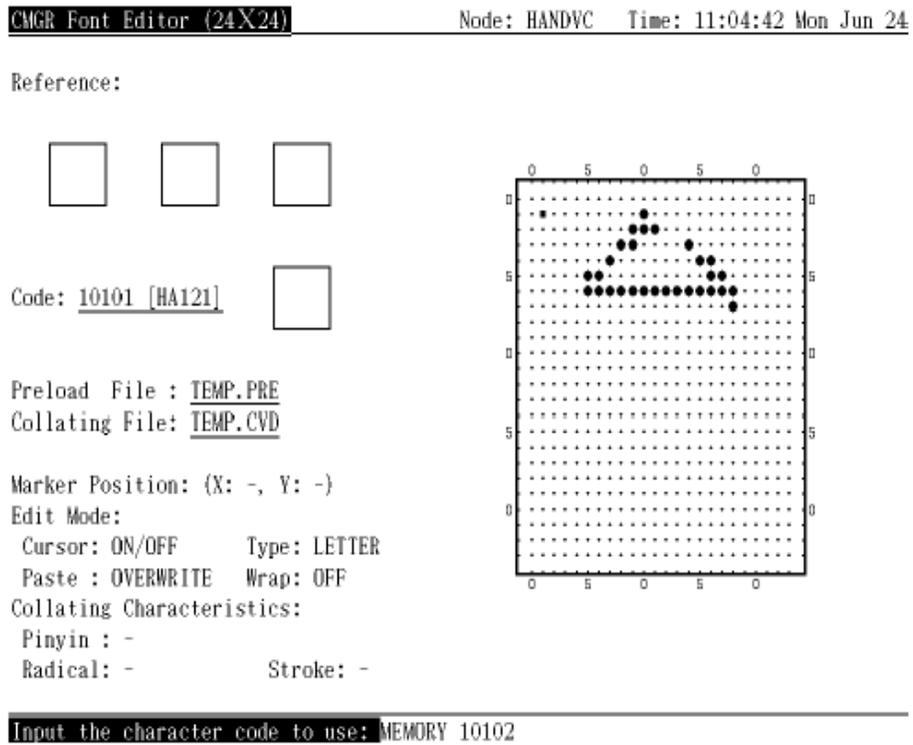
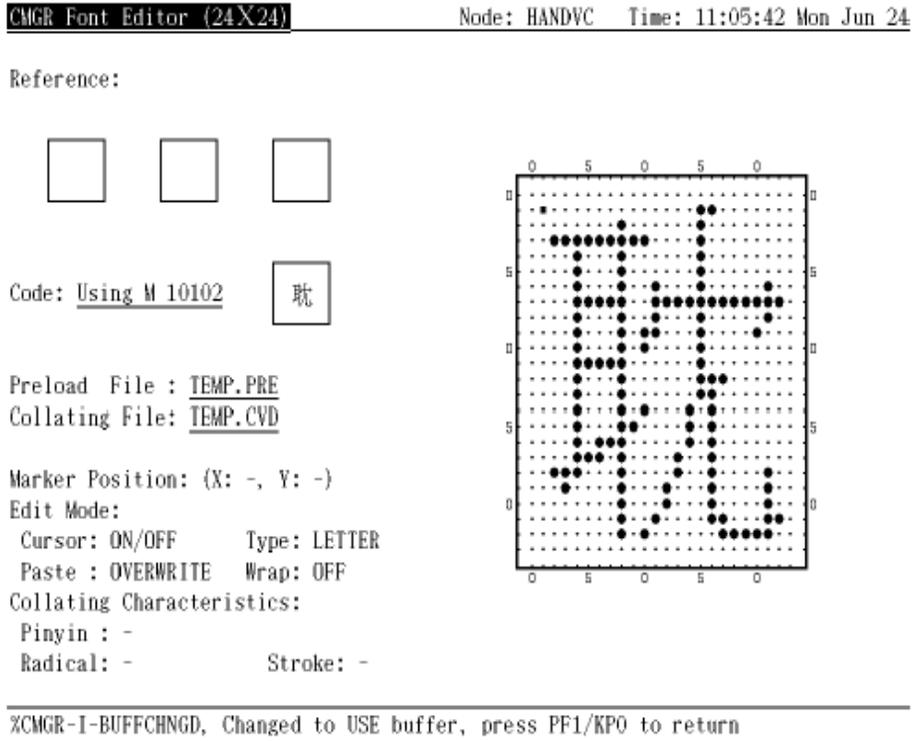


图 4-9 内存字形装入 USE 缓冲区



# 使用 CMGR

## 4.3 使用 CMGR 字形编辑程序

图 4-10 按 <KP.> 以实际大小显示当前字形

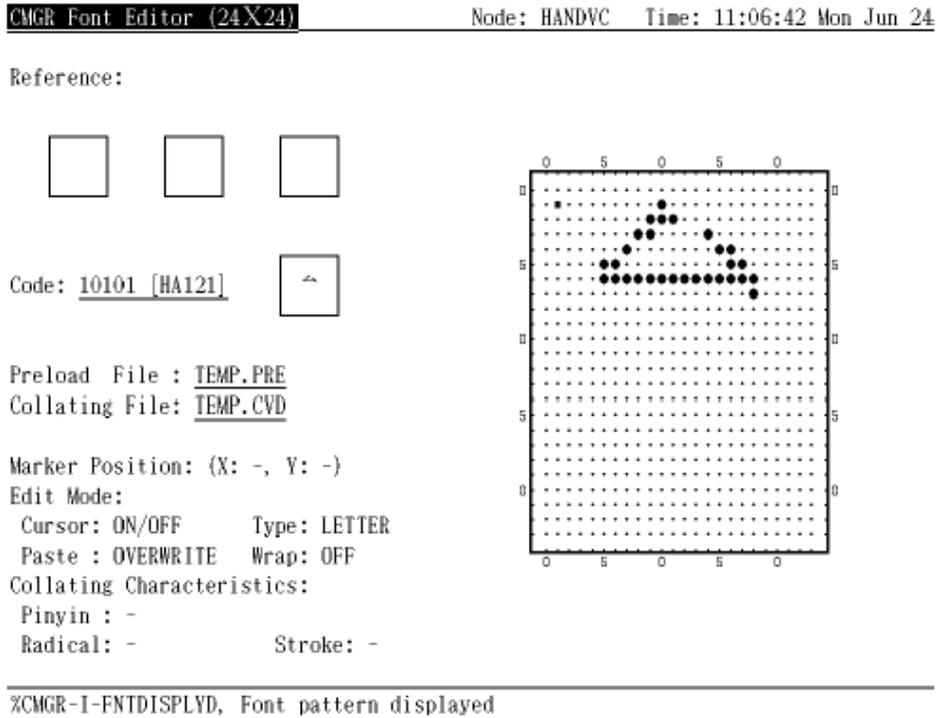
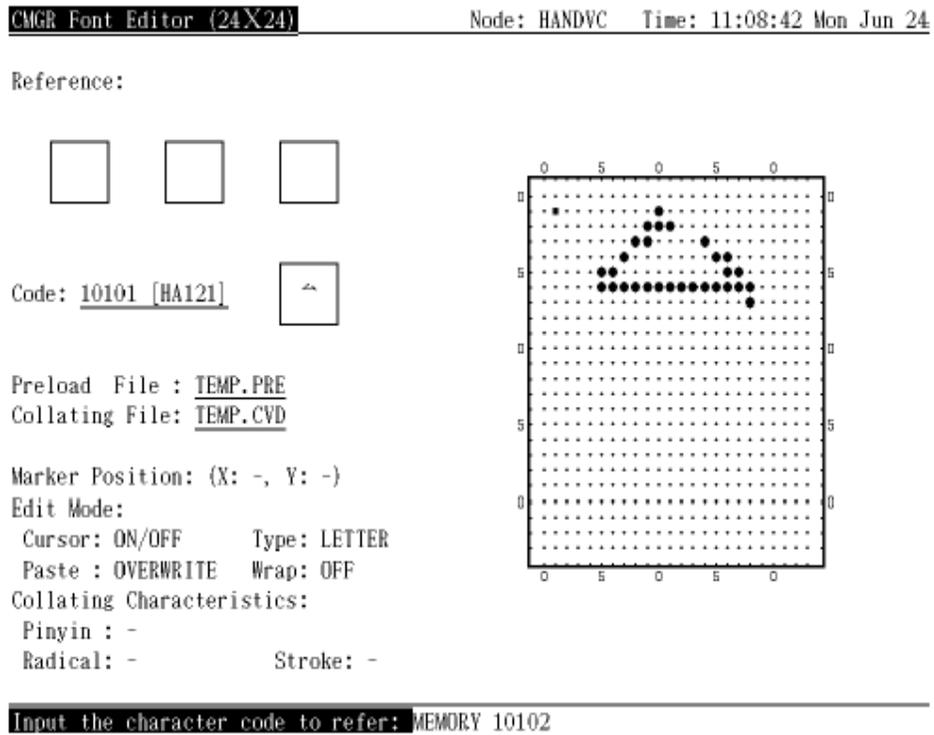


图 4-11 按 <Gold> + <KP.> 并键入 "MEMORY 10102" 以在其中一个 REFER 窗口显示内存字形



## 使用 CMGR

### 4.3 使用 CMGR 字形编辑程序

图 4-12 内存记录的 10102 字形显示在最左面的 REFER 窗口中

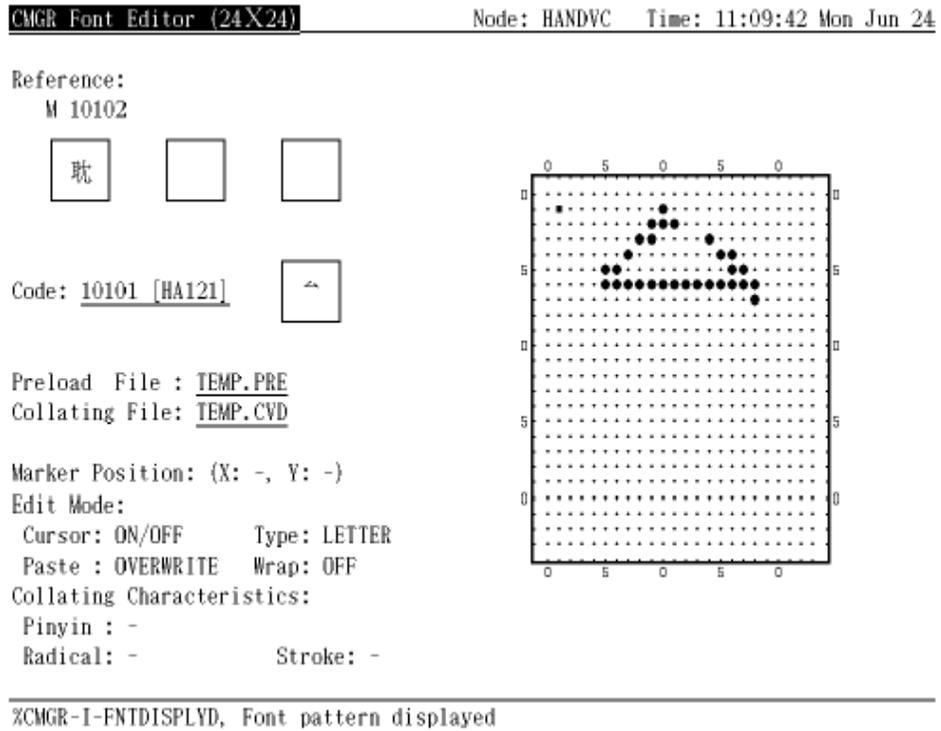
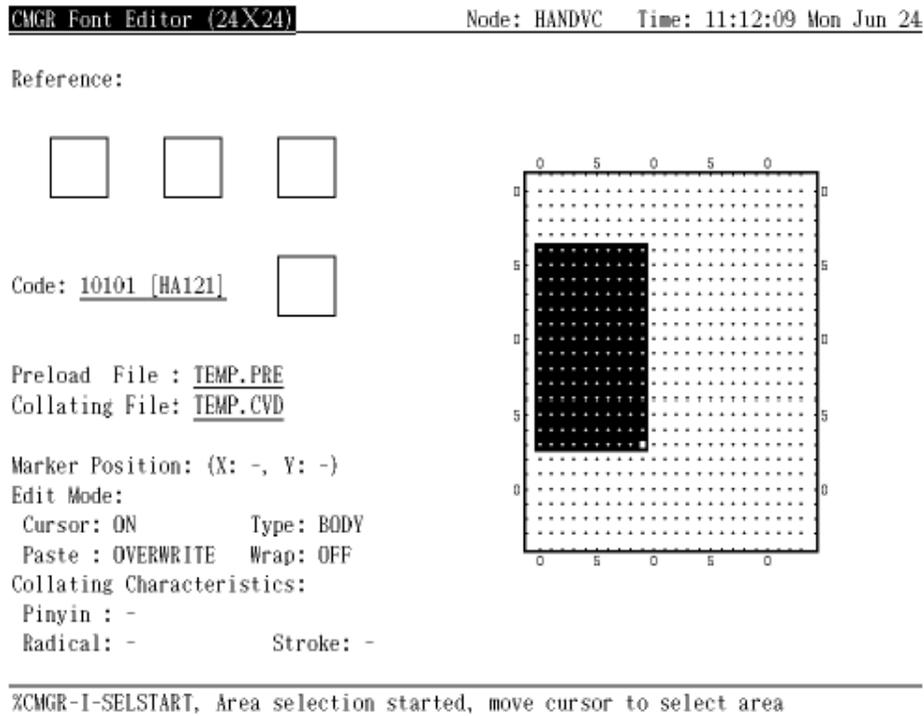


图 4-13 按 <0> 之前首先选择区域



1

1. 假定编辑态“类型”已设置为“BODY”以允许在边界象素上编辑。

# 使用 CMGR

## 4.3 使用 CMGR 字形编辑程序

图 4-14 按 <O> 画椭圆

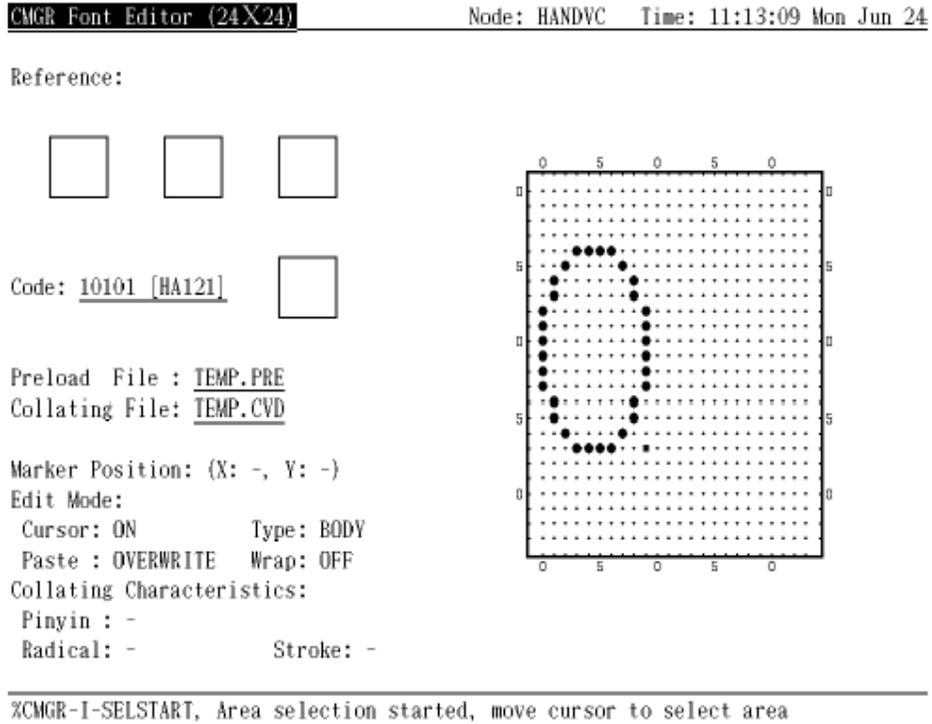
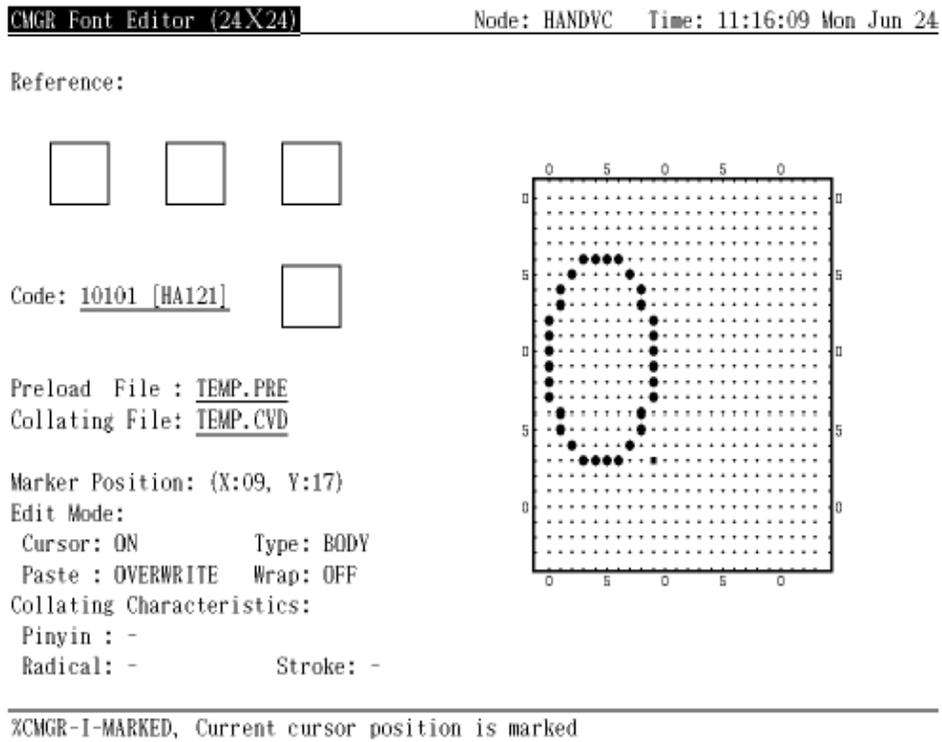


图 4-15 按 <Gold> + <M> 标记当前光标位置



## 使用 CMGR

### 4.3 使用 CMGR 字形编辑程序

图 4-16 使用 <KP6> 和 <KP8> 画三角形的两边

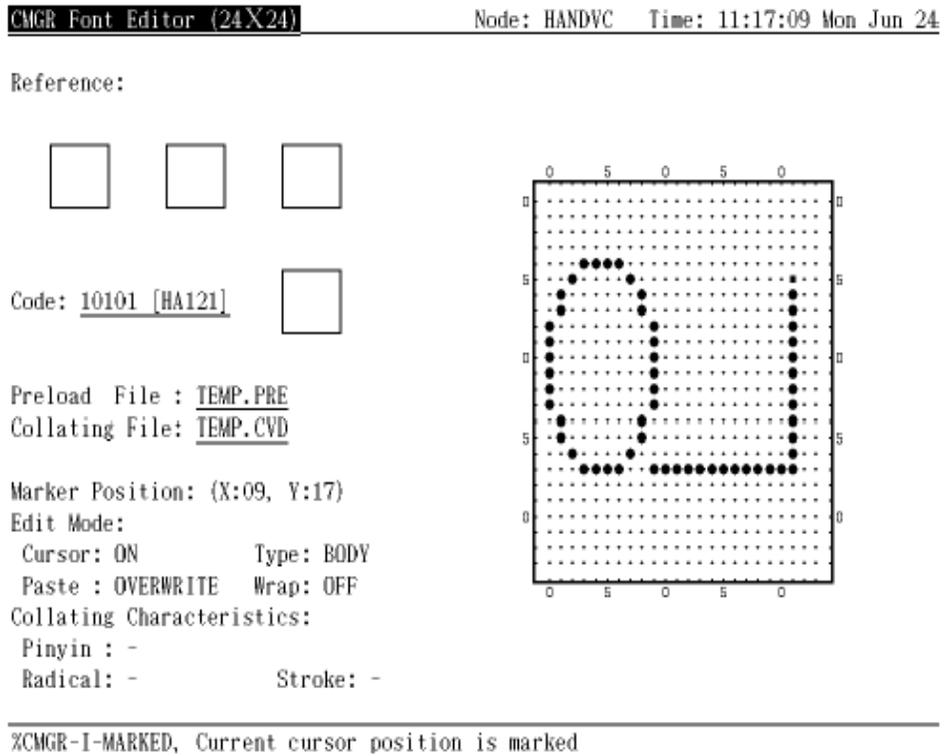
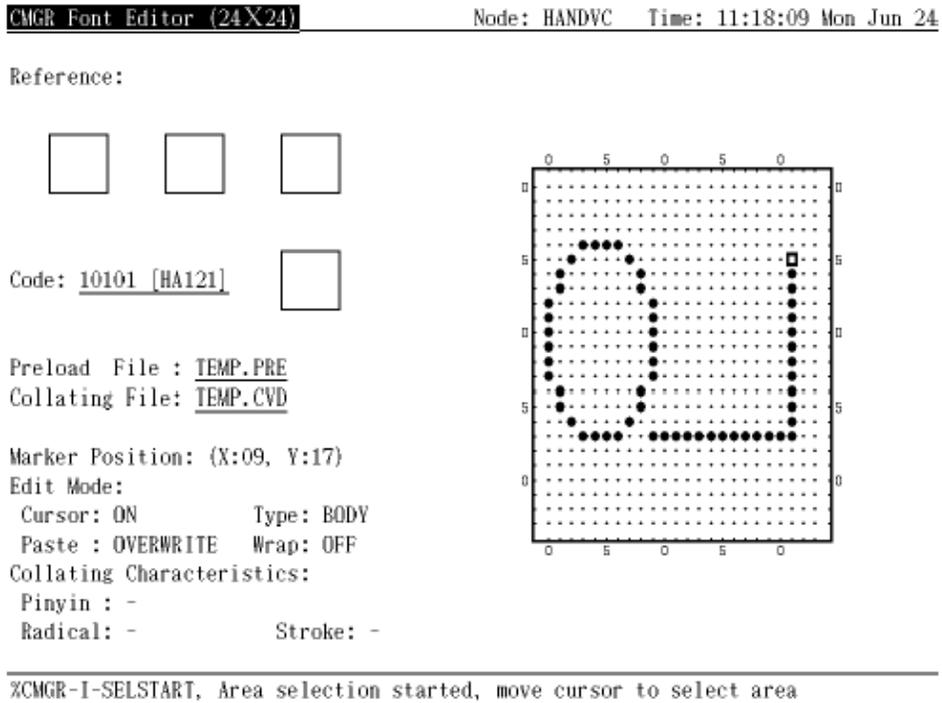


图 4-17 按 <Select> 开始选择



## 使用 CMGR

### 4.3 使用 CMGR 字形编辑程序

图 4-18 通过按 <Gold> + <G> 完成选择，光标会直接跳到标记位置

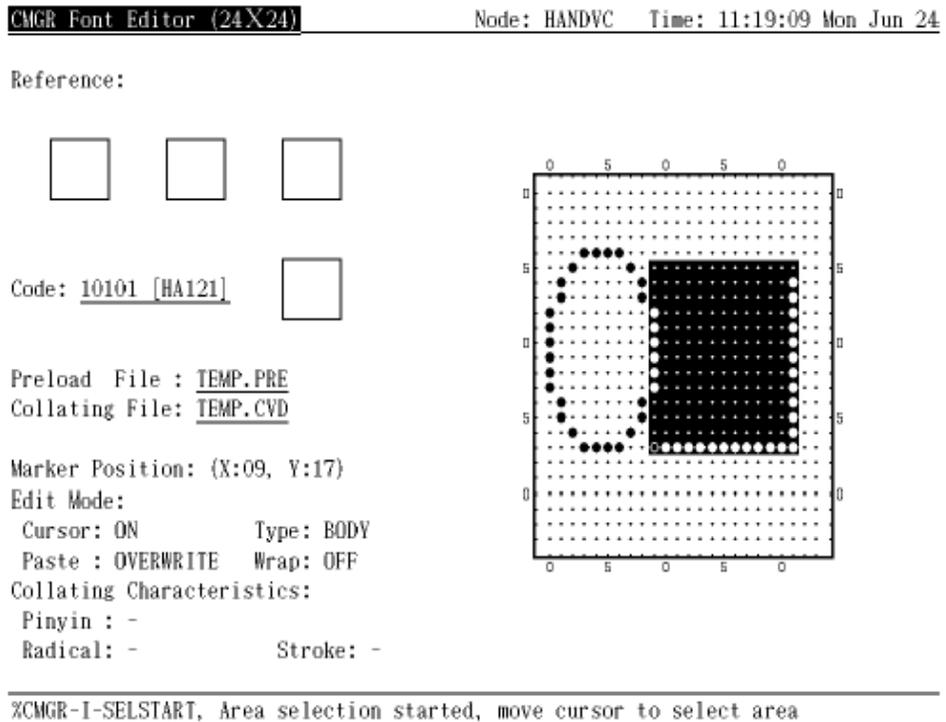
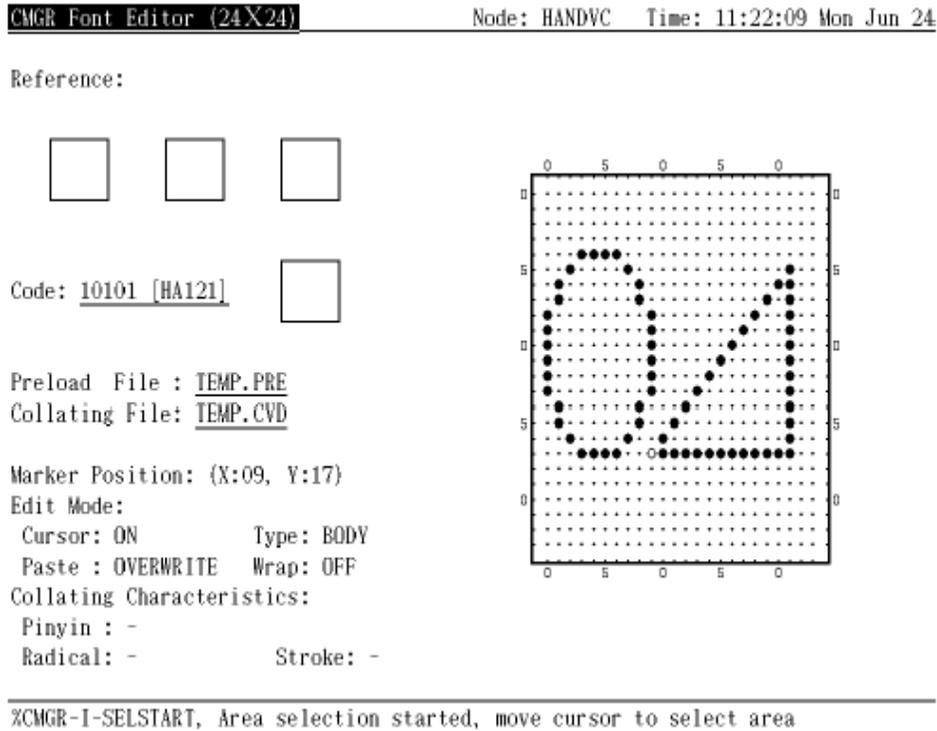


图 4-19 按 <L> 完成三角形



#### 第 4 步：

在 24x24 大小的字形模式的编辑完成之后，用户可能希望编辑其他大小的字形。要这样做，可按 <Next Screen>，编辑程序的编排设计便更改成图 4-20 那样。现在，用户可以编辑 32x32 大小的字形。

大小的次序是 24x24、32x32 和 40x40。按 <Next Screen> 便从 24x24 大小向前去到 40x40 大小，然后再回到 24x24 大小。按 <Prev Screen> 便从 40x40 大小反向前去到 24x24 大小，然后再回到 40x40。用户可使用这两个键来转换到任一个大小并编辑该大小的字形。

为了帮助用户建立不同大小的字形，CMGR 提供了一个转换大小的功能。用户可以把同一代码另一大小的字形转换成当前大小。例如，在使用 <Next Screen> 来转换成 32x32 大小之后，按 <F18>，便有提示询问转换前的大小。输入 24 并按 <Return>，24x24 大小的字形便会转换成 32x32 大小并装入 EDIT 缓冲区。图 4-21 和图 4-22 说明整个过程。

# 使用 CMGR

## 4.3 使用 CMGR 字形编辑程序

图 4-20 用于编辑 32x32 大小字形的屏幕编排设计

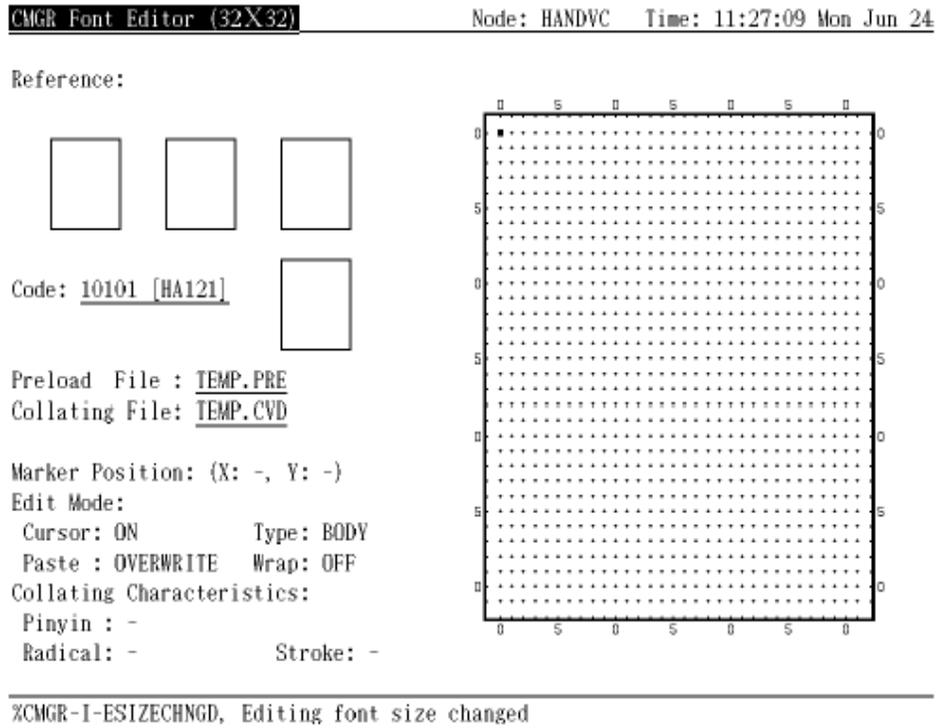
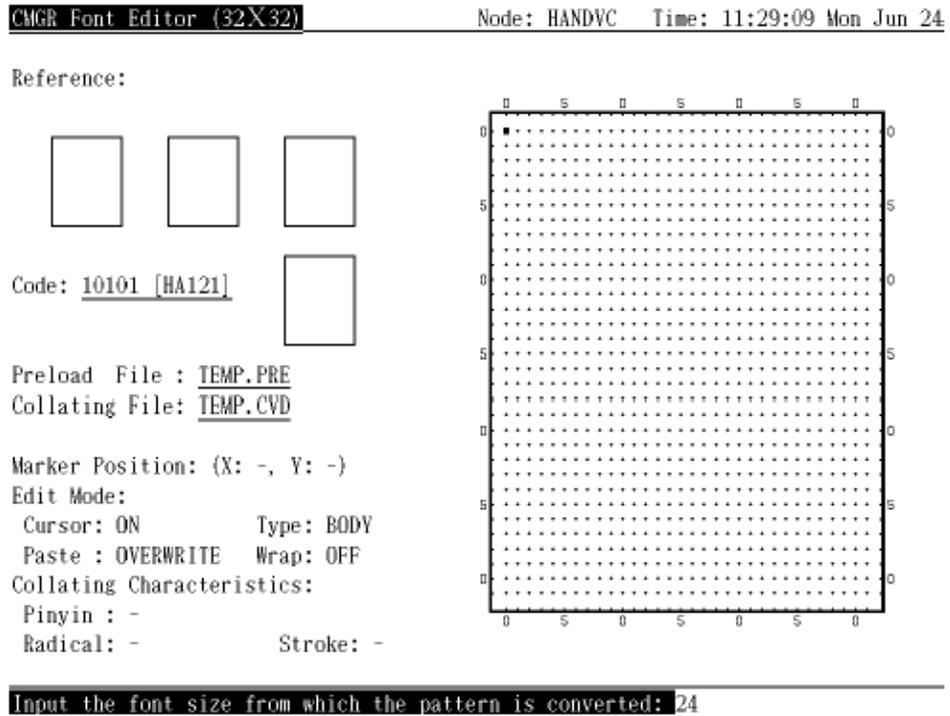


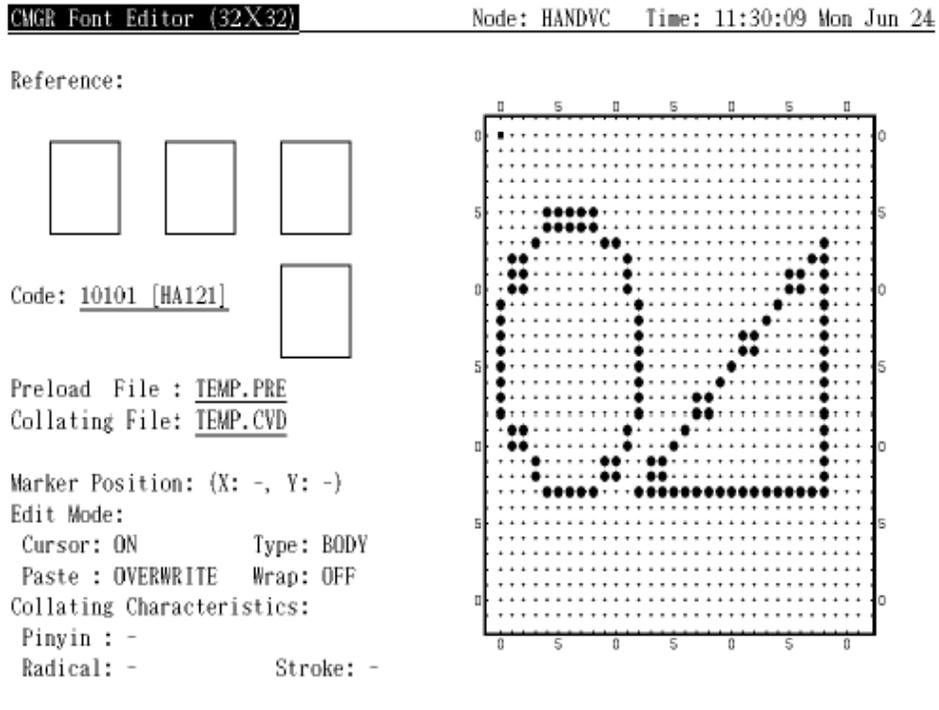
图 4-21 按 <F18> 并键入 24



## 使用 CMGR

### 4.3 使用 CMGR 字形编辑程序

图 4-22 把 24x24 大小的字形转换成 32x32

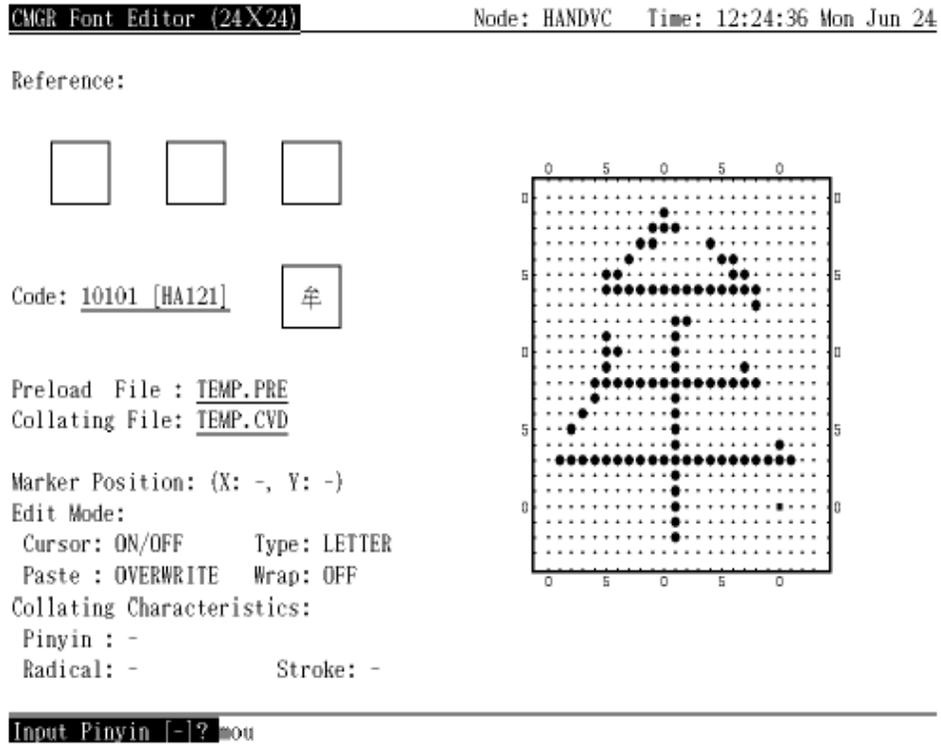


#### 第 5 步：

编辑字形后，按 <F17> 以输入整理特征。如果用户需要排序 / 合并字符，此步骤是必需的。用户会收到提示要求他输入每个整理特征的值（图 4-23）。输入所有整理特征后，便会象图 4-24 所示的那样显示出来。

参阅附录 A 以了解有效整理值的完整列表。

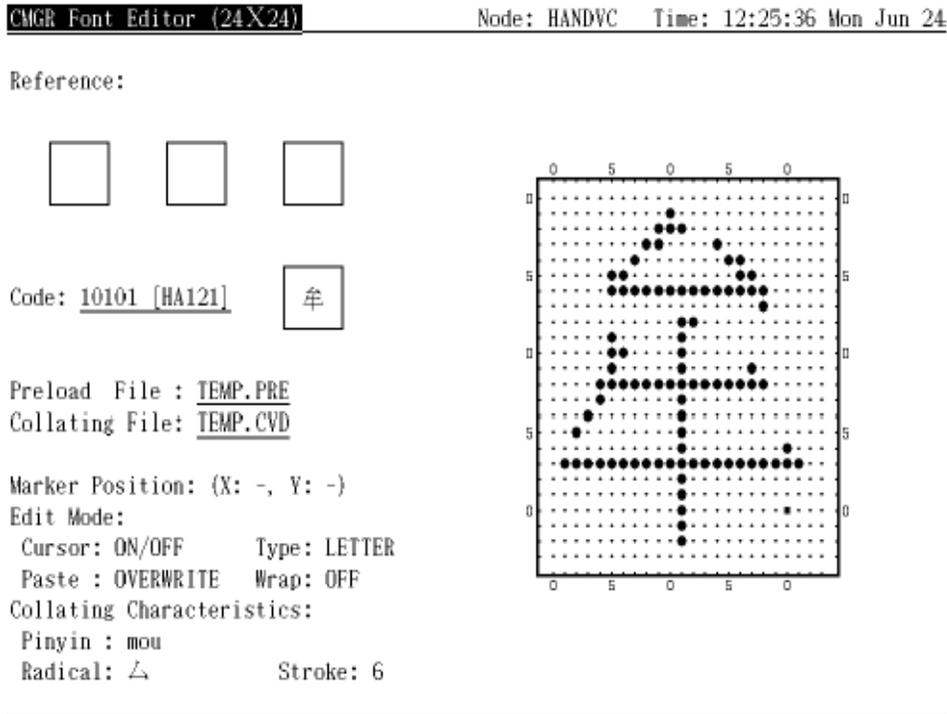
图 4-23 按 <F17> 输入整理特征



## 使用 CMGR

### 4.3 使用 CMGR 字形编辑程序

图 4-24 显示的整理特征



#### 第 6 步：

用户完成编辑后，他可以选择：

- 按 <ENTER> 来把当前字形保存到内存记录，并编辑另一个字形。
- 按 <GOLD> + <KP-> 来取消当前的编辑，自上次按 <ENTER> 以来的更改都会丢失。例如，用户先建立代码为 10101 的字形，然后编辑代码为 10102 的字形。之后，他回去编辑 10101 字形，但对该字形作了一些更改之后，他按 <GOLD> + <PF->。他所作的这些更改都会丢失，但原来的内存字形模式则不受影响。
- 按 <GOLD> + <PF4> 来删除内存记录，当前字符代码的所有数据都会丢失。
- 按 <GOLD> + <KP,> 来放弃编辑程序，而不把驻留在内存中的当前编辑结果写入文件。
- 按 <GOLD> + <ENTER> 来退出编辑程序，并把驻留在内存中的编辑结果保存到文件。

请参阅第 6 章以了解更多选项。

假定再编辑了两个用户定义字符 10102 和 10103，然后输入 EXIT (<GOLD> + <ENTER>)，便产生 TEMP.PRE 和 TEMP.CVD 两个输出文件。这两个输出文件保存了用户的编辑结果（TEMP.PRE 保存位映象字形模式，而 TEMP.CVD 保存整理特征）。之后，编辑对话期便结束，用户也返回 CMGR> 提示符。

## 4.4 CMGR UPDATE 命令

在产生用户属性文件 (TEMP.PRE 和 TEMP.CVD) 以后, 用户便可以用下述命令以这两个文件来更新当前的系统属性数据库:

```
CMGR> UPDATA TEMP
```

默认限定词是 /FULL/SIZE=ALL/CONFIRM=ALL/LOG, 意思是会按输入文件更新用户定义字符的所有属性, 并要求确认将要执行的所有更新。实际上, 用户可以输入 /NOCONFIRM 来撤销确认, 或输入 /CONFIRM=CONFLICT 来表示只在当前系统属性数据库中已有对应字符代码的属性时才需要确认。此外, 用户可以指定 /NOLOG 来撤销在更新操作期间显示通知信息。请注意, 此命令必须有 SYSPRV 特权才可执行。

与 EDIT 命令相同, 使用 /FULL 限定词时不能指定文件扩充名, 原因是 CMGR 会自动添加这两个扩充名。这说明这命令会以 TEMP.PRE 和 TEMP.CVD 来更新当前的系统属性数据库。

如果用户属性文件包含保留用户定义字符的属性, 用户便必须使用 /RESERVED 限定词, 否则, 更新操作便会失败。

如果用户只想更新预装入文件, 可输入:

```
CMGR> UPDATE/FONT TEMP
```

如果用户只要更新整理特征, 可输入:

```
CMGR> UPDATE/COLLATING TEMP
```

以下展示该 UPDATE 操作 (假定当前系统属性数据库是 CMGR\_DEFAULT 并且已包含字符 10102 的两个属性)。

```
CMGR> UPDATE TEMP
%CMGR-I-UPGFont, 正在用文件 DISK$:[USER]TEMP.PRE;1 更新 CMGR_DEFAULT
数据库的字形模式
更新代码 10101 的 24x24 大小的字形模式吗? [N]: Y
%CMGR-I-FONTUPD, 已更新代码 10101 的 24x24 大小的字形模式
%CMGR-I-FONTEXIST, 代码 10102 的 24x24 大小的字形模式已存在
更新代码 10102 的 24x24 大小的字形模式吗? [N]: ALL
%CMGR-I-FONTUPD, 已更新代码 10102 的 24x24 大小的字形模式
%CMGR-I-FONTUPD, 已更新代码 10103 的 24x24 大小的字形模式
%CMGR-I-FONTUPD, 已更新代码 10101 的 32x32 大小的字形模式
%CMGR-I-FONTEXIST, 代码 10102 的 32x32 大小的字形模式已存在
%CMGR-I-FONTUPD, 已更新代码 10102 的 32x32 大小的字形模式
%CMGR-I-FONTUPD, 已更新代码 10103 的 32x32 大小的字形模式
%CMGR-I-FONTUPD, 已更新代码 10101 的 40x40 大小的字形模式
%CMGR-I-FONTEXIST, 代码 10102 的 40x40 大小的字形模式已存在
%CMGR-I-FONTUPD, 已更新代码 10102 的 40x40 大小的字形模式
%CMGR-I-FONTUPD, 已更新代码 10103 的 40x40 大小的字形模式
%CMGR-I-UPGCOLL, 正在用文件 DISK$:[USER]TEMP.CVD;1 更新 CMGR_DEFAULT
数据库的整理值 ...
```

## 使用 CMGR

### 4.5 CMGR SHOW TABLE 和 CMGR SHOW BITMAP 命令

```
%CMGR-I-COLLUPD, 已更新代码 10101 的整理值
%CMGR-I-COLLEXIST, 代码 10102 的整理值已存在
%CMGR-I-COLLUPD, 已更新代码 10102 的整理值
%CMGR-I-COLLUPD, 已更新代码 10103 的整理值
%CMGR-I-DBUPD, 已更新 CMGR_DEFAULT 系统数据库
```

用户可以看到通知信息，这些信息说明您当前系统属性数据库已经包含字符 10102 的属性。

### 4.5 CMGR SHOW TABLE 和 CMGR SHOW BITMAP 命令

CMGR SHOW TABLE 命令允许用户建立字符属性表，CMGR SHOW BITMAP 命令建立字形位映象的输出。这两个命令的主要目的是以表或位映象的格式展示系统属性数据库和用户属性文件的内容。使用 SHOW TABLE 命令来展示系统属性数据库中的字形属性表时，可显示 DEC 汉字字符集内的所有区。至于在数据库上的其他 SHOW TABLE 和 SHOW BITMAP 操作，只限于展示用户定义字符或保留用户定义字符（如果指定了 /RESERVED）。

#### 4.5.1 展示用户属性文件的属性资料

如果用户只要展示属性文件（TEMP.PRE 和 TEMP.CVD）的内容但不展示位映象，他们可以输入：

```
CMGR> SHOW TABLE/USER=TEMP/OUTPUT=LIST1.LIS
```

/USER 限定词是指属性资料要从用户属性文件取出，而不是从系统属性数据库取出。用户属性文件说明是这限定词的值。同样地，如果指定 /FULL 限定词（是默认限定词），便不允许给用户属性文件指定扩充名。

如果用户要在屏幕上显示资料，由于 /OUTPUT 的默认是 /OUTPUT=SYS\$OUTPUT，因此他们可以省略限定词 /OUTPUT=LIST1.LIS。

图 4-25 展示输出列表文件 LIST1.LIS 的内容。

图 4-25 LIST1.LIS 的输出

```
Mon Jun 24 14:16:41 1991 Page 1
Preload File: DISK$:[USER]TEMP.PRE;1
Collating File: DISK$:[USER]TEMP.CVD;1
```

Code	Glyph	Existence			Collating Characteristics		
		24x24	32x32	40x40	Pinyin	Radical	Stroke
10101	[A121] 牟	o	-	-	mou	厶	6
10102	[A122] 耽	o	-	-	dan	耳	10
10103	[A123] 散	o	-	-	san	攴	12

如果用户只以字形编辑程序建立预装入文件而不建立整理数据文件，他们必须添加限定词 /FONT:



## 使用 CMGR

### 4.6 CMGR REMOVE 和 COPY 命令

```
CMGR> SHOW TABLE/OUTPUT=LIST3.LIS 1-87
```

要展示 CMGR\_DEFAULT 系统属性数据库的字符位映象，应输入：

```
CMGR> SET DATABASE/DEFAULT
```

```
CMGR> SHOW BITMAP/OUTPUT=LIST4.LIS 10101-18794
```

再一次，/FULL/SIZE=ALL 是这两个命令的默认。如果用户要在终端屏幕上显示资料，由于 /OUTPUT 的默认是 /OUTPUT= SYS\$OUTPUT，因此他们可以省略限定词 /OUTPUT。

每一个命令都接受一个参数。SHOW TABLE 的参数是区号，而 SHOW BITMAP 的参数是字符代码。

用户可以指定区的列表或区的范围，如下所述：

```
CMGR> SHOW TABLE 1-10, 30-40/QUADRANT=1
```

“1-10”和“30-40”指定区的范围。用户必须使用连字符“-”来分隔该范围的上下限。

SHOW BITMAP 的参数是一个代码列表或代码范围：

```
CMGR> SHOW BITMAP 10101-11094, 13001-14094
```

同样地，代码范围的上下限必须以连字符“-”分隔。

注意，没有 /RESERVED 限定词，便不能显示保留用户定义区域内的代码属性（仅用 SHOW TABLE 的字形除外）。因此，如果用户要展示保留用户定义区域内有关属性的资料，应使用下列命令：

```
CMGR> SHOW TABLE/RESERVED 88-94/QUADRANT=1
```

```
CMGR> SHOW BITMAP/RESERVED 18801-19494
```

用户应该尝试指定不同组合的 /FULL、/FONT、/COLLATING、/SIZE、/USER、/SYSTEM 和 /RESERVED 来看看输出有什么差别。

### 4.6 CMGR REMOVE 和 COPY 命令

REMOVE 和 COPY 命令都需要有 SYSPRV 特权来执行，它们在当前系统属性数据库上操作。REMOVE 用以除去当前系统属性数据库的字符属性，而 COPY 则用以把字符属性从一个代码复制到另一个代码。

这两个命令都以 /FULL/SIZE=ALL 和 /LOG 作为默认。另外，REMOVE 命令以 /CONFIRM 作为默认，而 COPY 命令则以 /CONFIRM=ALL 为默认。

REMOVE 命令接受代码列表或代码范围作为其参数。COPY 命令只接受两个参数。第一个是源代码列表，第二个是目的地代码列表。当源代码列表和目的地代码列表中的代码数目不同时，CMGR 便截断较长的那个参数。注意，除非指定 /RESERVED 限定词，否则两个命令都不允许处理保留用户定义字符。

以下展示每个命令的例子，当前数据库已设置为 CMGR\_DEFAULT。

```
CMGR> REMOVE 10101-10103
```

```
%CMGR-I-RMGFONT, 正在从 CMGR_DEFAULT 数据库除去 24x24 大小的字形模式...
```

```
除去代码 10101 的 24x24 大小的字形模式吗？[N]: Y
```

```
%CMGR-I-FONTRMD, 已除去代码 10101 的 24x24 大小的字形模式
```

```
除去代码 10102 的 24x24 大小的字形模式吗？[N]: N
```

```
除去代码 10103 的 24x24 大小的字形模式吗？[N]: ALL
```

```
%CMGR-I-FONTRMD, 已除去代码 10103 的 24x24 大小的字形模式
```

```

%CMGR-I-RMGFONT, 正在从 CMGR_DEFAULT 数据库除去 32x32 大小的字形模式 ...
%CMGR-I-FONTRMD, 已除去代码 10101 的 32x32 大小的字形模式
%CMGR-I-FONTRMD, 已除去代码 10102 的 32x32 大小的字形模式
%CMGR-I-FONTRMD, 已除去代码 10103 的 32x32 大小的字形模式
%CMGR-I-RMGFONT, 正在从 CMGR_DEFAULT 数据库除去 40x40 大小的字形模式 ...
%CMGR-I-FONTRMD, 已除去代码 10101 的 40x40 大小的字形模式
%CMGR-I-FONTRMD, 已除去代码 10102 的 40x40 大小的字形模式
%CMGR-I-FONTRMD, 已除去代码 10103 的 40x40 大小的字形模式
%CMGR-I-RMGCOLL, 正在从 CMGR_DEFAULT 数据库除去整理值 ...
%CMGR-I-COLLRMD, 已除去代码 10101 的整理值
%CMGR-I-COLLRMD, 已除去代码 10102 的整理值
%CMGR-I-COLLRMD, 已除去代码 10103 的整理值
%CMGR-I-DBUPD, 已更新 CMGR_DEFAULT 系统数据库
CMGR> COPY/FONT/SIZE=24 10101-10102 12101
%CMGR-I-CPGFONT, 正在复制 CMGR_DEFAULT 数据库中的 24x24 大小的字形模式 ...
%CMGR-I-FONTEXIST, 代码 12101 的 24x24 大小的字形模式已存在      把代码 10101 的 24x24 大小的字
形模式复制至代码 12101 ? [N]: Y
%CMGR-I-FONTCPD, 已把代码 10101 的 24x24 大小的字形模式复制至代码 12101
%CMGR-I-DBUPD, 已更新 CMGR_DEFAULT 系统数据库

```

## 4.7 CMGR EXTRACT CODE 和 CMGR EXTRACT REFERENCE 命令

EXTRACT 命令从系统属性数据库抽取字符属性资料来产生用户属性文件。用户可以用 EXTRACT CODE 命令来抽取字形和整理属性，或用 EXTRACT REFERENCE 命令来只抽取字形属性。

EXTRACT CODE 命令的主要功能是为编辑或预装入（只就预装入文件而言）建立用户属性文件。EXTRACT REFERENCE 命令则为预装入建立预装入文件。

EXTRACT CODE 接受两个参数。一个是代码列表，另一个是用户属性文件的说明：

```
CMGR> EXTRACT CODE 10101-10103 USER1
```

EXTRACT CODE 命令的默认限定词是 /FULL/SIZE=ALL，因此，上述命令会建立两个文件，即 USER1.PRE 和 USER1.CVD。这两个文件分别包含代码 10101 至 10103 的所有大小的预装入序列和整理值。而且，如果使用了 /FULL 限定词（或默认），便不允许在用户属性文件说明中指定文件扩充名。产生预装入文件和整理数据文件后，用户便可以使用字形编辑程序来编辑文件或把 USER1.PRE 预装入到终端以供显示。

如上所述，EXTRACT REFERENCE 只建立预装入文件，它不具有 /FULL、/FONT 或 /COLLATING 中的任何一个限定词。这命令的功能是检索一遍一些文本文件以找出用户定义字符，并给那些用户定义字符建立预装入文件。建立的预装文件可以预装入到终端/打印机上。因此，用户在发送文本文件前，应把这文件发送到设备以便显示或打印那些用户定义字符。当然，系统属性数据库必须包含那些用户定义字符代码的字形属性。

EXTRACT REFERENCE 也接受两个参数。一个是待检索的文本文件列表，另一个是预装入文件的说明：

```
CMGR> EXTRACT REFERENCE USERDATA1.TXT, USERDATA2.TXT USER2
```

## 使用 CMGR

### 4.8 CMGR CONVERT FONT\_SIZE

以上命令只建立一个文件 USER2.PRE。

用户可在这两个命令上使用限定词 /ENTRY\_LIMIT 来控制最多只可输出多少属性记录到用户属性文件。原因是所有设备（如 VT382）都只有有限的内存空间来存储预装入字形模式。如果用完了内存空间，任何输入的预装入字形模式便会重写到内存的现有字形模式上。因此，如果必需的话，用户应查看有关的设备参考手册来指定 /ENTRY\_LIMIT 的值，从而使预装入文件内的所有字形模式都可有效地预装入到设备上。如果不指定这限定词，CMGR 便不限制输出属性记录的数目。

再者，如果用户要抽取保留用户定义字符，便必须指定 /RESERVED。如果没有指定该限定词而 EXTRACT CODE 的代码列表却包含保留用户定义字符代码，命令便会失败。至于 EXTRACT REFERENCE，如果此命令在文本文件中找到一个保留用户定义字符，便会显示一个警告信息。

如果用户要抽取其中一种或两种大小的字形而不是所有大小的字形，可使用 /SIZE 限定词，并指定大小作为限定词的值。例如：

```
CMGR> EXTRACT REFERENCE/SIZE=24 USERDATA1.TXT, USERDATA2.TXT USER2
```

### 4.8 CMGR CONVERT FONT\_SIZE

CONVERT FONT\_SIZE 命令把预装入文件中的字形模式从一种大小转换成另一种大小。这命令接受两个参数，即输入预装入文件列表和输出预装入文件。系统会读取输入预装入文件中的预装入序列，并把它们转换成 /SIZE 限定词指定的字形大小。转换后的预装入序列会写入输出预装入文件。例如：

```
CMGR> CONVERT FONT_SIZE FONT.PRE FONT32.PRE/SIZE=32
```

这命令会读取 FONT.PRE 中的所有字形，并把它们转换成 32x32 的字形大小，然后输出到 FONT32.PRE 文件。如果 FONT.PRE 包含同一个代码的重复预装入序列（可能属不同的字形大小），那么会用最后一个预装入序列来执行转换。还有，如果 FONT.PRE 中的预装入序列已是 32x32 大小的，那么便只会复制该预装入序列到输出文件。

默认是 /SIZE=24。因此，如果不指定限定词，所有字形都会转换成 24x24 大小。

### 4.9 CMGR SHOW CHARACTER\_SET 和 CMGR SHOW VERSION 命令

SHOW CHARACTER\_SET 命令用以展示字符集资料。用户可以使用这些资料来确定 CMGR 操作的有效代码范围和区范围。例如，用户必须在 CMGR 的所有命令中指定有效的用户定义字符代码，或者用户必须在一些具有 /RESERVED 限定词的命令中输入保留用户定义字符的有效代码范围。SHOW CHARACTER\_SET 的输出展示如下。

```
CMGR> show character_set
```

DEC GB-2312 汉字字符集

代码范围	象限	区范围	字符种类
00101 - 00994	LR	1 - 9	DEC GB-2312 字符
01001 - 01594	LR	10 - 15	DEC GB-2312 保留字符
01601 - 08794	LR	16 - 87	DEC GB-2312 字符
08801 - 09494	LR	88 - 94	DEC GB-2312 保留字符
10101 - 18794	LL	1 - 87	用户定义字符
18801 - 19494	LL	88 - 94	保留用户定义字符

SHOW VERSION 命令展示当前 CMGR 的版本号，如下所示：

```
CMGR> SHOW VERSION
%CMGR-I-VERSION, CMGR 第 2.0 版
```

## 4.10 CMGR HELP 和 EXIT 命令

任何时候用户都可在 CMGR 中键入 HELP，然后在“Topic?”提示符处选择要阅读的题目。每当用户对使用任何一个 CMGR 命令有疑问，都可以使用这命令。

```
CMGR> help
Information available:
Character_attributes CONVERT COPY EDIT EXIT EXTRACT
HELP      REMOVE   SET      SHOW System_Attribute_Database
UPDATE    User_Attribute_Files
```

Topic? copy

COPY

COPY 命令把源字符代码的字符属性复制到当前“系统属性数据库”的目的地字符代码。对每对源/目的地代码执行操作时不设任何缓冲，因此，下列命令会使 10101 的属性被复制到从 10102 到 10194 的所有代码（10101 复制到 10102，然后 10102 复制到 10103，如此类推。）：

```
COPY 10101-10193 10102-10194
```

有关怎样选择当前数据库的详情，请参阅 SET DATABASE 命令的题目。

您必须有 SYSPRV 特权，才能执行这个命令。

语法：

```
COPY [/qualifiers] source-code[,...] destination-code[,...]
```

Additional information available:

Parameters Command\_Qualifiers

```
/COLLATING /CONFIRM /FONT /LOG /RESERVED /SIZE
```

Examples

COPY Subtopic?

EXIT 命令用以退出 CMGR 并把用户送回 DCL 提示。

```
CMGR> EXIT
```

```
$
```

用户也可以使用 <CTRL/Z> 来退出 CMGR。



## 第 5 章

### CMGR 命令参考

本章包含 CMGR 中所有可用命令的完整参考资料。

以下 DCL 命令可调用 CMGR 子系统：

```
$ CHARACTER_MANAGER
CMGR>
```

建议用户定义以下的外部命令：

```
$ CMGR ::= $HSY$SYSTEM:CMGR
```

从而，CMGR 可以下列命令调用：

```
$ CMGR
CMGR>
```

CMGR> 提示符表示已进入 CMGR 子系统，用户现在可以输入本区中描述的命令。

另一个使用 CMGR 命令的方法是在 DCL 命令级上直接地使用它们：

```
$ CMGR cmgr-commands [qualifiers] [parameters]
```

#### 5.1 CONVERT DATABASE

把旧字形数据库文件 FONT1.DAT 中的字形转换到当前系统属性数据库。

##### 语法

```
CONVERT DATABASE [/qualifiers]
```

命令限定词	默认
<i>/[NO]CONFIRM</i>	<i>/CONFIRM</i>
<i>/[NO]LOG</i>	<i>/LOG</i>

##### 限制

- 用户必须有 SYSPRV 特权，才能执行这命令。

##### 提示

没有。

##### 参数

没有。

##### 描述

## CMGR 命令参考

### 5.1 CONVERT DATABASE

推出 CMGR 后，便不再使用旧的 FONT.DAT、FONT1.DAT 系统字形数据库。要使用在 FONT1.DAT 中定义的字形模式，系统经理必须在系统升级后使用 CONVERT DATABASE 命令来执行转换。

CONVERT DATABASE 命令读取在 HSY\$SYSTEM:FONT1.DAT 中定义的字形模式，并把他们转换成新的格式，然后把字形存储到当前的系统属性数据库。

旧的字形系统仅支持 24x24 大小的字形。但采用 CMGR 后，32x32 和 40x40 大小的字形也得到支持。在执行 CONVERT DATABASE 时，系统会从 FONT1.DAT 中读取 24x24 大小的字形，并把它们转换成 32x32 和 40x40 的大小。转换后的字形会装入当前的系统属性数据库。

然而，旧的字形系统不支持整理特征。CMGR 无法从旧的字形系统中获得整理值。因此，在执行 CONVERT DATABASE 命令后，整理特征仍未定义。

可用 SET DATABASE 命令来设置当前数据库。在多数情况下，系统经理都需要把当前数据库设置为 CMGR\_DEFAULT，然后用 CONVERT DATABASE 来执行转换。这是因为 OpenVMS/Hanzi 的字形处理程序会在按需装入时使用 CMGR\_DEFAULT 数据库的字形。

#### 命令限定词

**/CONFIRM**

**/NOCONFIRM**

指定如果当前系统属性数据库内已有字形模式的定义时是否需要用户确认。

默认是 /CONFIRM。

回应确认请求时可接收的回答是：

YES, TRUE, 1	肯定的回答
NO, FALSE, 0	否定的回答
<RETURN>	默认的回答（否定）
<CTRL/Z>	终止该命令

回答可以缩写。

**/LOG**

**/NOLOG**

控制是否显示通知信息。

默认是 /LOG。

#### 例子

```
1 CMGR> CONVERT DATABASE
%CMGR-I-READFIL, 正在读文件 HSY$SYSTEM:FONT1.DAT
%CMGR-S-FONTCONVTD, 已转换 10101 的字形数据
%CMGR-S-FONTCONVTD, 已转换 10102 的字形数据
%CMGR-S-FONTCONVTD, 已转换 10103 的字形数据
%CMGR-I-FDBCONVEND, 总共把 3 个模式转换至 CMGR_DEFAULT 字形数据库
```

此命令转换 HSY\$SYSTEM:FONT1.DAT 中的三个字形模式并把它们存储到当前的系统属性数据库 - CMGR\_DEFAULT。

## 5.2 CONVERT FONT\_FILE

把一个或多个字形文件中的所有字形转换到一个预装入文件。

### 语法

```
CONVERT FONT_FILE [/qualifiers] font-file[,...] preload-file
```

命令限定词	默认
/[NO]LOG	/LOG

### 限制

没有。

### 提示

\_Font File: font-file[,...]  
\_Preload File: preload-file

### 参数

**font-file[,...]**

一个或多个字形文件的文件名。将转换这些字形文件中找到的用户定义字符和保留用户定义字符的所有字形模式。

**preload-file**

输出预装入文件的文件名。将建立这文件以包含转换的字形模式。

### 描述

在旧的字形系统中，用户字形存储在字形文件里。在 CMGR 中，用户字形存储在预装入文件里。要在 CMGR 下使用在字形文件中定义的字形，必须首先使用 CONVERT FONT\_FILE 命令来把这些字形文件转换成预装入文件。

用户可在 CONVERT FONT\_FILE 命令中输入字形文件列表，便可把文件中的所有字形转换成预装入序列。这命令可转换用户定义字符和保留用户定义字符。这里没有 /RESERVED 限定词。

指定的预装入文件会建立起来以存储这些已转换的预装入序列。

### 命令限定词

/LOG

/NOLOG

控制是否显示通知信息

默认是 /LOG。

### 例子

1

## CMGR 命令参考

### 5.3 CONVERT FONT\_SIZE

```
CMGR> CONVERT FONT_FILE MYFONT.FON MYFONT.PRE
```

这命令把 MYFONT.FON 中的位映象字形模式转换成预装入序列，并把它们放入 MYFONT.PRE。

### 5.3 CONVERT FONT\_SIZE

把一个或多个预装入文件中的所有字形转换成其中一种支持的字形大小。

#### 语法

```
CONVERT FONT_SIZE [/qualifiers] source-file[,...] destination-file
```

---

命令限定词	默认
/[NO]LOG	/LOG
/SIZE=font-size	/SIZE=24

#### 限制

没有。

#### 提示

\_Input File: source-file[,...]

\_Output File: destination-file

#### 参数

**source-file[,...]**

一系列源预装入文件，里面包含要转换的预装入序列。默认扩充名是 .PRE。

**destination-file**

存储已转换的预装入序列的输出预装入文件。转换后，这文件中的所有预装入序列都会有指定的大小。

#### 描述

CONVERT FONT\_SIZE 命令把一个或多个预装入文件中的所有字形转换为其中一个支持的字形大小，并把它们输出到一个目的地预装入文件。这命令可转换用户定义字符和保留用户定义字符。这里没有 /RESERVED 限定词。

要指定输出字形大小，可使用 /SIZE 限定词。如果输入预装入文件已包含指定大小的字形，预装入序列就只复制到输出文件。如果输入文件包含一个以上的同码预装入序列，便使用最后的预装入序列来作转换。

#### 命令限定词

**/LOG**

**/NOLOG**

控制是否显示通知信息。

默认是 /LOG。

**/SIZE=font-size**

指定要转换的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。

默认是 /SIZE=24

## 例子

1

```
CMGR> CONVERT FONT_SIZE/SIZE=32 MYFILE24.PRE MYFILE32.PRE
```

假定 myfile24.pre 包含 10101 的 24x24 字形。这命令会把该字形转换为 32x32 大小，并把它存储入 myfile32.pre。

## 5.4 COPY

把指定的源字符代码的字符属性复制到当前系统属性数据库的目的地字符代码。

### 语法

```
COPY [/qualifiers] source-code[,...] destination-code[,...]
```

命令限定词	默认
<i>/COLLATING</i>	<i>/FULL</i>
<i>/[NO]CONFIRM=[condition]</i>	<i>/CONFIRM=ALL</i>
<i>/FONT</i>	<i>/FULL</i>
<i>/FULL</i>	<i>/FULL</i>
<i>/[NO]LOG</i>	<i>/LOG</i>
<i>/RESERVED</i>	<i>None</i>
<i>/SIZE=(font-size[,...])</i>	<i>/SIZE=ALL</i>

### 限制

- 用户必须有 SYSPRV 特权，才能执行这命令。

### 提示

```
_From:    source-code[,...]
_To:      destination-code[,...]
```

### 参数

**source-code**[,...]

**destination-code**[,...]

代码或代码范围的列表。代码可以是“区位”码或内码。代码范围是两个代码，以连字符分隔，中间无空格。

“区位”码由 5 个十进制数字组成，第一位数字是任选的，可以是“0”或“1”。“0”指定右下象限，而“1”指定左下象限。如果不指定，默认为“0”。接下来的两位数字指定“区”，最后两位数字指定“位”。

## CMGR 命令参考

### 5.4 COPY

内码是前面带“H”的十六进制代码。例如，HA121 即等于 10101。

如没有 /RESERVED 限定词，只能对属于用户定义区域的代码执行复制。有关详情，请参阅 /RESERVED 的描述。

星号“\*”可用来指定用户定义区域的所有代码。也就是说，“\*”等于“10101-18794”。

代码范围可以跨越几区，但不可以跨越象限。

#### 描述

COPY 命令把在 source-code 代码列表中指定的每个代码的字符属性复制到在 destination-code 代码列表中的相应代码。如果两个代码列表具有不同的长度，较长列表中的任何多余代码都会被忽略。

对于每对源 / 目的地代码执行复制操作时不设任何缓冲，因此，下列命令会使 10101 的属性复制到从 10102 到 10194 的所有代码（10101 复制到 10102，然后 10102 复制到 10103，依此类推）：

```
CMGR> COPY 10101-10193 10102-10194
```

按照默认，会复制所有字符属性。然而，用户可以明确指定要复制那个字符属性。如果要复制字形模式，用户可以指定要复制那种字形大小。

注意，用户只在当前数据库中代码间复制字符属性。当前数据库可以 SET DATABASE 命令来设置。

#### 命令限定词

##### /COLLATING

这字符属性限定词指定要复制整理特征。

这限定词不可以与 /FULL 一起使用。

##### /CONFIRM[=condition]

##### /NOCONFIRM

指定是否需要用户确认。确认级取决于条件。可能的条件有：

ALL	对每个事项都要求确认
CONFLICT	如果相应的字符属性已在目的地代码中登记，就要求确认。

默认是 /CONFIRM=ALL。

回应确认请求时可接受的回答是：

YES, TRUE, 1	肯定的回答
NO, FALSE, 0	否定的回答
<RETURN>	默认的回答（否定）
<CTRL/Z>, QUIT	终止该命令

回答可以缩写。

##### /FONT

这字符属性限定词指定要复制字形模式。

这限定词不可以与 /FULL 一起使用。

#### **/FULL**

指定要复制所有字符属性。

如果在命令行中没有指定字符属性限定词，便假定为 /FULL。

这限定词不可以与其他字符属性限定词一起使用。

#### **/LOG**

#### **/NOLOG**

控制是否显示通知信息。

默认是 /LOG。

#### **/RESERVED**

要指定“保留用户定义字符”的代码时，便指定这个限定词。

如果指定了这个限定词，便可用星号“\*”来指定用户定义区域内的所有代码以及保留用户定义区域中的代码。也就是说，“\*”等于“10101-19494”。

#### **/SIZE=(font-size[,...])**

指定要复制的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。用户也可以使用关键字 'ALL' 来表示要复制所有支持的字形大小。

默认是 /SIZE=ALL

这个限定词只与 /FONT 或 /FULL（默认）一起使用时才有效。

### 例子

1

```
CMGR> COPY 10101 10102
```

这个命令把字符代码 10101 的所有字符属性复制到 10102。它将请求确认并通知您复制操作是成功还是失败。

2

```
CMGR> COPY/NOLOG/CONFIRM=CONFLICT/FONT HA121-Ha122 10111-10113
```

source-code 代码列表扩展到 10101、10102。

destination-code 代码列表扩展到 10111、10112、10113。

这个命令将把字形模式从 10101 复制到 10111，同时从 10102 复制到 10112。目的地代码列表中的最后一个代码（10113）会被忽略。除非发生错误，否则不会打印信息。此外，除非两个 destination-code 已有字形模式定义，否则不会请求确认。

## 5.5 EDIT

调用字形编辑程序来编辑用户属性文件中的字形模式。同时也可以编辑字形模式时编辑其他字符属性。

### 语法

## CMGR 命令参考

### 5.5 EDIT

**EDIT** *[/qualifiers] user-attribute-file*

命令限定词	默认
<i>/COLLATING</i>	<i>/FULL</i>
<i>/FONT</i>	<i>/FULL</i>
<i>/FULL</i>	<i>/FULL</i>
<i>/[NO]LOAD_GLYPH</i>	<i>/LOAD_GLYPH</i>
<i>/[NO]LOG</i>	<i>/LOG</i>
<i>/RESERVED</i>	<i>None</i>
<i>/SIZE=(font-size[,...])</i>	<i>/SIZE=ALL</i>

#### 限制

- 字形编辑程序只支持 VT382 终端。

#### 提示

\_Input File: user-attribute-file

#### 参数

user-attribute-file

用户属性文件的文件名。要编辑一个以上的属性，文件名不能有扩充名，而是使用默认的扩充名。支持的用户属性文件：

字符属性	文件扩充名	文件类型
字形模式	.PRE	预装入文件
整理特征	.CVD	整理数据文件

#### 描述

EDIT 命令调用 CMGR 字形编辑程序来编辑预装入文件中的字形模式。用户也可以使用字形编辑程序来编辑其他字符属性，但不能在没有编辑字形时去编辑它们。

字形编辑程序只支持 VT382 终端。

如没有 /RESERVED 限定词，只可以编辑属于用户定义区域的代码。有关详情请参阅 /RESERVED 的描述。

有关使用字形编辑程序的详情，请参阅第 6 章“字形编辑程序参考”。

#### 命令限定词

**/COLLATING**

这字符属性限定词指定要编辑整理特征。

这限定词不可没有 /FONT 而单独使用，原因是字形编辑程序不可以单独编辑整理特征。

这限定词不可以与 /FULL 一起使用。

#### **/FONT**

这字符属性限定词指定要编辑字形模式。

这限定词不可以与 /FULL 一起使用。

#### **/FULL**

指定要编辑所有字符属性。

如果在命令行中没有指定字符属性限定词，就假定为 /FULL。

这个限定词不可以与其他字符属性限定词一起使用。

#### **/LOAD\_GLYPH**

#### **/NOLOAD\_GLYPH**

为了显示，字形编辑程序使用一些特殊的字符字块。每当启动编辑程序，便会把这些字块装入用户的终端。然而，一旦装入字块，除非用户重置终端，否则它们不会被擦除。装入需花时间，因此，如果确信字块已装入，为了节省时间，就可以指定 /NOLOAD\_GLYPH。

默认是 /LOAD\_GLYPH。

#### **/LOG**

#### **/NOLOG**

控制是否显示通知信息。

默认是 /LOG。

#### **/RESERVED**

要编辑保留用户定义字符时，便指定这个限定词。

如果调用编辑程序时不指定这个限定词，便只能编辑用户定义区域的代码。

#### **/SIZE=(font-size[,...])**

指定要编辑的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。用户也可以使用关键字 'ALL' 来表示要编辑所有支持的字形大小。

默认是 /SIZE=ALL。

### 例子

1

```
CMGR> EDIT/FONT/COLLATING/SIZE=ALL MYFILE
```

这命令调用字形编辑程序来编辑用户属性文件 myfile.pre; 以及 myfile.cvd;。如果默认目录中没有这两个文件，便会把它们建立起来。在进入编辑对话期前，会把特殊字块装入用户终端。同时用户可以编辑任何支持的字形大小。此外，还会有信息显示以通知您在对话期内执行的操作是成功还是失败。

## 5.6 EXIT

终止 CMGR 对话期

## CMGR 命令参考

### 5.7 EXTRACT CODE

#### 语法

**EXIT**

---

**命令限定词**

**默认**

没有

没有

#### 限制

没有。

#### 提示

没有。

#### 参数

没有。

#### 描述

EXIT 命令终止 CMGR 对话期并从 CMGR 子系统返回 DCL。  
用户也可以按 <CTRL/Z> 来终止 CMGR 对话期。

#### 限定词

没有。

#### 例子

1

```
CMGR> EXIT
```

```
$
```

EXIT 命令终止 CMGR 对话期并返回 DCL "\$" 提示符。

## 5.7 EXTRACT CODE

按指定的字符代码从当前的系统属性数据库中抽取字符属性到用户属性文件。

#### 语法

**EXTRACT** [*/qualifiers*] *source-code*[...] *user-attribute-file*

---

**命令限定词**

**默认**

*/COLLATING*

*/FULL*

*/ENTRY\_LIMIT=number-of-characters*

*Nones*

*/FONT*

*/FULL*

*/FULL*

*/FULL*

*/[NO]LOG*

*/LOG*

*/RESERVED*

*None*

*/SIZE=(font-size[,...])*

*/SIZE=ALL*

## 限制

没有。

## 提示

`_Code:` source-code[,...]

`_Output file:` user-attribute-file

## 参数

### **source-code[,..]**

代码或代码范围的列表。代码可以是“区位”码或内码。代码范围是两个代码，以连字符分隔，中间无空格。

“区位”码由 5 个十进制数字组成，第一位数字是任选的，可以是“0”或“1”。“0”指定右下象限，而“1”指定左下象限。如果不指定，默认为“0”。接下来的两位数字指定“区”，最后两位数字指定“位”。

内码是前面带“H”的十六进制代码。例如，HA121 即等于 10101。

如没有 /RESERVED 限定词，只能抽取属于用户定义区域内的代码。有关详情，请参阅 /RESERVED 的描述。

星号“\*”可用来指定用户定义区域的所有代码。也就是说，“\*”等于“10101-18794”。

代码范围可以跨越几区，但不可以跨越象限。

### **user-attribute-file**

用户属性文件的文件名。要抽取一个以上的属性，文件名不能有扩充名，因为会添加默认的扩充名。

支持的用户属性文件：

字符属性	文件扩充名	文件类型
字形模式	.PRE	预装入文件
整理特征	.CVD	整理数据文件

## 描述

EXTRACT CODE 命令把在 source-code 代码列表中所指定的每个代码的字符属性抽取到用户属性文件。如果文件已经存在，便建立更高的版本。

这命令严格依据指定的代码列表来执行抽取并且会逐个抽取。

如果指定输入极限，便不会抽取多于指定数目的字符属性。代码列表末尾的多余代码会被忽略。

按照默认，会抽取指定代码的所有字符属性。然而，用户可以明确指定要抽取的字符属性。如果要抽取字形模式，用户可以指定要抽取的字形大小。

注意，字符属性会从当前数据库抽取，而当前数据可以由 SET DATABASE 命令来设置。

## 命令限定词

## CMGR 命令参考

### 5.7 EXTRACT CODE

#### **/COLLATING**

这字符属性限定词指定要抽取整理特征。

这限定词不可以与 /FULL 一起使用。

#### **/ENTRY\_LIMIT=number-of-character**

如果指定的话，可限制抽取字符属性的最大数目。

#### **/FONT**

这字符属性限定词指定要抽取字形模式。

这限定词不可以与 /FULL 一起使用。

#### **/FULL**

指定要抽取所有字符属性。

如果在命令行中没有指定字符属性限定词，就假定为 FULL。

这个限定词不可以与其他字符属性限定词一起使用。

#### **/LOG**

#### **/NOLOG**

控制是否显示通知信息。

默认是 /LOG。

#### **/RESERVED**

要指定保留用户定义字符的代码时，便指定这个限定词。

如果指定了这个限定词，可以用星号 "\*" 来指定用户定义区域的所有代码以及保留用户定义区域中的代码。也就是说，"\*" 等于 "10101-19494"。

#### **/SIZE=(font-size[,...])**

指定要抽取的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。用户也可以使用关键字 'ALL' 来表示要抽取所有支持的字形大小。

默认是 /SIZE=ALL。

这个限定词只与 /FONT 或 /FULL（默认）一起使用时才有效。

## 例子

1

```
CMGR> EXTRACT CODE/FONT/ENTRY_LIMIT=5 10101-10107 MYFILE
```

source-code 代码列表扩展到 10101、10102、10103、10104、10105、10106、10107。

因为输入极限为 5，10101 到 10105 内所有大小的字形模式都被抽取，只忽略最后两个代码 10106 和 10107。

2

```
CMGR> EXTRACT CODE * MYFILE
```

这个命令抽取所有用户定义字符的全部字符属性。

## 5.8 EXTRACT REFERENCE

为指定文本文件中的用户定义字符把字形模式从当前的系统属性数据库中抽取到用户预装入文件。

### 语法

```
EXTRACT REFERENCE [/qualifiers] text-file[,...] preload-file
```

命令限定词	默认
<i>/ENTRV_LIMIT=number-of-characters</i>	<i>Nones</i>
<i>/[NO]LOG</i>	<i>/LOG</i>
<i>/RESERVED</i>	<i>None</i>
<i>/SIZE=(font-size[,...])</i>	<i>/SIZE=ALL</i>

### 限制

没有。

### 提示

\_Text File: text-file[,...]

\_Output file: preload-file

### 参数

**text-file[,...]**

文本文件名的列表。这些文本文件中的用户定义字符将会抽取其预装入序列。如果一个字符在文本文件中出现多次，便仅抽取其预装入序列一次。

如果指定 */RESERVED*，便抽取保留用户定义字符。

**preload-file**

输出预装入文件的文件名。这个文件将被建立起来，并包含抽取的预装入序列。

默认预装入文件的扩充名是 *.PRE*。

### 描述

EXTRACT REFERENCE 命令为指定文本文件列表中的每个用户定义字符抽取字形模式，并把它们输出到指定的预装入文件。如果预装入文件已经存在，便会建立更高版本。

如果指定了 */RESERVED* 限定词，便也可抽取保留用户定义字符。否则，如在文本文件中找到保留用户定义字符，便会显示警告信息。

它有助于在文本文件中找出用户定义字符的预装序列，使用户可在打印文本文件前把序列预装到终端或打印机，从而节省按需装入的时间。

抽取次序取决于用户定义字符在文本文件中的出现频率。频率越高，抽取越迟。这保证了当用户把字形预装入到没有足够字形 RAM 的终端时，只丢掉频率较低的预装入序列。

如果指定了输入极限，便不能抽取多于指定数目的字形模式。如果用户定义字符的个数大于指定的输入极限，便会首先丢掉频率较低的多余用户定义字符。

## CMGR 命令参考

### 5.9 HELP

按照默认，会抽取所有可支持其大小的字形模式。然而，用户也可用 /SIZE 限定词来指定要抽取的字形大小。

注意，字形模式是从当前数据库中抽取的，而当前数据库可以由 SET DATABASE 命令来设置。

#### 命令限定词

**/ENTRY\_LIMIT=number-of-character**

如果指定的话，可限制要抽取字形模式的最大数目。

**/LOG**

**/NOLOG**

控制是否显示通知信息。

默认是 /LOG。

**/RESERVED**

需要抽取保留用户定义字符时便指定这个限定词。

如果指定了这个限定词，便会抽取文本文件中找到的用户定义区域的所有代码以及保留用户定义区域的代码。

**/SIZE=(font-size[,...])**

指定要抽取的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。用户也可使用关键字 'ALL' 来表示要抽取所有支持的字形大小。

默认是 /SIZE=ALL。

#### 例

1

```
CMGR> EXTRACT REFERENCE/SIZE=24 MYTEXT.TXT MYFILE.PRE
```

这个命令会根据 'mytext.txt' 中找到的用户定义字符，从当前系统属性数据库中将 24x24 大小的字形模式抽取到预装入文件 'myfile.pre'。然后，用户便可以在发送文件 'mytext.txt' 前，将文件预装入到终端 / 打印机。字形将可用于终端 / 打印机而无须按需装入。

## 5.9 HELP

提供有关 CMGR 命令和其他相关题目的求助。

#### 语法

**HELP** [*topic*]

**命令限定词**

没有。

**默认**

没有。

#### 限制

没有。

#### 提示

Topic?      topic

## 参数

[**topic**]

指定用户想要求助的题目。如果没有指定题目，便会显示所有题目的列表。

要离开求助，可使用 <CTRL/Z> 或按 <RETURN> 几次。

## 描述

HELP 命令可显示某特定题目的求助。如果没有指定题目，便会列出包含所有 CMGR 命令及一些有关题目的列表。

## 限定词

没有。

## 例子

1

```
CMGR> HELP COPY
```

HELP 命令显示 COPY 命令的求助文本。

## 5.10 REMOVE

把指定字符代码的字符属性从当前系统属性数据库中除去。

## 语法

```
REMOVE [/qualifiers] character-code[,...]
```

命令限定词	默认
<i>/COLLATING</i>	<i>/FULL</i>
<i>/[NO]CONFIRM</i>	<i>/CONFIRM</i>
<i>/FONT</i>	<i>/FULL</i>
<i>/FULL</i>	<i>/FULL</i>
<i>/[NO]LOG</i>	<i>/LOG</i>
<i>/RESERVED</i>	<i>None</i>
<i>/SIZE=(font-size[,...])</i>	<i>/SIZE=ALL</i>

## 限制

- 用户必须有 SYSPRV 特权才能行这个命令。

## 提示

\_Code:        *character-code[,...]*

## 参数

**character-code[,...]**

## CMGR 命令参考

### 5.10 REMOVE

代码或代码范围的列表。代码可以是“区位”码或内码。代码范围是两个代码，以连字符分隔，中间无空格。

“区位”码由 5 个十进制数字组成，第一位数字是任选的，可以是“0”或“1”。“0”指定右下象限，而“1”指定左下象限。如果没有指定，默认为“0”。接下来的两个数字指定“区”，最后两个数字指定“位”。

内码是前面带“H”的十六进制代码。例如，HA121 即等于 10101。

如没有 /RESERVED 限定词，只能除去属于用户定义区域的代码。有关详情，请参阅 /RESERVED 的描述。

星号“\*”可用来指定用户定义区域的所有代码。也就是说，“\*”等于“10101-18794”。

代码范围可以跨越几区，但不可以跨越象限。

#### 描述

REMOVE 命令除去 character-code 代码列表中指定的每个代码的字符属性。它按照指定代码列表的次序逐个除去属性。

如果指定了 /RESERVED，便可除去用户定义区域和保留用户定义区域的属性。否则，只可除去用户定义区域的属性。

按照默认，会除去所有字符属性。然而，用户可以明确地指定要除去哪些字符属性。如果已除去字形模式，用户可以指定哪些字形大小要删除。

注意，字形模式是从当前数据库中除去的，而当前数据库可以由 SET DATABASE 命令来设置。

#### 命令限定词

##### /COLLATING

这个字符属性限定词指定要除去整理特征。

这个限定词不可以与 /FULL 一起使用。

##### /CONFIRM

##### /NOCONFIRM

指定是否需要用户确认。

默认是 /CONFIRM

回应确认请求时可接受的答案是：

YES, TRUE, 1	肯定的回答
NO, FALSE, 0	否定的回答
<RETURN>	默认的回答（否定）
<CTRL/Z>, QUIT	终止该命令

回答可以缩写。

##### /FONT

这个字符属性限定词指定要除去字形模式。

这个限定词不可以与 /FULL 一起使用。

**/FULL**

指定要除去所有字符属性。

如果在命令行中没有指定字符属性限定词，就假定为 /FULL。

这个限定词不可以与其他字符属性限定词一起使用。

**/LOG**

**/NOLOG**

控制是否显示通知信息。

默认是 /LOG。

**/RESERVED**

要指定保留用户定义字符的代码时，可指定这个限定词。

如果指定了这个限定词，可用星号 "\*" 来指定用户定义区域内的所有代码以及保留用户定义区域内的代码。也就是说，"\*" 等于 "10101-19494"。

**/SIZE=(font-size[,...])**

指定要除去的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。用户也可以使用关键字 'ALL' 来表示要除去所有支持的字形大小。

默认是 /SIZE=ALL

这个限定词仅与 /FONT 或 /FULL（默认）一起使用时才有效。

**例子**

1

```
CMGR> REMOVE 10101
```

这个命令将从当前系统属性数据库中除去 10101 代码的所有字符属性。在它实际删除属性之前，会请求用户确认，并且通知用户操作是成功还是失败。

## 5.11 SET DATABASE

把一个数据库设置为当前数据库。

**语法**

```
SET DATABASE [/qualifiers] [database-name]
```

命令限定词	默认
<i>/DEFAULT</i>	
<i>/LOG</i>	<i>/LOG</i>

**限制**

没有。

**提示**

```
_Database: database-name
```

## CMGR 命令参考

### 5.12 SHOW BITMAP

#### 参数

**[database-name]**

指定要设置为当前数据库的数据库名。支持的数据库有 CMGR\_DEFAULT、SONG 和 HEI。

如果已指定 /DEFAULT 限定词，便不允许使用该参数，否则，就必须指定这个参数。

#### 描述

CMGR 有三个系统属性数据库，名为 CMGR\_DEFAULT、SONG 和 HEI。用户注册时，会以默认数据库 CMGR\_DEFAULT 作为当前数据库（除非由系统经理明确地另行设置）。所有随后的 CMGR 命令将在当前数据库上操作。要选择另一个数据库作为当前数据库，可使用 SET DATABASE 命令。

SET DATABASE 命令使用逻辑名 CMGR\$DATABASE\_NAME 来保存当前数据库设置以供下次调用。

系统经理或用户可以在系统、组、作业或进程逻辑名表中，把其中一个支持的数据库名定义为 CMGR\$DATABASE\_NAME 这逻辑名来取代默认。以后调用 CMGR，便会以此数据库作为当前数据库。但如果把逻辑名定义为一个不支持的名，便不能调用 CMGR。

要展示有关数据库的资料，可使用 SHOW DATABASE 命令。

#### 命令限定词

**/DEFAULT**

要重置当前数据库为默认数据库，可使用这个限定词。

如果指定了这个限定词，便不要指定 database-name 参数。

**/LOG**

控制是否显示通知信息。

默认是 /LOG。

#### 例子

1

```
CMGR> SET DATABASE HEI
```

这个命令设置当前数据库为 HEI。会显示通知信息。

### 5.12 SHOW BITMAP

以位映象格式显示当前系统属性数据库和用户属性文件中指定代码的字形模式。

#### 语法

```
SHOW BITMAP [/qualifiers] [character-code[,...]]
```

---

**命令限定词**

**默认**

*[/[NO]BACKGROUND*

*/BACKGROUND*

*/COLLATING*

*/FULL*

*/FONT*

*/FULL*

*/FULL*

*/FULL*

<i>/OUTPUT=output-file</i>	<i>/OUTPUT=SYS\$OUTPUT</i>
<i>/RESERVED</i>	<i>None</i>
<i>/SIZE=(font-size[,...])</i>	<i>/SIZE=ALL</i>
<i>/SYSTEM</i>	<i>/SYSTEM</i>
<i>/USER</i>	<i>/SYSTEM</i>
<i>/WIDTH=80 or 132</i>	<i>/WIDTH=132</i>

## 限制

没有。

## 提示

`_Code:` [character-code[,...]]

## 参数

[character-code[,...]]

代码或代码范围的列表。代码可以是“区位”码或内码。代码范围是两个代码，以连字符分隔，中间无空格。

“区位”码由 5 个十进制数字组成，第一位数字是任选的，可以是“0”或“1”。“0”指定右下象限，而“1”指定左下象限。如果不指定，默认为“0”。接下来的两个数字指定“区”，最后两个数字指定“位”。

内码是前面带“H”的十六进制代码。例如，HA121 等于 10101。

如没有 /RESERVED 限定词，只可指定属用户定义区域的代码。有关详情，请参阅 /RESERVED 的描述。

星号“\*”可用来指定用户定义区域的所有代码。也就是说，“\*”等于“10101-18794”。

代码范围可以跨越几区，但不可以跨越象限。

如果已在命令行中指定 /USER，这个参数便是任选的。

## 描述

SHOW BITMAP 命令以位映象格式显示当前系统属性数据库中定义的以及来自预装入文件的预装入序列的字形的模式。其他字符属性也可和字形模式一起显示。用户可以用 /width 限定词来挑选 80 列或 132 列显示。

要显示当前系统属性数据库的位映象，可使用 /SYSTEM 限定词并指定要显示的字符代码作为参数。注意，当前数据库可用 SET DATABASE 命令来选择。要从用户属性文件来显示位映象，可使用 /USER 限定词并指定用户属性文件作为 /USER 限定词的值。在这种情形下，字符代码列表参数是任选的。如果指定了字符代码列表，便只显示属于该列表的代码位映象。如果没有指定字符代码列表，便会显示指定预装入文件中找到的所有字形模式。

如果需要指定保留用户定义字符的代码，可使用 /RESERVED 限定词。

## CMGR 命令参考

### 5.12 SHOW BITMAP

#### 命令限定词

##### **/BACKGROUND**

##### **/NOBACKGROUND**

控制背景点是否显示在位映象中。

SHOW BITMAP 使用 ‘@’ 来代表位映象上的 ON 象素（前景），“.” 来代表 OFF 象素（背景）。

##### **/COLLATING**

这个字符属性限定词指定要显示整理特征。

因为 SHOW BITMAP 不可以单独显示整理特征，所以这个字符属性限定词不可以没有 /FONT 而单独使用。

这个限定词不可以与 /FULL 一起使用。

##### **/FONT**

这个字符属性限定词指定要显示字形模式。

这个限定词不可以与 /FULL 一起使用。

##### **/FULL**

指定要显示所有字符属性。

如果在命令行中没有指定字符属性限定词，就假定为 /FULL。

这个限定词不可以与其他字符属性限定词一起使用。

##### **/OUTPUT=output-file**

按照默认，位映象会打印到 SYS\$OUTPUT。使用这个限定词，用户可以指示它打印到输出文件。

##### **/RESERVED**

要指定保留用户定义字符的代码时，可指定这个限定词。

如果指定了这个限定词，星号 “\*” 可用来指定用户定义区域内的所有代码以及保留用户定义区域内的代码。也就是说，“\*” 等于 “10101-19494”。

##### **/SIZE=(font-size[,...])**

指定要展示的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。用户也可以使用关键字 “ALL” 来表示要展示所有支持的字形大小。

默认是 /SIZE=ALL。

##### **/SYSTEM**

使用 SHOW BITMAP 打印当前系统属性数据库中的字形模式和 / 或其他字符属性。使用这个限定词时，必须指定 character-code 代码列表参数。然后，便打印指定字符代码的所有位映象和其他字符属性。

这个限定词不可以与 /USER 一起使用。

如果在命令行中没有指定 /SYSTEM 或 /USER，就假定为 /SYSTEM。

##### **/USER=user-attribute-file**

使用 SHOW BITMAP 来打印存储在用户属性文件中存储的字形模式和 / 或其他字符属性。使用这个限定词时，character-code 代码列表参数是任选的。如果指定该代码列表，仅显示属于列表的那些代码位映象。如果没有指定，便打印用户属性文件中的所有位映象及其他字符属性。

要显示一个以上的属性，user-attribute-file 的文件名不能有扩充名，而是使用默认的扩充名。支持的用户属性文件：

字符属性	文件扩充名	文件类型
字形模式	.PRE	预装入文件
整理特征	.CVD	整理数据文件

这个限定词不可以与 /SYSTEM 一起使用。

**/WIDTH=80 或 132**

指定要使用的显示列宽。

用户可以指定 80 或 132 列宽。如果不指定这个限定词或指定限定词却不指定值，默认是 132 列。

### 例子

1

```
CMGR> SHOW BITMAP/FONT/OUTPUT=MYBITMAP.LIS/SIZE=(32,40) 10101
```

这个命令将以位映象格式把 10101 的 32x32 和 40x40 大小的字形模式打印到文件 'mybitmap.lis'。位映象会使用 '.' 作为背景。

2

```
CMGR> SHOW BITMAP/FONT/COLLATING/USER=MYFILE/NOBACKGROUND
```

这个命令会打印预装入文件 'myfile.pre' 中找到的所有代码的字形模式，并且也会打印整理数据文件 'myfile.cvd' 中代码的整理特征。位映象会以空白作为背景。

## 5.13 SHOW CHARACTER\_SET

以表的格式显示当前的字符集。

### 语法

**SHOW CHARACTER\_SET**

命令限定词	默认
没有。	没有。

### 限制

没有。

### 提示

没有。

### 参数

## CMGR 命令参考

### 5.14 SHOW DATABASE

没有。

#### 描述

SHOW CHARACTER\_SET 命令以表的格式显示正在使用的字符集。它展示分配了哪个部分的字符平面给哪个字符种类。

#### 限定词

没有。

#### 例子

1

```
CMGR> SHOW CHARACTER_SET
```

SHOW CHARACTER\_SET 命令显示 DEC 汉字字符集。

### 5.14 SHOW DATABASE

展示当前数据库以及所有支持的数据库名。

#### 语法

```
SHOW DATABASE [/qualifiers]
```

---

命令限定词

默认

*/ALL*

#### 限制

没有。

#### 提示

没有。

#### 描述

SHOW DATABASE 命令展示当前数据库名。

要展示系统上的所有数据库名，可使用 */ALL* 限定词。这些名字会以表的格式展示。

#### 命令限定词

*/ALL*

如果指定这个限定词，便会展示系统中的所有数据库名。当前数据库会有箭头标记。

#### 例子

1

```
CMGR> SHOW DATABASE
```

这个命令展示当前数据库。当前数据库可能是 CMGR\_DEFAULT、SONG 或 HEI。

### 5.15 SHOW TABLE

显示当前系统属性数据库中指定区内或用户属性文件中的字符代码表。

## 语法

```
SHOW TABLE [/qualifiers] [/positional-qualifiers]
            [section-number[/positional-qualifiers][,...]]
```

命令限定词	默认
/COLLATING	/FULL
/FONT	/FULL
/FULL	/FULL
/OUTPUT=output-file	/OUTPUT=SYS\$OUTPUT
/RESERVED	None
/SIZE=(font-size[,...])	/SIZE=ALL
/SYSTEM	/SYSTEM
/USER=user-attribute-file	/SYSTEM
定位限定词	默认
/QUADRANT=quadrant-name	/QUADRANT=ALL

## 限制

没有。

## 提示

\_Section No. [section-number [/positional-qualifiers] [...]]

## 参数

[section-number [/positional-qualifiers] [...]]

区号或区范围的列表。区号是从 1 到 94 的整数。区范围是两个区号，以连字符分隔，中间无空格。

星号 “\*” 可用来指定一个象限的所有区。也就是说，“\*” 等于 “1-94”。

区号的解释取决于它所在的象限。默认是 DEC 汉字字符平面的左下象限。要指定右下象限中的区，可使用 /QUADRANT 定位限定词。

/QUADRANT 限定词可置于两处以修改区号的意义。它可以置于 “TABLE” 一字之后及区号列表之前。这样可把限定词应用到整个区号列表。/QUADRANT 限定词也可以与每个区号或区范围并排放置。这样仅把限定词应用到该具体的区号或区范围。第二种情形会取代第一种情形。

在命令行中指定 /USER 时，便不允许使用这个参数。

## 描述

SHOW TABLE 命令显示当前系统属性数据库中指定区内或用户属性文件中的字符代码表。这些代码表展示当前系统属性数据库或用户属性文件中有关字符属性的资料。

## CMGR 命令参考

### 5.15 SHOW TABLE

要显示当前系统属性数据库的代码表，可使用 /SYSTEM 限定词。注意，当前数据库可用 SET DATABASE 命令来选择。

要显示用户属性文件的代码表，可使用 /USER 限定词。

有关所有区中的字形模式资料可随时显示。然而，有关其他字符属性的资料，则只可显示具有用户定义字符和保留用户定义字符的那些区的资料。（如果指定了 /RESERVED）。

#### 命令限定词

##### **/COLLATING**

这个字符属性限定词指定要显示有关整理特征的资料。

这个限定词不可以与 /FULL 一起使用。

##### **/FONT**

这个字符属性限定词指定要显示有关字符模式的资料。

这个限定词不可以与 /FULL 一起使用。

##### **/FULL**

指定要显示有关所有字符属性的资料。

如果在命令行中没有指定字符属性限定词，就假定为 /FULL。

这个限定词不可以与其他字符属性限定词一起使用。

##### **/OUTPUT=output-file**

代码表通常打印到 SYS\$OUTPUT。使用这个限定词，用户可以指示把它打印到输出文件。

##### **/RESERVED**

要显示保留用户定义字符的所有属性时，可指定这个限定词。

如果没有指定这个限定词，便只显示保留用户定义字符的字形模式是否存在。

##### **/SIZE=(font-size[,...])**

指定要展示的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。用户也可以使用关键字 'ALL' 来表示要展示所有支持的字形大小。

默认是 /SIZE=ALL。

这个限定词仅与 /FONT 或 /FULL（默认）一起使用时才有效。

##### **/SYSTEM**

使用 SHOW TABLE 来打印当前系统属性数据库中的代码表。要使用这个限定词，就必须指定区号列表参数，然后便会打印指定区的代码表。

这个限定词不可以与 /USER 一起使用。

在命令行中没有指定 /SYSTEM 或 /USER 时，就假定为 /SYSTEM。

##### **/USER=user-attribute-file**

使用 SHOW TABLE 来打印指定用户属性文件的代码表。要使用这个限定词，不可指定区号参数。用户属性文件中所有字符属性的资料将被打印。

要显示一个或以上的属性，user-attribute-file 的文件名就不能有扩充名，因为会使用默认的扩充名。

支持的用户属性文件：

字符属性	文件扩充名	文件类型
字形模式	.PRE	预装入文件
整理特征	.CVD	整理数据文件

这个限定词不可以与 /SYSTSEM 一起使用。

#### 定位限定词

**/QUADRANT=quadrant-name**

指定要使用字符平面上的哪个象限。象限名可接收的值是 LOWER\_LEFT 和 LOWER\_RIGHT。用户也可以使用 LL 或 1 来代表 LOWER\_LEFT，使用 LR 或 0 来代表 LOWER\_RIGHT。

默认是 /QUADRANT=LL

这个限定词不可以与 /USER 一起使用。

#### 例子

1

```
CMGR> SHOW TABLE/FONT/COLLATING 1/QUADRANT=1
```

这个命令显示当前系统属性数据库左下象限中区 1 的代码表。所有大小的字形模式和整理特征都将展示表中。

2

```
CMGR> SHOW TABLE/COLLATING/USER=MYFILE.CVD/OUTPUT=MYTABLE.LIS
```

这个命令把整理数据文件 'myfile.cvd' 的代码表打印到文件 'mytable.lis'。代码表包含整理数据文件中定义的所有整理特征的资料。

## 5.16 SHOW VERSION

显示 CMGR 的软件版本号。

#### 语法

**SHOW VERSION**

**命令限定词**

没有。

**默认**

没有。

#### 限制

没有。

#### 提示

没有。

#### 参数

没有。

## CMGR 命令参考

### 5.17 UPDATE

#### 描述

SHOW VERSION 命令显示 CMGR 的软件版本号。  
当用户需要知道他们使用的 CMGR 的版本时，这个命令便很有用。

#### 限定词

没有。

#### 例子

1

```
CMGR> SHOW VERSION
```

SHOW VERSION 命令显示 CMGR 的当前版本号。

### 5.17 UPDATE

使用用户属性文件中定义的字符属性来更新当前系统属性数据库。

#### 语法

```
UPDATE [/qualifiers] user-attribute-file[,...]
```

---

命令限定词	默认
<i>/COLLATING</i>	<i>/FULL</i>
<i>/[NO]CONFIRM[=<i>condition</i>]</i>	<i>/CONFIRM=ALL</i>
<i>/FONT</i>	<i>/FULL</i>
<i>/FULL</i>	<i>/FULL</i>
<i>/[NO]LOG</i>	<i>/LOG</i>
<i>/RESERVED</i>	<i>None</i>
<i>/SIZE=(<i>font-size[,...]</i>)</i>	<i>/SIZE=ALL</i>

#### 限制

- 用户必须有 SYSPRV 特权才能执行这个命令。

#### 提示

\_Input File: *user-attribute-file[,...]*

#### 参数

***user-attribute-file[,...]***

用户属性文件的文件名列表。要更新一个以上的属性，文件名不能有扩充名，因为会添加默认的扩充名。

支持的用户属性文件：

字符属性	文件扩充名	文件类型
字形模式	.PRE	预装入文件
整理特征	.CVD	整理数据文件

## 描述

UPDATE 命令用存储在用户属性文件中的字符属性插入或替换当前系统属性数据库的字符属性。如果没有指定 /RESERVED, 便只可更新用户定义字符。否则, 可更新用户定义字符和保留用户定义字符。

该命令会顺序地读取用户属性文件中找到的字符属性, 并在系统属性数据库中逐一更新这些属性。

按照默认, 会更新所有字符属性。然而, 用户可以明确地指定要更新哪个字符属性。如果已更新字形模式, 用户可以指定哪些字形大小要更新。

注意, 字符属性会更新到当前数据库, 而当前数据库可用 SET DATABASE 命令来设置。

## 命令限定词

### /COLLATING

这个字符属性限定词指定要更新的整理特征。

这个限定词不可以与 /FULL 一起使用。

### /CONFIRM[=condition]

#### /NOCONFIRM

指定是否需要用户确认。确认级取决于条件。可能的条件有:

ALL	对每个事项都要求确认。
CONFLICT	如果相应字符属性已在目的地代码中登记, 就要求确认。

默认是 /CONFIRM=ALL。

回应确认请求时可接受的答案是:

YES, TRUE, 1	肯定的回答
NO, FALSE, 0	否定的回答
<RETURN>	默认的回答 (否定)
<CTRL/Z>, QUIT	终止该命令

回答可以缩写。

### /FONT

这个字符属性限定词指定要更新的字形模式。

这个限定词不可以与 /FULL 一起使用。

## CMGR 命令参考

### 5.17 UPDATE

#### **/FULL**

指定要更新所有字符属性。

如果在命令行中没有指定字符属性限定词，就假定为 /FULL。

这个限定词不可以与其他字符属性限定词一起使用。

#### **/LOG**

#### **/NOLOG**

控制是否显示通知信息。

默认是 /LOG。

#### **/RESERVED**

要更新保留用户定义字符的属性时，可指定这个限定词。

#### **/SIZE=(font-size[,...])**

指定要更新的字形大小。font-size 的有效值是 24、32 和 40，它们分别表示 24x24、32x32 和 40x40 的字形大小。用户也可以使用关键字 'ALL' 来表示要更新所有支持的字形大小。

默认是 /SIZE=ALL。

这个限定词仅与 /FONT 或 /FULL（默认）一起使用时才有效。

### 例子

1

```
CMGR> UPDATE/FONT/COLLATING MYFILE
```

这个命令用 'myfile.pre' 中定义的字形模式和 'myfile.cvd' 中定义的整理特征来更新当前系统属性数据库。它将请求确认，并且通知用户更新是成功还是失败。

2

```
CMGR> UPDATE/NOLOG/CONFIRM=CONFLICT/FONT/SIZE=24 MYFILE.PRE
```

这个命令用 'myfile.pre' 中定义的 24x24 大小的字形模式来更新当前系统属性数据库。除非发生错误，否则不打印信息。此外，除非目的地代码已定义字形模式，否则不会请求确认。

## 第 6 章

# 字形编辑程序参考

CMGR 提供一个全屏幕字形编辑程序以建立和修改字符字形模式。编辑程序也允许用户输入和更改用户定义字符的整理特征。本章提供使用字形编辑程序的全面参考资料。同时，也在章末提供一个快速参考，介绍所有字形编辑程序的命令（键关联）。

### 6.1 终端支持

VT382 是 CMGR 字形编辑程序支持的终端类型。字形编辑程序启动时，会把一些字块装入 VT382 终端的动态浮动字符集 (DRCS)。字块是一些图形字符，用以显示编辑程序屏幕的编排设计。如果在编辑对话期间重置终端，字块便会丢失，屏幕显示也会搞乱。如要恢复，必须先退出编辑程序然后重新启动。

字形编辑程序要求 VT382 的小键盘态需要设置为“应用小键盘”。在调用 CMGR 时，其实已设置了该小键盘态。所以，如果小键盘在 CMGR 对话期间给设置为“数字小键盘”，用户便必须把它重置为“应用小键盘”以便编辑程序能恰当地操作。

除上述情况外，通常只需要按 <CTRL/W> 便可恢复编辑程序屏幕。

### 6.2 调用字形编辑程序

有两种调用字形编辑程序的方式：

1 在 CMGR 子系统级，

```
$ CMGR
CMGR> EDIT TEMP
%CMGR-W-NOSUCHFIL, 没有这文件, DISK$USER:[MYDIR]TEMP.PRE;
%CMGR-W-NOSUCHFIL, 没有这文件, DISK$USER:[MYDIR]TEMP.CVD;
%CMGR-I-LOADFONT, 正在把字块装入您的终端, 请稍候...
```

2 在 DCL 级，

```
$ CHARACTER_MANAGER EDIT TEMP
%CMGR-W-NOSUCHFIL, 没有这文件, DISK$USER:[MYDIR]TEMP.PRE;
%CMGR-W-NOSUCHFIL, 没有这文件, DISK$USER:[MYDIR]TEMP.CVD;
%CMGR-I-LOADFONT, 正在把字块装入您的终端, 请稍候...
```

有关 EDIT 命令的详情，请参阅第 5 章中的全面描述。

#### 6.2.1 字符属性

OpenVMS/Hanzi 的字符系统中，每个字符代码都与若干个字符属性相关。CMGR 是一个公用程序，用以操纵用户定义字符和保留用户定义字符的这些属性。CMGR 当前支持的属性有：

- 字形模式

## 字形编辑程序参考

### 6.3 内部操作

- 整理特征

为了要在 CMGR 命令中指定操纵哪个属性，于是定义了一些字符属性限定词。此外，为了便于在系统属性数据库以外操纵这些属性，便使用预定义格式的文件来存储这些属性。这些文件合称为用户属性文件。

表 6-1 支持的字符属性

字符属性	限定词	文件类型	文件扩充名
字形模式	/FONT	预装入文件	.PRE
整理特征	/COLLATING	整理数据文件	.CVD

字形模式以预装入序列来存储，整理数据则以文本文件格式的编码数字来存储。

CMGR 字形编辑程序支持编辑 24x24、32x32 和 40x40 大小的字形模式以及整理特征。用户可以使用该编辑程序来建立或修改预装入文件和整理数据文件。用户输入命令启动编辑程序时，可以使用相应的属性限定词来指定编辑哪个属性。

#### 6.2.2 装入字块

如 6.1 节的描述，编辑程序使用一些装入到 VT382 动态浮动字符集中的字块来显示它的屏幕编排设计。字块装入以后，除非把终端重置，否则它们不会丢失。因此，无须在每次调用编辑程序时都执行装入。此外，装入需要花一段颇长的时间，因此，CMGR 允许用户选择装入字块或不装入字块。如果确已装入字块，输入以下命令以避免重新装入：

```
CMGR> EDIT/NOLOAD_GLYPH TEMP
```

### 6.3 内部操作

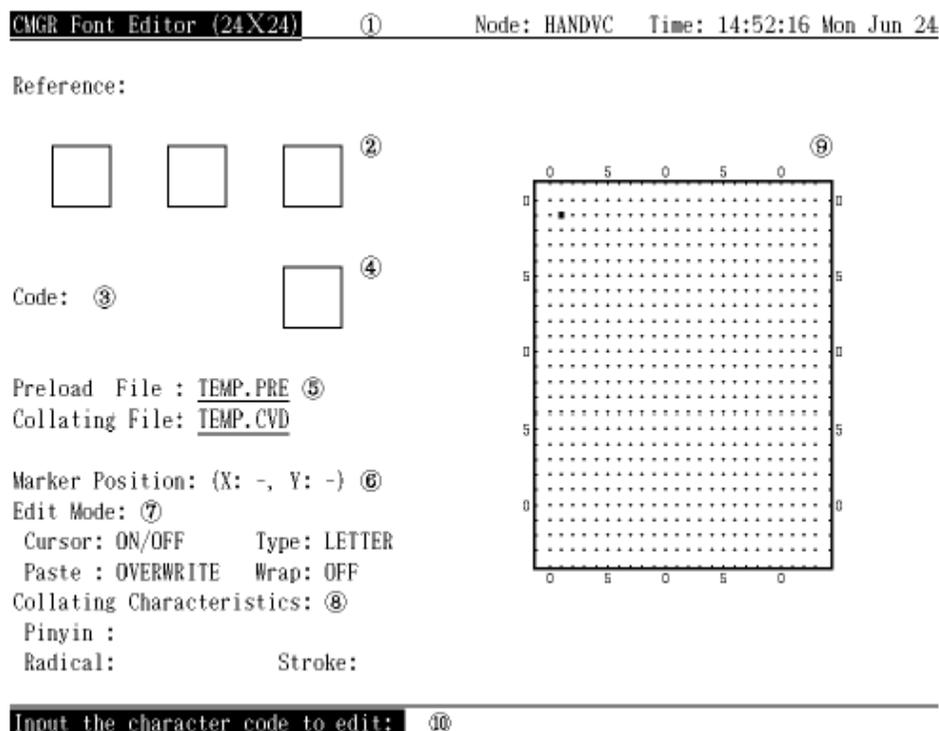
字形编辑程序会根据指定的属性限定词来打开指定的预装入文件和 / 或整理数据文件。如果文件存在，并为有效的预装入文件和整理数据文件，字形编辑程序便会把所有预装入序列和整理数据读入内存。从此之后，便只会用内存记录来执行所有的处理。

用户正常退出字形编辑程序时，或按 <GOLD> + <PF3> 来输出编辑结果时 (6.6.15 节)，便会建立新的预装入文件和 / 或整理数据文件以存储编辑结果。

### 6.4 屏幕编排设计

在 VT382 终端上成功地启动编辑程序以后，用户应该可以看见一个类似图 6-1 的屏幕：

图 6-1 编辑程序编排设计



1. 标题行：  
屏幕的顶行是标题行，包含如正编辑的字形大小、CMGR 正在其上运行的机器（节点名）以及当前时间等资料。时间不会校准至秒，但会大约每分钟更新一次。
2. REFER 窗口：  
标题“参考”下面的左上角有三个小窗口，是给 REFER 功能用来显示当前系统属性数据库或内存记录的字形的。字形按其实际大小显示，用户在设计新字形时可以这些字形作为参考。有关如何使用 REFER 的详情，请参阅 6.6.5 节中的描述。
3. 代码：  
REFER 窗口下面有一个字段，用以展示正在处理哪个字符代码。区位码和内码都会显示出来。有关如何输入代码的详情，请参阅 6.6.1 节。
4. DISPLAY 窗口：  
在三个 REFER 窗口下面另有一个小窗口。这窗口用于显示当前正在编辑的字形。字形按实际大小显示。用户可通过此窗口看见字形实际上在终端上使用时的样子。有关如何 DISPLAY（显示）字形的详情，请参阅 6.6.5 节。
5. 用户属性文件  
在代码和 DISPLAY 窗口下面，展示了用户正在处理的文件。如果有空间的话，便会显示完整文件名，否则仅显示文件名及其文件扩充名。
6. 标记位置：  
标记位置指明 EDIT 或 USE 缓冲区的标记的 X-Y 坐标。标记由 Mark 功能设置（请参阅 6.6.9 节）。坐标的原点是左上角。X 坐标从左到右增加，Y 坐标则向下增加。

## 字形编辑程序参考

### 6.4 屏幕编排设计

#### 7. 编辑态

共有四个编辑态，用户可选择设置其中一个来控制编辑功能的特征。光标态可以是 ON、OFF、ON/OFF 或 MOVE，它控制光标移动时是否会开关光标下的像素。贴补态可以是 OVERLAY 或 OVERWRITE，它控制贴补操作的结果。类型态可以是 BODY 或 LETTER，它指定是否保留包围字符的边界。绕回态可以是 ON 或 OFF，它允许或撤销绕回光标。此外，光标态和贴补态可以影响几何图形的绘画。有关这些态的详情，请参阅 6.6.4 节。

#### 8. 整理特征：

如果同时编辑整理特征和字形，在“整理特征”的标题下面便会有三个字段。这三个字段是支持的整理特征，即拼音、部首和笔画。如果其中一个特征采用默认值，便显示一个“-”，否则便显示该特征的值。

#### 9. 主窗口

右手边的是主窗口。用户可在该窗口内移动光标或使用所提供的编辑功能（如剪切与贴补、自动建立几何图形等）来建立或修改字形。窗口内的每一点代表一个像素，因此，24x24 大小的字形便有 24x24 的点矩阵。

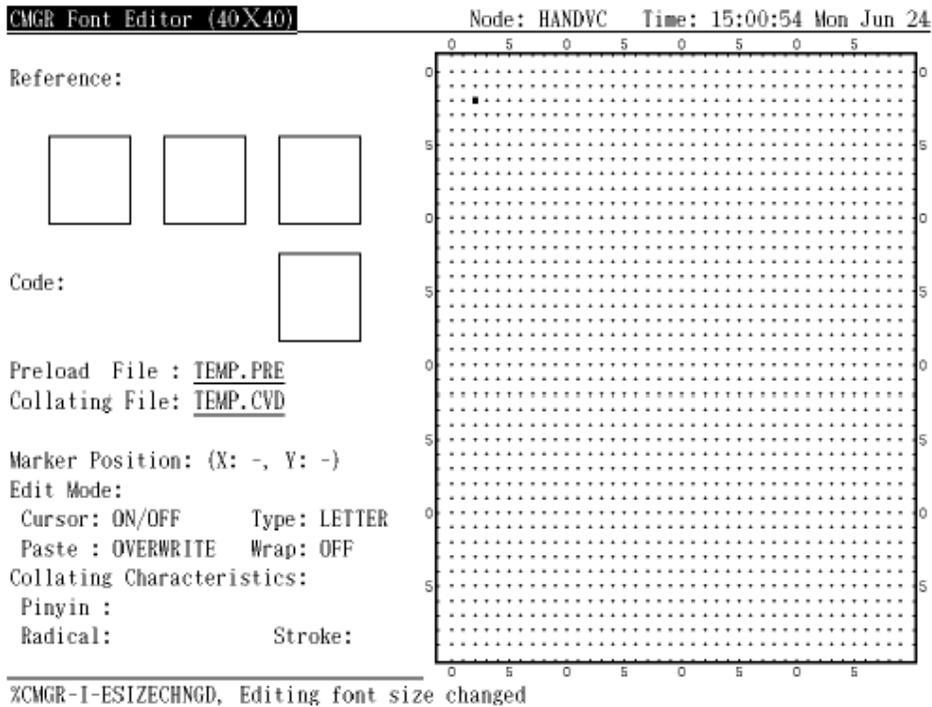
窗口周围的数字帮助用户找出矩阵内每一点的 X-Y 坐标。X 坐标从左到右地增加，而 Y 坐标则向下增加。左上角是原点。

#### 10. 信息和输入窗口

屏幕的底行用以显示信息，也让用户输入字形编辑程序请求的资料。

字形编辑程序也支持编辑 32x32 和 40x40 大小的字形。图 6-2 展示编辑 40x40 大小的字形时的编排设计。图 4-20 是编辑 32x32 大小的字形时的编排设计。有关如何编辑大小的详情，请参阅 6.6.12 节。有关如何转换字形大小的详情，请参阅 6.6.13 节。

图 6-2 编辑 40x40 大小字形时的编辑程序编排设计



## 6.5 缓冲区

CMGR 字形编辑程序中的缓冲区是用以存储位映象的区域。共有三个缓冲区：

- EDIT 缓冲区是用户执行编辑的主缓冲区，通常出现在主窗口（6.4 节）上。此缓冲区也保存正在编辑的整理特征。
- USE 缓冲区是调用 USE 功能时（6.6.6 节）出现在主窗口上的缓冲区，用以保存从当前系统属性数据库或内存记录装入的字形模式。在此缓冲区上，只可执行剪切与贴补操作。
- 第三个缓冲区是 CUT & PASTE 缓冲区。CUT & PASTE 缓冲区的内容是不可见的，但用户可以从其他两个缓冲区剪切一个字形模式到此缓冲区，或者把此缓冲区的内容贴补到其他两个缓冲区（6.6.8 节）。

## 6.6 编辑功能

CMGR 字形编辑程序提供一组功能以帮助用户编辑他们的字形。所有这些功能都可在键盘上按一、两个键来调用。在本节中，您可以找到有什么支持的功能，这些功能是什么以及如何调用它们。

### 6.6.1 启动

在用户开始建立字形或输入一些整理值之前，他们必须首先指定与该字形或整理值相关的字符代码。要指定该代码，可使用应用小键盘上的 <Enter> 键。按 <Enter> 之后，会有提示询问代码。如果输入了用户定义区域或保留用户定义区域<sup>1</sup>中的一个有效代码，编辑便会启动。有关如何在

1. 如果调用编辑程序时指定了 /RESERVED。

## 字形编辑程序参考

### 6.6 编辑功能

CMGR 中指定有效代码的详情，请参阅 4.1 节。要编辑另一个代码的属性，再次按 <Enter> 并指定该代码。当前的更改会保存（6.6.16 节）在内存中，然后可开始编辑另一代码。

#### 6.6.2 Gold 状态

有些功能是以两次击键来调用的。第一个键是称为 GOLD 键的 <PF1>。按 GOLD 键之后，便处于 COLD 状态，小键盘上的键定义都会更改。按了第二个键后，GOLD 状态便会自然被清除。一般来说，用户也可再按一次 <PF1> 来清除 GOLD 状态。

#### 6.6.3 移动光标

在主窗口中移动光标是建立或修改字形的基本方法。以下小节描述三组用以移动光标的键。

##### 6.6.3.1 步进

除了 <KP5> 以外，<KP1> 至 <KP9> 是用以移动光标的一组基本键。光标移动时，会把它下面的象素 ON 或 OFF，这要视乎当时的光标态。有关详情，请参阅 6.6.4.1 节。以 <KP5> 为中心的八个键正好对应于光标移动的八个方向。每次击键都会把光标往那方向移动一个象素。能否移入或移过边界则取决于类型态和绕回态。有关详情，请参阅 6.6.4.3 节和 6.6.4.4 节。

按 <KP5> 可互换光标下面象素的状态，但不移动光标。

##### 6.6.3.2 仅移动

编辑小键盘上的四个箭头键可以用来移动光标，但它们是独立于光标态设置的。用户可以用它们来把光标移向四个方向而不更改任何象素的状态。这四个箭头键即等于光标态设置为 MOVE 时的 <KP2>、<KP4>、<KP6> 和 <KP8>。

##### 6.6.3.3 跳跃

要以较少的击键来把光标从当前位置移到较远的位置，可使用转移键。转移键是在 Gold 状态中的 <KP1> 至 <KP9>。<GOLD> + <KP1>、<GOLD> + <KP3>、<GOLD> + <KP7> 和 <GOLD> + <KP9> 把光标跳到四个角。<GOLD> + <KP2>、<GOLD> + <KP4>、<GOLD> + <KP6> 和 <GOLD> + <KP8> 把光标跳到四边的中心。最后，<GOLD> + <KP5> 把光标定位到主窗口的中心。

#### 6.6.4 编辑态

共有四个编辑态，即光标态、贴补态、类型态和绕回态。这些编辑态会影响一些编辑功能的特征。

##### 6.6.4.1 光标态

光标态设置有 ON、OFF、ON/OFF 或 MOVE，可使用 <PF3> 来互换。光标态处于 ON 态时，以小键盘键来移动光标（6.6.3.1 节）可开启光标下的象素。光标态处于 OFF 态时，便会关掉象素。光标态是 ON/OFF 时，便会把象素从开互换为关或从关互换为开。最后，MOVE 态允许用户移动光标而不更改任何象素的状态。

光标态也会影响绘画几何图形。有关详情，请参阅 6.6.10 节。

##### 6.6.4.2 贴补态

贴补态影响剪切与贴补中的贴补操作。贴补态可以是 OVERLAY 或 OVERWRITE。在 OVERLAY 态，PASTE（贴补）缓冲区的模式便叠加在编辑缓冲区中已经存在的模式上。在 OVERWRITE 态中，PASTE 缓冲区的模式将重写原来的模式上。

贴补态是以 <PF4> 来互换。

贴补态也影响绘画几何图形。有关详情，请参阅 6.6.10 节。

### 6.6.4.3 类型态

类型态可以是 LETTER 或 BODY。一般情况下，当建立中文字符以供在终端上显示时，通常会在每个字符的周围留空一条空白的边界，这样邻接的字符便不会相连。然而，对图形字符（如画线字符）来说，便可能不需要有边界。在第一种情形中，用户可以设置类型态为 LETTER，不允许编辑最外边的像素。在第二种情形中，类型态可以设置为 BODY 以便编辑整个位映象。

注意，24x24 大小字形的边界为一个象素宽，而 32x32 和 40x40 大小字形的边界则为两个象素宽。

类型态可由 <KP-> 来互换。

### 6.6.4.4 绕回态

如果绕回态是 ON，当用户试图移动光标超越边界时，光标便从位映象的一边跳到另一边（绕回）。如果绕回态是 OFF，光标便停留在边界上。

绕回态可由 <KP,> 来互换。

### 6.6.5 DISPLAY 与 REFER 窗口

EDIT 缓冲区中的字形模式可以按其实大小显示在 DISPLAY 窗口中（6.4 节）。按 <KP.> 或 <CTRL/R> 便可执行以下操作。

为了帮助用户建立或修改字形，编辑程序提供了 REFER 功能。REFER 窗口每次最多只可显示当前系统属性数据库或内存记录中的三个不同的字形。（6.4 节）。按 <Gold> + <KP.> 便有提示要求用户输入要引用的代码。在应答中，用户应输入一个关键字（任选的）和一个字符代码。有效的关键字有“DATABASE”和“MEMORY”，可以缩写及用小写。如果没有指定关键字，默认是“DATABASE”（见图 4-11）。

如果关键字是“DATABASE”或没有指定关键字，并输入了一个用户定义区域或保留用户定义区域<sup>1</sup>内的有效代码，而在当前系统属性数据库中已具该字形的定义，该字形便会按其实大小显示在其中一个 REFER 窗口中。

同样地，如果关键字是“MEMORY”，并输入了一个有效的代码，而内存记录中已具有该字形，那么，该字形便会显示在其中一个 REFER 窗口中（见图 4-12）。

### 6.6.6 USE 缓冲区

编辑程序允许用户把当前系统属性数据库中定义的字形或内存记录中的字形检索到 USE 缓冲区。在那里，用户可以把字形的一部分剪切到 Paste 缓冲区，然后把它贴补到 EDIT 缓冲区。

按 <KP0>，便会有提示要求输入一个字符代码。在应答中输入一个关键字（任选的）和一个字符代码。有效的关键字有“DATABASE”和“MEMORY”，可以缩写和用小写。如果没有指定关键字，默认是“DATABASE”（见图 4-7）。

如果关键字是“DATABASE”或没有指定关键字，并输入了一个用户定义区域或保留用户定义区域<sup>2</sup>中的有效代码，而当前系统属性数据库已具有该字形的定义，该字形便会给检索到 USE 缓冲区。USE 缓冲区会显示在主窗口中（6.4 节）。

---

1. 如果在调用编辑程序时指定了 /RESERVED。  
2. 如果调用编辑程序时指定了 /RESERVED。

## 字形编辑程序参考

### 6.6 编辑功能

同样地，如果关键字是“MEMORY”，并输入了一个有效代码，而内存记录已具有该字形，那么，该字形便会装入 USE 缓冲区并显示在主窗口中（6.4 节）。

现在，部分或全部的装入字形便可剪切到 Paste 缓冲区内，但不可用光标移动键来更改 USE 缓冲区中字形。

要从 USE 缓冲区返回 EDIT 缓冲区，可按 <GOLD> + <KP0>。

#### 6.6.7 选择区域

有些编辑程序操作只可在选择区域上执行。选择区域是主窗口内以反相显示的矩形区域。要开始选择，按编辑小键盘上的 <Select> 或主小键盘上的 <s>。光标的当前位置是该选择区域的一角。接着，把光标移到另一个位置。该位置便是选择区域的对角，现以反相显示。要取消选择，可再按 <Select> 或 <s>。

要选择整个位映象，按 <GOLD> + <Select> 或按 <GOLD> + <s>。击键两次便可选择整个位映象。

#### 6.6.8 剪切与贴补

编辑程序提供一个 Paste 缓冲区以使用户把一部分的模式剪切到缓冲区，随后再把模式贴补到另一位置。要把模式剪切到该缓冲区，首先选择一个区域（6.6.7 节），然后按 <Remove> 或 <c>。选择区域中的模式便从当前缓冲区中消失，并存储到 Paste 缓冲区。要把该模式贴补到当前缓冲区，可使用 <Insert Here> 或 <p> 键。

如果当前缓冲区没有足够空间插入 Paste 缓冲区的整个位映象，它会在执行插入之前要求您确认。如果要插入的话，也只会插入部分的位映象。

#### 6.6.9 标记

编辑程序提供一个 MARK 功能来协助编辑。用户可按 <GOLD> + <m> 来标记 EDIT 缓冲区或 USE 缓冲区的当前光标位置。标记坐标会展示于标记位置下面（6.4 节）。稍后，用户只需要击键两次（也就是按 <GOLD> + <g>），便可从缓冲区的任何位置跳到标记位置。

标记位置的 X-Y 坐标取左上角作为原点。X 坐标从左至右增加。Y 坐标则向下增加。

#### 6.6.10 几何图形

编辑程序提供一些能助用户绘画几何图形的功能。支持的几何图形有线、矩形和椭圆形。图形只可在用户选择的矩形区域内（6.6.7 节）绘画，结果则取决于当前的光标态和贴补态设置（6.6.4.1 节和 6.6.4.2 节）。

要画一个图形，首先挑选适当的光标态和贴补态。一般规则是：如果光标态是 ON，图形便以 ON 像素的方式画出；如果光标态是 OFF，图形将用 OFF 像素的方式画出；如果光标态是 ON/OFF，图形将用 ON 像素互换为 OFF 像素和 OFF 像素互换为 ON 像素的方式画出；如果光标态是 MOVE，便不会画出任何图形。如果是空心图形（空心矩形和空心椭圆形），贴补态会确定该空心图形是重写还是重叠原来的像素。有关态如何影响绘画图形功能的详情，请参阅下列的表。

表 6-2 编辑态与画几何图形

光标态	贴补态	画图形功能的特征
ON	OVERWRITE	图形以 ON 像素的方式画出。如果是空心矩形或空心椭圆形，图形内原来的模式便会以 OFF 像素重写。

表 6-2 (续) 编辑态与画几何图形

ON	OVERLAY	图形以 ON 象素的方式画出。如果是空心矩形或空心椭圆形，图形内原来的模式不会更改。
OFF	OVERWRITE	图形以 OFF 象素的方式画出。如果是空心矩形或空心椭圆形，图形内原来的模式便会以 ON 象素重写。
OFF	OVERLAY	图形以 OFF 象素的方式画出。如果是空心矩形或空心椭圆形，图形内原来的模式不会更改。
ON/OFF	OVERWRITE	图形以 ON 互换为 OFF 和 OFF 互换为 ON 象素的方式来画出。如果是空心的矩形或空心的椭圆形，图形内原来的模式不会更改。
ON/OFF	OVERLAY	同上
MOVE	OVERWRITE	不画任何图形
MOVE	OCERLAY	不画任何图形

设置适当的态后，开始选择一个区域，然后按下面列出的任何一个键来绘画对应的图形。

键	几何图形
'L', 'l'	一条线
'B', 'b'	一个空心矩形
'W', 'w'	一个实心矩形
'O', 'o'	一个空心椭圆形
'Q', 'q'	一个实心椭圆形

### 6.6.11 特殊功能

有几个特殊的编辑功能可助用户建立或修改字形，包括在选择区域内倒置象素状态的功能和环绕轴线对称地重新排列模式的功能。

所有这些功能都只能在选择区域内操作。有关如何建立选择区域的详情，请参阅 6.6.7 节。选择之后，按下面的其中一个键来执行所需的功能。

键	特殊功能
'I', 'i'	倒置选择区域中的象素状态
','	在选择区域中，环绕纵向中心线，对称地重新排列模式
'-'	在选择区域中，环绕横向中心线，对称地重新排列模式
'/'	在选择区域中，环绕连接左下角到右上角的对角线，对称地重新排列模式。选择区域必须是正方形。

## 字形编辑程序参考

### 6.6 编辑功能

'\' 在选择区域中，环绕连接左上角到右下角的对角线，对称地重新排列模式。  
选择区域必须是正方形。

---

#### 6.6.12 编辑不同大小的字形

字形编辑程序支持编辑 24x24、32x32 和 40x40 大小的字形。用户第一次进入编辑程序时，所编辑的是 24x24 大小的字形。要改为编辑其他大小，可按编辑小键盘上的 <Prev Screen> 或 <Next Screen>。字形大小依次是 24x24、32x32、40x40，然后再返回 24x24。按 <Prev Screen> 便反向前去下一个大小。按 <Next Screen> 则正向前去下一个大小。之后，下一个大小的字形便会装入主窗口以供编辑。

#### 6.6.13 转换不同大小的字形

字形编辑程序让用户把一个种小的字形转换成同样字符代码但有另一大小的字形。假定用户建立了大小为 24x24 的字形，然后，他要建立大小为 32x32 的字形。他按 <Next Screen> (6.6.12 节) 后再按 <F18>，便有提示要求输入字形转换前的大小。他应回答 '24' 并按 <Return>，字形便被转换并装入主窗口以供进一步编辑。

#### 6.6.14 输入整理值

只有在命令行指定了 /COLLATING 或 /FULL (默认) 限定词后，编辑程序才允许输入整理值。

按 <F17> 便有提示要求用户给每个支持的整理特征输入整理值。在应答中，用户应输入有效的整理值。如果没有输入任何值，便不更改当前值。如果输入 '-'，便会把该整理特征设置为默认值。

附录 A 列出所支持的整理特征的有效整理值。

#### 6.6.15 输出、删除、取消

正如 6.3 节和 6.6.16 节所述，所有修改都只会保存在内存的记录中。要把字符属性输出到用户属性文件而不终止编辑对话期，用户应按 <GOLD> + <PF3>。当前更改便会保存到内存记录，再把所有内存记录写入新版本的用户属性文件。

要删除当前字符代码的内存记录，用户应按 <GOLD> + <PF4>。系统会要求确认。如果回答是 'y'，该字符代码的所有字符属性都会从内存除去。如果回答是 'n'，便会询问是否要删除该字符代码的某一具体属性。这样，用户便可以选择只删除字形或整理值。

如要取消对当前字符代码所作的更改，可按 <GOLD> + <KP->。系统会要求确认。如果回答是 'yes' 或 'y'，便可取消从输入该字符代码 (6.6.1 节) 至按 <GOLD> + <KP-> 期间所作的任何更改。然而，如果在 '取消' 前已执行 '保存' (6.6.16 节)，在保存之前所作的更改便不会取消。

#### 6.6.16 保存

调用编辑程序时，会把用户属性文件中的所有字符属性读入内存 (内存记录)。用户按 <Enter> 来开始编辑字符代码后，只可在 EDIT 缓冲区中执行修改。内存记录不受影响。只有在执行 '保存' 后，才会用这些对字符属性所作的修改来更新内存记录。然而，没有明确的命令来执行 '保存'。'保存' 操作会在用户执行下列任何一项时自动执行。

- 按 <Enter> 以编辑另一字符代码的属性 (6.6.1 节)。
- 按 <GOLD> + <PF3> 来把内存记录输出到用户属性文件 (6.6.15 节)。
- 按 <GOLD> + <Enter> 或 CTRL/Z 以退出字形编辑程序 (6.6.18 节)。

### 6.6.17 取得求助

用户可以按 <Help> 或 <GOLD> + <PF2> 来取得关于如何使用编辑程序的求助。要显示小键盘图，用户应按 <PF2>。

### 6.6.18 终止字形编辑程序

按 <CTRL/Z> 或 <GOLD> + <Enter> 便可正常地终止编辑对话期。当前 EDIT 缓冲区内的更改会保存下来，所有内存记录也都会写入用户属性文件的新版本中。然后，系统会返回 DCL 级或 CMGR 子系统级，视调用编辑之处而定。

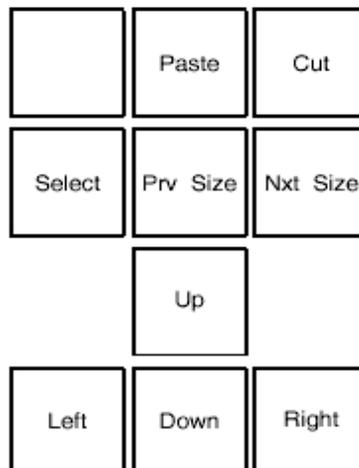
用户也可按 <CTRL/C> 或 <GOLD> + <KP,> 来异常终止编辑对话期。所不同的是系统不会把内存记录输出到文件。

## 6.7 快速参考

本节包含几个小键盘图和一个 CMGR 字形编辑程序中所有命令的列表。每个命令下面的描述只供快速观看之用。有关详情，请参阅 6.6 节中的描述。

### 6.7.1 小键盘图

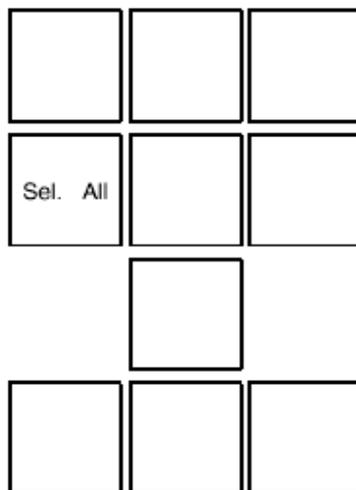
编辑小键盘



## 字形编辑程序参考

### 6.7 快速参考

#### GOLD 状态的编辑小键盘



#### 应用小键盘



GOLD 状态的应用小键盘

Gold	Help	Output	Delete
ULcorner	Top	URcorner	Cancel
L side	Center	R side	Quit
LLcorner	Bottom	LRcorner	Exit
Return from USE		REFER	

6.7.2 字形编辑程序功能列表

表 6-3 应用小键盘

键	缩写	功能
<Help>		提供关于 EDIT 的求助
<F17>		输入整理值
<F18>		把字形转换成当前大小
<PF1>	Gold	互换 GOLD 状态
<PF2>	Key Help	提供关于 EDIT 小键盘的求助
<PF3>	Curs Mo.	互换光标态
<PF4>	Past Mo.	互换贴补态
<KP->	Type Mo.	互换类型态
<KP,>	Wrap Mo.	互换绕回态
<Enter>	Code	输入字符代码
<KP7>	U&L	依据光标态，向左上移动

## 字形编辑程序参考

### 6.7 快速参考

表 6-3 (续) 应用小键盘

<KP8>	UP	依据光标态, 向上移动
<KP9>	U&R	依据光标态, 向右上移动
<KP4>	Left	依据光标态, 向左移动
<KP6>	Right	依据光标态, 向右移动
<KP1>	D&L	依据光标态, 向左下移动
<KP2>	Down	依据光标态, 向下移动
<KP3>	D&R	依据光标态, 向右下移动
<KP5>	Toggle	互换光标下像素的 ON/OFF 状态
<KP0>	USE	把字形模式装入 USE 缓冲区
<KP. >	DISPLAY	以实际大小显示字形模式
<GOLD> + <PF2>	Help	提供关于 EDIT 的求助
<GOLD> + <PF3>	Output	把编辑结果写入用户属性文件
<GOLD> + <PF4>	Delete	删除字形模式和 / 或其他字符属性
<GOLD> + <KP->	Cancel	取消对当前代码的编辑
<GOLD> + <KP, >	Quit	放弃编辑程序
<GOLD> + <Enter>	Exit	退出编辑程序
<GOLD> + <KP7>	ULcorner	跳到左上角
<GOLD> + <KP8>	Top	跳到顶部
<GOLD> + <KP9>	URcorner	跳到右上角
<GOLD> + <KP4>	L side	跳到左边
<GOLD> + <KP5>	Center	跳到中心
<GOLD> + <KP6>	R side	跳到右边
<GOLD> + <KP1>	LLcorner	跳到左下角
<GOLD> + <KP2>	Bottom	跳到底部
<GOLD> + <KP3>	LRcorner	跳到右下角
<GOLD> + <KP0>	Return from USE	从 USE 缓冲区返回到 EDIT 缓冲区
<GOLD> + <KP. >	REFER	在 REFER 窗口内显示字形模式

---

表 6-4 编辑小键盘

键	缩写	功能
<Select>	Select	互换启动 / 取消选择区域
<Insert Here>	Paste	插入 Paste 缓冲区的内容
<Remove>	Cut	把选择区域剪切到 Paste 缓冲区
<Prev Screen>	Prv Size	转换到上一个字形大小
<Next screen>	Nxt Size	转换到下一个字形大小
<Up arrow>	Up	独立于光标态, 向上移动
<Down arrow>	Down	独立于光标态, 向下移动
<Left arrow>	Left	独立于光标态, 向左移动
<Right arrow>	Right	独立于光标态, 向右移动
<GOLD> + <Select>	Sel. All	选择整个位映象区域

表 6-5 控制键

键	功能
<CTRL/C>	放弃编辑程序
<CTRL/R>	以实际大小显示字形模式
<CTRL/W>	刷新屏幕
<CTRL/Z>	退出编辑程序

表 6-6 主键盘键

键	功能
<B>, <b>	在选择区域中画一个空心矩形
<C>, <c>	把选择区域剪切到 Paste 缓冲区
<I>, <i>	倒置选择区域
<L>, <l>	连接两个选择点画线
<O>, <o>	在选择区域中画一个空心椭圆
<P>, <p>	插入 Paste 缓冲区的内容
<Q>, <q>	在选择区域中画一个实心椭圆

## 字形编辑程序参考

### 6.7 快速参考

表 6-6 (续) 主键盘键

<S>, <s>	互换启动 / 取消选择区域
<W>, <w>	在选择区域中画一个实心矩形
<;>	在选择区域中环绕纵向中心线对称地重新排列字形模式
<->	在选择区域中环绕横向中心线对称地重新排列字形模式
</>	在选择区域中环绕左下到右上的对角线对称地重新排列字形模式。选择区域必须是正方形。
<\>	在选择区域中环绕左上到右下的对角线对称地重新排列字形模式。选择区域必须是正方形。
<GOLD> + <G>, <GOLD> + <g>	跳到标记
<GOLD> + <M>, <GOLD> + <m>	标记当前光标位置
<GOLD> + <S>, <GOLD> + <s>	选择整个位映象区域

---

# 附录 A

## 整理值列表

### A.1 拼音整理值

拼音值可以是下面表中所列出的常规拼音或特殊拼音。

**表 A-1** 常规拼音

a	ai	an	ang	ao	ba	bai	ban	bang
bao	bei	ben	beng	bi	bian	biao	bie	bin
bing	bo	bu	ca	cai	can	cang	cao	ce
cen	ceng	cha	chai	chan	chang	chao	che	chen
cheng	chi	chong	chou	chu	chuai	chuan	chuang	chui
chun	chuo	ci	cong	cou	cu	cuan	cui	cun
cuo	da	dai	dan	dang	dao	de	deng	di
dia	dian	diao	die	ding	diu	dong	dou	du
duan	dui	dun	duo	e	ei	en	er	fa
fan	fang	fei	fen	feng	fo	fou	fu	ga
gai	gan	gang	gao	ge	gei	gen	geng	gong
gou	gu	gua	guai	guan	guang	gui	gun	guo
ha	hai	han	hang	hao	he	hei	hen	heng
hong	hou	hu	hua	huai	huan	huang	hui	hun
huo	ji	jia	jian	jiang	jiao	jie	jin	jing
jiong	jiu	ju	juan	jue	jun	ka	kai	kan
kang	kao	ke	ken	keng	kong	kou	ku	kua
kuai	kuan	kuang	kui	kun	kuo	la	lai	lan
lang	lao	le	lei	leng	li	lia	lian	liang
liao	lie	lin	ling	liu	long	lou	lu	luan
lue	lun	luo	m	ma	mai	man	mang	mao
mei	men	meng	mi	mian	miao	mie	min	ming
miu	mo	mou	mu	na	nai	nan	nang	nao
ne	nei	nen	neng	ng	ni	nian	niang	niao

整理值列表  
A.1 拼音整理值

表 A-1 (续) 常规拼音

nie	nin	ning	niu	nong	nou	nu	nuan	nue
nuo	o	ou	pa	pai	pan	pang	pao	pei
pen	peng	pi	pian	piao	pie	pin	ping	po
pou	pu	qi	qia	qian	qiang	qiao	qie	qin
qing	qiong	qiu	qu	quan	que	qun	ran	rang
rao	re	ren	reng	ri	rong	rou	ru	ruan
rui	run	ruo	sa	sai	san	sang	sao	se
sen	seng	sha	shai	shan	shang	shao	she	shen
sheng	shi	shou	shu	shua	shuai	shuan	shuang	shui
shun	shou	si	song	sou	su	suan	sui	sun
suo	ta	tai	tan	tang	tao	te	teng	ti
tian	tiao	tie	ting	tong	tou	tu	tuan	tui
tun	tuo	wa	wai	wan	wang	wei	wen	weng
wo	wu	xi	xia	xian	xiang	xiao	xie	xin
xing	xiong	xiu	xu	xuan	xue	xun	ya	yan
yang	yao	ye	yi	yin	ying	yo	yong	you
yu	yuan	yue	yun	za	zai	zan	zang	zao
ze	zei	zen	zeng	zha	zhai	zhan	zhang	zhao
zhe	zhen	zheng	zhi	zhong	zhou	zhu	zhua	zhuai
zhuan	zhuang	zhui	zhun	zhuo	zi	zong	zou	zu
zuan	zui	zun	zuo					

表 A-2 特殊拼音

---

bao gai tou	bao zi tou	bing zi pang
cao zi tou	ce dao pang	chu zi tou
dan er duo	dan ren pong	dong zi dou
fan quan pang	fan wen pang	hu zi tou
jian zhi pang	jiao si pang	jin zi pang
liang dian shui	nong zi di	ping bao gai
qu zi kuang	san dian shui	san pie
shan zi di	shi zi pang	shu xin pang

表 A-2 特殊拼音

si dian shui	ti shou pang	tong zi kuang
wen zi tou	xun zi tou	yan zi pang
yi zi pang	yu zi tou	zhuang zi pang
zou zhi pang	zuo er duo	

## A.2 部首整理值

下表中列出有效的部首值：

图 A-1 部首列表

一	丨	丿	丶	乙	二	十	厂	匸	卜	冫	冂	彳	人	八
勹	儿	儿	勹	彡	冂	讠	冂	冂	刀	力	又	廴	口	厶
工	土	士	艹	井	大	尢	扌	寸	弋	口	口	巾	山	彳
彡	彡	夕	夕	彡	广	卜	门	斗	彡	宀	辶	冂	尸	弓
己	中	女	小	子	马	彡	幺	叀	王	韦	木	犬	歹	车
戈	比	瓦	止	支	日	贝	见	牛	手	毛	气	攴	片	斤
爪	月	欠	风	攴	文	方	斗	火	父	灬	户	礻	心	聿
水	毋	示	石	龙	业	目	田	四	皿	钅	矢	禾	白	瓜
用	鸟	疒	立	穴	礻	疋	皮	矛	耒	老	耳	臣	西	页
虍	虫	缶	舌	竹	白	自	血	舟	衣	羊	米	艮	羽	糸
麦	走	赤	豆	酉	辰	豕	鹵	里	足	身	采	豸	角	言
辛	谷	青	其	雨	齿	黽	隹	金	鱼	革	骨	鬼	食	音
彡	麻	鹿	黑	鼠	鼻									

## A.3 笔画数整理值

笔画数的有效值是从 1 至 255 的正整数。



## CMGR 词汇表

**位映象 (Bitmap):** 一个位映象由一系列字节表示, 而每字节的每一位均代表一个象素。在 CMGR 中, 字形模式以位映象来表示以供用户编辑。CMGR 支持三种字形大小。24x24 的字形模式需要 72 个字节的位映象来表示、32x32 的字形模式需要 128 个字节来表示、40x40 的字形模需要 200 个字节来表示。

**整理特征 (Collating Characteristics):** 这是指用户定义字符的整理属性。在 OpenVMS/Hanzi 中, 只支持三种整理特征, 分别是“拼音”、“部首”和“笔画数”。

**整理数据文件 (Collating Data File):** 整理数据文件是用来存储字形编辑程序或一些 CMGR 命令建立的整理特征。只由 CMGR 使用并具有默认文件扩充名“.CVD”。

**字形模式、位映象字形模式 (Font Pattern, Bitmap Font Pattern):** 这术语是指用户定义字符的字形属性。位映象字形模式由二维矩阵建立, 里面的每个元素就是一个象素。象素可以是 ON 或 OFF。这二维矩阵是以位映象格式存储的。

**内码 (Internal Code):** 内码是指 DEC 汉字字符的一种表示法。在 CMGR 中, 大多数命令都接受内码作为参数或限定词的值。内码以一个前面带有“H”的十六进制值字符串来表示。请参阅本手册的 4.1 节。

**按需装入 (On Demand Loading):** 按需装入是在硬件设备上实施的一种机制。它允许在设备发出请求时把字形模式动态地装入该设备。

**象素 (Pixel):** 一个图象元素。

**预装入 (Preload, Preloading):** 预装入是指在需要实际使用或显示字形模式前, 把一些字形模式装入原来没有该些字形模式的硬件设备。

**预装入文件 (Preload File):** 预装入文件以预装入序列格式存储位映象字形模式。预装入文件可以供 CMGR 使用或作预装入之用。它有一个默认文件扩充名“.PRE”。

**预装入序列 (Preload Sequence):** 预装入序列是一个存储位映象字形模式的换码序列。预装入序列的格式与位映象不同, 除了位映象资料外, 字形的大小和字符代码也会编写到预装入序列里。任何可以执行按需装入的硬件设备都可以解释此序列。

**区位码 (Qu Wei Code):** 区位码表示 DEC 汉字字符平面两个象限中的字符代码。每个象限再分成 94 区, 并且每区均有 94 个位置。区号是“区”, 位置号是“位”。请参阅本手册的 4.1 节。

**保留用户定义, 保留用户定义字符 (Reserved User-Defined Area, Reserved User-Defined Character):** DEC 汉字字符集中有一个区域, 可以让用户使用 CMGR 命令的 /RESERVED 限定词来建立自己的字符或特殊字形模式。该区域称为“保留用户定义区域”, 而那些定义位置则称为“保留用户定义字符”。

**系统属性数据库 (System Attribute Database):** 这是指用来存储位映象字形模式和整理特征的“全系统数据库”。共有三个支持的系统属性数据库, 分别命名为 CMGR\_DEFAULT、SONG 和 HEI。有了这三个数据库便可让用户存储不同字体的字形和整理值。这与系统数据库的定义相同。系统属性数据库可以再分成系统字形数据库和系统整理数据库。

**字体 (Typeface):** 同一中文字符可以用不同的式样写出。中文字符的这些不同外观就称为字体。例如, 有 SONG 体 (宋体) 和 HEI 体 (黑体)。使用 CMGR, 用户可以在不同的系统属性数据库中存储和操纵不同字体的字形和整理值。

## CMGR 词汇表

**用户属性文件 (User Attribute File):** 这是用户自己或由一些 CMGR 命令建立的数据文件。与属性文件的定义相同。共有两种用户属性文件：一种是预装入文件，另一种是整理数据文件。

**用户定义区域、用户定义字符 (User-Defined Area, User-Defined Character):** DEC 汉字字符集中有一个让用户建立自己的字符或特殊字形模式的区域，这区域称为“用户定义区域”，而这些定义位置则称为“用户定义字符”。