
# Series 100/Condor®
# 20-1
# User's Manual


Manual Part No.
45415-90001

Printed in U.S.A.
9/83

# NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.



# Printing History

New editions of this manual will incorporate all material since the previous edition. Update packages may be used between editions and contain replacement and additional pages to be merged into the manual by the user.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.)

First Edition - September 1983

# Preface

This manual describes the Condor Database Management System for the HP Series 100 Personal Office Computers. It is written for the business manager or professional, and no prior familiarity with database management systems or concepts is assumed. The early chapters provide a conceptual overview of database management and a step-by-step introduction to database creation and use. Experienced readers may want to skim these chapters, and concentrate on the later chapters and the *Condor Reference Manual.*

This manual does assume familiarity with the general operation of an HP 150 system, as described in the *HP 150 Owner's Guide.* If you are not familiar with HP 150 operation and terminology, you should review this manual before proceeding.

Condor is available in two levels. Level 1 (20-1) provides a powerful set of commands for creating, managing and manipulating individual databases. Level 3 (20-3) adds relational commands for interrelating data from several different databases. Level 3 also adds advanced report writing and database indexing capabilities. This manual describes the full set of available Level 1 commands. Level 3 commands are included in a 20-3 manual section which you will receive if you purchase Condor 20-3. Features not available to the Level 1 user are clearly noted. An upgrade product is available for Level 1 users who later desire the full capabilities of Level 3.

# HP Computer Museum
# www.hpmuseum.net

**For research and education purposes only.**

# Table of Contents

## Chapter 1: Database Concepts



## Chapter 2: Getting Started with Condor

## Chapter 3: Defining a Database



## Chapter 4: Entering and Updating Data



## Chapter 5: Simple Inquiries and Reports

## Chapter 6: Additional Database Capabilities





## Chapter 7: Database Utilities



## Chapter 8: Using Databases with Other Applications

## Chapter 9: Restructuring and Reorganizing Databases



## Chapter 10: Database Command Procedures

## Chapter 11: Transaction Processing



## Appendix A: Example Databases

## Index

# Chapter 1

# DATABASE CONCEPTS


Condor is a powerful and easy to use database management system for the HP Series 100 Personal Computers. In almost any business situation where you might say "I need to keep track of...", Condor can help you keep track faster and more effectively.

With Condor, you can store important information and then manipulate it in many useful ways. For example, you might track important customers and their orders. On demand, you can look up information about individual customers, print customer reports alphabetically or by geography, or combine customer information with order data to summarize orders by customer, by sales region, or by product line. You can transfer that information to Series 100/ WordStar or Series 100/Graphics to create documents or plot color charts. And that is just the beginning of Condor's capabilities!

Applications of Condor are as varied as the different types of information that you use every business day. Condor can track personal information, such as client names and addresses. A department can use it to keep track of office equipment or employee time cards. You can use it to track spare parts inventory, or a stock of office supplies. And it can form the foundation for a complete transaction processing application.

This manual introduces the capabilities of Condor, step-by-step. The early chapters will help you "get going" quickly, with a few commands that let you build and use simple databases. Later, you can move through the manual to learn more about Condor's features, as you need them. This chapter begins by discussing a few fundamental database concepts.

# What Is a Database?

First of all, what is a **database**? A **database** is an organized collection of related information that can be retrieved and manipulated on demand. Chances are that you already use many non-computerized "databases" every day, without calling them databases.

One of the best examples of a database is a secretary's card file. The card file keeps track of people—their names, addresses and phone numbers, etc. Figure 1-1 shows a typical card from a card file. Notice how the card file fits the database definition: it is organized alphabetically, it stores related, highly structured information, and it is used to retrieve information many times each day.

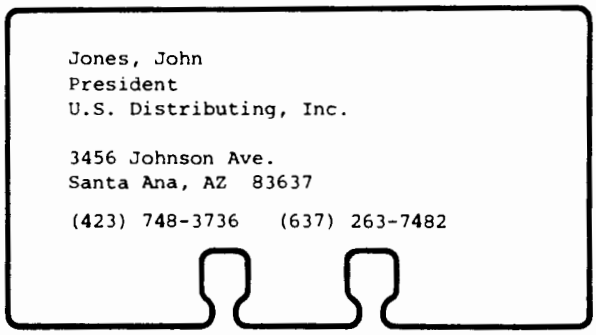


Figure 1-1. A Typical Card from a Card File

Another familiar database is the clerk's file folder holding the day's orders. Each order is recorded on a single order form, on which is written all the pertinent information about the order: customer name, address, product ordered, quantity, price, totals, etc. Figure 1-2 shows a typical order form from this **ORDERS database**.

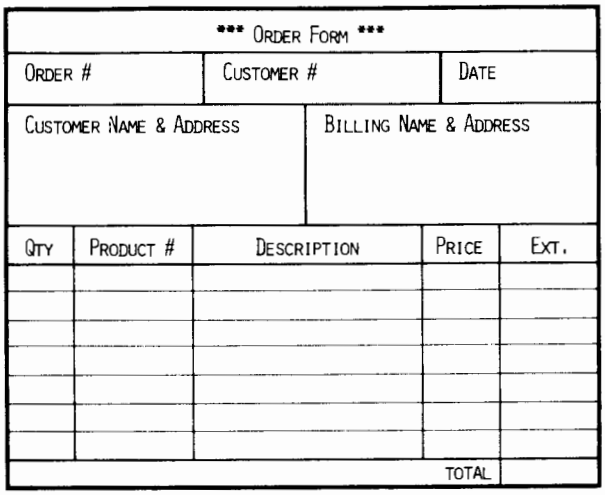


Figure 1-2. A Typical Order Form

Figure 1-3 shows another typical business document, a regional sales report. Could this data be viewed as a database? Yes! And you will often examine the information in Condor databases with printed reports like this one.

***** REGIONAL SALES REPORT FOR APRIL, 1982 ******

| REGION.NAME | ACTUAL.MTD | QUOTA.MTD | ACTUAL.YTD | QUOTA.YTD |
|---|---|---|---|---|
| CANADA | 2445.00 | 2400.00 | 14253.56 | 15000.00 |
| MID-ATLANTIC | 9475.45 | 10000.00 | 48394.39 | 50000.00 |
| MIDWEST | 6738.46 | 7000.00 | 41200.00 | 35000.00 |
| MOUNTAIN | 4393.00 | 4500.00 | 29382.45 | 27500.00 |
| NEW ENGLAND | 5238.34 | 5000.00 | 23057.39 | 26750.00 |
| N. PACIFIC | 7636.34 | 7000.00 | 53627.34 | 50000.00 |
| S. PACIFIC | 5346.34 | 5000.00 | 37463.34 | 38000.00 |
| SOUTHERN | 1927.78 | 2000.00 | 9372.92 | 10000.00 |
| Total | 43200.71 | 42900.00 | 256751.39 | 252250.00 |

Figure 1-3. Regional Sales Report

Actually, the row/column report is just the **card file** format of the first example, in disguise. Imagine that the data in each row of the report was written on a separate index card and then filed away in a card file. The data remains the same as in the report, of course, but the information is now in a **database** much more like the other two examples.

A card file is one of the best and simplest ways of thinking about a database. Card files and a **CARDFILE** database will be used often as examples in this manual.

# Database Structure

How is a database organized? What is its structure and form? The CARDFILE database provides an excellent example to answer these questions.

A database is organized as a collection of **records**, just as a card file is organized as a collection of cards. Each card, for example, is one record. The card stores all the information related to one person—name, title, address, firm, etc. The records correspond one-to-one with the people they represent. This will be true of most databases that you will use with Condor; there will be a **one-to-one correspondence between database records** (the **index cards**) **and some physical or logical entity**—a customer, an order, a product, a sales region, etc.

Each CARDFILE record also has its own individual structure. It is made up of several pieces of information—a name, an address, etc. We call these individual pieces of information **data items**. Each CARDFILE record contains **exactly the same** data items as the others. This will also be true for **every** Condor database: every record in the database contains exactly the same data items as the others. Figure 1-4 summarizes the structure of a database—a database is comprised of records; each record contains the same data items as the others.

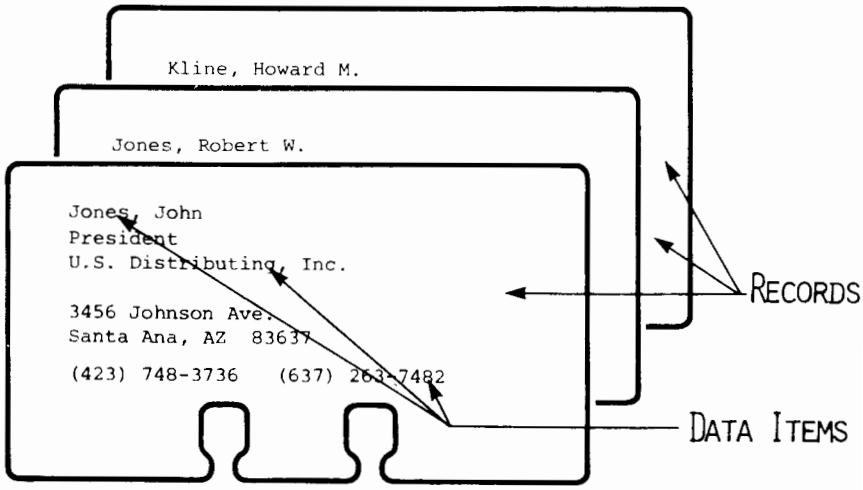


Figure 1-4. Records and Data Items

# What Does Condor Do?

Condor is an HP Series 100 software package that stores databases on magnetic discs instead of card files. Using Condor, information can be stored more compactly, and more accurately than with manual databases. Inquiries or reports are available in a fraction of the time it would take to prepare them with a manual system. And information that would be too expensive or time-consuming to generate by hand is available in minutes, at the touch of a few keys.

Condor makes an excellent **personal assistant** for a business manager or professional, helping to keep track of important information. It can also be an important tool for the secretary. And in the hands of a system designer, it can form the basis for an entire transaction processing application.

# Database Forms

Every Condor database has an associated **screen form**—a format for presenting database information on the HP Series 100 screen. The form presents information from a single record of the database at a time. You create the form when you first define a new database to Condor. From then on, the form is used to enter, update and display information in the database.

There is nothing mysterious about a database form. A card file, for example, has a fixed, consistent format for each card. Figure 1-5 shows how this card format can easily become an HP Series 100 screen form for an **electronic CARDFILE** database.

```
************************************************************
**                                                        **
**      ***** HP SERIES 1ØØ CARDFILE DATABASE *****       **
**                                                        **
**    NAME :        Jones, John                           **
**    TITLE :       President                             **
**    FIRM :        U. S. Distributing, Inc.              **
**                                                        **
**    STREET :      3456 Johnson Ave.                     **
**    CITY :        Santa Ana                             **
**    STATE :       AZ        ZIPCODE :   83637           **
**                                                        **
**    BUS-PHONE :   (423) 748-3736                        **
**    HOME-PHONE :  (637) 263-7482    REV-DATE : Ø1/26/83 **
**                                                        **
************************************************************
```

Figure 1-5. CARDFILE Screen Form

# How Does Condor Work?

How do you actually use Condor to create and access a database? The first step is to make some decisions—what to name the database and what data items it will contain. With this information, you are ready to create a new database.

First, you must **define** the database. You will **paint** the database form on the HP Series 100 screen, using the editing keys on the keyboard. Condor obtains much of the information it needs about the database directly from the form. It will also ask you for additional information. After you supply all the needed information, Condor creates the database on the disc.

Next, you will **enter information** into the database. The form you have just created is used for data entry, record by record. You can also update records in the database, or delete records, all using the same, familiar form.

Later, you will want to **retrieve data** from the database. Again, the form can be used to display retrieved information on demand. Or, you may want to **print reports**, based on database contents.

You can use information from Condor databases with other HP Series 100 applications, too. For example, you might plot a pie chart from an ORDERS database, include a database report as part of a memo, or generate a mailing list from a CARDFILE database.

After a while, you will have many different databases, and will want to **relate these databases to one another**. In this way, even complex reports and inquiries can be handled on demand, drawing on stored information to gain new insights or to look at a problem in a new way.

But why talk about Condor when you're ready to use it yourself? Go on to the next chapter, and get started on building your first Condor database!

# Chapter Summary

- A **database** is an organized collection of related information, like a card file or folder full of order forms.
- Condor maintains databases electronically on discs, with the benefits of **speed, accuracy,** and **powerful, fast, flexible inquiry and reporting**.
- A Condor database is comprised of **records**. Each **record** is comprised of **data items**, and all records in a database have the same data items.
- Data from Condor databases can be exchanged with other Series 100 applications, such as Word Processing, Graphics and Data Communications.
- Each Condor database also has an associated **screen form**, which is used to enter, update, and display database information.

# Chapter 2

# GETTING STARTED WITH CONDOR


In this chapter, you will install Condor onto your HP Series 100 system, and build and use your first Condor database. By carefully following the instructions presented here, you should be using your own database in less than an hour.

## Installing Condor

The one or two flexible discs that are packaged with this manual contain the Condor Database Management Software. Before you can use Condor on your HP Series 100, you must **install** Condor onto **work discs** which you will use in everyday operation. The installation procedure is fully described in the Series 100 Owner's Guide that was shipped with your system.

---

**NOTE**

Condor is a large and advanced software product. When Condor 20-1 is installed on a flexible disc, **it must be the only application on the disc**. No other programs or files except the operating system may be present on the **work disc** on which Condor is installed. If you purchased Condor 20-3, you will have to install it onto two separate work discs. Either product can be installed onto a hard disc along with other applications

---

## Installing Your Condor 20-1 Disc

The Condor 20-1 software you received is packaged with a single flexible application disc. As mentioned earlier, this application disc must be installed onto another flexible disc (called a work disc) or onto a Winchester disc. To install Condor 20-1;

1. Create a work disc with only the Series 100 operating system.
2. Install 20-1 by following the procedure described in the system Owner's Manual.

When Condor 20-1 is correctly installed, you should see the label `Condor 1` appear on the Program Applications Manager (P.A.M.) menu.

## Installing Your Condor 20-3 Discs

The Condor 20-3 software you received is packaged with two flexible application discs. These must be installed onto two flexible work discs or they may be installed onto a Winchester hard disc. To install Condor 20-3 onto flexible discs:

1. Create two work discs with only the Series 100 operating system.
2. Install the two application discs onto the two work discs by following the procedure described in the HP 150 Owner's Guide. When you are finished doing the installation, the discs will contain the following functions:

   Disc 1—Commands for building databases. Commands present on this disc are:

| | | |
|---|---|---|
| COPY | DATE | DEFINE |
| DESTROY | DIC | ENTER |
| FORMAT | HELP | INDEX |
| READ | RENAME | REORG |
| REPORT | SAVE | UPDATE |
| WRITE | DBMS | SET |
| SYSTEM | DIR | ABORT |
| EDIT | LIST | LOG |
| RESTART | | |

Disc 2—Commands for running database applications. Commands present on this disc are:

| | | |
|---|---|---|
| APPEND | CHANGE | COMBINE |
| COMPARE | COMPUTE | COPY |
| DATE | DELETE | DIC |
| DISPLAY | EMPTY | ENTER |
| HELP | INDEX | JOIN |
| LIST | POST | PRINT |
| PROJECT | RENAME | REPORT |
| RUN | SELECT | SORT |
| STAX | TABULATE | TITLE |
| UPDATE | ABORT | DIR |
| LOG | RESTART | SAVE |
| DBMS | SET | SYSTEM |

When these discs are correctly installed, either Condor 3_Build. or Condor 3_Run will appear on the P.A.M. menu, depending on which disc is logged in.

To install Condor 20-3 onto a Winchester disc, install both discs using the standard installation procedure. When you are finished, two labels should appear on the P.A.M. menu: Condor 3_Build and Condor 3_Run. Touching either of these labels allows you to use all of the Condor 20-3 commands.

---

**NOTE**

**After you have completed installation of Condor, it is extremely important that you save the original Condor application disc in a safe place.** The application disc is your backup disc in case you should accidentally damage your work disc(s).

---

# Starting Condor

To begin processing databases with Condor, insert the 20-1 work disc or the Condor 3 "Build" work disc you have just created into the **A** drive in your system. Insert a blank, **formatted** disc into the **B** drive; this disc will contain your first few databases. (If your HP Series 100 uses a Winchester disc, then you can ignore these two steps.)

Follow the normal system start up procedure (see the *HP 150 Owner's Guide* that accompanied your system for details). The P.A.M. Menu will appear, and one of the application labels will be either `Condor 1` or `Condor 3_Build` . Touch this label. In a few seconds, a copywrite notice will appear, and then Condor will ask you for your first command by displaying the prompt:

```
A>>
```

You are now ready to begin using Condor. You will follow this same procedure each time you start a session with Condor.

Whenever you want to start Condor and immediately use any of the functions that are on the second disc of Condor 20-3, start up your system using the above procedures with the Condor 3 "Run" work disc. If you have already started Condor 20-3 and wish to use these functions, then simply remove Condor 3 "Build" from Drive A, insert Condor 3 "Run," and type in the appropriate command.

# Using HP Touch

HP Touch is offered on the HP 150 to make command entry and cursor movement as easy as touching the screen. Condor uses HP Touch in both of these ways.

There are several places in Condor where softkeys are used to select options or enter commands. In all of these situations, the softkey may be activated either by touching the softkey label on the screen or by pressing the key.

HP Touch is also available in Condor during screen creation and modification for moving the cursor. Touch a point on the screen and the cursor will move to that location.

# Building Your First Condor Database

As a **warm-up** exercise, we will build a simple Condor database to keep track of birthdays. Don't worry if you don't understand every detail as you go through the exercise. By following the instructions carefully, you will discover some powerful Condor features, and see how quickly you can apply Condor to your own tasks.

Do the following:

1. Type `B:` Return
   This signals Condor that you will create a database on the **B** drive. Condor will respond with the prompt `B>>`.

2. Type `DEFINE BIRTHDAY` Return
   The system will ask if you want to create a new form.

3. Type `Y`
   In a few moments, a screen for creating the new form will be displayed. It will be blank except for the softkeys at the bottom of the screen.

4. Type Return Return `[NAME]` _____
   Type Return Return `[BIRTHDATE]` _____
   Type 16 underscores **(important: not hyphens)** for NAME, and 8 underscores for BIRTHDATE.

5. Touch the End function key label.
   This ends form creation, and stores the BIRTHDAY form on the disc.

6. The system will now prompt you for the definition of each data item in the form. In a few moments, it will display the prompt `>1. NAME:`, indicating that NAME is the first data item to be defined.

7. Type `AN` Return
   You have specified that NAME contains alphanumeric data. Next, the system will display `>2. BIRTHDATE:`, requesting a definition for the BIRTHDATE data item.

8. Type `J` Return
   You have specified that BIRTHDATE is a Julian (date) data item. Since this is the last data item, the system requests that you review the data definitions and asks if they are correct.

9. Type `Y`
   The system will indicate Busy while it creates the database. It will then ask you if you want to create an index for this database.

10. Type `N`
    It will then ask if you want a printed copy of the data definitions. If you have a printer on your HP Series 100...

11. Type `Y`
    The system prints attributes for the database BIRTHDAY, including the number of data items, their definitions, and the

total record length. A B>> prompt will be displayed, and you have finished creating your first Condor database. Congratulations!

# Entering Data Into Your Database

With the BIRTHDAY database now defined, you are ready to enter data. Data is entered directly into the form, record by record. Simply follow the steps below:

1. Type ENTER BIRTHDAY [Return]
   The system will display the BIRTHDAY form on the screen. The fields on the screen where data may be entered are highlighted in inverse video, and the cursor is positioned in the first field, ready for you to enter a name.

2. Type John Doe [Return]
   When you press the [Return] key, the cursor moves to the next field, ready for you to enter a birth date.

3. Type 10/15/45 [Return]
   When you press the [Return] key, the form is complete, and a new set of softkey labels appear. By pressing the appropriate softkey, you may return to revise the data in the form, abort the data entry, end the data entry, or continue by entering a new set of data for a new record.

4. Touch the Continue softkey label.
   The data in the form is now added to the database. The form is cleared, ready to accept the next record.

5. Repeat steps 1-4 several times.
   With each form you enter, use a different name, but repeat some of the birthdays, to make the later reports interesting. If you like, you may use the data listed below:

| Name | Birthdate |
|---|---|
| Jane Doe | 10/20/50 |
| Paul Anderson | 2/5/62 |
| Bob Greene | 10/20/55 |
| Mary Roberts | 2/5/59 |

6. Touch the End softkey label.
   When you have entered a few records, end data entry by touching the softkey label or by pressing the End softkey. The form will disappear and the B>> prompt will be displayed. Data entry is now complete.

# Manipulating Data in a Database

Now that you have seen how easy it is to create and enter data into a Condor database, try some Condor commands for manipulating and reporting data:

1. Type `SORT BIRTHDAY BY BIRTHDATE` Return
   The system will print messages indicating progress of the sort operation. After completion, the database will be in ascending date sequence.

2. Type `LIST BIRTHDAY BY NAME, BIRTHDATE` Return
   The system will display a two-column report on the screen, listing NAME and BIRTHDATE for each record. The data appears in sequence by birthdate.

3. Type `SORT BIRTHDAY BY NAME` Return
   This command will sort the database into ascending sequence by (first) name.

4. If you have a printer connected to your computer, then type `PRINT BIRTHDAY BY BIRTHDATE, NAME` Return
   This time the report will appear on the system printer. The data is in sequence by name, and the columns are in the opposite order.

This exercise has given only a brief introduction to the data manipulation and reporting power of Condor. The following chapters describe in detail the Condor capabilities for entering, updating, manipulating, and reporting information.

# Database Commands

Each time Condor displays the prompt:

A>> or B>> or C>> etc.

it is waiting for a **command** from you. Commands are the way that you tell Condor what to do; each of the available commands performs its own unique function. Most Condor commands have the same format:

```
verb   database-name   qualifying-information  [option]
```

VERB — tells what action is to be taken; ENTER, PRINT, SORT, etc.

DATABASE-NAME — tells what database to act upon

QUALIFYING-INFORMATION — varies depending on the command; tells which data items to list, how to sort a database, etc.

[OPTION] — selects different command options, such as whether information is displayed on the screen or printer

Examples:

```
DEFINE BIRTHDAY
```
(define the database named BIRTHDAY)

```
LIST BIRTHDAY
```
(list the database contents on screen)

```
UPDATE BIRTHDAY WHERE NAME IS JOHN*
```
(update the database record for JOHN DOE)

```
TABULATE BIRTHDAY BY DATE [P]
```
(print database statistics by date, on the printer)

Condor commands can be up to 128 characters long. If you type a command longer than one line, Condor automatically displays the last part of the command on a separate line. You must only press the Return key when you have completely finished typing the command. If you make a mistake while typing a command, use the Back space key to move the cursor back to the character in error, and then retype the command from that character onward. The arrow keys **cannot** be used to edit Condor commands.

# The Current Disc Drive

The letter before the >> in Condor's command prompt tells you which disc is the **current disc drive** on which Condor is working. Unless you specifically tell Condor otherwise, it will always process databases on the current disc drive. You can change the **current disc drive** by typing the name of another disc drive following the command prompt.

Examples:

To change the current disc to drive B, type

```
B: Return
```

when the `A>>` prompt appears.

To switch to drive C, type

```
C: Return
```

when the `A>>` prompt appears.

You can also ask Condor to process a database that is **not** on the current disc. Type the name of the disc drive and then the name of the database when you type in a command.

Examples:

`LIST D:BIRTHDAY`
(list the BIRTHDAY database on disc drive **D**)

`LIST B:BIRTHDAY`
(list the BIRTHDAY database on drive **B**)

`LIST BIRTHDAY`
(list the BIRTHDAY database on the current disc drive)

# Exiting Condor

When you are finished using Condor, and wish to return to P.A.M., type the command:

SYSTEM Return

After a few seconds, the HP 150 P.A.M. Menu will reappear. This is the way you should always exit Condor, to be certain that your databases are maintained in an orderly manner.

---

**CAUTION**

**Never exit Condor by performing a hard-reset on your system or turning off the system power. Doing so may result in loss of data and/or loss of your databases.**

---

# Chapter Summary

- Condor is installed on **work discs**, like all HP Series 100 applications.
- To start Condor, touch the `Condor 1` or `Condor 3_Build` or `Condor 3_Run` labels on the P.A.M. Menu.
- Condor uses simple, English-like commands to define databases, to enter data into them, to sort them, print them, etc.
- Unless instructed otherwise, Condor uses the **current disc** for all database processing. You can change the current disc to a different drive by typing the drive name in response to the command prompt.
- Databases not on the current disc drive can be processed by including the drive name before the database name in Condor commands.
- Use `SYSTEM` command to exit Condor and return to the P.A.M. Menu. **Always** exit Condor this way. Never perform a hard-reset while Condor is in operation.

# Chapter 3

# DEFINING A DATABASE


Before you can use a database, you must first **define** the database for Condor. That is, you must design the database form, and provide information on each data item in the database. You have already defined one simple database (the BIRTHDAY database) in the warm-up example of the previous chapter. This chapter describes the steps for defining a new database in more detail.

The general procedure for defining a database is:

1. Choose a name for the new database.
2. Decide what data items the database will contain; choose a name for each data item.
3. Design a screen form for the database. The form will be used to enter, update and display database information on the screen.
4. Decide on which disc the database will be stored.
5. Use the `DEFINE` command to paint the database form on the HP Series 100 screen.
6. Continuing with the `DEFINE` command, describe each data item to Condor in detail.

When you have completed these six steps, your database will be defined, and ready to accept data. The following sections describe each step in detail. (The example databases used in this Chapter, and throughout the manual, are fully described in Appendix A, for reference.)

# Naming the Database

The name that you choose for a new database should be descriptive and easy to remember. The name may have from one to eight alphabetic or numeric characters.

Example:

Here are some good examples of database names:

```
CARDFILE — an electronic CARDFILE database
ORDERS   — a database of customer orders
PRODUCTS — a database of product information
REGIONS  — a sales region database
CLIENTS  — a database of client information
GLJOURNL — a General Ledger journal database
CHTACCTS — a Chart of Accounts database
```

# Naming Data Items

The most important decision when planning a new database is deciding what data items it will contain. You can add data items to a database later, but it is much, much easier to make certain that you have included all the needed data items at the beginning. Remember, each record in the database will usually correspond to a physical entity—a customer, a part, an order, etc. Ask yourself "What information is important to remember about this entity? What information might you want to use to retrieve, compare, accumulate, or select database records later?" That information should be included in the database as one or more data items.

An individual data item is the smallest bit of information that you can access in a database. If you were to store a client address as a single data item, for example, then you could not break the address down into street, city, and state later. You could not, for example, sort the database by state. If this type of sorting or selecting is important to your application, split the address into separate street, city, and state data items.

Each data item in the database has its own **data item name**. Data item names can be up to fifteen characters long. Try to use short, descriptive names, since you will be typing them often as you use the database.

**NOTE**

The **special** characters *, ?, #, <, >, [, ], _, and the single and double quote characters all have special meaning in certain Condor commands. If you use these characters in data item names, they may lead to confusion when using these commands. Avoid them if at all possible. Data item names that contain embedded blanks (such as **NEW DATE**) should also be avoided if at all possible. It is better to "artificially" eliminate the blanks with a period or hyphen (e.g., use **NEW.DATE**). If you **must** use embedded blanks, the data item name must always be enclosed in quotes when used in Condor commands.

---

Here are some examples of good data item names:

```
ADDRESS    — for a street address
ZIP.CODE   — for a zip code
PROMDATE   — for promised date
ORDERDATE  — for order date
```

Example:

The CARDFILE database will consist of the following data items:

| | |
|---|---|
| NAME | — the name of the person who is described in this CARDFILE record. Up to 20 characters will be allowed. |
| TITLE | — the person's job title. Up to 20 characters will be allowed. |
| FIRM | — the name of the company the person represents. 30 characters allowed. |
| STREET | — the street address. 30 characters. |
| CITY | — the city. 20 characters. |
| STATE | — the state. 2 characters. |
| ZIPCODE | — the zip code. 5 characters. |
| BUS-PHONE | — a 10-character business phone no. |
| HOME-PHONE | — a 10-character home phone no. |
| REV-DATE | — revision date for this information. 8 characters, in mm/dd/yy format. |

# Designing the Database Form

The database form determines how database information is displayed on the HP Series 100 screen. The form is used to enter data into the database, to update the data, and to display selected records in response to a database inquiry. For each data item, the form will contain the **data item name**, followed by a **data entry field** that is used to enter or display the data for that data item.

For simple databases, it is often most effective to simply list the data items down the left side of the screen, one data item to each row. If a paper business form is already being used in your firm to process the data in a database, you should design the screen form to resemble the paper form as closely as possible, for ease of use. Otherwise, arrange the data items in the most logical sequence on the display.

Example:

The CARDFILE database form should closely resemble the typewritten card from the card file. One possible design is shown in Figure 3-1.

```
**************************************************************
**                                                          **
**          *****  HP SERIES 100 DATABASE  *****            **
**                                                          **
**    [NAME]:          ____________________                 **
**    [TITLE]:         ____________________                 **
**    [FIRM]:          ____________________________         **
**                                                          **
**    [STREET]:        ____________________________         **
**    [CITY]:          ____________________                 **
**    [STATE]:         __      [ZIPCODE]:  _____           **
**                                                          **
**    [BUS-PHONE]:     ______________                       **
**    [HOME-PHONE]:    ______________   [REV-DATE] ________ **
**                                                          **
**************************************************************
```

Figure 3-1. CARDFILE Screen Form

Each data item name in the database **must** appear somewhere on the form, enclosed in parentheses (see example above). Condor searches the form for these characters to determine the data item names for the database.

Following each data item name on the form is a **data entry field** for that data item. This field is used to enter or display data for that particular data item. The field must be long enough to accommodate the largest data item value that will be entered or displayed (e.g., the longest customer name, or the largest dollar amount). Data entry fields are indicated on the form by a series of underscore characters "_".

Other information on the form (i.e., all the information except the data item names and the data entry fields) is simply treated as **part of the form**. This information may be informative, or may make the form more readable, but it is purely cosmetic in function. The asterisks and title of the CARDFILE example form are examples of good **form cosmetics**.

With experience, you will be able to design simple forms directly on the HP Series 100 screen. For the first few forms, it is probably a good idea to design the form on paper first.

# Choosing the Disc to Store the Database

Generally, you will place a new database on a disc other than the one that contains your work copy of Condor software. The reason is simple—Condor is a rather large program, and in most cases, there will not be enough space remaining on the Condor work disc to store a database of any significant size. This is especially true if your system uses flexible discs as opposed to a Winchester disc.

On dual floppy disc systems, you will normally place your work disc with Condor software in drive A, and store the database on a flexible disc in drive B. On Winchester disc systems, the database is often located on the Winchester disc B, C, or D.

Be certain that the disc on which you store your database is formatted before attempting to use it (see directions for formatting a disc in the *HP 150 Owner's Guide*). Label the disc and insert it into the proper drive for processing.

Regardless of which drive will contain the database, you should usually select that drive as the **current drive** for ease of operation.

Example:

The sample CARDFILE database will be stored on a flexible disc dedicated to that purpose. The disc is labelled **CARDFILE** with a felt tip pen, and is inserted into disc drive B. Condor is loaded, and displays the prompt: `A>>`

In response, type `B:` Return

Condor responds with the prompt:

```
B>>
```

indicating that drive B is now the current drive.

# Creating the Database Form

With planning for the database complete, and the proper disc selected, you are ready to define the database to Condor. The `DEFINE` command is used to define new databases as shown in the steps below:

1. Type `DEFINE database-name` Return
   The system will ask if you wish to create a new form.
2. Type `Y`
   The screen will blank, ready for form input.

3. Type in the database form.

   Use HP Touch or the arrow keys on the keyboard to move anywhere on the screen, and **paint** the form on the screen by typing characters where they belong. Position titles, headings, data item names and data entry fields as desired. To help in positioning fields, you can display a grid on your screen by pressing the Grid softkey. To erase the grid, press the Refresh Screen softkey. If you make a mistake, use the cursor keys or the Back space key to move to the error and correct it by over typing on the screen.

   Remember to enclose data item names in square brackets ([ ]). Data entry fields are indicated as a sequence of underscores "__" (note, do **not** use hyphens). Underscores for a single data entry field must be consecutive, with no intervening blanks or other characters. Other characters may come between the data item name and the data entry field, however.

4. Press the End softkey.

   This softkey ends form creation and automatically stores the form on the current disc. If you are dissatisfied with the form, and do not wish to save it, press the Abort softkey instead of End.

5. Enter data item definitions for each data item on the form.

Example:

Enter the CARDFILE database form with the following steps:

Type `B:` Return

This sets drive B as the current drive.

Type `DEFINE CARDFILE` Return

The screen will blank, ready for you to **paint** the form as shown in Figure 3-1.

# Defining Data Items

The data item names for a new database all appear on the database form, enclosed in square brackets. However, the form provides only a limited amount of information about each data item: its name and its length (indicated by the number of underscores in the data entry field). The form does not tell whether the data item is alphabetic or numeric, whether it is a dollars-and-cents value, or a date, etc. The next step is to define each data item, in detail, to Condor.

Condor can handle ten different types of data items:

| CODE | TYPE | MINIMUM | MAXIMUM |
|---|---|---|---|
| A | Alphabetic | 1 character | 127 characters |
| AR | Alphabetic-Required | 1 character | 127 characters |
| AN | Alphanumeric | 1 character | 127 characters |
| ANR | Alphanumeric-Required | 1 character | 127 characters |
| $ | Dollars | -$21,483,736.47 | +$21,483,736.47 |
| $R | Dollars-Required | -$21,483,736.47 | +$21,483,736.47 |
| J | Julian Date | 01/01/00 | 12/31/99 |
| JR | Julian Date-Required | 01/01/00 | 12/31/99 |
| N | Numeric | -2,148,373,647 | +2,148,373,647 |
| NR | Numeric-Required | -2,148,373,647 | +2,148,373,647 |

After you have finished entering the database form, Condor will prompt you, data item by data item, for the **type** of data that the data item represents. Enter the appropriate code (from the table above), to indicate the type of data. Press Return after typing each code. Condor will take the default (built-in) minimum and maximum values for each type of data, and display them on the screen. (You will see how you may select other minimum, maximum and default values in the next section).

Example:

Continuing from the previous example, Condor will ask for the data type of each data item in the CARDFILE database. Respond as follows:

```
NAME:          ANR Return
TITLE:         AN  Return
FIRM:          AN  Return
STREET:        AN  Return
CITY:          AN  Return
STATE:         AN  Return
ZIPCODE:       N   Return
BUS-PHONE:     AN  Return
HOME-PHONE:    AN  Return
REV-DATE:      JR  Return
```

After you press the Return key each time, Condor will fill in the default minimum, maximum, and default values for the data item, based on its type. For the CARDFILE database, these values will serve our purposes well.

Condor will next ask if the data item definitions are correct. Type `N` if a data item definition is incorrect, and repeat the procedure. If all the definitions are correct, type `Y`. The system will create the database on the disc, displaying a `Busy` message until it is finished.

Next, Condor will ask if you want to create an index for this database. Indexing will be discussed in detail in Chapter 13 (Condor 20-3 only). For our purposes here, respond with an N to indicate that the database is not to be indexed.

Finally, Condor will ask if you want a printed copy of the definitions. Type `Y` to get a printed copy. The database is now defined; you may enter data into the database.

The type of each data item is used to check data for correctness when you are entering data into a database. For example, Condor will not allow you to enter a name into a dollars field, or a dollars-and-cents quantity into a date field. The Specifications section of the Condor Reference Manual describes in detail, for each data type, what data may be entered.

Fields that are **required** cannot be skipped over during data entry; Condor will force you to enter data in these fields. Fields that are not **required** may be skipped when entering data.

If you later discover that there is an error in the database form, or in the data item definitions, you can use the `FORMAT` or `DEFINE` commands to edit the incorrect information. How to edit the form and definitions, as well as several advanced features of these commands, are described in Chapter 9.

# Minimum and Maximum Values

When data is being entered into a database, Condor checks each data item value, as it is entered, against the minimum and maximum values allowed for that data item. If the entered data falls outside the acceptable range, an error message is displayed and the data must be re-entered correctly.

The minimum and maximum values have slightly different meanings, depending on the type of data item:

- For **Alphabetic** or **Alphanumeric** data items, they refer to the minimum and maximum number of characters that may be entered in the data entry field.
- For **Julian** (date) data items, they refer to the earliest and latest dates that may be entered in the field.
- For **Numeric** and **Dollars** data items, they refer to the smallest amount (including negative numbers) and largest amount that may be entered in the field.

For most fields, Condor will automatically pick the correct minimum and maximum values to use (its defaults). These are:

- For **Alphabetic** or **Alphanumeric** fields, a minimum of zero (0) characters and a maximum equal to the number of character positions in the data entry field.
- For **Julian** data items, the earliest and latest dates that Condor can represent.
- For **Numeric** or **Dollars** data items, the largest negative amount and largest positive amount that can be stored by Condor in a data entry field that size.

However, there will be data items where you want to restrict the entered data to a much narrower range. For these data items, you enter the data item **type** and the minimum and maximum values to be used, separated by commas, when Condor prompts you for the data item definition.

Example:

Suppose that the STATE data item on the CARDFILE database **must** have a 2-character State abbreviation entered. You can specify a minimum of 2 characters, maximum of 2 characters, and required field, with the data item definition:

```
STATE: AR,2,2,2 |Return|
```

Example:

Suppose that an AMOUNT data item must be greater than or equal to $0, but less than $5000. These minimum and maximum values can be set with the definition:

```
AMOUNT: $10,0.00, 5000.00
```

# Default Values

If a data item in a database has one value that is more common than all the others, or has a value that should be **filled in** on the form when no value is available, you should define that value as the **default** value for that data item. Later, when you are entering data into the database, you can merely press the |Return| key when you come to the data entry field for that item, and Condor will **automatically** insert into place on the form the default value. You specify a default value by listing it after the minimum and maximum values, separated by a comma, when Condor asks for a data item definition. The examples show several typical situations where **default** values can be used.

Example:

In the CARDFILE database, suppose that the value "N/A" is to appear in the TITLE data item whenever a title is not available. You can define "N/A" to be the default value for TITLE with the data item definition:

```
TITLE: AN, 20, 0, 20, "N/A"
```

Remember, the 0 and 20 specify a minimum of zero and maximum of 20 characters for the title. With this definition, you press the Return key when prompted for data for the TITLE field, and the system displays "N/A" in the field automatically.

Example:

Suppose the AMOUNT data item in a database is frequently zero, and we want to be able to enter a zero simply by pressing the Return key when it comes time to enter the AMOUNT. The definition:

```
AMOUNT: $, 10, 0.00, 500.00, 0.00
```

will make 0.00 the default amount.

For additional information on default values for data items and how they are used in data entry, see Chapter 4.

# Chapter Summary

- Defining a database is the first step in using a new database.
- A database is defined with the `DEFINE` command:

  ```
  DEFINE database-name
  ```

- The database form is painted on the HP Series 100 screen with HP Touch, the cursor keys, editing keys, and typewriter keyboard. Data item names are enclosed in square brackets, and data entry fields are indicated with a series of underscores.
- Each data item is further defined by telling Condor what type of data it contains. Data items can be alphabetic, alphanumeric, numeric, dollars-and-cents, or dates.
- A data item can be required or optional. If it is required, Condor will force the user to enter a value for that data item when data is entered.
- Minimum and maximum values may be optionally specified for each data item. Condor will not allow data outside this range to be entered into the database.
- Default values may be optionally specified for each data item, to speed data entry.

# Chapter 4

# ENTERING AND UPDATING DATA


After a database has been defined, you may enter data into it. Until the data is entered, the database is empty. This chapter describes the use of the ENTER command to enter new records into a database. The UPDATE command, which is used to modify data that is already stored in a database, is also described.

## Data Entry

Entering data into a Condor database is as simple as filling out an ordinary business form. Just as you might use a pen to "fill in the blanks" of a paper form (such as an order form or purchase order), you use the HP Series 100 keyboard to "fill in the blanks" of a Condor database form.

To enter information into a database, type the command:

```
ENTER database-name
```

The form for the database will be displayed on the screen. The name of each data item in the database will appear on the form, followed by an inverse-video field. This **data entry field** is the place where you will "fill the blank" with data.

Figure 4-1 shows the database form for the example CARDFILE database:

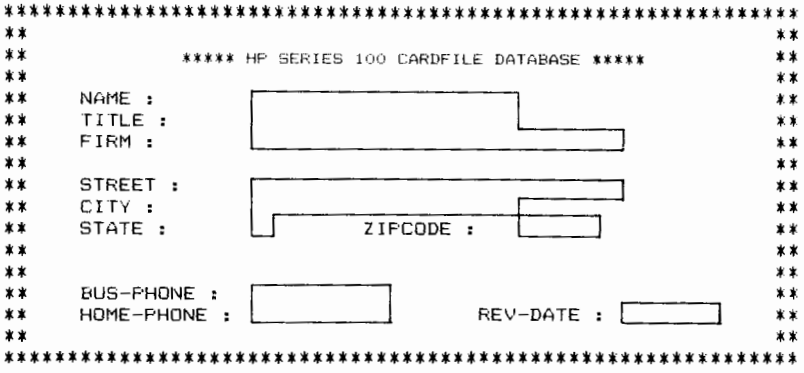


Figure 4-1. CARDFILE Data Entry Form

To fill in the form, just type! When you reach the end of a field, the cursor will automatically move to the next field. You can also use the HP Series 100 editing keys to edit data in the form:

| | |
|---|---|
| Tab | — moves forward to the next field on the form |
| Shift Tab | — moves backward to the previous field on the form |
| Return | — same as Tab ; moves forward to the next field |
| Back space | — backspaces one character, erasing that character |
| Arrow keys | — move the cursor up, down, left, and right within the form |

Normally, you will enter data onto the form, field by field, until the entire form is completed. You can signal completion of the form in several different ways:

- Typing into the last character position of the last field on the form.
- Pressing the Tab or Return key to exit the last field of the form.
- Pressing the End function key, which is always displayed while entering data onto a database form.

# Data Entry Function Keys

When the form is completed, Condor will display a new set of function key labels on the screen. Figure 4-2 shows these function keys.

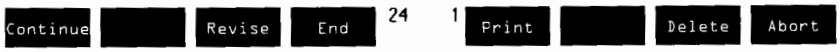


Figure 4-2. Data Entry Function Keys

Stop to review the data for accuracy at this point. Then, use the function keys to select what will happen next:

Press the Print function key to get a printed copy of the form and data. After the form is printed, the function keys are displayed again, for another choice.

Press the Revise function key if the data is incorrect and you wish to correct it. The data may be revised by moving the cursor to the character(s) in error, and overtyping. When all corrections have been made, press the End function key and the function key choices in Figure 4-2 will be displayed once again.

Press the Delete function key if the data is incorrect and you wish to "start again" with a clean form. The form will be cleared and the data will **not** be added to the database. You may now re-enter the data.

Press the Continue function key if the data is correct and you wish to continue entering data. The new record will be added to the database, and the form will be cleared. You may continue, entering the next record.

Press the Abort function key if the data is incorrect, and you wish to stop entering data now. The data in the form will not be added to the database. The screen will clear, and in a few seconds, the command prompt will appear.

Press the End function key if the data in the form is correct and this is the last record you wish to enter at this time. The new record will be added to the database, and the form will disappear from the screen. After a few seconds of "cleanup", the command prompt will appear, ready for your next command.

Example:

To enter data into the CARDFILE sample database defined in the last chapter, follow the following procedure:

Type `B:` in response to the `A>>` prompt
(this specifies drive B: as the current drive)

Type `ENTER CARDFILE` Return
(the CARDFILE form will appear on the screen)

Type `Jones, John` Return
(this will fill in the NAME field)

Type `President` Return
(this will fill in the TITLE field)

Type `U. S. Distributing, Inc.` Return
(this will fill in the FIRM field)

Type `3456 Johnson Ave.` Return
(the street address)

Type `Santa Ana` Return
(the city)

Type `AZ`
(the state—note that the cursor automatically moves to the next field)

Type `83637`
(the zip code)

Type `4237483736`
(the business phone)

Type `6372637482`
(the home phone)

Type `04/02/82`
(the revision date)

This completes entry of the first record. The function key labels of Figure 4-2 will appear at the bottom of the screen. Press Continue to add the record to the CARDFILE database. The form will be cleared, waiting for other records to be entered. Try entering other data, and try the Revise, Print, and other function keys. When you have entered four or five records, press the End function key to complete data entry.

# Range Checking

When a database form is defined, you may optionally specify a range of acceptable values for each data item. For example, you might restrict an amount to be less than $5000, or restrict a date to be between March 1 and June 30. As data is entered into the form, it is checked to be certain that it falls within the acceptable range. If you try to enter data that is outside the acceptable range, an error message is displayed, and the data is not accepted.

Normally, if you try to enter data that is outside the acceptable range, you are making an error, and you will want to re-enter the data correctly. However, you can manually override the range check and force Condor to accept data that is outside the acceptable range. This is done by pressing the Range Override function key when the `data out of range` error message appears, and then re-entering the data.

---

**NOTE**

You should use this feature with great caution, since entering out of range data may result in errors when database computations or comparisons are attempted later.

---

# Default Data Values

When a database form is defined, you may optionally set up a **default** value for each data item on the form. Usually, the default will be the most commonly entered value for that item. For example, if 90 percent of the people in the CARDFILE database will have the same city, state, and zip code, it may be helpful to have these values set up as **default values**.

When entering data, you may enter the default value for any field by pressing Return at the beginning of the data item field. The default value, if there is one, will be displayed in the field, and the cursor will move to the next field.

Example:

Suppose the CARDFILE database form had been defined with default values for CITY (Santa Ana), STATE (AZ), and ZIPCODE (83637). Then, while entering the CARDFILE record in the preceding example, you could simply press Return when it came time to enter the city, state, and zipcode fields. The default values would be automatically displayed and used for each of the three fields.

# The Auto-Default Feature

In some data entry applications, it is useful to have the default value for each data item appear in the form **before** beginning data entry for each record. This allows you to see the defaults while entering data, and either accept them, or type over them to enter other data. This **Auto Default** feature may be selected at any time by pressing the `Auto-Default` function key that is displayed during data entry. Whenever the **Auto-Default** feature is in operation, `Auto-Default` will appear at the bottom of the screen. Pressing the function key again will turn off the **Auto-Default** feature.

Example:

If the CARDFILE form is defined with defaults for the CITY, STATE, and ZIPCODE data items, then pressing the `Auto-Default` function key would cause these defaults to appear each time the form is displayed or cleared for entering a new record. Usually, the defaults will be the correct data, so you may simply skip the fields, saving data entry time. When the defaults are not the correct data for a particular record (e.g., a person is from Los Angeles), you may type over the default values to correct them.

# The Auto-Repeat Feature

In some data entry applications, much of the data that is entered for one record will be exactly the same for the following record, and the one following, etc. In this case, it is very useful if the form is **not** cleared when preparing to enter the next record. Rather, the data from the previous record is allowed to remain on the form. Only those few data items that change from record to record need to be entered (by typing over the information from the previous record).

This feature of Condor is termed **Auto-Repeat**, since the data from each data record is automatically repeated on the form for the next record. The **Auto-Repeat** feature can be selected at any time during data entry by pressing the Auto-Repeat function key. Auto-Repeat will appear at the bottom of the screen, indicating the **Auto-Repeat** feature is in operation. To turn off the **Auto-Repeat** feature, press the function key again.

Example:

Suppose your firm has just acquired a new customer, and thirty people from the firm are to be added to the CARDFILE database. All the people have exactly the same firm name, address, business phone, etc. The only data that varies from person to person is the name, title, and home phone.

**Auto-Repeat** mode will be very useful here. Enter the first person into the database normally and then press the Auto-Repeat function key. For each of the other 29 people, you can enter the name and title, Tab to the home phone field, and you have finished data entry! Each of the other fields (firm, city, state, etc.) will still be filled with the data from the preceding record, which is the same for all the records. In this way, data entry time for the new records will be cut dramatically.

The **Auto-Default** and **Auto-Repeat** features are mutually exclusive—both may not be selected at the same time. Pressing either the Auto-Default or Auto-Repeat function key will automatically turn off the other feature (if it was turned on).

# Updating Data

What happens when someone in the CARDFILE database has a new telephone number or changes address? The UPDATE command lets you retrieve the appropriate record from the database, display the data in the database form, revise the data as required, and then put the revised record back into the database.

To retrieve database records for updating, type the UPDATE command:

```
UPDATE database
```

Condor will ask you for the **search condition** for the update—that is, **which records would you like to update? The simplest form of search condition takes the form:**

```
data-item IS value
```


Examples:

```
NAME IS ''JONES, J.''
```
(find the record for J. Jones)

```
FIRM IS ''HEWLETT-PACKARD''
```
(find all of the records for HP employees)

```
REV-DATE IS 04/01/82
```
(find records revised on April 1)

```
CITY IS SEATTLE
```
(find the records for people from Seattle; notice that quotes are not needed when there are no embedded blanks)

Condor will search the database for records that match the search condition. If no records in the database match the search condition, Condor will print a message to that effect and ask for another search condition.

If a record is found that matches the search condition, it will be displayed on the screen in the database form. Note that more than one record may meet the search condition. (In the example above, perhaps more than one person in the CARDFILE database works for Hewlett-Packard). The first record found in the database that matches the search criterion is displayed.

At this point, you may review the data displayed on the form. The function key labels in Figure 4-3 are displayed on the screen.

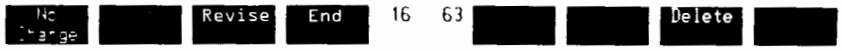


Figure 4-3. Update Function Keys

Use the function keys to select what to do next:

Press the **No Change** function key if you decide to leave the data unchanged, as it is displayed in the form.

Press the **Revise** function key if you wish to edit or modify the data in the form. You may then use the HP Series 100 editing keys to edit the displayed data, character by character, until it is correct. When the data is correct, press the **End** function key. The function keys in Figure 4-2 will be displayed, and you may print the revised record, revise it further, delete the record, etc. using these keys. Pressing the **Continue** function key at this point continues the update operation.

Press the **Delete** function key to delete the displayed record from the database completely. The record will be marked as "deleted", and the data in the record will no longer be accessible. Use the **Delete** function key with caution! If a record is deleted accidentally, you will have to re-enter all the data; there is no way to recover the deleted information.

Press the **End** function key to stop the UPDATE operation. If you have made revisions to the data in the form, the revised record will be replaced in the database. The form will disappear from the screen, and Condor will ask for another search criterion.

If more than one record in the database meets your search condition, pressing the Continue or Delete function key automatically brings the **next record** meeting the search condition into the form. This continues, record by record, until no more records meet the search condition. Condor will then ask you for another search condition.

If you wish to perform another search and update another set of records, enter a new search condition and repeat the steps above. If you are finished updating the database, press the Return key when Condor asks for a new search condition. The screen will clear and the command prompt will request the next command.

Example:

Suppose that the telephone number for U.S. Distributing (whose President was added to the CARDFILE database in the previous example) has changed. We wish to change it in the database:

Type `B:` in response to the `A>>` prompt
(select drive B: for processing)

Type `UPDATE CARDFILE`
(Condor will ask for a search condition)

Type `FIRM IS ''U. S. Dist*''`
(the form will appear, with the record of a U.S. Distributing employee displayed)

Press the Revise function key
(to revise the record)

Press Tab to get to the `BUS-PHONE` field
(change the number by typing over it)

Press the End function key
(indicating the end of the revisions for this record)

Press the Continue function key
(the revised record is replaced in the database)

Repeat the procedure with the next record displayed in the form. Experiment with the Revise, Delete, and No Change function keys as well.

# Search Conditions

Search conditions for the UPDATE command can become quite complex. In addition to testing for a match with a certain value, we can ask for records with a greater than/less than relationship as well. The table lists the tests that can be done to select records:

| Relationship | Code | Meaning |
|---|---|---|
| Equals | EQ<br>IS | Search for data item values that are identical to the specified value |
| Less Than | LT | Search for data item values that are less than the specified value |
| Less Than or Equals | LE | Search for data item values that are less than or equal to the specified value |
| Greater Than | GT | Search for data item values that are greater than the specified value |
| Greater Than or Equals | GE | Search for data item values that are greater than or equal to the specified value |
| Inequality | IS NOT<br>NE | Search for data item values that are not equal to the specified value |

In addition to these relationships, compound search conditions can be created by combining simple search conditions with an **AND** or **OR**. However, **AND** and **OR** conditions cannot be combined in a single search condition. Extensive abbreviation is also possible; see the Reference Manual for complete details.

Examples:

```
NAME LT ''B''
```
(selects records with names that begin with "A")

```
REV-DATE GE 02/01/82
```
(selects records revised since February 1, 1982)

```
FIRM IS ''HEWLETT-PACKARD'' AND DATE GT 02/01/82
```
(selects records that meet both criteria)

As the examples show, quotes (") may be used in search conditions to clarify the search for alphabetic data. Quotes **must** be used when the data item value being searched for includes embedded spaces or special characters. Thus, most names, addresses, etc. must be enclosed in quotes when used in search conditions.

Experienced Condor users may "shortcut" the dialogue by including the search condition directly in the UPDATE command itself, following the word **WHERE**:

Example:

```
UPDATE CARDFILE WHERE NAME IS ''JOHNSON, R.''

UPDATE CARDFILE WHERE DATE LT 01/01/82

UPDATE CARDFILE WHERE FIRM IS ''HEWLETT-PACKARD'' AND
NAME IS ''JONES, JOHN''
```

Search conditions are used in many Condor commands. Whenever a search condition is used, the rules and examples given here apply.

# Deleting Unwanted Records

You can use the UPDATE command to delete a small number of unwanted records from a database. Use the search condition to select the records to be deleted. Then, as each record appears in the form, choose either the No Change function key to keep the record, or the Delete function key to delete it. Condor-3 also provides a DELETE command for deleting large numbers of records automatically. This command is discussed in Chapter 6 of the Condor 20-3 User's Manual.

# Chapter Summary

- Database data is entered using the ENTER command. Data entry and update both use the database form to display data, a record at a time.
- As each record is entered, function keys are used to:

  Continue to the next record,

  Print the record,

  Revise the record,

  Abort the record, or

  End data entry
- Database update takes place through the UPDATE command.
- Records are selected for update through a **search condition**. The database is searched, and records matching the condition are displayed in the database form, a record at a time, for potential revision.
- The UPDATE command provides a Delete function key, which deletes the displayed record.

# Chapter 5

# SIMPLE INQUIRIES AND REPORTS


Condor's inquiry and reporting features let you obtain the data you need from your databases on demand. Condor can help you with everything from simple inquiries (e.g., "What is the address for U.S. Distributing?") to very complex reporting needs ("Please summarize sales trends by product line for our 20 largest customers over the last six months"). The next several chapters describe this full spectrum of inquiry and reporting capabilities. This chapter begins with the simplest inquires.

## Simple Database Inquiry

The simplest form of Condor database inquiry occurs when all the information to answer a question is contained in a single database. In the CARDFILE database, examples would include:

What is John Jones' address and phone number?
Which clients live in Arizona?
Who are the salesmen from U.S. Distributing?

You already know one simple way of answering this kind of question—with the UPDATE command of the last chapter! To pose a simple inquiry:

1. Type UPDATE database-name
   (Select the database that answers the inquiry)
2. Enter a **search condition**
   (Use the search condition to pose your question)
3. **Review** the displayed record(s)
   (Your answer comes in the form of a database record that matches the search condition, displayed in the database form.)
4. Press the No Change function key
   (If several records match the search condition, the No Change function key brings each record successively into the form for viewing.)
5. Press the End function key
   (When you have viewed all the records, the End function key will complete the inquiry.)
6. Enter another search condition
   (Condor gives you the opportunity for another inquiry in the same database. If you do not want to make another inquiry, press Return)

Example:

To see all the salesmen from U. S. Distributing, enter the command:

```
UPDATE CARDFILE
```

When Condor asks for a search condition, type:

```
TITLE IS Salesman AND FIRM IS ''U. S. Distributing''
```

The CARDFILE record for each person meeting the search condition will be displayed on the screen, in succession.

# The Display Command (Condor 20-3 Only)

For Condor 20-3 users, the `DISPLAY` command provides the same inquiry functions as the `UPDATE` command, without the danger of accidentally changing the database contents. The command is entered as:

```
DISPLAY database
```

Like the `UPDATE` command, the `DISPLAY` command searches the database for all records that meet a **search condition**. Condor asks for the search condition immediately after you enter the `DISPLAY` command. Identical rules apply for search conditions for both the `DISPLAY` and `UPDATE` commands. Here are some examples of search conditions:

`NAME IS ''JONES, JOHN''`
(John Jones' form would be displayed)

`STATE IS AZ`
(Several records would qualify, and you could browse through them)

`STATE IS NOT AZ`
(The opposite inquiry from the preceding example)

`ZIPCODE LT 02999`
(Zipcodes less than 02999 are selected)

`STATE IS NY AND FIRM IS ''W.E.I.''`
(Select people from W.E.I. in New York)

`FIRM IS ''W.E.I.'' OR FIRM IS ''R.X.I.''`
(Select W.E.I. or R.X.I. employees)

As the examples show, you have a great deal of flexibility to tailor the database inquiry to your exact needs. Search conditions can also be combined with **AND** and **OR**, to build up more complex search criteria. (Note: **AND** and **OR** cannot both be used in the same search condition).

After you enter the search condition, Condor will display the database form on the screen, and begin searching the database for records matching the condition. The first qualifying record is displayed in the form, and the function key labels in Figure 5-1 are displayed.

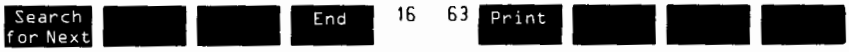


Figure 5-1. Display Command Function Keys

Use the function keys to select what will happen next:

Press the **Search for Next** function key to continue searching the database; the next record matching the search condition will be found and displayed in the form.

Press the **Print** function key to print the form and the displayed record on the system printer.

Press the **End** function key to stop the search.

When the **End** function key is pressed, or when Condor can find no more records that match the search condition, the screen is cleared. The command prompt appears, and Condor is ready to accept the next command.

# Approximate Matches in Search Conditions

In the examples thus far, searches have been based on an **exact match**, character by character between data item values and the value supplied as the search criterion. What happens if we wish to retrieve the CARDFILE record for a person named Robertson, but have forgotten his first name? Using an exact match, we cannot—if we enter a command searching for **NAME IS "ROBERTSON"**, and the name is stored as **"ROBERTSON, A."** in the database, no match will be found, and the record will not be retrieved.

Condor solves this problem with its **approximate match** feature. This feature is accomplished by using two "wild card" characters—a question mark ("?") or an asterisk ("*"). These characters function just like "wild cards" in a card game—**they will match any other character**. The question mark and asterisk work in slightly different ways:

**Question Mark:** A question mark matches any other **single character** in the same position. For a match to occur, the number of question marks must be exactly the same as the number of matching characters.

**Asterisk:** An asterisk matches all of the remaining characters in the data item. It may only be used at the end of the search string in the search criterion.

Examples:

| | |
|---|---|
| Search condition: | `NAME IS RICHARDSON???` |
| Matches: | RICHARDSON, J<br>RICHARDSON,JB |
| Does not match: | RICHARDSON, ANDREW<br>RICHARDSON, J.B. |

Search condition: `NAME IS RICHARDSON*`

Matches:
RICHARDSON, J
RICHARDSON,JB
RICHARDSON, ANDREW
RICHARDSON, J.B.

Characters following an asterisk are ignored. For example,

Search condition: `NAME IS RICH*EW`

Matches:
RICHARDS, EW
RICHARDSON, EW
RICH, ANDREW

And also matches:
RICHARDS, E.W.
RICHELIEU

An asterisk in the first position will match every record in the database.

Approximate matches can be used with both the `DISPLAY` and `UPDATE` commands, as well as several other Condor commands. In fact, with only a few exceptions (noted under individual commands in the Command Description section of the Condor Reference Manual), an approximate match can be used anywhere in a Condor command! Approximate matches can be very useful when searching for names, street names, etc., where exact spelling may be unknown. They are also useful for selecting records based upon just part of a data item (e.g., select the "VP-Sales", "VP-Finance", "VP-Manufacturing", etc. by searching for **TITLE IS "VP*"**).

# Database Report Formats

Simple database inquiries may be effectively answered by selecting records and viewing them in the database form, one record at a time. But what happens if you need to look at data from many different records at once? A printed report, organized into familiar rows and columns, is often the most convenient way of looking at this type of information.

With Condor's extensive reporting capabilities, you can organize data the way you need it, generating many different kinds of reports. Reports can be printed on the printer or displayed on the screen, on demand. There are five basic Condor report formats:

**Screen Format:** Records are printed in the database form; one record per form.

**Columnar Format:** Records are printed in a row/column report; one row per record; each column displays one data item.

**Columnar Format with Statistics:** Columnar format, with totals, subtotals, averages, etc.

**Summary Format:** Selected data items are tabulated (totalled, counted, etc.); summary statistics are included.

**Statistical Report:** Summary statistics (totals, counts, averages) are printed for the entire database.

Examples of each type of report and instructions on how to print it, are given in the sections that follow. There are three steps for generating any report from a Condor database:

1. Make sure that the database contains all the data needed for the report. (Chapter 6 tells how to combine data from different databases to generate more complex reports).
2. Sort the database into the order in which you want it to appear in the report (using the `SORT` command).
3. Print the actual report (using the `PRINT`, `LIST`, `TABULATE`, `STAX` and/or `TITLE` commands).

# Sorting a Database

Databases are sorted into order for a report using the SORT command. The database records are sorted based on the values of the data items named in the SORT command:

```
SORT database BY data-item1, ... data-item32 [option]
```

Up to 32 data items can be used for sorting (for example, to sort a database by state, and within each state by city, and within each city by firm name, etc.). If more than one data item is used, the first data item listed becomes the major sort item, with each successive data item being sorted within the preceding items. An ascending sort is normally used (A before B, 1 before 2, etc.). To sort into descending order instead, the **[D]** option of the SORT command is used (see the examples).

Examples:

```
SORT CARDFILE BY NAME
```
(Sorts into ascending order by name)

```
SORT CARDFILE BY FIRM, STATE, CITY
```
(Sorts into ascending order by firm, and within each firm, by state, and within each state, by city)

```
SORT CARDFILE BY DATE [D]
```
(Sorts into descending order by date, placing most recent dates first)

The SORT command may require some time to complete the sorting operation, especially on large databases. Progress messages are displayed on the screen as the sort operation proceeds.

Note: While sorting a database, Condor requires temporary working space. The SORT command obtains this space by creating a temporary disc file **on the current disc drive**. This file will be equal in size to the database being sorted, and it is automatically erased when the sort is complete.

If you are sorting a large database, you should check to make sure there is enough space on the current disc drive to accommodate this working space. If there is not enough space, insert a blank disc into a spare disc drive, and make that drive the current disc drive before beginning the sort. After sorting is complete, you can return to your previously-used current disc drive.

# Screen Format Report

Printing a report in screen format is very simple. First, be certain the database is sorted into the order you wish for printing. (To see if the database is sorted correctly, see the section on producing Columar Reports later in this chapter.) Then, enter the command:

```
PRINT database
```

Records from the database will be printed in the database form, on the system printer.

Example:

Entering the commands:

```
SORT CARDFILE BY NAME
PRINT CARDFILE
```

will produce the screen format report in Figure 5-2.

```
**********************************************************************
**                                                                  **
**            ***** HP SERIES 100 CARDFILE DATABASE *****           **
**                                                                  **
**    NAME :         Anderson, Harold E.                            **
**    TITLE :        Mfg. Rep.                                      **
**    FIRM :         Arizona Test Equipment                         **
**                                                                  **
**    STREET :       47 Harrison St.                                **
**    CITY :         Middletown                                     **
**    STATE :        AZ       ZIPCODE :    84749                   **
**                                                                  **
**                                                                  **
**    BUS-PHONE :    4634837473                                     **
**    HOME-PHONE :   9384737483        REV-DATE :   04/20/82        **
**                                                                  **
**********************************************************************


**********************************************************************
**                                                                  **
**            ***** HP SERIES 100 CARDFILE DATABASE *****           **
**                                                                  **
**    NAME :         Cartwright, Mary                               **
**    TITLE :        VP - Sales                                     **
**    FIRM :         Interna          l Freight C                   **
```

Figure 5-2. Screen Format Report

The function key labels of Figure 5-3 are displayed during printing of the report, and allow control over the printing operation:

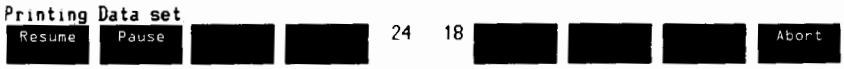


Figure 5-3. Report Control Function Keys

Press the **Pause** function key to temporarily halt the printer. (The printer will complete the current record before stopping.)

Press the **Resume** function key to start printing again after pressing the **Pause** function key.

Press the **Abort** function key to abort the report.

---

**CAUTION**

Care must be taken to make sure your Series 100 is connected to a printer before entering any print routine. If you print to a non-existent HP-IB printer (serial printer or thermal printer) you will have to hard reset the system. If you print to a non-existent serial printer (letter quality, or internal printer) control will be passed back to Condor.

---

# Screen Format Reports on the Screen

If you want to display a screen format report on the HP Series 100 screen instead of printing it on the system printer, use the `LIST` command:

```
LIST database
```

The report will look exactly the same as it would if printed on the printer. However, only a small part of the report can be displayed on the Series 100 screen at a time. The `LIST` command will display each record of the report on the screen, and then pause with the function keys labels of Figure 5-4 displayed:

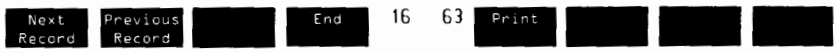


Figure 5-4. Screen Report Control Function Keys

Choose the next step by pressing the appropriate function key label:

Press the **Next Record** function key to see the next record of the report.

Press the **Previous Record** function key to see the preceding record of the report.

Press the **Print** function key to print the currently displayed record on the system printer.

Press the **End** function key to end the report.

If you accidentally press Next Record when the last record of the report is displayed, or press Previous Record when the first record of the report is displayed, the **LIST** command responds with the function keys in Figure 5-5.

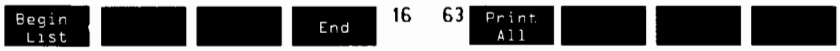


Figure 5-5. Screen Report Control Function Keys

These function keys provide the following choices:

Press the Begin List function key to begin the report, starting with the first record.

Press the Print All function key to print a report of all the records on the system printer. Note that this has the same effect as the **PRINT** command for screen format reports.

Press the End function key to end the report.

# Columnar Reports

Columnar reports are the common row/column reports that are used in business every day. To print a columnar report, first sort the database into the proper order for the report. Then enter the PRINT command:

```
PRINT database BY data-item1, data-item2, ...
```

Records from the database will be printed in sequence, each record appearing as one row of the report. The columns of the report are the selected data items from the PRINT command, appearing left to right, as they are listed in the command.

Example:

We want to print a report of the CARDFILE database, sorted geographically. The report is to show the STATE, CITY, FIRM, and NAME of each person in the database. Entering:

```
SORT CARDFILE BY STATE, CITY
PRINT CARDFILE BY STATE, CITY, FIRM, NAME
```

will produce the columnar report in Figure 5-6.

| STATE | CITY | FIRM | NAME |
|---|---|---|---|
| AZ | Middletown | U. S. Distributing, Inc. | Williams, Jeff |
| AZ | Middletown | Arizona Test Equipment | Harris, Bill |
| AZ | Middletown | Arizona Test Equipment | Jameson, Paula |
| AZ | Middletown | Arizona Test Equipment | Anderson, Harold E. |
| AZ | Phoenix | Arizona Test Equipment | Smith, John W. |
| AZ | Santa Ana | U. S. Distributing, Inc. | Jones, John |
| AZ | Santa Ana | U. S. Distributing, Inc. | Harrison, Jacob |
| AZ | Tucson | International Freight Co. | Robertson, Michael |
| AZ | Tucson | International Freight Co. | Kline, Paul J. |
| AZ | Tucson | International Freight Co. | Cartwright, Mary |

Figure 5-6. Columnar Report

While the report is printing, the function key labels in Figure 5-3 provide the same printing control as described under "Screen Format Reports" above.

The headings above each column in the report will be the data item names. You can prevent these headings from being printed by specifying the **[X]** option of the `PRINT` command.

Example:

```
PRINT CARDFILE BY STATE, CITY, FIRM, NAME [X]
```

# Columnar Reports With Statistics

Often, a columnar report will be more useful if the report contains totals, subtotals, averages, etc. Condor will automatically compute and print these statistics, if you request them in the `PRINT` command. The statistics that can be computed are:

**TOTAL** — The total of a column is printed
**MIN** — The minimum value in a column is printed
**MAX** — The maximum value in a column is printed
**AVERAGE** — The average of a column is printed
**STAX** — All the above statistics are printed

Statistics can be accumulated for one or more columns in a report. More than one type of statistic can also be requested for a given column in the report.

To generate a columnar report with statistics, first sort the database into the proper order for printing. Then, enter the PRINT command:

```
PRINT database BY data-item1, data-item2, ...
AND COMPUTE statistic data-item3, data-item4 ...
```

Remember, enter the command as one long command. Only press [Return] at the end of the command.

Examples:

To total the orders in an ORDERS database, enter:

```
PRINT ORDERS BY ORDER.NO, DATE, AMOUNT AND COMPUTE
TOTAL AMOUNT
```

The report in Figure 5-7 will be generated:

```
ORDER.NO  DATE          AMOUNT

56473624  12/01/81      456.56
35264537  12/01/81       56.45
64736453  12/02/81       50.00
56384637  12/03/81       47.75
46374673  12/03/81      100.00

Total                   710.76
```

Figure 5-7. Sample Columnar Report

The ''AND COMPUTE...'' phrase of the PRINT command can request several different statistics, and name several different data items for each different type of statistic. For example:

```
PRINT ORDERS BY ORDER.NO, AMOUNT, DATE, TAX
AND COMPUTE TOTAL AMOUNT, TAX AVERAGE AMOUNT
```

This will produce the report in Figure 5-8.

| ORDER.NO | AMOUNT | DATE | TAX |
|---|---|---|---|
| 56473624 | 456.56 | 12/01/81 | 56.56 |
| 35264537 | 56.45 | 12/01/81 | 6.45 |
| 64736453 | 50.00 | 12/02/81 | .00 |
| 56384637 | 47.75 | 12/03/81 | 7.75 |
| 46374673 | 100.00 | 12/03/81 | 10.00 |
| Total | 710.76 | | 80.76 |
| Average | 142.15 | | |

Figure 5-8. Sample Columnar Report with Statistics

In addition to totaling columns at the end of a report, you can request that subtotals be computed during the report. Subtotals will be printed each time the value of a specified data-item changes, when you add the phrase:

```
SUBTOTAL USING data-item
```

to the end of a PRINT command.

Example:

```
PRINT ORDERS BY ORDER.NO, DATE, AMOUNT, TAX
AND COMPUTE TOTAL AMOUNT, TAX SUBTOTAL USING DATE
```

This will produce the report in Figure 5-9.

| ORDER.NO | DATE | AMOUNT | TAX |
|---|---|---|---|
| 56473624 | 12/01/81 | 456.56 | 56.56 |
| 35264537 | 12/01/81 | 56.45 | 6.45 |
| Subtotal | | | |
| | | 513.01 | 63.01 |
| 64736453 | 12/02/81 | 50.00 | .00 |
| Subtotal | | | |
| | | 50.00 | .00 |
| 56384637 | 12/03/81 | 47.75 | 7.75 |
| 46374673 | 12/03/81 | 100.00 | 10.00 |
| Subtotal | | | |
| | | 147.75 | 17.75 |
| Total | | 710.76 | 80.76 |

Figure 5-9. Sample Columnar Report with Subtotals

If you request subtotals in the PRINT command, the subtotals will normally be **non-cumulative**. You can request **cumulative** subtotals, by specifying the **[A]** option on the command. As with columnar format reports, the **[X]** option will suppress column headings.

Example:

```
PRINT ORDERS USING ORDER.NO, DATE, AMOUNT, TAX
AND COMPUTE TOTAL AMOUNT, TAX SUBTOTAL USING DATE [A]
```

This will print the report as above, but with each day's order subtotal showing cumulative order dollars for the month, rather than the subtotal for that day only.

As you can see, `PRINT` commands can be quite lengthy. Extensive abbreviation is possible. See the description of the `PRINT` command in the Command Reference Section of the Condor Reference Manual for details.

# Columnar Reports on the Screen

Columnar format reports, with or without statistics, can be displayed on the Series 100 screen as well as printed on the printer. To produce the columnar reports on the screen, use the `LIST` command instead of the `PRINT` command:

```
LIST database BY data-item1, data-item2, ...
```

The report will appear on the HP Series 100 screen exactly as it would on the printer. All the options of the `PRINT` command for columnar format reports and columnar format reports with statistics also apply to the `LIST` command.

# Summary Reports

In the previous section, you saw how to print subtotals on a columnar report. If you are only interested in subtotal or statistical information (and not the detail report itself), then a summary report can be used.

Summary reports are generated with the TABULATE command:

```
TABULATE database BY data-item1, data-item2 ...
AND COMPUTE statistic data-item3, data-item4 [option]
```

This command displays, on the screen, the statistics requested, for the data items listed. The data in the database **must be sorted** into sequence by data-item1, data-item2, etc. before the TABULATE command is given.

Example:

To tabulate how many people in the CARDFILE database are from each city and state, enter:

```
SORT CARDFILE BY STATE, CITY
TABULATE CARDFILE BY STATE, CITY
```

The resulting report is shown in Figure 5-10.

| STATE | CITY | {Count} |
|---|---|---|
| AZ | Middletown | 4 |
| AZ | Phoenix | 1 |
| AZ | Santa Ana | 2 |
| AZ | Tucson | 3 |
| CA | Los Angeles | 1 |
| CA | San Francisco | 1 |
| CN | Toronto | 1 |
| CO | Denver | 1 |
| GA | Atlanta | 1 |
| IL | Chicago | 1 |
| MA | Boston | 1 |
| NY | New York | 1 |
| Total | | 18 |

Figure 5-10. Sample Summary Report

Example:

To tabulate orders by day, enter:

```
TABULATE ORDERS BY DATE AND COMPUTE TOTAL AMOUNT
```

The resulting report is shown in Figure 5-11.

| DATE | AMOUNT {Subtotal} |
|---|---|
| 12/01/81 | 513.01 |
| 12/02/81 | 50.00 |
| 12/03/81 | 147.75 |
| Total | 710.76 |

Figure 5-11. Sample Summary Report

The statistics available through the TABULATE command are the same as those for the PRINT command **(TOTAL, MIN, MAX, AVERAGE)**. As with the columnar reports, the **[A]** option can be used to tabulate cumulative subtotals.

# Statistical Reports

If summary statistics for an entire database are needed, a statistical report will provide all the necessary information. The statistical report is requested with the STAX command:

```
STAX database BY data-item1, data-item2 ... [option]
```

A variety of statistical information can be compiled through the options of the command:

[A] — calculate averages only for selected data items
[C] — calculate counts only for selected data items
[M] — calculate only the minimums & maximums
[T] — calculate totals only for selected data items
[P] — send the output to the printer

Examples:

To display total and average order statistics on your screen, enter:

```
STAX ORDERS BY AMOUNT [T] [A]
```

To print the same statistics to your printer, enter:

```
STAX ORDERS BY AMOUNT [T] [A] [P]
```

and generate the report in Figure 5-12.

```
                AMOUNT

Total
                710.76
Average
                142.15
```

Figure 5-12. Sample Statistical Report

If no option is specified, all the available statistics are printed for each data item.

# Printing Summary and Statistical Reports

The TABULATE and STAX commands normally display their results on the screen. To print the results on the printer instead, use the [P] option of the commands. For example:

```
TABULATE ORDERS BY DATE AND COMPUTE TOTAL AMOUNT [P]
```

Prints the report of Figure 5-11 on the printer instead of displaying it on the screen.

In addition to displaying reports on the screen and printing them on the printer, the TABULATE command can place a summary report into a **database**. Specifying the [S] option of the TABULATE command causes it to create a special database, whose records correspond to the rows of the summary report, and whose data items correspond to the report's columns. The database generated is the **RESULT** database, which is described fully in Chapters 6 and 9.

# Report Titles

Titles can be printed on Condor reports by using the TITLE command:

```
TITLE ''...text...''
```

The text (enclosed in quotes) will appear at the top of each page of the report when it is printed.

Example:

```
TITLE ''Outstanding Orders Summary Report''
```

More elaborate titles, with page numbering, dating, multi-line titles, and pagination, are available through various command options:

P or PAGE means print page n, where n is the current page
L or LINE means carriage return-line feed
D or DATE means print date set by the Condor DATE command
T or TOP means do form feed after nn lines
S or SKIP means skip a line, i.e., insert blank line
nn = number of lines between titles (default is 66)

For example,

```
TITLE ''...text...'',PAGE,''...text...''
```

causes the page number to appear in the midst of the title line, in exactly the position where the word PAGE appears in the command.

```
TITLE ''...text...'',DATE,''...text...''
```

similarly causes the date to appear in the midst of the title line.

```
TITLE ''...text1...'',LINE,''...text2...''
```

causes a line feed after printing text1, for a two-line title.

```
TITLE TOP, nn
```

causes a top-of-form after nn lines. Use this option to control the number of printed lines per page of the report. From 2 to 99 lines per page are allowed.

```
TITLE SKIP, nn
```

causes a skip of nn lines between every line. This allows you to perform double spacing (when nn = 2), triple spacing (when nn = 3), etc.

The various options of the TITLE command can be combined in a single command, leading to intricate control over report titling.

Examples:

The commands:

```
TITLE DATE,'' Daily Order Report '',PAGE
TITLE TOP, 66, ''Order Analysis'',LINE,''By Product
Line''
```

produce the titles shown in Figure 5-13.

```
 01/26/83  Daily Order Report  Page 1

 Order Analysis
 By Product Line
```

Figure 5-13. Report Title Examples

# Chapter Summary

- The UPDATE and DISPLAY commands are used for simple database inquiry. Each uses search criteria and displays the matching records in the database form.
- Screen-format reports display each record in the database form, and are produced by the commands:

```
PRINT database
LIST database
```

- Columnar reports display data in row/column format, and are produced by the commands:

```
PRINT database BY data-item1, ...
LIST database BY data-item1, ...
```

- Statistics (totals, averages, etc.) may be added to columnar format reports with the clause:

```
AND COMPUTE statistic data-item ...
```

- Statistical and summary reports suppress the report detail and provide summary information only. They are produced with the commands:

```
TABULATE database BY data-item
STAX database BY data-item
```

- The TITLE command is used to control printed report formats, and to print report titles.

# Chapter 6

# ADDITIONAL DATABASE CAPABILITIES


For many questions that arise in daily business use, the commands that were covered in the previous chapters will provide the needed information. This chapter covers additional commands for combining two databases, selecting certain information from a database, and performing arithmetic on databases.

## The Result Database

Two of the database commands in this chapter take data from one or more databases and put it into a temporary holding database. This database is appropriately named the `RESULT` database. Figure 6-1 shows the role of the `RESULT` database:

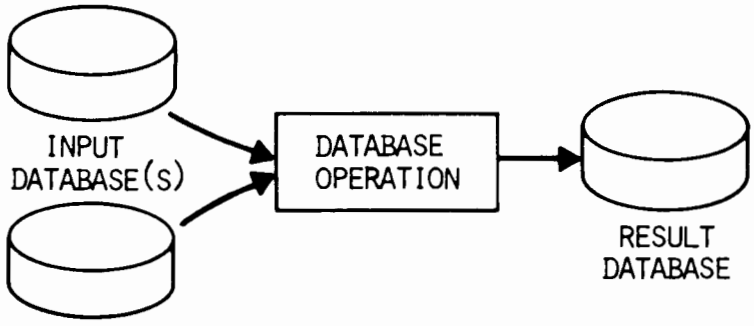


Figure 6-1. The RESULT Database

The RESULT database is always created on the **current disc drive**. It is **automatically** created by commands that produce a RESULT database.

Creating the RESULT database on a disc requires that a **Data Dictionary** already exists on that disc. This will automatically be true if any database has been defined or copied onto the disc. Chapter 9 fully explains the Data Dictionary and its function.

# Selecting Database Records

One of the most common types of inquiries is "show me all the records that meet the following criteria. . .". For example, you may want to select all the people in the **CARDFILE** database who live in Texas; or select all parts in an **INVENTORY** database where stock-on-hand is less than six months supply.

This type of selection is done with the SELECT command:

```
SELECT database
```

The SELECT command is very similar to the UPDATE command. Like this command, it asks for a search condition, and searches the database for records that match the condition. But with the SELECT command, the records that match the search condition are placed in the **RESULT** database, so that you may process them further. Figure 6-2 illustrates the SELECT command.

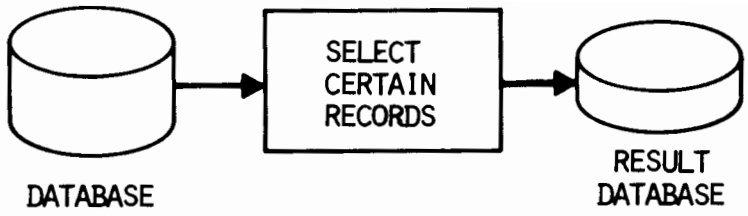


Figure 6-2. The SELECT Command

You can shorten the SEARCH command dialogue by including the search condition in the command, following the word "WHERE":

```
SELECT database WHERE search-condition
```

Example:

```
SELECT CARDFILE WHERE STATE IS AZ
```
(records for Arizona residents are copied to the RESULT database)

```
PRINT RESULT BY NAME, CITY, STATE
```
(print the RESULT database)

The rules for search conditions in the SELECT command are the same as those for the UPDATE command (see Chapter 4).

# Saving the RESULT Database

Whenever you use a command that creates a **RESULT** database, the former contents of the **RESULT** database are replaced and lost. If you want to preserve the **RESULT** database, and return to it for later processing, you must save its contents. The SAVE command lets you save the **RESULT** database under another name:

```
SAVE database
```

The **RESULT** database is copied into the database named in the command. Figure 6-3 illustrates the effect of the **SAVE** command.

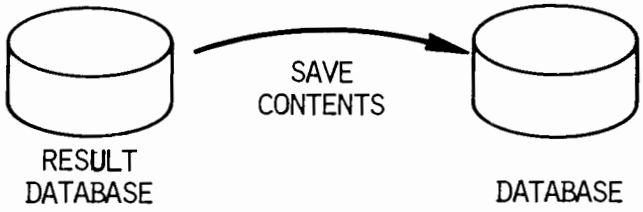


Figure 6-3. The SAVE Command

The SELECT and SAVE commands are often used together for complex inquiries that ask "what if" questions. A common inquiry technique is to successively narrow the search condition until an acceptably small number of records is found, then review the records selected. The following example illustrates this database inquiry technique.

Example:

Suppose we wish to search an EMPLOYEE database, to find employees eligible for the position of Manager of Engineering. Candidates should meet the following qualifications:

Mandatory conditions:
BS or MS degree
Management experience

Desirable conditions:
MSEE degree
Not currently at or above salary level
5 or more years experience with the company

First find those matching the mandatory conditions:

```
SELECT EMPLOYEE WHOSE DEGREE IS BS* OR DEGREE IS MS*
```

Database: EMPLOYEE 2820 Records.
Busy
Total records in Result Set = 152

Two SELECTs must be used because AND and OR cannot be combined in the same search condition:

```
SELECT RESULT WHOSE POSITION > 1999
```

Database: RESULT 152 Records
Busy
Total records in Result Set = 51

We assume position codes 2000 and above are management. Note that the RESULT database is used as the source of the search, and is overwritten with the result of the search. Candidates meeting the mandatory conditions are saved in case succeeding searches yield too few candidates and we wish to backtrack:

```
SAVE POTENTL
SELECT POTENTL WHOSE SALARY < 40000
```

Database: POTENTL 51 Records
Busy
Total records in Result Set = 30

Close, let's try for MSEE's:

```
SELECT RESULT WHOSE DEGREE IS MSEE
```

Database: RESULT 30 Records
Busy
Total records in Result Set = 2

Probably the best candidates. Find out who they are:

```
LIST RESULT BY NAME, DEPT
```

Now, backtrack to get secondary candidates:

```
SELECT POTENTL WHOSE SALARY < 40000 AND DEGREE IS BS?E

SELECT RESULT WHOSE HIRE.DATE < 6/1/76

LIST RESULT BY NAME, DEPT
```

As the example shows, the SELECT/SAVE combination allows powerful ad hoc inquiry, which can explore many different paths before reaching its conclusion.

# Combining Two Databases

Data from two databases can be combined together into a single database if both databases have the same structure (identical number of fields and record sizes). For example, clients who attended each of two different seminars may have been registered into two different databases, and you can combine the two databases to analyze the entire group of attendees.

The APPEND command permits the combining of records from two databases.

# The Append Command

The APPEND command adds the records of one database to another:

```
APPEND database1 database2
```

Both databases must have an identical structure; i.e., they must contain the same number of data items, in the same order, with the same lengths. The records of database 2 are appended (added) to database1. Database2 remains unchanged by the command. Figure 6-4 illustrates the command.

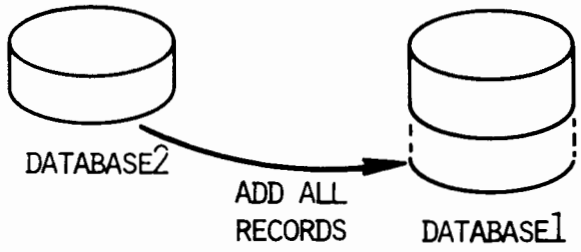


Figure 6-4. The APPEND Command

Example:

Orders for the day are contained in an ORDERS database. At the end of the day, these order records must be added to the other order records for the month, in the month-to-date orders database (MTDORDER), from which month-to-date reports are generated. The command:

```
APPEND MTDORDER ORDERS
```

appends the day's orders to the month-to-date database.

# Database Arithmetic

Data item values within a database can be changed in many different ways. The UPDATE command (described in Chapter 4) allows you to change values, one-by-one, for selected records in a database. Updating an entire database in this way would be very tedious, however. The COMPUTE command allows you to change a data item value throughout an entire database with one command.

# The COMPUTE Command

The COMPUTE command calculates a new value for a data item in every record of a database. The data item to be computed, and the formula for computing its new value, are both given in the command:

```
COMPUTE database ST data-item = expression
```

For each record in the database, the COMPUTE command performs the computation indicated, and assigns the resulting value to the data-item. The expression in the command must be an algebraic expression using addition (+), subtraction (−), multiplication (*), and division (/).

Examples:

```
COMPUTE ORDERS ST AMOUNT = QUANTITY * UNIT.PRICE
COMPUTE ORDERS ST TAX = AMOUNT * 6/100
```

The individual variables that are added, subtracted, multiplied and divided must be either numeric constants (e.g., 100) or numeric other data items from the same record. The expression is evaluated according to parentheses within the equation. If there are no parentheses, Condor performs multiplication and division before addition and subtraction. Dollars-and-cents constants in the expression should always have a decimal point and two decimal places. Constants or data items without decimal points are treated as cents when combined with dollars-type data items (i.e., 1067=$10.67). Refer to the COMPUTE command in the Command Reference Section of the Reference Manual if you would like more details.

# The POST Command

The COMPUTE command performs its arithmetic within each individual database record. Data item values within the record determine the new value for the data item being changed. Sometimes, data item values must be changed based on information from **outside** the database.

For example, suppose we were accumulating regional month-to-date order statistics in a REGIONS database, with one record per sales region. As each individual order is received during the day, it is entered into an ORDERS database. Each order includes information identifying the sales region from which it came. How can the REGIONS database be updated at the end of the day to reflect the day's orders?

The POST command performs this updating function. Using the POST command, records from one database (usually referred to as the **transaction** database) can be POSTED to another database (usually referred to as the **master** database), altering the contents of the master database. In the example, we would **POST** the **ORDERS** database to the **REGIONS** database, updating regional order statistics.

Figure 6-5 illustrates the operation of the POST command.

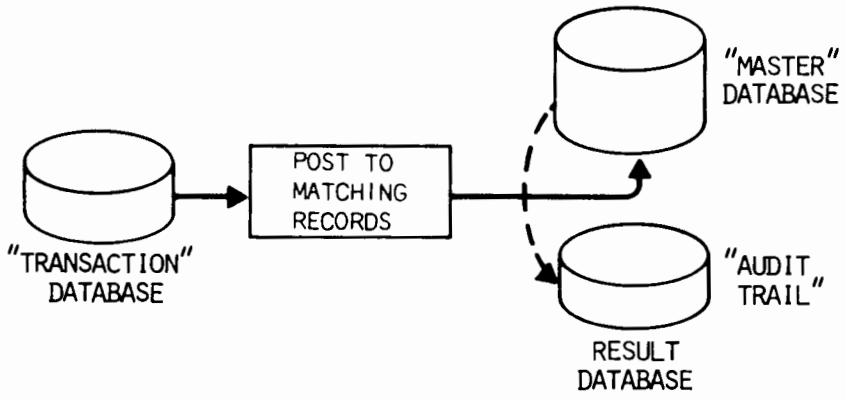


Figure 6-5. The POST Command

For each record in the **transaction** database, the POST command:

1. Finds the matching **master** database record, using one or more **matching fields. The matching fields must have the exact same structure (same data type, length, etc.).** In our example, the **matching field** is the REGION, which determines which master database record will be posted.
2. POSTs the **master** database record, as requested in the POST command. Several different data items in the **master** database can be updated with a single POST command.

The POST command can perform several different kinds of posting:

ADD — the data item value from the **transaction** record is added to the corresponding data item in the **master** database record. You would use this option to accumulate total orders, for example.

SUBTRACT — the data item value from the **transaction** record is subtracted from the corresponding data item in the **master** database record. You would use this option to process withdrawals from a total inventory count, for example.

REPLACE — the data item value from the **transaction** record replaces the corresponding data item value in the **master** database record. You would use this option to replace the **last update** date in a master database, for example.

The databases, matching fields, posting operation, and data items for posting are all named in the POST command:

```
POST database1 database2 MATCHING data-item1, ...
posting-operation data-item2 ...
```

**Database1** is the **master** database, and **database2** the **transaction** database for the posting operation. **Data-item1** specifies the matching field. The posting operation may be any of those listed above; the operation is carried out on the data-item(s) listed as **data-item2**.

Example:

Daily orders from the ORDERS database are to be posted to the REGIONS database. The command:

```
POST REGIONS ORDERS MATCHING REGION ADD AMOUNT
```

causes each order record to be matched against the correct regional record in the REGIONS database. The AMOUNT from the ORDERS record will be added to the AMOUNT in the REGIONS record. At the completion of the `POST` command, the REGIONS database is up-to-date with the day's new orders.

Example:

Withdrawals from a PARTS inventory database are entered into a TRANSACT database during the day. At the close of business, the PARTS inventory must be updated, and reports generated for ordering new parts. The command:

```
POST PARTS TRANSACT MATCHING PART-NO SUBTRACT
QUANTITY
```

causes each TRANSACT record to be matched with the master PARTS inventory record for the part withdrawn. The quantity on hand is decreased by the quantity of parts withdrawn. At the end of the `POST` command, the PARTS database has been updated to reflect the day's withdrawals.

You may use more than one matching field by listing several data-items, separated by commas, following the word **MATCHING** in the command. If several matching fields are listed, **all** the fields must match for the posting to occur. Similarly, the posting-operation may be applied to more than one data-item. You can also combine different posting options in the same command.

Examples:

```
POST PARTS TRANSACT MATCHING PART-NO, BIN-NO SUBTRACT
QUANTITY
```

(two matching data items)

```
POST PARTS TRANSACT MATCHING PART-NO SUBTRACT QUANTITY,
REPLACE DATE
```

two posting operations)

The POST command creates a **RESULT** database as a by-product of the POSTing operation. Before a **master** database record is updated, it is copied to the **RESULT** database. The **RESULT** database thus serves as an audit trail of the POSTing operation. This can be saved or printed using the appropriate command.

The POST command must search the **master** database for each record of the **transaction** database, which can be a time-consuming task. The command will operate much more rapidly if both the **master** and **transaction** databases are sorted into the same sequence by matching fields before POSTing.

For more examples of the POST command, see the General Ledger example in Chapter 11.

# Comparing Two Databases

Two databases can be compared to one another using the Condor COMPARE command. The compare command lists the individual data-items that are to be compared, record by record:

```
COMPARE database1, database2 MATCHING data-item1, ...
```

The COMPARE command works in the following way:

1. A record of database1 is read.
2. A search is made through database2, to find a record whose data-items match those of the database1 record. All the data-items listed in the COMPARE command must match in order for the database2 record to be considered a "match".
3. If a matching record is found, the **database1 record** is added to the **RESULT** database.
4. Steps 1-3 are carried out for each record in database1.
5. At the end of the COMPARE command, the **RESULT** database will contain those database1 records for which matching database2 records were found.

Figure 6-6 illustrates the COMPARE command.

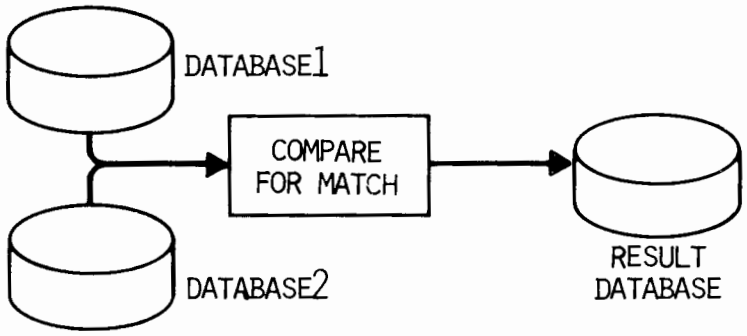


Figure 6-6. The COMPARE Command

Example:

Two databases of prospective customers have been produced from two different sources. We wish to compare the databases, to find duplicates. The command:

```
COMPARE LIST1, LIST2 MATCHING NAME, FIRM
```

will produce a **RESULT** database containing all the records in LIST1 that had a duplicate record in LIST2 (i.e., the same NAME and FIRM).

The `COMPARE` command can also be used to find the records in a database that do **NOT** match any records in another database:

```
COMPARE database1, database2 NOT MATCHING data-item,
...
```

The `COMPARE...NOT MATCHING` command follows exactly the same five-step process described for the `COMPARE` command. However, the records placed in the **RESULT** database will be the records from database1 for which **no** matching database2 record can be found.

Example:

Daily orders have been entered into an ORDERS database, and a data-item in each record tells from which sales region the order came. Before posting the REGIONS database with the ORDERS, we should insure that only valid regions have been entered into the ORDERS database records. The command:

```
COMPARE ORDERS REGIONS NOT MATCHING REGION
```

will place in the **RESULT** database any ORDERS record which does not have a matching region in the REGIONS database. These records should be corrected before posting.

Note that both databases in a `COMPARE` command must contain the data-items to be compared, and the data items must be defined the same way (data item name, type and size) in both databases. However, the other data items in the two databases may be quite different.

The order of the databases in the `COMPARE` command is also important. Note that

```
COMPARE ORDERS REGIONS MATCHING REGION
```

is not the same as

```
COMPARE REGIONS ORDERS MATCHING REGION
```

In the first example, records from the ORDERS database will be placed in the **RESULT** database; in the second, records from the REGIONS database will be placed in the **RESULT** database.

More examples of the `COMPARE` command can be found in the General Ledger example of Chapter 11.

# Chapter Summary

- Some commands place their results in a temporary database called the **RESULT** database.
- The `SELECT` command selects specific records from a database and places them in the **RESULT** database.
- The `SAVE` command saves the **RESULT** database for later use.
- The `APPEND` command adds the records of one database to another.
- The `COMPUTE` command computes a new value for a data-item from an algebraic expression.
- The `POST` command posts **transaction** records to a **master** database, updating selected data items.
- The `COMPARE` command compares two databases, placing in the **RESULT** database the records that do or do not match.

# Chapter 7

# DATABASE UTILITIES


In everyday use, there are several "housekeeping" functions that are critical for managing and maintaining your databases. Functions such as database backup, copying a database from one disc to another, switching among several different discs with databases on them, etc., are an important, if mundane, part of daily database use. This chapter describes the "utility" commands that help you perform these functions.

## Changing Discs

After a short while, it is likely that you will have databases on several different flexible discs. During a single session with Condor, you may wish to process several of these databases. The LOGDISK command allows you to switch discs in an orderly fashion.

**CAUTION**

You must **never** remove a disc from the disc drive while Condor is processing a database command. **Removing a disc while Condor is reading or writing data on it can result in loss of data and may result in loss of all the databases on the disc.**

---

**The only time you should remove and insert discs when using Condor is when the command prompt:**

```
A>>, B>>, C>>, etc.
```

appears on the screen. When this prompt appears, Condor is waiting for a command from you and is not accessing any of the discs.

At this point you may remove a flexible disc and insert another in its place. After switching discs in this way, you must inform Condor that a new disc is present, by **logging in** the disc through the `LOGDISK` command:

```
LOGDISK drive:
```

Example:

You have removed the flexible disc in drive B: and replaced it with another. Enter the command:

```
LOGDISK B:
```

to have Condor recognize the new disc.

Failure to properly **log in** the new disc can produce erroneous results and error messages from Condor.

# Listing Database Names

A listing of all the databases that are defined on a disc may be obtained through the DICtionary command:

```
DIC drive:
```

This command may be especially useful to find the name of a database that you had forgotten, or to locate unused databases that should be destroyed. The names of all the defined databases on the requested drive will be displayed on the screen. If you wish to see a database listing for the current disc, you do not have to type the drive name in the command.

Examples:

`DIC D:`
(lists databases on drive D)

`DIC`
(lists databases on the current disc)

The information displayed comes from the **Data Dictionary**, and includes the database name ("title") for each database, along with the names of the corresponding form file, definition file, and data file. (Chapter 9 describes the Data Dictionary and the role of these component files).

# Listing Disc Contents

The names of the files stored on a disc can be listed using the `DIR` (**directory**) command:

```
DIR drivename:
```

This command performs the same function as the system directory command. The names of the files on the requested disc drive are listed on the screen, in four columns. Both the filename and the extension, if any, are listed, for each file.

If the drivename is not given in the command, a directory listing for the current disc drive is displayed. Note that the directory command deals directly with files on the disc, and not with databases. Thus, each database stored on the disc will be listed as its component files (form file, data file, etc.). See Chapter 9 for a description of these files.

# Changing the Name of a Database

The name of a database can be changed with the `RENAME` command:

```
RENAME newname = oldname
```

This command changes the name of the database, and renames all its associated files to reflect the new name.

Examples:

```
RENAME CUSTOMER = CLIENTS
RENAME MTDACCTS = DAYACCTS
```

# Copying a Database

There are several reasons why it may be necessary to copy a database from one disc to another. Frequently-used databases stored on flexible discs should be copied to fresh diskettes when the original disc becomes worn. Critical databases should also be copied frequently, and a **backup** copy stored in a safe place. Or, a copy of a database may be produced on a flexible disc, to be mailed to another HP Series 100 user for processing.

A duplicate copy of a database is made with the `COPY` command:

```
COPY newdatabase = olddatabase
```

Examples:

`COPY E:ORDERS = B:ORDERS`
(copy the orders database from drive B to drive E)

`COPY E:ORDR29 = B:ORDERS`
(copy to a database with a different name)

`COPY E:ORDR29 = E:ORDERS`
(duplicate database on the same disc)

# Destroying Unwanted Databases

When a database is no longer needed, you can destroy it with the DESTROY command:

```
DESTROY database
```

The DESTROY command frees all the disc storage that the database required, for use by other databases or files. After a database has been destroyed, the data in the database is no longer accessible. The form and data item definitions for the database are also destroyed, and are no longer accessible.

Example:

```
DESTROY MYDATA
```
(destroys the database named MYDATA)

The DESTROY command has drastic effects! Condor will ask you to confirm that you actually want to destroy the database named in the command before carrying out the destruction. Use the command with care.

# Emptying a Database

From time to time, you may need to purge a database of all its contents, but keep the database form and data item definitions. For example, a database that contains month-to-date order statistics must be cleared of its data as part of the end-of-month processing, to prepare for the following month. You can **empty** a database of its data with the EMPTY command:

```
EMPTY database
```

The EMPTY command logically deletes all the records from a database. The data is no longer accessible following the command. However, unlike the DESTROY command, the database form and data item definitions for the database are preserved.

Example:

```
EMPTY ORDERS
```
(deletes all records in the ORDERS database)

Because of its drastic effects, confirmation is required before the EMPTY command is carried out.

# Setting the Current Date

Condor maintains a **current date** for printing the date in report titles, and for comparing and manipulating dates in databases. You may use the current system date or you may set the date to some other date by using the DATE command:

```
DATE mm/dd/yy
```

This changes the **current date** to the date shown. You can also use the DATE command to view the current date, by entering:

```
DATE
```

without typing a new date. Condor will display its **current date**, and then request a new date from you. At this point, you may either enter a new date, or press Return , in which case the **current date** remains unchanged.

Examples:

```
DATE 12/25/81

DATE 1/2/82

DATE
```

# Printing Reports to a Disc File

Normally, when you instruct Condor to print a report (using the `PRINT` command, for example), the report is printed on the system printer. The printer used is selected with the HP Series 100 configuration menu, as described in your system **Owner's Guide**. You can ask Condor to print its reports to a disc file instead, with the command:

```
SET PRINTER filename
```

After entering this command, all subsequent printed output will be written in the named file instead of being printed on the printer. To return printed output back to the printer, enter the command without a filename:

```
SET PRINTER
```

The `SET PRINTER` command may be especially useful for reports that must be edited (e.g., with Series 100/WordStar) before being printed. It also allows printed reports to be incorporated into documents, transmitted to a host system electronically, etc. See Chapter 8 for information on using the `SET PRINTER` command to pass information from Condor to other applications.

# Setting Condor Operating Characteristics

Several features of Condor operation can be selectively switched on or off by the user for convenience in database processing. These features are selected through the SET command, which has many different variations.

The SET DATE command controls how Condor will display and print dates. The available formats of this command are:

SET DATE MDY — dates appear as mm/dd/yy
SET DATE DMY — dates appear as dd.mm.yy
SET DATE YMD — dates appear as yy-mm-dd

The SET ECHO command controls the echoing of Condor commands and messages to the screen, especially during command procedure file processing. It has two forms:

SET ECHO ON — commands and messages are displayed
SET ECHO OFF — commands and messages are not displayed

The SET ABORT command controls whether a command procedure automatically aborts when an error occurs. Normally, the command procedure is aborted. (See Chapter 10 for a description of the Condor Command Procedure capability.) The SET ABORT command has two forms:

SET ABORT ON — errors cause command procedures to abort
SET ABORT OFF — command procedures continue despite errors

The SET AUTOSEL command controls the system's search for its command files. Each time you enter a command, Condor searches for the corresponding **command file**, which contains the HP Series 100 instructions needed to carry out the command. With the AUTOSELECTION feature, Condor first searches the current drive and, failing to find the command file there, will search the master drive for the command file. Normally, the command file will be found on the master drive.

You may disable this feature (restricting the command file search to the current disc only) with the commands:

```
SET AUTOSEL ON  — Autoselection will occur
SET AUTOSEL OFF — Autoselection will not occur
```

Normally, autoselection should be left ON. Note: you may override the entire command search by explicitly designating the drive on which the command file is located as part of the command name (e.g., `C:LIST` executes the `LIST` command from the command file on drive C).

The `SET MASTER` command allows you to set the **master drive** that will be used for the autoselection feature. Normally, the master drive is drive A, which contains the Condor programs. Under unusual circumstances, you may switch to another master drive, with the command:

```
SET MASTER drive:
```

where `drive` is the desired drive name.

Example:

```
SET MASTER C:
```

Sets drive C as the new master drive.

# Exiting Condor

After finishing a session with Condor, you may exit with the SYSTEM command:

```
SYSTEM
```

This command causes an orderly shutdown of Condor, and returns you to P.A.M. so that you may choose another application. If you had originally started Condor by using the DBMS command, then you will return back to the operating system instead of P.A.M. The prompt

```
A>
```

will appear instead of the P.A.M. menu.

# Running Non-Condor Programs

Occasionally, you may want to run a non-Condor program while in the midst of a Condor database management session. You may of course, exit from Condor, run the program, and then re-enter Condor control. However, you can also execute the program while under Condor control, by typing the program name, preceded by a dollar sign ($), in response to the Condor command prompt.

Examples:

`$CHKDSK` — runs the MS™-DOS utility, $CHKDSK, which displays disc status information

`$MYPROG` — runs the application program MYPROG

This capability is limited to small programs that can fit into HP Series 100 memory along with Condor's main program. Large programs that make use of all or nearly all the available user space on the HP Series 100 must be executed by exiting Condor.

*MS™-DOS is a trademark of MicroSoft Corporation.*

# Chapter Summary

- Several utility commands provide "housekeeping" functions for Condor databases.
- The `LOGDISK` command is used when switching discs, to **log in** the new disc.
- The `DIC` command lists all the currently-defined databases on a disc.
- The `DIR` command lists the files on a disc.
- The `RENAME` command is used to change the name of a database.
- The `COPY` command copies databases.
- The `DESTROY` command destroys a database, including its form and data item definitions.
- The `EMPTY` command deletes all the records in a database.
- Condor uses the current system date. You may enter another date by using the `DATE` command.
- Other Condor operating characteristics are set with the `SET` command.
- The `SYSTEM` command exits Condor, and returns to P.A.M.
- Small non-Condor programs can be run without exiting Condor, by entering a dollar sign followed by the program name, as a command. (e.g., `$CHKDSK`).

# Chapter 8

# USING DATABASES WITH OTHER APPLICATIONS


Condor databases can be used as a source of data for other HP Series 100 applications, from word processing to business graphics. For example, you might use a customer database to generate a customized mailing through Series 100/WordStar. Or, you might use Series 100/Graphics to plot data from an orders database.

Data from other HP Series 100 applications can also be entered into Condor databases for processing and reporting. You might have a Basic program that generates data for database processing by Condor. Or, you may want to feed financial data from Series 100/VisiCalc into Condor for tracking purposes.

This chapter shows how to exchange data with some of the other HP Series 100 applications.

# Printing Reports to a File

A simple way of passing data from Condor to other applications is by printing Condor reports into a disc file instead of printing them on the system printer. Once a report is stored in a disc file, it may be incorporated into a Series 100/WordStar document, or transmitted to a host system via Series 100/DSN/Link. Printed output is directed to a disc file with the command:

```
SET PRINTER filename
```

All subsequent printed reports will be stored in the disc file instead of being printed on the printer. If the file extension is not specified in the filename, Condor will give it the extension".PRT" (e.g. SET PRINTER TEMP goes to TEMP .PRT). To return to normal printing, enter the command:

```
SET PRINTER
```

The file is closed, and normal printing resumes.

Example:

To print a report to a disc file, for later transmission to a host system, enter the following command sequence:

```
SET PRINTER RPTFILE
(the report will be stored in the file RPTFILE.PRT)
PRINT CARDFILE BY NAME, FIRM, TITLE, DATE
(the report is actually "printed" in the file)
SET PRINTER
(return to normal printing)
```

The file RPTFILE.PRT can now be transmitted to a host via DSN/Link.

# READ and WRITE Commands

The Condor READ and WRITE commands provide a more flexible way to exchange data between a database and other applications. Figure 8-1 shows how these commands work.

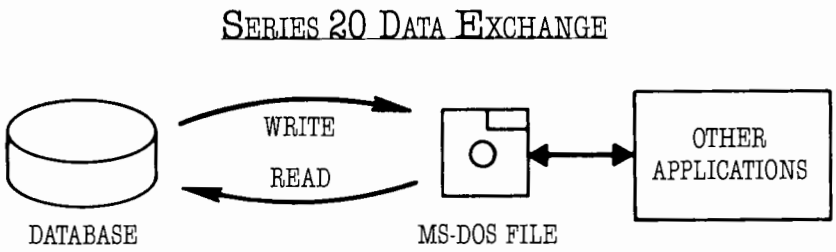


Figure 8-1. READ and WRITE Commands

The WRITE command extracts data from a database, and places it in a disc file. Once the data is in the file, applications such as Series 100/WordStar, Series 100/Graphics, and your own programs can process it. Different HP Series 100 applications will require that the data file be created in slightly different formats. An option (which must be in uppercase) in the WRITE command controls the format that is used to write the file:

```
WRITE database output-filename [option]
```

The READ command takes data from a disc file and enters it into a database. The data in the file may have come from a Series 100/WordStar non-document, a Series 100/VisiCalc spreadsheet, a DSN/Link file transfer, your own programs, etc. Once entered into the database with the READ command, the data can be manipulated using all the other Condor commands.

The syntax of the READ command is:

```
READ database input-filename
```

Specific instructions for using databases with other HP Series 100 applications are given below. For complete details on the READ and WRITE commands and their file formatting options, refer to the Command Reference Manual.

# Series 100/WordStar Documents

Data from Condor databases can be incorporated in tabular form into memos and other documents prepared using Series 100/WordStar. The [CG] option of the WRITE command is used to present database information in row/column format, exactly as it would appear in a columnar report. Data from each single record appear as a single row; data items from the records are displayed in columns, one data item per column, in correct consecutive order, from left to right. A double space separates columns from one another.

After the Condor WRITE command has created its disc file, WordStar can access the file through its Read a File function key. Note that no column headings are written to the output file by the WRITE command; if required, they may be edited into the document using WordStar.

Example:

Summary data from the ORDERS database is included in a Series 100/WordStar memo:

```
SORT ORDERS BY REGION
TABULATE ORDERS BY REGION AND COMPUTE TOTAL AMOUNT
AVERAGE AMOUNT [S]
```

(Orders are tabulated by region, with the results stored in the **RESULT** database. Each record in the **RESULT** database contains a region name, a total of orders, and an average of orders.)

```
WRITE RESULT REPORT [G]
```

(Transfer data to a file named REPORT)

Now, leave Condor, enter Wordstar, and compose the memo. When you reach the point when you wish to insert the table, press the Read a File function key on the word processor.

When prompted for the filename, enter REPORT

The table of data from the database will appear in the document at this point.

If you want a report with Condor report headings, subtotals, etc., use the SET PRINTER filename command described in Chapter 7 and read it into WordStar using the same Read a File function key (while in WordStar).

# Plotting Database Data

Using Series 100/Graphics, data from a Condor database can be plotted in a wide variety of pie, bar, and linear charts. The [G] option of the WRITE command is used to organize database data into the row/column format required by Series 100/Graphics. Pie, Bar and Line charts can all plot Condor data. Simply write one column for the labels and 1-5 columns for the data to be plotted. The labels may be up to 20 characters in length and data points may be up to 10 digits long.

Example:

A pie chart of orders by region is required, with data drawn from the ORDERS database:

```
SORT ORDERS BY REGION
TABULATE ORDERS BY REGION AND COMPUTE TOTAL AMOUNT [S]
```

(generate order statistics by region, and store them in the **RESULT** database)

```
WRITE RESULT DATAFILE.DAT [G]
```

(put statistics in the file DATAFILE.DAT)

Now, exit Condor and enter Series 100/Graphics Pie Charts

Press the Get a Chart function key.

Press the Transfer Data In function key; an asterisk (*) should appear in the function key label.

Enter the filename (DATAFILE.DAT in this example) and path name in the datafile information fields.

Press the Data function key to transfer the data into Series 100/Graphics.

Example:

Test results have been stored in a TESTDATA database, and are to be used to generate a Series 100/Graphics line chart:

```
SET PRINTER DATAFILE.DAT
SORT TESTDATA BY MONTH
PRINT TESTDATA BY MONTH, TEST1, TEST2, TEST3...[X]
```

(Select only month and data points as data items. Month must be designated by 1, 2, . . ., 12, . . .)

Now exit Condor and enter Series 100/Graphics Linear Charts.

Press the **Get a Chart** function key.

Press the **Transfer Data In** function key; an asterisk (*) should appear in the function key label.

Press the **Data** function key to transfer the data into Series 100/Graphics.

Bar Chart data (also row/column in organization) can be generated in the same way.

# Databases and User Application Programs

Data from Condor databases can be processed by user-written HP Series 100 programs, whether written in Basic, Cobol, Pascal, Fortran or other languages. Basic programs use the [B] option of the WRITE command to obtain database data in a usable format. Other languages typically prefer to process data in an "ASCII card image" format, which is selected with the [R] option.

Example:

Place all the data items from the CARDFILE database into a file for processing by a Basic program:

```
WRITE CARDFILE NEWFILE [B]
```

When data is stored using the [B] option, a Basic statement such as:

```
INPUT #1, A$, B$, C$, D$, E$, F$, G, H$, I$, J$
```

will read successive data items into the string and numeric variables shown. Condor does this by separating alphabetic data with quotes and commas and numeric data with commas (e.g., "Joe Smith","President","ACME Company","123 4thStreet","South Ridgeport","Washington",96045,...etc.). Note, option letters need to be capitalized (e.g., [B] not [b]).

Example:

Place all the data items from the CARDFILE database into a file for processing by a COBOL program:

```
WRITE CARDFILE NEWFILE [R]
```

Each record from the database becomes a fixed-length record in the resulting file. Data items occupy exactly as many positions in the record as their data entry fields on the database form, and appear consecutively in the record with no intervening spaces. The COBOL program would define an input **record** structure to read the data using Picture XXX... to read alphabetic/alphanumeric data and Picture 9999... to read numeric data. Similar record structures would be used by other languages.

# Exchanging Data with "Host" Computers

Data from Condor databases can be sent to a wide range of **host computer systems using DSN/Link and the [R] option** of the WRITE command. In addition, data from host computer files can be transferred to the Series 100 and read into Condor databases for storage and processing. DSN/Link also provides easy access to the Image/3000 database management system inquiry facility on the HP 3000 system, allowing data from Image/3000 databases to be entered into Condor databases for local processing.

To transfer data to a host computer, use the [R] option of the WRITE command. The data from the database will be stored in fixed-length records in the resulting file, one database record per file record. Data items appear consecutively in the record, in fixed-length fields like those in the database form, with no intervening spaces. Most programming languages on host computer systems find this data format the easiest to process.

Example:

Transfer detail data from the ORDERS database to the corporate HP 3000 system for processing:

```
WRITE ORDERS DATAFILE [R]
```
(the data is written to a file named DATAFILE)

Exit Condor, and enter DSN/Link.

Press the Transfer to Host function key.
Enter DATAFILE as the source filename, and use the destination filename of your choice.

Press the Begin Transfer function key.

The data will be transferred to the HP 3000 system.

File transfer from a host system is accomplished the same way, using the `READ` command:

Example:

Transfer ORDERS data from a host HP 3000 system into the Condor ORDERS database:

Using DSN/Link:

Press the `Transfer from Host` function key.

Enter the source file to be transferred, and enter `DATAFILE` as the name of the destination file.

Press the `Begin Transfer` function key.

When transfer is complete, exit DSN/Link and enter Condor.

Type `READ ORDERS DATAFILE`

To use Image/3000 data in Condor, you must first format the data with Query/3000. After Query/3000 has generated a report with fixed length records, use DSN/Link to transfer the Query/3000 report as in the previous example. Remember to create the Query/3000 report in a format usable by the Condor READ command.

The information on **READ/WRITE** file formats in the Command Reference Section of the Reference Manual is especially useful when exchanging information with an application program on a host computer. Depending upon the host system being used, one or another of the possible formats may make the host's processing burden easier or eliminate the need to rewrite an application program.

# Using Data from Series 100/WordStar Documents

Series 100/WordStar is frequently used to prepare reports, and the row/column data in a report may become a source of data for a Condor database. However, you will have to edit the report (using the word processor) to make its format agree exactly with the data item definitions in the database that is to receive the data. The example shows how to do this. To serve as a source of data for a database, a report must have the following two characteristics:

1. The rows of the report must represent the records of the database to be created; one row per record.
2. The columns of the report must represent the data items of each record, one data item per column, in the correct consecutive order, left to right.

The example shows how to transfer data into a database from a report that meets the criteria. **(Note: When creating the report in Series 100/Wordstar, it must be created as a NON-DOCUMENT file.)**

Example:

A printed report shows the following information:

```
REGION      TARGET     ACTUAL     DIFFERENCE

WEST         30000      35000           5000
SOUTH       240000     235000          -5000
EAST        150000     152000           2000
             --- etc ---
```

The data is to be entered into a database with the following data item definitions:

```
REGION:   AN, 5 characters
TGT-AMT:  N,  6 character data entry field
ACT-AMT:  N,  7 character data entry field
DIFF:     N,  6 character data entry field
```

The report meets the conditions necessary to use it as a source of data for this database, so you can proceed:

1. First, call up the file in WordStar using the Open Non Document function key. (You may want to first rename the file so that it is called **REPORT**.) Then use WordStar to strip away all information except the lines of the report that actually contain data. Specifically, delete report titles, column headings, etc.
2. Eliminate any total lines, or any blank lines in the report. After this step, there should be exactly as many lines in the report as there will be records in the database when you are finished.

3. Each column of the report must now be edited so that it occupies exactly as many characters as the corresponding data entry field on the database form. All intervening spaces between columns must be eliminated. For the example report, the REGIONS column must occupy exactly 6 spaces, followed immediately by the TARGET column at 7 spaces, etc. Note: data will probably be "smashed together" horizontally by this step. Although it appears strange, this format is required to read the data into Condor. The example report will look like this:

```
WEST  30000 35000  5000
SOUTH240000235000 -5000
EAST 150000152000  2000
```

4. Save the (edited) report by using the Save and Exit function key.
5. Enter Condor and type the command:

   READ ORDERS REPORT

   to load the data.

# Reading VisiCalc Data into a Database

Data from a VisiCalc spreadsheet can also be entered into a Condor database through the `READ` command. The data must be formatted in VisiCalc so that each row in the VisiCalc spreadsheet corresponds to one record in the database. Each data item must occupy one column. When you have organized the data in this way, use VisiCalc's **Print File** (`/PF`) command to print the spreadsheet into a row/column ASCII file. Print only the rows with actual data that are to become records of the database; do not include title rows, blank rows, etc.

The width of each VisiCalc column must correspond **exactly** to the size of the corresponding data item's data entry field in order for the VisiCalc print file to be entered directly into Condor. Set each column width in VisiCalc to be exactly the same as the field width of the corresponding data item in the database.

# Interfacing with Series 100/Personal Card File (PCF)

Condor data can be used by PCF. The [B] option of the WRITE command is used to represent Condor data which PCF can transfer in. PCF will insert Condor data items into its card files in left to right and top to bottom order.

PCF will left justify and pad with blanks any Condor data items which are shorter than the PCF field. If the PCF field is too short, PCF will throw away the right portion of the data item. If Condor writes more data items than will fit in a PCF card, PCF will only use the first data items.

If PCF has more fields than Condor data items, PCF will leave the remaining fields blank.

Example:

Transfer the Condor Customer data base into a PCF card file.

```
WRITE CUSTOMER DATAFILE [B]
```

Exit Condor and enter PCF. If a cardfile has not been created, then one must be created now using the `Create Card File` function key.

Press the `Cardfile Tasks` function key.

Press the `Transfer In` function key and type the filename (DATAFILE).

The data will be transferred into PCF.

PCF data may be entered into Condor by using the PCF Transfer Out function key and the Condor READ command. The fields of the Condor database and the PCF card file must be in the same order.

Example:

Transfer PCF data into the Condor CUSTOMER database.

From PCF:

Press the Card file Tasks function key.

Press the Copy Card file function key.

Press the Transfer Out function key and then type the filename PCFDATA.

Exit PCF and enter Condor.

```
READ CUSTOMER PCFFILE
```

The PCF data will be transferred into Condor.

# Chapter Summary

- Condor can exchange information with other HP Series 100 applications, greatly increasing the power and flexibility of database applications.
- The `WRITE` command takes information from a Condor database, and writes it in an ASCII file. It is used to pass Condor data to other HP Series 100 applications.
- Different file formats of the output ASCII file are available through options of the `WRITE` command.
- The `READ` command takes information from an ASCII file and enters it into a database. It is used to process with Condor, data generated by other applications. Exact formatting of the data must first be done in the source file.

# Chapter 9

# RESTRUCTURING AND REORGANIZING DATABASES


In the preceding chapters, a database has been treated as a single entity, which could be sorted, printed, copied, displayed, updated, etc. This is a very powerful feature of Condor, since you can use databases, even in very complex ways, without having to understand the underlying files and data structures.

However, Condor becomes an even **more** powerful tool if you have an understanding of how a Condor database is "put together". This knowledge is also required if a database is to be redefined or restructured after it contains data. This chapter explains the underlying structure of a Condor database, and how it can be used to effectively restructure and manipulate the database in advanced ways.

# Database Structure

A Condor database is a complex structure, made up of data items, records, a database form, etc. Condor uses the following files to store the component parts of a database:

| File | Purpose |
|---|---|
| Form File | Stores the database form. This file is edited by the `FORMAT` and `DEFINE` commands. The form file is named **xxxxxxxx.FRM**, where **xxxxxxxx** is the database name. |
| Definition File | Stores the data item definitions, including the data type, length, maximum, minimum, and default values for each data item. The definition file is named **xxxxxxxx.DEF**, where **xxxxxxxx** is the database name. |
| Data File | Stores the actual data of the database, along with the current record count. The data file is named **xxxxxxxx.DAT**, where **xxxxxxxx** is the database name. |

Working together, these three component files store the data of the database, define the data so you may access it by name, and determine how the database data is displayed for your review.

# The Result Database

The **RESULT** database was first discussed in Chapter 6, which described its role in relational database operations. Like all other databases, the **RESULT** database has three component files, but with the following unique names:

| | |
|---|---|
| Form File: | R$.FRM |
| Definition File: | R$.DEF |
| Data File: | R$.DAT |

The Condor commands that create a **RESULT** database use these filenames to generate the appropriate form, data item definitions, and data for the **RESULT** database.

# Other Database Files

In addition to the component files associated with each database, Condor also uses special files for other purposes. In each case, the filename extension shows at a glance the purpose of the file. The other files used by Condor are summarized in the table below:

| **Filename** | **Purpose** |
|---|---|
| xxxxxxxx.HLP | Stores a HELP screen. (See Chapter 10) |
| xxxxxxxx.CMD | Stores a Command Procedure. (See Chapter 10) |
| xxxxxxxx.DBM | Condor Command File—stores instructions needed to execute Condor commands. |
| DATA.DIC | Stores the Data Dictionary. |
| R$.DBM | Stores the work file used by the `RUN` command in processing command procedures. (See Chapter 10) |
| R$.ASC | Stores the ASCII file used by the `READ` and `WRITE` commands when no filename is given. (See Chapter 8) |

# Manipulating Database Files

Most of the **utility** commands available through Condor can be used with individual database files, as well as entire databases. To manipulate individual database files, simply specify the full filename (including the filename extension) in the command.

Examples:

```
COPY NEWDBASE.DAT = OLDDBASE.DAT
```
(copies data files only)

```
DESTROY NEWDBASE.DAT
```
(erases the data file, leaving the definition file and form files intact)

These commands may be useful for database backup, and they are required for some of the advanced database reorganization and restructuring described later in the chapter. The Command Descriptions in the Reference Manual lists, for each Condor command, whether it may be used with files as well as entire databases.

# Revising a Screen Form

When you create a new database, the `DEFINE` command steps you through creating a new database form and entering a new set of data item definitions. If you need to revise the form later, the `FORMAT` command allows you to edit the form file directly, without affecting the data item definitions or data in the database. You must be careful only to edit the form cosmetically. The lengths of the data entry fields, and the order in which the data items appear on the screen must not be changed after data has been entered into the database! If you change any of these characteristics of the form, the data stored in the database cannot be displayed correctly.

The FORMAT command may also be used to create a new form file, as an alternative to the DEFINE command. Additionally, it is used to create and edit **HELP** screens. The FORMAT command names the file to be created or edited:

```
FORMAT filename.FRM
or
FORMAT filename.HLP
```

The FORMAT command displays the form to be edited on the screen (or displays a blank screen, if the form is being created for the first time). The HP Series 100 editing keys or HP Touch may be used to move the cursor about the screen and edit the form at will:

Arrow keys or HP Touch — move the cursor about the screen

Delete char — deletes the character at the current cursor position

Delete line — deletes 80 characters from the form, beginning at the cursor position, and wrapping around to the next line.

Insert line — inserts 80 blanks into the form, beginning at the cursor position, and wrapping around to the next line.

The Insert char key toggles the FORMAT command back and forth between its two modes of operation, **replace** mode (the normal mode) and **insert** mode.

In **replace** mode, the characters you type replace (over type) the characters already on the screen. Data item names, underscores, and comments may all be entered on the screen in this manner. If you need to add a character to the screen and not replace other characters (e.g., insert a word between two words), the **insert** mode should be used.

Pressing the Insert char keys causes the system to enter **insert** mode. In this mode, each character you type is **inserted** into position on the screen, at the current cursor position. All subsequent characters on the line are shifted one character position to the right, to make room for the inserted character. The character in column 80 of the row is shifted off the screen and lost. Pressing Insert char again returns the system to replace mode.

Pressing the Grid function key will paint a grid on your screen, making it easier to line up items. The grid will not appear on your saved form. The Refresh Screen function key will erase the grid from the screen.

When you have finished editing a form, press the End function key. The form file will be replaced on the disc. To end editing without storing the revisions, press the Abort function key. The Refresh Screen function key will "repaint" the screen.

# Printing Data Item Definitions

You can print out the data item definitions of a database (from the definition file) with the DEFINE command. Simply enter:

```
DEFINE database-name
```

Since the database already exists, you are asked whether you want to print its description, redefine it, or end. Typing D (for describe) lists the data item definitions on the screen or the printer, at your option.

Example:

To list the CARDFILE data item definitions:

Type `DEFINE CARDFILE` Return

Condor responds:

```
This Database exists already.
Choose Option: Describe (D), Re-define (R), End <C/R>
```

Type `D`
(select the "Describe" option)

Condor responds:

```
Send output to printer (Y/N)?
```

Type `Y`

The data item definitions, record size, and record count are printed on the printer.

# Revising Data Item Definitions

If a database is empty (i.e., it contains no records), the `DEFINE` command can be used to revise its data item definitions. Follow the sequence:

Type `DEFINE database` Return

Condor will respond:

```
This Database exists already.
Choose Option: Describe (D), Re-define (R), End <C/R>
```

Type `R`
(to redefine the data items)

Each data item in the database will be displayed with its data item type, field size, minimum value, maximum value, and default value. To leave the data item definition unchanged, press Return and the next data item will be displayed.

To change the definition of the data item, use the left and right arrow keys to move the cursor and edit the definition. Simply type over the current definition and press Return when you have edited the definition to your satisfaction. If you change the type of a data item, it is a good idea to let the system recalculate the field size, minimum, and maximum by "wiping out" these definitions with the space bar before pressing Return.

When redefining data items, you can jump to a specific data item by typing a "#" symbol. The system will prompt with a ">", to which you should respond with the data item name to be changed. To end the revision without pressing Return for each of the remaining definitions, type a # and, when prompted, type another # followed by Return.

**CAUTION**

If you lengthen an item's definition field length, be sure to lengthen its field length in the .FRM by using the `FORMAT` command. Otherwise, when displaying or updating your database, you will receive the error message: `Insufficient space for data item value.`

# Changing Maximum, Minimum, and Default Values

You cannot directly change the data item definitions for a database that contains data. However, a special version of the `UPDATE` command allows you to change the maximum, minimum and default values for data items—even for a database containing data. This special `UPDATE` command is used to edit the definition file of the database directly:

```
UPDATE filename.DEF ST FIELD IS nn, ...
```

The filename specified in the command should be the definition file whose maximum, minimum, and default values are to be changed. Specify the field numbers for the data items to be changed in the command.

To change the default value for FIRM (the third data item) in the CARDFILE database, enter the command:

```
UPDATE CARDFILE.DEF ST FIELD IS 3
```

The form shown in Figure 9-1 is displayed, together with the data item type, length, maximum, minimum, and default values for the selected data item. The form operates just like any `UPDATE` command—you may revise the data, or leave it unchanged. Other fields may be selected by entering further `FIELD IS nn` selection criteria.

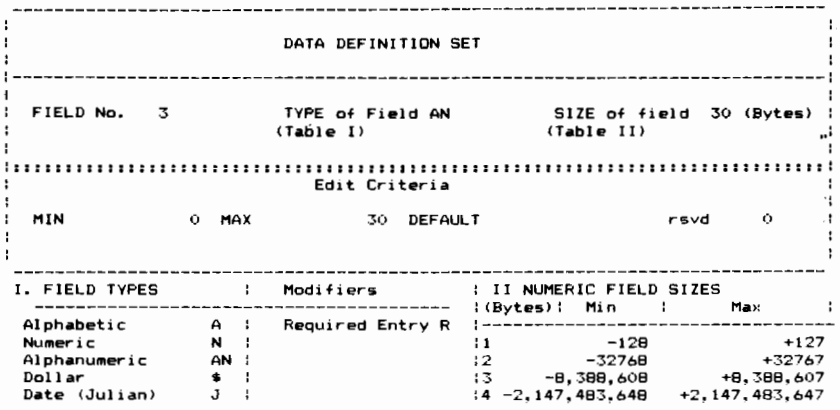


Figure 9-1. Updating Maximum, Minimum, and Default Values

This version of the UPDATE command allows you to edit the definition file as if it were itself a database. For a complete description of the function keys available in the UPDATE command, see Chapter 4.

# The Data Dictionary

How does Condor keep track of all the component files of a database, given the dozens of databases you will define and use? It uses a special file called the **Data Dictionary**. Condor automatically creates a Data Dictionary on each disc the first time you define a database on the disc. From that point on, each time you add a new database or destroy a database on the disc, the Data Dictionary is updated to reflect the change.

The Data Dictionary is organized so that it appears as a database to you. Each **record** of the Data Dictionary shows the structure of one database—its database name (title), and the names of its component form file, definition file, and data file. Figure 9-2 shows a Data Dictionary listing for some of the example databases in this manual.

```
Title                    Form      Definition Data

CUSTOMER                 CUSTOMER  CUSTOMER   CUSTOMER
PRODUCTS                 PRODUCTS  PRODUCTS   PRODUCTS
REGIONS                  REGIONS   REGIONS    REGIONS
CARDFILE                 CARDFILE  CARDFILE   CARDFILE
ORDERS                   ORDERS    ORDERS     ORDERS
RESULT                   R$        R$         R$
```

Figure 9-2. Sample Data Dictionary

When you use a database name in a command, Condor uses the Data Dictionary to locate the files needed to process your request. Each record in the Data Dictionary **defines** one database, by specifying its form file, its definition file, and its data file.

Because the Data Dictionary appears as a database, you can use Condor commands to process it. For example, the commands:

```
ENTER DATA.DIC and
UPDATE DATA.DIC
```

can be used to add new database definitions to the Data Dictionary, and to modify database definitions. Similarly, the commands:

```
LIST DATA.DIC BY TITLE, FORM, DEFINITION, DATA and
PRINT DATA.DIC
```

will print the Data Dictionary contents. Figure 9-3 shows the form that is used to enter or update information in the Data Dictionary directly. More examples of manipulating the Data Dictionary are given later in this chapter.

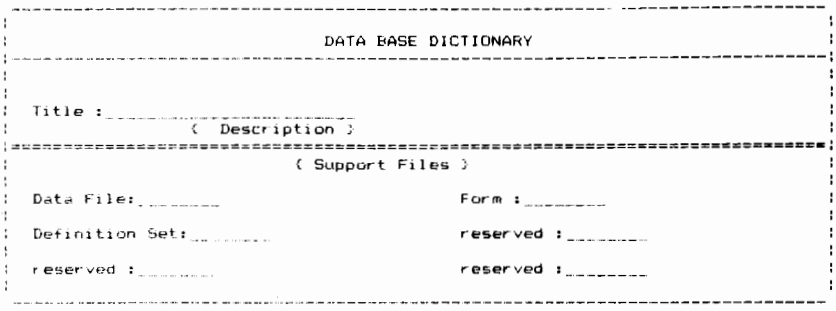


Figure 9-3. Data Dictionary Form

# Alternate Database Descriptions

Thus far, each database you have seen has had its own unique form file, definition file and data file, separate from all other databases. With the Data Dictionary, however, a single component file can be shared among several different databases. For example, by sharing a single form file, definition file, or data file among multiple databases, you can:

- look at several different sets of data through the same form
- look at one set of data through several different forms
- use two different sets of data item names to describe the same set of data

The Data Dictionary thus allows a database to have several different "alternate" definitions, to meet different needs. Many database applications can benefit from this capability, using it to describe a single database in two or more different ways. For example, you may want to use a form with elaborate data entry instructions to enter orders into a database, but use a much simpler form to display the orders for inquiry. Or, you may need to use two different sets of data item names to relate a database to other databases, if data item naming had not been planned carefully in advance. The two examples that follow illustrate how you may create alternate database descriptions through the data dictionary.

Example:

A separate ORDERS database is required for each of two subsidiaries. Both will contain the same data items, and both will use the same form for data entry. We can use the `DEFINE` command to create two databases that share common forms and data item definitions, but each contains its own data:

Define the first database normally:

```
DEFINE ORDERS Return
```

Type in the form, and the data item definitions.

For the second database, answer the questions in the `DEFINE` command dialogue as follows:

Type `DEFINE ORDERS2` Return

Condor responds:

```
Do you wish to create a new form (Y/N)?
```

Type `N`
(you want to use the form from the ORDERS database)

Condor responds:

```
ORDERS2 FRM does not exist –
Enter different name or End Program <C/R>:
```

Type `ORDERS` Return
(you tell Condor that the ORDERS2 will share the ORDERS.FRM form file)

Condor responds:

```
Do you wish to create a new definition file (Y/N)?
```

Type `N`
(you want to use the data item definitions from the ORDERS database)

Condor responds:

```
Enter Filename or End Program <C/R>:
```

Type `ORDERS` Return
(you tell Condor that the ORDERS2 database will share the ORDERS.DEF definition file)

Condor responds:

```
Do you wish to create an index for this Database
(Y/N)?
```

Type `N`

Condor responds:

```
Do you want a printed copy of the data definitions
(Y/N)?
```

Type `N`

Condor will display a summary of the database attributes, and a new data dictionary record is created for the database ORDERS2. You can see the new entry:

Type `LIST DATA.DIC BY TITLE, FORM, DEFINITION, DATA`
(ORDERS2 is listed, composed of the ORDERS.FRM, ORDERS.DEF, and ORDERS2.DAT files)

As this example shows, a new database that shares a form file or data definition file with other databases can be created by naming those files in the `DEFINE` command dialogue. The form and definition files must already exist when the `DEFINE` command is entered. A new database created in this way will always have its own, unique data file.

If two databases are to share the same data file, the Data Dictionary records must be edited directly to create the alternative database definition. The following example illustrates how this can be done.

Example:

A CUSTOMER database has been created with the following form:

```
*************************************************
*  [NAME] _____________________________          *
*  [COMPANY] _____________________________      *
*  [ADDRESS] _____________________________      *
*  [CITY] ____________________                  *
*  [STATE] __                                   *
*  [ZIP] _____                                  *
*************************************************
```

We wish to print mailing labels for the customers. One way to do this is to define a new database (named MAILLABL) that has a different form, but shares the same data and data item definitions as the CUSTOMER database. The MAILLABL form will position the data items correctly for printing on the mailing label. To define the alternative MAILLABL database description, do the following:

Use the `FORMAT` command to create a MAILLABL.FRM file with the proper form for the mailing label. The form will have underscores (for data fields), but no data item names, as shown below:

```
*************************************************
*                                               *
*    [ ] _____________________________          *
*    [ ] _____________________________          *
*    [ ] _____________________________          *
*    [ ] ____________________ [ ] __ [ ] _____  *
*                                               *
*************************************************
```

When the system asks if you wish to define a database using this form:

Type `N`

With the form defined, you can create the Data Dictionary entry for the new database:

Type `ENTER DATA.DIC`

The Data Dictionary form from Figure 9-3 will appear. Enter `MAILLABL` for the database title, and `MAILLABL` for the database form file name. Enter `CUSTOMER` for the definition file name, and `CUSTOMER` for the data file name.

Now, you can print the mailing labels:

```
SORT CUSTOMER BY ZIP
PRINT MAILLABL
```

The preceding example also illustrates how you could define an alternative database description with different data item names than the original description. This is a common requirement when two databases are first related using Condor's relational capabilities. Unless great care was taken when the databases were first defined, you may find that the data items you wish to use as matching data items (for the `JOIN`, `COMPARE` or `POST` commands) have different names in the two databases. An alternative database definition for one of the databases solves the problem easily, without reorganizing one of the databases unnecessarily.

You must use extreme caution when creating alternative database descriptions using the `ENTER` and `UPDATE` commands directly on the Data Dictionary. It is possible to create alternative descriptions where the database form, descriptions, and data make no sense together, resulting in nonsense output when you try to print or use the database. Be certain that the form, definition and data files that you combine describe a consistent, logical structure for the new database.

# Reorganizing a Database

Great care should be taken when initially defining a database, to insure that it contains the needed data items, that they are named correctly for relating them to other databases, and that they have the right data type for the processing required. Nonetheless, you may occasionally need to restructure a database that contains data. Using the `REORG` command, you can delete data items from a database, add new data items, or change the order in which the data items appear.

The idea behind a `REORG` is as follows:

1. The form file is to be altered so as to add, delete, or reorder data items.
2. The definition file will be changed to reflect any additional or deleted data items.
3. The data that currently exists is to be transferred into the newly reorganized database.

To reorganize a database, the `REORG` command is followed by the database to be altered:

```
REORG database
```

`REORG` will automatically bring up the form file for modification. At this time, new items or titles may be added, reordered, and/or deleted. All of the current data items are remembered by Condor, so if an item name is changed, it is assumed that the item has been deleted and a new one added.

Care should be taken when altering underscores. If there are less underscores than the item definition specifies, an error message will be issued when you try to display data: `Insufficient space for data item value.`

After the form file has been altered, press the End function key. REORG begins modification of the database definition file. If an item is added, Condor prompts you for the specifications of that item. Enter definition values in the same manner as when a database is defined. If an item is deleted, Condor will denote that the item is to be deleted, then ask for a confirmation. **If you answer Y, all of the data for that item will be deleted from the database.** If you decide that you don't want to delete that item, answer **N**, and REORG will be aborted. You may then use the REORG command again and add the item name back onto the form.

Example:

We wish to delete the HOME-PHONE data item from the CARDFILE database, and add a new data item for EXTENSION-NBR. This is accomplished as follows:

```
REORG CARDFILE
```

The form file is displayed on the screen, just as is seen using the FORMAT command. Using the cursor and screen control keys, delete the HOME-PHONE item and replace it with the EXTENSION-NBR item. When finished, press the End function key. The system will respond by asking you to confirm that the HOME-PHONE item is to be deleted.

Type Y

Now, Condor prompts you for the specification of only the new item that has been added, EXTENSION-NBR. Enter specifications the same way as is done using the DEFINE command. The system will ask Definitions OK (Y/N)?

Type Y

After listing the attribute summary of the new definition to the screen, Condor will ask if it is okay to replace the current database with the REORG result.

Type Y

# Reorganization Using Read/Write

When changing an item's field definition length or a field name, the reorganization **must** be accomplished by copying the database contents out of the database (into an ASCII file), and then copying the contents back into another (restructured) database. The `READ/WRITE` command pair is used to copy the database contents out of, and back into a database. Extreme caution must be exercised when using this procedure. Condor cannot check the `WRITE/READ` operation to be sure the "new" database definition is compatible with the "old" definition; and if the two definitions are not compatible, data will appear "garbled" when displayed. Be careful.

The procedure to perform this type of database restructuring is:

1. Be sure you have a backup of the database to be restructured. If not, make a duplicate, using the Condor `COPY` command.
2. Write out the database contents to a disc file using the `WRITE` command.
3. Use the `EMPTY` command to delete all the data from the database; now use the `DEFINE` command to redefine the data item definitions in the empty database (the `DEFINE` command will not work if the database is not empty).
4. Read the database contents back into the redefined database using the `READ` command.
5. Destroy the temporary files used by the `WRITE` and `READ` commands.

Example:

We wish to increase the length of the NAME field from 20 to 30 characters. The following sequence of commands will restructure the database:

```
COPY TEMP = CARDFILE
```
(we will work with a duplicate copy, just in case!)

```
WRITE TEMP TEMPFILE [B]
```
(write the database to an external file called TEMPFILE.)

```
EMPTY TEMP
DEFINE TEMP
```
Condor responds that the database already exists. Indicate that you want to redefine the database by typing an R. Then change the length and maximum value of the NAME field to 30 as follows:

1. NAME: ANR, 30, 0, 30

Note: You can always increase the field size, expand the range of your minimum and maximum values, and assign new default values. If you want to change the field type or reduce the field size, be sure the records in your current dataset conform to the new definitions. If they do not, you must change the records or else the READ procedure will not work as you expect.

Also, if you want to change the field size, check to see whether your maximum value is the same as the field size. If the two values are the same and you change the field size, you should change the maximum value too. The field size for numeric (N) or dollar ($) fields should never exceed 4 bytes; the field size for date (J) fields should always be 3 bytes.

```
FORMAT TEMP.FRM
```
Add 10 underscores to the NAME field to increase the length to 30.

```
READ TEMP TEMPFILE
```
(read the external back into the database.)

```
LIST TEMP
```
(check to see that everything worked.)

```
COPY CARDFILE = TEMP
DESTROY TEMP
DESTROY TEMPFILE
```

# Chapter Summary

- A database is made up of three files:
  - a FORM file
  - a DEFINITION file
  - a DATA file
- Many Condor commands can be used to process individual database files as well as entire databases.
- The `FORMAT` command is used to edit or revise database forms or `HELP` screens.
- The `DEFINE` command can be used to list and revise data item definitions.
- The **Data Dictionary** defines the databases on a disc; it keeps track of their form, data, and definition files.
- The `ENTER`, `UPDATE`, `DELETE`, `LIST`, `PRINT`, and other commands can be used with the Data Dictionary (DATA.DIC) to allow databases to share common files.
- The `DEFINE` command can be used to reorganize an empty database.
- Reorganization of a database containing data can be accomplished with the `REORG` command.
- The `READ/WRITE` commands can be used for other kinds of reorganization when a database contains data.

# Chapter 10

# DATABASE COMMAND PROCEDURES


One of the most powerful features of Condor is the ability to manipulate information on demand. Using the commands described in the preceding chapters, databases can be queried, combined, interrelated, and manipulated to produce information as it is needed.

Often, however, the same procedure for processing a database is repeated over and over again. For example, the month-end reporting procedure that generates reports from an ORDERS database may consist of twenty or thirty commands, which must be entered every month. Or the daily posting procedure for the database may require a long sequence of commands that is repeated every day.

Condor allows very efficient processing of repetitive database procedures through two powerful capabilities—**Command Procedures** and **Help Screens**. Using these facilities, repetitive database operations can be defined once and then performed over and over again with the press of a few keys.

Command Procedures and Help Screens can also be used to make Condor the foundation of an entire transaction processing application, complete with data entry and inquiry functions, reports, and daily, weekly, and monthly procedures. Application users (who are typically clerical personnel unfamiliar with the structure of the database) can select the procedures they need through simple menus and access database data through familiar business forms.

# Command Procedures

A Command Procedure is a sequence of one or more Condor commands stored in a **command procedure** file on a disc. Once the command procedure has been defined, it may be retrieved and processed at a later time, using the RUN command. In this way, long and complex command sequences can be defined once, and then executed whenever they are needed, with a single command.

Command Procedures are processed by the Condor RUN command. The RUN command is similar to the MSDOS batch processing capability. It allows you to tailor the execution of a command procedure through symbolic parameters, and a powerful set of command procedure directives. Among its capabilities are:

- Conditional processing of a group of commands, through an *IF...*ENDIF structure.
- Assignment of parameter values through the RUN command, through user prompts, or through assignment directives within the command procedure.
- Display of informational messages for the user.
- Dynamic suppression/enabling of screen echo of command processing.
- Orderly end, abort, and restart of command procedures.

Command procedures are started by typing the RUN command:

```
RUN procedure param1 param2 ... param9
```

When the RUN command is entered, the procedure must have been previously defined. The RUN command can process up to 9 parameters (variables). These parameters may optionally be assigned values by listing the values in the command line, following the procedure name.

Example:

```
RUN ENDMONTH FEBRUARY 3/3/82
```
(run end-of-month procedure; parameter #1 = "FEBRUARY", parameter #2 = "3/3/82")

Command procedure files are stored with a file extension of ".CMD". For example, the procedure "MONTHEND" is stored in the command procedure file "MONTHEND.CMD". The extension is not specified in the RUN command.

# Command Procedure Processing

It is very important that the designer of a Condor command procedure understand **exactly** how the RUN command will go about processing that procedure. The RUN command, and its associated directives, actually form a small macro processor capability within Condor. Command procedure processing by the RUN command takes place in two steps.

First, the command procedure is scanned, and a work file of commands to be executed is built. (The work file is named R$.DBM, and is created on the current disc drive.) Several functions are performed during the scan:

1. Parameters ($1, $2, ... $9) are replaced with the value given in the RUN command line, assigned with the *LET or entered through the *GET directive. $TODAY is replaced with Condor's current date.
2. When a *IF directive is encountered, the specified condition is tested to determine if the command lines between the *IF and the *ENDIF will be written to the work file. Command directives appearing between the *IF and *ENDIF directives are processed only if the specified condition is satisfied.
3. When a *MESSAGE directive is encountered, the message is displayed on the screen.
4. When a *GET directive is encountered, a colon (:) prompt is displayed on the screen. Processing is suspended until the requested parameter values are entered. The value entered is used for all subsequent appearances of a parameter until a subsequent *GET directive is processed for the same parameter.
5. When a *LET directive is encountered, the parameter on the left side of the equals sign (=) is assigned the value appearing on the right side of the equals sign. If a parameter appears on the right side, its value is assigned. The *LET directive is particularly useful for setting parameter default values.

6. When an *END directive is encountered, scanning terminates. No further lines in the command procedure file are scanned. Commands that have been written to the work file prior to encountering the *END are now executed in sequence.
7. When an *ABORT directive is encountered, scanning terminates. No commands are written to the work file. No commands will be executed.
8. The *ECHO directive has the effect of writing a SET ECHO command as the first line of the work file. If several *ECHO directives are processed, the ECHO status during execution of the commands in the work file will be that of the last *ECHO processed during scanning.
9. Comments (lines beginning with a semicolon) are not written to the work file.
10. Directives are not written to the work file. Therefore, no conditional testing occurs during execution. Conditional testing applies only during the scanning operation.

After the scan is complete, the second step is to execute the command procedure. The commands in the work file are read one by one, and executed. When the last command in the work file has been executed, the command procedure is complete.

A command procedure may exit Condor and return to the operating system by using a SYSTEM command. The next commands that follow need to be operating system level commands. With this feature you may call other applications.

# Comments in Command Procedures

Command procedures may include comment lines, which do not affect procedure processing. Comments improve command procedure readability, and may be used to make procedures self-documenting to aid later maintenance. Any command procedure line which begins with a semicolon (;) is treated as a comment by the RUN command.

Example:

```
; Command Procedure to print Month-End Reports
; These are comment lines
;
; First, print orders by region report
SORT ORDERS BY REGION
TITLE ''REGIONAL ORDER REPORT''
TABULATE ORDERS BY REGION, AMOUNT, TYPE
   SUBTOTAL USING REGION
;
; Follow with orders by order type report
SORT ORDERS BY TYPE
TITLE ''MONTHLY ORDERS BY TYPE''
PRINT ORDERS BY TYPE, ORDER-NO, AMOUNT
   AND COMPUTE TOTAL AMOUNT SUBTOTAL USING TYPE
*END
```

# Echoing Command Procedure Execution

Normally, when Condor executes a command procedure, the individual commands and their results are echoed to the Series 100 screen during execution. This allows tracing of the command procedure execution as it occurs. If no echo is desired, include the directive:

```
   *ECHO OFF
```

in the command procedure. This has the effect of issuing a SET ECHO OFF command before execution of the command procedure begins. Echo is always turned back on again automatically at the end of the command procedure execution, or if an error or abort condition occurs, or if a HELP command is executed.

The opposite effect may be obtained by including the directive:

```
*ECHO ON
```

in the command procedure. If multiple ECHO directives are processed during scanning of a command procedure, the last one processed before actual procedure execution commences is the directive that will be used.

# Messages in Command Procedures

Messages may be displayed for the user during processing of a command procedure through the *MESSAGE directive:

```
*MESSAGE message-text
```

The message is displayed on the Series 100 screen when the directive is encountered during initial scan of the command procedure. No resulting command is written into the work file for later execution. The *MESSAGE directive is especially useful in combination with the *GET directive, prompting the user with an easy-to-understand message for the parameters which must be supplied.

# Command Procedure Parameters

A command procedure may have up to nine parameters (variables), which can be used to make its use more flexible. Parameters may be used, for example, to pass a report date to a procedure, or to pass a yes/no response to a question, or to select which of several report options is to be printed.

The parameters are indicated within the procedure by the symbols $1, $2, ... $9. They are treated as variable-length character strings. During command procedure processing, as each line is scanned, the current value of all parameters used in that line are substituted for the parameters, in the exact position where they appear. The parameter-substituted command line is then copied to the work file for later execution. The parameters thus serve exactly the same role as macro parameters in the classic definition of a macro processor.

In addition to the nine user-defined parameters, Condor provides a special parameter named $TODAY. $TODAY always has the value of current date (as entered by the user through the DATE command). $TODAY may be freely used in command procedures, for comparison with other dates, etc.

---

**CAUTION**

In a command procedure, Condor treats any word beginning with a dollar sign ($) as a parameter, unless the dollar sign is immediately followed by another dollar sign, in which case it is treated as a literal character. A word such as $AMOUNT must therefore be specified as $$AMOUNT when referenced in a command procedure.

---

# Conditional Processing

Commands in a command procedure can be selectively included or excluded from execution through the *IF and *ENDIF directives. The syntax of these directives is:

```
*IF condition
...other commands & directives go here
*ENDIF
```

The condition may be any logical condition, with the same rules applying as those used for the condition in the SELECT command. *IF...*ENDIF directives can be nested within a command procedure, up to seven levels deep.

When an *IF directive is encountered during procedure scanning, the condition is tested. If true, scanning continues with the command immediately following the *IF directive, and the *ENDIF is ignored when encountered. If the condition is false, then the commands and directives that follow the *IF are skipped over, until the matching *ENDIF is found. Scanning then continues normally with the command following *ENDIF.

It is important to remember that conditional testing takes place at scanning time, and not at actual procedure execution.

# Ending or Aborting a Procedure

The *END directive is used to halt scanning of a command procedure, and cause actual execution of the procedure (from the work file) to begin. Multiple *END directives may, of course, be used within a single command procedure—the first *END encountered during scanning will halt the scan. An *END directive is not required at the physical end of the command procedure file; one is assumed.

If an abnormal condition is encountered during procedure scanning, the *ABORT directive may be used to halt scanning immediately and prevent execution from taking place. When the *ABORT directive is encountered during scanning, no commands are written to the work file, and none are executed. Processing of the RUN command immediately halts.

# Changing Parameter Values

Command procedure parameters get their initial values from the RUN command. If parameter values are specified in the RUN command, they are assigned to the parameters before procedure scanning begins.

The values of parameters can also be altered during the scanning of the command procedure. The *LET directive serves as an **assignment statement** to assign a parameter a value within the procedure:

```
*LET parameter = value
```

Example:

```
*LET $4 = $5
*LET $4 = ''Boston''
```

Another way of altering parameter values is to request a value for the parameter from the user. This is done with the *GET directive:

```
*GET parameter
```

Example:

```
*GET $4
*GET $3, $5
```

When the *GET directive is encountered during procedure scanning, the user is prompted with a colon (:) on the Series 100 screen, requesting that he supply a value for the parameter. Multiple parameter values may be assigned with a single *GET directive by listing the parameters in sequence.

# Command Procedure Example

The sample command procedure that follows prints a report of sales within a specified range of dates, with optional totals and subtotals on a user-selected data item. It illustrates use of all the command procedure directives discussed in this chaper. Use the EDIT command to create the following procedures.

```
 1.;THIS PROCEDURE PRINTS A SALES REPORT
 2.;Syntax: RUN SALEREPT start-date end-date
 3.;   $1 IS THE BEGINNING DATE OF THE REPORT PERIOD
 4.;   $2 IS THE ENDING DATE OF THE REPORT
 5.*LET $4 = N
 6.*MESSAGE The report will show sales from $1 to $2
 7.*MESSAGE Are these the dates you want to use? (Y/N)
 8.*GET $3
 9.*IF $3 NE Y
10.    *ABORT
11.*ENDIF
12.SELECT SALEDATA ST DATE GE $1 AND DATE LE $2
13.*MESSAGE Do you want Totals? (Y/N)
14.*GET $3
15.*IF $3 = Y
16.    *MESSAGE Do you want Subtotals? (Y/N)
17.    *GET $4
18.    *IF $4 = Y
19.        *MESSAGE Enter Subtotal Data Item
20.        *GET $5
21.        SORT RESULT BY $5
22.    *ENDIF
23.*ENDIF
24.*IF $3 = Y AND $4 = Y
25.    PRINT RESULT BY REGION @ TOT SALES SUBTOT BY $5
26.    *END
27.*ENDIF
28.*IF $3 = Y
29.    PRINT RESULT BY REGION @ TOTAL SALES
30.    *END
31.*ENDIF
32.PRINT RESULT BY REGION SALES
```

Explanation of Example:

Lines 1-4 are comments; they are not written to the work file.

Line 5 sets the value of parameter $4 to N. If $4 is not given a value, the *IF on line 24 will cause an error. The user may change $4 on line 17.

Lines 6 and 7 display a message with the two dates entered in the RUN command.

Line 8 requests confirmation of these dates.

Lines 9-11 test the user response, and abort the procedure if the response is not ''Y''.

Line 12 writes the SELECT command to the work file.

Lines 13 and 14 determine if totals are desired in the report. Note that parameter $3 is reused.

If totals are not desired, the scan skips to line 32, because of the *IF directives on lines 15, 24, and 28. Two commands will be written to the work file and executed:

```
SELECT SALEDATA ST DATE GE 1/1/81 AND DATE LE 6/1/81
PRINT RESULT BY REGION SALES
```

If totals are desired, lines 16-17 ask if subtotals are desired.

If subtotals are requested, the *IF directive on line 18 is satisfied, and the Subtotal data items are requested (lines 19 and 20).

Once the subtotal items are entered, the data item names ($5) are substituted in the SORT command on line 21, and it is written to the work file.

If both totals and subtotals have been requested, (*IF directive on line 24), the PRINT statement on line 25 is written to the work file, with the Subtotal data items ($5) substituted. No further scanning will occur, because of the *END directive on line 26.

If only totals are requested, the *IF directive on line 24 causes scanning to skip to line 28. The *IF directive on line 28 is satisfied, and the PRINT statement on line 29 is written to the work file. The *END directive on line 30 halts scanning.

Note that the line numbers are used here only for reference; they must not be entered into the command procedure file.

# Restarting a Command Procedure

If no error conditions are encountered during execution of a command procedure, all commands in the work file will be executed, and control is returned to Condor. If one of the commands in the work file completes abnormally (aborts), however, execution will stop immediately, and the work file will contain all commands following the one that was aborted. Possible reasons for abnormal termination include:

1. The operator enters a control-C.
2. A command terminates abnormally (e.g., the user enters N to a confirmation to erase a database.
3. A power failure or hardware reset occurs.
4. An invalid command line is encountered.

An aborted command procedure can be restarted. However, before restarting, the command procedure should be reviewed to identify the command that was aborted. The user must then determine whether to repeat the command that was aborted. For example, repeating an APPEND, POST, or READ command may cause undesired effects.

Once the status is determined, execution of commands from the work file may be continued using the RESTART command:

```
RESTART
```

If the work file is on a drive other than the current drive, the command must be entered as RESTART drv: (e.g., RESTART B:). The current drive will not be the same as when the command procedure aborted if the current drive was reset by the command procedure.

# Creating and Editing Command Procedures

Condor command procedures are created and edited using the `EDIT` command. This command provides a powerful, pushbutton-oriented **on-screen editor** for the HP Series 100.

The syntax of the `EDIT` command is:

```
EDIT procedurename
```

After you enter the `EDIT` command, the screen is blanked, and a ">" prompt is displayed, indicating that the `EDIT` command is ready to accept the first line of your command procedure file. If you are editing a command procedure file that already exists, the first several lines of the file will appear on the screen, again ready for editing. The lines of the command procedure are displayed on every other line of the display to make inserting lines easier.

The HP Series 100 editing keys can be freely used to edit the command procedure. New command procedure lines are entered line by line. Press the [Return] key to move to the next line. You may use the arrow keys to return to a previously-entered line and correct mistakes simply by "typing over" the text in error on the screen. In addition, the editing keys provide useful editing functions:

[Delete char] — deletes one character, at the cursor position.

[Delete line] — deletes 80 characters, starting at the cursor position, and wrapping around to the next line.

[Insert line] — inserts 80 blank characters, starting at the cursor position and wrapping around to the next line.

The [Insert char] key toggles the `EDIT` command back and forth between **replace** mode (the normal mode) and **insert** mode. These modes operate in the same way for the `EDIT` command as described for the `FORMAT` command. See Chapter 9 for details.

In addition to the editing keys, the `EDIT` command displays the following function keys to aid in command procedure editing:

Press the **Next Page** function key to bring the next "screenful" of the command procedure file onto the screen for editing.

Press the Previous Page function key to bring the previous "screenful" of the command procedure file onto the screen for editing.

Press the Top Page function key to bring the first "screenful" of the command procedure file onto the screen.

Press the Bottom Page function key to bring the last "screenful" of the command procedure file onto the screen.

Press the Refresh Screen function key to cause Condor to re-paint the screen, if it becomes garbled.

Press the Print function key to print the edited command procedure file on the printer.

Press the Abort function key to stop editing the command procedure and leave the command procedure file on disc unchanged.

Press the End function key to stop editing the command procedure and replace the command procedure file on disc with the edited version.

In addition to Condor's built-in editor, you may use Series 100/Wordstar (in NON-DOCUMENT mode) to edit command procedures.

# Help Screens

Condor HELP screens simplify computer operations by giving the user a menu to use to select among processing options. The HELP screen lists, by number, the choices available to the user; each choice corresponds to a Condor command. Pressing the appropriate number on the keyboard automatically executes the associated command (often a RUN command), without any entry of commands by the user. Condor application designers will find HELP screens to be useful "glue" that holds their individual application modules together.

Figure 10-1 shows how HELP screens can simplify processing in a routine daily application such as Accounting.

```
*****************************************************************
*         GENERAL LEDGER ACCOUNTING SYSTEM                      *
*****************************************************************
*                                                               *
*              SELECT ONE OF THE FOLLOWING OPTIONS              *
*                                                               *
*     1.  INPUT CHART OF ACCOUNTS                               *
*                                                               *
*     2.  UPDATE GENERAL LEDGER                                 *
*                                                               *
*     3.  ENTER JOURNAL ENTRIES                                 *
*                                                               *
*     4.  JOURNAL TRIAL BALANCE                                 *
*                                                               *
*     5.  POST GENERAL LEDGER                                   *
*****************************************************************
```

Figure 10-1. A HELP Screen

The HELP screen is requested by the user with the `HELP` command:

```
HELP helpfile
```

When the HELP menu has been displayed, an option is selected by typing a number followed by a carriage return. Condor executes the command (or command procedure) that is associated with the number typed.

The command associated with each choice on a HELP screen is defined when the HELP screen is created by the application designer. The associated command is placed in square brackets ([ ]) following the command number on the screen. Figure 10-2 shows the HELP screen from figure 10-1, as defined by the application designer. The bracketed commands are automatically removed when the HELP screen is displayed.

The commands enclosed in brackets on a HELP screen may be any Condor command except DIC, DIR, LOG, RESTART, SET, and SYSTEM. These "built-in" commands are an integral part of Condor, and may be selected through a HELP screen by causing the appropriate selection to RUN a command procedure that includes them.

```
*****************************************************************
*        GENERAL LEDGER ACCOUNTING SYSTEM                       *
*****************************************************************
*                                                               *
*              SELECT ONE OF THE FOLLOWING OPTIONS              *
*                                                               *
*     1.  INPUT CHART OF ACCOUNTS  [RUN B:ACCNTS]               *
*                                                               *
*     2.  UPDATE GENERAL LEDGER  [RUN B:UPGLEDG]                *
*                                                               *
*     3.  ENTER JOURNAL ENTRIES  [RUN B:JOURNIN]                *
*                                                               *
*     4.  JOURNAL TRIAL BALANCE  [RUN B:TRIALB]                 *
*                                                               *
*     5.  POST GENERAL LEDGER  [RUN B:POSTGL]                   *
*****************************************************************
```

Figure 10-2. A HELP Screen

HELP screens are created using the FORMAT command. Enter the command:

```
FORMAT filename.HLP
```

and use the HP Series 100 editing keys to "paint" the HELP form on the screen. You may also use Series 100/WordStar (in NON-DOCUMENT mode) to edit HELP screens. See Chapter 9 for a description of the FORMAT command and its editing capabilities.

# Aborting a HELP Screen

A HELP screen may include an option to abort the HELP screen and return to the Condor command prompt by including the ABORT command as one of the choices in the screen. The ABORT command serves only in this special function, and consists of the single word:

```
ABORT
```

# Chapter Summary

- Condor command procedures and HELP screens are used to create applications systems, or to ease repetitive database chores.
- The RUN command processes a command procedure, a pre-stored sequence of Condor commands and special directives.
- RUN command directives allow for parameter-driven processing, conditional processing, and user interaction during command procedures.
- HELP screens provide a pushbutton-oriented menu choice for the user; each choice executes a corresponding command on the user's behalf.

# Chapter 11

# TRANSACTION PROCESSING


Chapter 10 showed how HELP screens and command procedures can be used to streamline repetitive Condor procedures. In the hands of a data processing professional, these features also make Condor an attractive foundation for a complete transaction processing application. This chapter describes the use of Condor for transaction processing, using a basic General Ledger Accounting System as a comprehensive example. Design considerations for transaction processing with Condor are presented through the example. In addition, the application provides an excellent, integrated summary of the Condor capabilities discussed in earlier chapters.

# General Ledger Application Example

The example used throughout this chapter is a simple General Ledger Accounting System. The application uses three databases, with the following names and functions:

**GLEDGER** — the General Ledger Database; stores data on each account in the General Ledger (account#, name, balance, etc.).

**JOURNAL** — the General Journal Database; stores journal entries as they are entered, until they are eventually posted to the General Ledger accounts.

**JOURAUDT** — The Journal Audit Trail Database; stores all journal entries after they are posted, providing an audit trail of all general ledger transactions.

The application accumulates journal entries as they are made, verifies their correctness, and ultimately posts them to the appropriate general ledger accounts. Various reports are also generated.

# Defining the Example Databases

The first steps in designing the example application is to define the databases. Figure 11-1 shows the **GLEDGER** database form. The `DEFINE` command is used to "paint" the form on the HP Series 100 screen, and define the database to Condor.

```
*********************************************************************
*              GENERAL LEDGER ACCOUNTING SYSTEM                     *
*********************************************************************
*                                                                   *
*           GENERAL LEDGER RECORD                                   *
*                                                                   *
*      [ACCOUNT]:_________      [ASSET]:__     [LIABILITY]:__       *
*                                                                   *
*      [ACCNAME]:_____________________________                      *
*                                                                   *
*      [DEBIT]:_____________      [CREDIT]:_____________            *
*                                                                   *
*      [YTDAMT]:_____________ (Year - To - Date Amount)             *
*                                                                   *
*      [MONTH]:_____________ (Last Accounting Month                 *
*                             Posted to General Ledger)             *
*********************************************************************
```

Figure 11-1. GLEDGER Database Form

The **JOURNAL** database is also defined to Condor using the `DEFINE` command. Its database form is shown in Figure 11-2.

```
*****************************************************************
*            GENERAL LEDGER ACCOUNTING SYSTEM                   *
*****************************************************************
*                                                               *
*          JOURNAL RECORD                                       *
*                                                               *
*     [ACCOUNT]:_________                                       *
*                                                               *
*     [DESC]:_____________________________                      *
*                                                               *
*     [DEBIT]:_____________     [CREDIT]:_____________          *
*                                                               *
*     [MONTH]:____________________ (Accounting Month)           *
*                                                               *
*     [DATE]:_____________________ (Date of Journal Entry)      *
*****************************************************************
```

Figure 11-2. JOURNAL Database Form

The **JOURAUDT** database, consisting of journal records which have already been posted, has the same data items as the **JOURNAL** database. The **JOURAUDT** database can therefore use exactly the same form and the same data item definitions as the **JOURNAL** database. This is also specified through the `DEFINE` command:

Type `DEFINE JOURAUDT` Return
Condor will indicate that no form file is found, and ask if you want to create a new one. Responding `N` signals that an existing form file will be used; Condor asks for its name:

Type `JOURNAL` Return
Condor next asks if a new data item definitions file is to be created. Responding `N` signals that an existing definition file will be used; Condor asks for its name:

Type `JOURNAL` Return
Condor creates the JOURAUDT database, with its own unique data file (JOURAUDT.DAT), but with a shared form file (JOURNAL.FRM) and data item definition file (JOURNAL.DEF). The Data Dictionary entry reflects these file names.

The above steps may also be accomplished with the one line command:

```
DEFINE JOURAUDT JOURNAL.FRM, JOURNAL.DEF Return
```

This completes the definition of the example databases.

# Preparation For Entering Data Into the Example Databases

In any transaction processing application, there are typically two types of data entry operations; entry of new records in a **master** database, and entry of new **transaction** records. In addition, data in a **master** database is generally updated from one or more **transaction** databases through a **transaction posting** operation. These three operations are described below, and illustrated with the General Ledger example in the sections that follow.

**Entering Unique Records** (Generally **master** records)

When entering new records to an existing database, duplicate records may not be allowed. For example, in general ledger account records, two records having identical account numbers are not allowed.

**Entering Matching Records** (Generally **transaction** records)

When entering records to a **transaction** database, there must be a **matching record** in the **master** database. For example, a payment from a client would be entered in an accounts receivable database with an account number, which must match the account number of a record in the general ledger database.

**Posting** (Generally transactions are posted to master records)

New records that have been entered into a **transaction** database are posted to records in a master database. The database posted to (master database) has unique records, and the database with posting data (transaction database) may have multiple records, though it is restricted to matching records in the master database. One example is a purchase order receipt transaction that updates the purchase order database.

# Entering Unique Records

Inputting the General Ledger Chart of Accounts illustrates the entry of unique records.

In the general ledger **master** database, GLEDGER, there is to be one record for each general ledger account number. If the same account number is entered into the database twice, it is an error. To eliminate this error, a data entry procedure is followed which will ensure that only unique records are added to the database.

To ensure unique records, new records are never entered directly into a master database. A holding database is defined, into which new records are entered. Upon completion of entry into the holding database, the records in the holding database are compared to the records in the master database to find duplicates. Only the unique records are appended to the master database. The holding database need only be defined once. After each entry of data, the holding database is emptied.

Example:

The procedure outlined below enters unique records into the GLEDGER database:

1. `DEFINE GLTEMP`

   GLTEMP will be the holding database. The system will ask if a new form is to be created. Answer `N`, and tell Condor that GLTEMP uses the same form as the GLEDGER database, `GLEDGER.FRM`.

   The system will ask if a new definition file is to be created. Answer `N`, and tell Condor that GLTEMP will use the same data item definitions as the GLEDGER database, from `GLEDGER.DEF`.

2. `ENTER GLTEMP`

   New account numbers and other information are entered through the GLEDGER form, and accumulated in the GLTEMP holding database. All new accounts are added in this way; existing accounts may be updated directly with the `UPDATE GLEDGER WHERE ACCOUNT IS xxxxxxx` command. Note: For proper accounting practice, always enter Credits, Debits, and Totals in the Journal database, never in the General Ledger database. An accountant can give you additional information on accounting practices.

3. `COMPARE GLTEMP GLEDGER NOT MATCHING ACCOUNT`

   The holding database is compared to the master database to find account numbers that do not match. These records are written to the **RESULT** database, and are valid accounts to be added to the master database. The **RESULT** database has the same form and data item definitions as the GLEDGER database.

4. ```
   TITLE T,'GENERAL LEDGER REPORTS',S,S,
     'NEW ACCOUNT NUMBER ADDITIONS FOR ',DATE,S,S
   PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT
   ```

   A report of the new accounts to be added is printed. The headings and date will be printed at the top of each page of the report. The heading is printed on two lines, and the `S`'s will cause lines to be skipped.

5. `APPEND GLEDGER RESULT`

   New and unique general ledger account numbers are appended to the master (GLEDGER) database. The `APPEND` command appends the `R#.DAT` file to the `GLEDGER.DAT` file.

6. `SORT GLEDGER BY ACCOUNT`

   The GLEDGER database is sorted in account number sequence.

7. ```
   TITLE T,'GENERAL LEDGER REPORTS',S,S,
     'GENERAL LEDGER CHART OF ACCOUNTS DATE ',DATE,S,S
   PRINT GLEDGER BY ACCOUNT ACCNAME DEBIT CREDIT
   ```

   A revised chart of accounts report is printed in account number sequence. The headings and date are printed at the top of each page.

8. `EMPTY GLTEMP`

   The new accounts have been added, so the GLTEMP database can be emptied.

# Entering Matching Records

Inputting journal entries illustrates the entry of matching records.

Journal entries in the general journal database must match, by account number, records in the general ledger database. If a record entered in the JOURNAL database does not match a record in the GLEDGER database, it is an error.

Before being posted to GLEDGER, JOURNAL records are compared to the GLEDGER records to verify that they match an account number in the chart of accounts. Those JOURNAL records that do not match are corrected before posting.

Example:

The procedure outlined below enters matching records into the JOURNAL database.

1. `ENTER JOURNAL`

   Journal transactions are entered, using the `JOURNAL.FRM` form.

2. `SORT JOURNAL BY ACCOUNT`
   `SORT GLEDGER BY ACCOUNT`

   Sorting the JOURNAL and GLEDGER databases before comparing and posting can speed the time for these commands by a large margin over unsorted operation.

3. `COMPARE JOURNAL GLEDGER NOT MATCHING ACCOUNT`

   JOURNAL records are compared to GLEDGER records by matching ACCOUNT. JOURNAL records that do not match are written to the **RESULT** database. The form and data items for the **RESULT** database are the same as for the JOURNAL database.

4. `TITLE T,'GENERAL JOURNAL REPORTS',S,`
   `JOURNAL EXCEPTIONS FOR ',DATE`
   `PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT`

   Journal records that do not match a record in the GLEDGER database are printed in an exception report. The problem in either the GLEDGER or JOURNAL database is determined, and the respective databases are updated, if needed. If either database is updated, the procedure restarts with step #2 above.

Entry of correct JOURNAL transactions is complete when the printed report shows zero records; that is, when all JOURNAL records match records in the GLEDGER database. The next step is posting the JOURNAL to the GLEDGER database.

# Posting Transactions to a Master Database

The posting of journal entries to the general ledger serves as an example of transaction posting.

In the general ledger accounting system, journal entries are collected for a month accounting period. At the end of a month, the journal entries are posted to the general ledger. After posting, journal entries are accumulated in an audit trail database, and the JOURNAL database is emptied in preparation for the next month's entries.

Example:

The procedure outlined below posts journal entries to the general ledger.

1. `STAX JOURNAL BY DEBIT, CREDIT`

   Statistics are calculated for DEBIT and CREDIT, showing the total debits and credits in the JOURNAL database. For a "trial balance", the total credit must equal the total debit. If these totals are not equal, the error must be found and the journal file updated with the `UPDATE` command.

2. `POST GLEDGER JOURNAL BY ACCOUNT AND ADD DEBIT, CREDIT REPLACE MONTH`

   For each matching ACCOUNT, DEBIT, and CREDIT amounts from the JOURNAL database record are added to the DEBIT and CREDIT amounts in the correct GLEDGER record. The MONTH in the GLEDGER record is replaced by the MONTH from the JOURNAL record.

3. `COMPUTE GLEDGER ST YTDAMT = DEBIT - CREDIT`

   An account balance (YTDAMT) is calculated by subtracting CREDITs from DEBITs.

4. 
```
SELECT GLEDGER ST ASSET IS Y
TITLE T,'GENERAL LEDGER MONTHLY REPORTS',S,
 'ASSET ACCOUNTS REPORT FOR ',DATE,S
PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT
 AND COMPUTE TOTAL YTDAMT
```

   General ledger records for asset accounts are selected and copied to the **RESULT** database. A report showing asset account debits and credits, and the total of the asset accounts is printed. The total will be compared to the total liabilities in the next report.

5. 
```
SELECT GLEDGER ST LIABILITIES IS Y
TITLE T,'GENERAL LEDGER MONTHLY REPORTS',S,
 'LIABILITIES ACCOUNTS REPORT FOR ',DATE,S
PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT
 AND COMPUTE TOTAL YTDAMT
```

   Liabilities accounts are selected, and the corresponding report is printed for liabilities. If the total year-to-date totals from the assets and liabilities reports are not equal, the general ledger database is in error. Correcting entries must be entered into the JOURNAL database and the JOURNAL re-posted to the GLEDGER as described above. Prior to any adjusting entries, the JOURNAL database is emptied as described below.

6. 
```
APPEND JOURAUDT JOURNAL
EMPTY JOURNAL
```

   Finally, the JOURNAL database's journal entries are added to the audit trail database (JOURAUDT), and the JOURNAL database is emptied for the next cycle.

# Transaction Processing HELP Screens

The General Ledger Accounting System could be used repetitively, day by day, by clerical personnel following the procedures in the preceding sections. However, daily use of the application would be greatly simplified through the use of a **HELP** screen. The **HELP** screen in Figure 11-3 illustrates how simple data entry and posting can become with this approach.

```
*****************************************************************
*          GENERAL LEDGER ACCOUNTING SYSTEM                     *
*****************************************************************
*                                                               *
*      SELECT ONE OF FOLLOWING OPTIONS                          *
*                                                               *
*                                                               *
*      1.  INPUT CHART OF ACCOUNTS  [RUN B:ACCNTS]              *
*                                                               *
*      2.  UPDATE GENERAL LEDGER  [RUN B:UPGLEDG]               *
*                                                               *
*      3.  ENTER JOURNAL ENTRIES  [RUN B:JOURNIN]               *
*                                                               *
*      4.  JOURNAL TRIAL BALANCE  [RUN B:TRIALB]                *
*                                                               *
*      5.  POST GENERAL LEDGER  [RUN B:POSTGL]                  *
*****************************************************************
```

Figure 11-3. General Ledger HELP Screen

The **HELP** Screen is defined with the `FORMAT` command:

```
FORMAT ACCNTG.HLP
```

Note that when the **HELP** screen is displayed for function selection in response to the `HELP` command, the information in square brackets on the screen is not displayed.

# Example Command Procedure Files

The **HELP** screen of Figure 11-3 responds to each selection by **RUN**ning a corresponding Condor command procedure file. The example requires five command procedures:

`ACCNTS.CMD` — Entering Chart of Accounts

`UPGLEDG.CMD` — Updating General Ledger

`JOURNIN.CMD` — Entering General Journal

`TRIALB.CMD` — General Journal Trial Balance

`POSTGL.CMD` — Posting Journal to General Ledger

Each command procedure would be created in turn using the `EDIT` command. The command procedures to implement the required functions are given below.

1. `ACCNTS.CMD COMMAND` Procedure

```
COMPARE GLTEMP GLEDGER NOT MATCHING ACCOUNT
TITLE T,'GENERAL LEDGER REPORTS',S,S,'NEW ACCOUNT
 NUMBER ADDITIONS DATE', DATE,S,S
PRINT RESULT BY ACCOUNT NAME DEBIT CREDIT
APPEND GLEDGER RESULT
COMPARE GLTEMP RESULT NOT MATCHING ACCOUNT
TITLE T,'GENERAL LEDGER REPORTS',S,S'DUPLICATE
 ACCOUNT NUMBERS',DATE,S,S
PRINT RESULT BY ACCOUNT NAME DEBIT CREDIT
SORT GLEDGER BY ACCOUNT
TITLE T,'GENERAL LEDGER REPORTS',S,S,'GENERAL
 LEDGER',DATE,S,S
PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT
EMPTY GLTEMP
HELP ACCNTG
```

2. `UPGLEDG.CMD COMMAND` Procedure

```
UPDATE GLEDGER
SORT GLEDGER BY ACCOUNT
TITLE T,'GENERAL LEDGER REPORTS',S,S,
 'GENERAL LEDGER CHART OF ACCOUNTS DATE',DATE,S,S
PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT
HELP ACCNTG
```

3. JOURNIN.CMD COMMAND Procedure

```
ENTER JOURNAL
SORT JOURNAL BY ACCOUNT
SORT GLEDGER BY ACCOUNT
COMPARE JOURNAL GLEDGER BY ACCOUNT
*MESSAGE 'ENTER ACCOUNTING MONTH'
*GET $1
TITLE T,'GENERAL LEDGER REPORTS FOR $1',PAGE,S,S,
 'JOURNAL EXCEPTIONS FOR DATE', DATE,S,S
PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT
HELP ACCNTG
```

4. TRIALB.CMD COMMAND Procedure

```
*MESSAGE 'ENTER ACCOUNTING MONTH'
*GET $1
TITLE T,'GENERAL LEDGER REPORTS FOR $1',PAGE,S,S,
 'JOURNAL TRIAL BALANCE FOR DATE',DATE,S,S
STAX JOURNAL BY DEBIT CREDIT (T)
TITLE T,'GENERAL LEDGER REPORTS FOR $1',PAGE,S,S,
 'GENERAL LEDGER TRIAL BALANCE FOR DATE',DATE,S,S
STAX GLEDGER BY DEBIT CREDIT (T)
HELP ACCNTG
```

5. POSTGL.CMD COMMAND Procedure

```
*MESSAGE 'ENTER ACCOUNTING MONTH'
*GET $1
POST GLEDGER JOURNAL BY ACCOUNT AND ADD DEBIT, CREDIT
 AND REPLACE MONTH
COMPUTE GLEDGER ST YTDAMT = DEBIT - CREDIT
SELECT GLEDGER ST ASSET IS Y
TITLE T,'GENERAL LEDGER REPORTS FOR $1',
 PAGE,S,S'ASSETS ACCOUNT REPORT FOR DATE',DATE,S,S
PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT AND
 COMPUTE TOTAL YTDAMT
SELECT GLEDGER ST LIABILITY IS Y
TITLE T,'GENERAL LEDGER REPORTS',DATE,S,S,
 LIABILITIES ACCOUNT REPORT FOR DATE',DATE,S,S
PRINT RESULT BY ACCOUNT ACCNAME DEBIT CREDIT AND
 COMPUTE TOTAL YTDAMT
HELP ACCNTG
```

# Daily Transaction Processing

This concludes the steps required to set up the General Ledger example for use. Clerical personnel can now enter the single command:

```
HELP ACCNTG
```

to gain access to the **HELP** screen at the start of each data entry session. Further database processing takes place by selecting an option (from 1-5) from the **HELP** screen, and using the database form for data entry and update. Of course, if an unanticipated inquiry arises, Condor's powerful ad hoc capabilities can be used to answer it quickly, drawing on the data stored in the three databases. It is this combination of easy daily operation with powerful ad hoc facilities that makes Condor an attractive "programming language" for implementing transaction processing applications.

# Chapter Summary

- HELP screens and command procedures are used to streamline repetitive Condor procedures.
- In a general ledger application, separate databases are used to store records for each account, daily journal entries and an audit trail of journal records.
- When entering new records into an existing master database, specific command sequences should be used so as to not allow duplicate master records (by account number or whatever data item is selected).
- When entering a matching record in a transaction database, specific command sequences should be used to assure there will be a matching record in the master database.
- New records entered into a transaction database are posted to records in the master database and also saved in the audit trail database.
- The HELP screen is a menu from which the operator may select the operations to be performed.
- Command procedures are programmed sequences of instructions by which the system performs the operations selected from the HELP screen menu.

# Index

# TO 20-1 USER MANUAL


A

## B



## C

D

## F



## G



## H

I



J



L



M

N



O



P



Q

S

T

U



V



W



X