FORTUNE:WORD™

RECORDS PROCESSING
USER'S GUIDE

Office Automation

FORTUNE SYSTEMS

Records Processing
User's Guide

Contents

3   Creating the Format Document

5   Creating Control-Glossary Entries

6   Advanced Control-Glossary Functions   6-1

About This Guide

This guide is a learning and reference tool for Fortune:Word Records Processing. Records Processing is an integral part of Fortune:Word that allows you to quickly and economically produce customized form letters, labels, or lists. This guide teaches you how to use Records Processing as a mail-merge program to automate mass mailings. It also teaches you how to use Records Processing as a simple data base management system.

Anyone who knows how to use Fortune:Word can learn to use Records Processing. For detailed reference information and a procedural summary of any Fortune:Word feature, refer to the *Fortune:Word Reference Guide*. Each feature is listed alphabetically in Chapter 3 of the Guide. Cross-references to other features are provided.

This guide describes glossary features that can be used with Records Processing control-glossary entries. It does not contain complete descriptions of all the glossary functions. Use the *Fortune:Word Glossary User's Guide* as a complement to the *Fortune:Word Records Processing User's Guide*.

You do not need to be familiar with Fortune:Word glossary functions to use Chapters 1 through 4 of this guide. Chapters 5 through 7 contain instructions on how to use control-glossary entries to increase the power of Records Processing. You can learn how to write control-glossary entries from the information in this guide. A familiarity with glossary functions is useful, but is not essential to understand the information in this guide.

HOW THIS BOOK IS ORGANIZED

This book has seven chapters. Appendices and an index are also provided. Following is a brief description of each part.

Chapter 1: Introduction: This chapter explains the concepts of Records Processing and briefly introduces the four types of documents you can create. It explains how to use the Records Processing menu. At the end of the chapter a short Records Processing tutorial shows you now to produce a short form letter.

Chapter 2: Creating the List Document: Records Processing list documents act as the data base that contain the variable information you manipulate with other parts of the Records Processing feature. This chapter contains an in-depth

discussion of how to create and use list documents, and also contains training exercises.

Chapter 3: Creating the Format Document: The format document contains the unchanging data and text that merge with the variable data and text from records in the list document. This chapter contains an in-depth discussion of the parts of a format document, and also contains training exercises. Information is presented on using **repeat** and **include** statements in format documents.

Chapter 4: Creating Format Documents for Address Labels, Envelopes, and File Cards: As you develop expertise in Records Processing, you will create more types of format documents. This chapter tells you how to set up format documents for three special types of mail-merge activities: address labels, envelopes, and file cards.

Chapter 5: Creating Control-Glossary Entries: Control-glossary entries can be used to sort records or to sort and select specific records based on criteria that you specify. This chapter describes control-glossary syntax and basic functions, and provides complete examples of entries for each function.

Chapter 6: Advanced Control-Glossary Functions: This chapter describes how to use arithmetic operators and string-manipulation functions in control-glossary entries. Specific examples of entries using these functions are provided.

Chapter 7: Error Messages and Troubleshooting: This chapter contains an alphabetical list of Records Processing error messages and troubleshooting suggestions.

Appendices

- **Appendix A String-Manipulation Functions**: This appendix provides an alphabetical listing of the string-manipulation functions and syntax for quick reference.

- **Appendix B ASCII Collating Sequence**: This appendix shows the case-insensitive ASCII collating sequence that is used by Records Processing.

- **Appendix C Creating a List Document Using Glossary Entries**: This appendix gives some examples of glossaries you can use to create list documents.

CONVENTIONS USED IN THIS BOOK

The following conventions are used throughout this document:

- The names of keyboard function and editing keys you press are capitalized, as shown in the following examples.

RETURN   SEARCH   INDENT   GO TO PAGE
INSERT

- The four directional arrow keys (up, down, left, and right) that move the cursor are identified as follows:

    UP   DOWN   LEFT   RIGHT

- To perform certain word processing functions, you have to hold down one key while you press a second key. This combination of keystrokes is shown in the following way:

    Press SHIFT/COPY
    Press SHIFT/MERGE

    In the examples above, you hold down SHIFT while you press COPY or MERGE.

- Words or phrases you type are in boldface type, as shown in the following example:

    Press MERGE, type **first name**, then press SHIFT/MERGE

- Variable words or phrases that you replace with your own selection are in italic type, as shown in the following example:

    Type: *field label*

- Screen prompts and messages are in italic type as shown in the following example:

    Press EXECUTE to continue

- Fortune:Word document names are in bold type as shown in the following example:

    Create a new document named **add.form**

Text and Program Display

In the control-glossary entries in this guide, keywords are in italics, **functions** are boldfaced, and ***variables*** are boldfaced italics. This convention is only used for emphasis in examples. It is not necessary for you to observe these conventions in your glossary entries.

RELATED FORTUNE DOCUMENTS

Following is a list of other Fortune publications that will be of interest to you.

- *How to Use Fortune:Word, Self-Paced Learning*
- *Fortune:Word Reference Guide*
- *Fortune:Word Glossary User's Guide*
- *FOR:PRO User's Guide*

Chapter 1
Introduction

Records Processing is a major feature of Fortune:Word. To access Records Processing you select Document Processing Tools from the Fortune:Word Main menu, then select Records Processing. Records Processing can be used in many different ways. As an important example, you can use Records Processing as a mail-merge program to automate mass mailings. In this application, a list of addresses is merged into a document to quickly and economically produce customized form letters, labels, or lists. You only need to type the address list and a master form letter one time. Records Processing merges the individual addresses into the form letter for you automatically. You can then use the same address list to print labels or envelopes, and to generate reports.

Once you find out how easy it is to use Records Processing as a mail-merge feature, you may want to learn some of its more sophisticated capabilities. You can use Records Processing as a simple data base management system to maintain parts lists, handle client lists, or update employee records. You can merge selected information into reports produced with Fortune:Word, sorting and selecting specific items from a list document that contains your data base.

This chapter contains information on the following topics:

- How Records Processing works
- The four types of Records Processing documents
- The Records Processing menu
- A Records Processing tutorial

HOW RECORDS PROCESSING WORKS

When you use Records Processing for mail merge, you create two Fortune:Word documents. One document—the *list document*—contains the list of addresses. The other document—the *format document*—contains the letter, with markers to indicate where the variable information from the list document is to be inserted. Figure 1-1 shows a sample list document, and Figure 1-2 shows a sample format document. You will learn more about list and format documents in later chapters of this guide.

```
Doc add.list    Page 1   Line 21  Pos 1
word Format 1 Spacing 1 Length 54
+(1►....►1...►....►2...►....►3...►....►4...►....►5...►....►6....◄
<first name>Ronald◄
<last name>Parsons◄
<address>4390 Alealoa◄
<city>Honolulu◄
<state>HI◄
<zipcode>99301◄
<>◄
<first name>Dorothy◄
<last name>Adams◄
<address>389 N. 25th St.◄
<city>Seattle◄
<state>WA◄
<zipcode>96890◄
<>◄
<first name>Averil◄
<last name>Moore◄
<address>390 N. Birch St.◄
<city>Fresno◄
<state>CA◄
<zipcode>93109◄
<>◄
```
A2382

Figure 1-1. A Sample List Document

```
Doc add.form    Page 1   Line 1   Pos 39
word Format 1 Spacing 1 Length 54
1(1).....1.........2.........3.........►4.........5.........6....◄
                                    ►<date1>◄
◄
<first name> <last name>◄
<address>◄
<city>, <state> <zipcode>◄
◄
Dear <first name>:◄
◄
Thank you for your membership in the Central Valley Art
Association.  As a member, you will receive our newsletter
keeping you informed of meetings, workshops, exhibits, and
special events.◄
◄
                              ►Sincerely,◄
◄
◄
                              ►Sandra Simons◄
                              ►Public Relations◄
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1(1).....1.........2.........3.........►4.........5.........6....◄
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
(Document End)
```
A2383

Figure 1-2. A Sample Format Document

After you have created these two documents, you select Records Processing from the Document Processing Tools menu. You then enter the names of the two documents on the Records Processing menu. Records Processing combines the list and format documents to create an *output document*. You can do further editing on this document or print it as it is. Figure 1-3 shows a sample output document.

You do not have to know how Records Processing merges documents to use this feature successfully. However, reading the explanation of the process that follows may help you remember what information should go into a list document and what should go into a format document. Suppose you have a list document containing a list of addresses, and a format document containing a one-page form letter. In combining these two documents, Records Processing automatically follows these steps:

1.   Begin at the top of the format document.

2.   Put text from the format document into the output document.
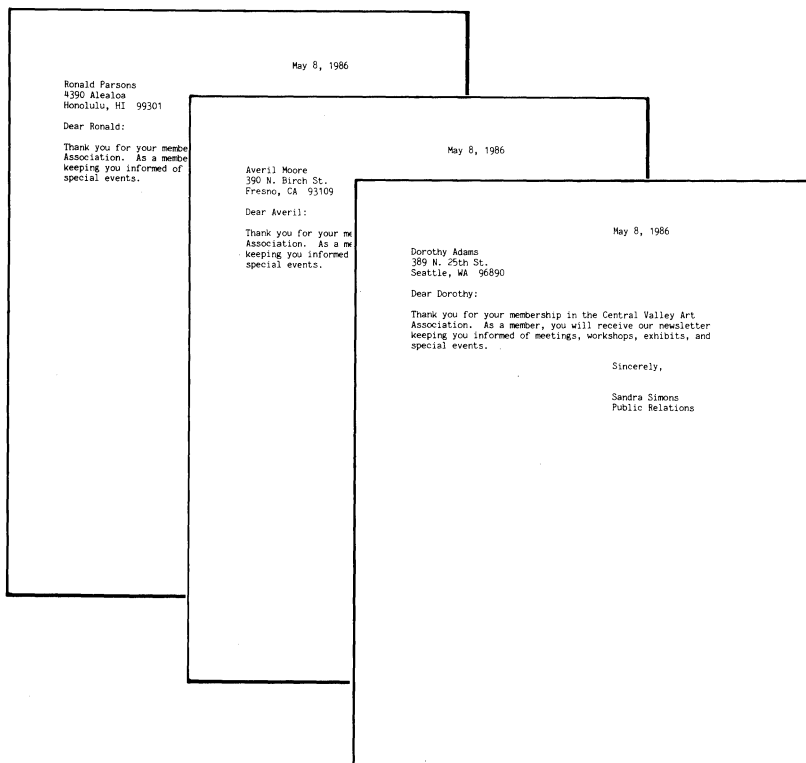
Figure 1-3. A Sample Output Document

3.   When a merge marker is found indicating that variable information is to be inserted, go to the list document.

4.   Find the variable information from the first address in the list document, and insert it into the output document.

5.   Go back to the format document, and repeat steps 2 through 4 until the end of the format document is reached.

6.   Repeat steps 1 through 5 until the end of the list document—the last address—is reached.

When this process has been completed, each page of the output document will contain a letter with a different address. Note that in this example the address also includes the name. There will be one page in the output document for each address in the list document, plus one blank page at the end of the document. You can edit, print, and archive the output document as you would any other Fortune:Word document.

There is another kind of document, called a *control glossary*, that you can use with Records Processing. A control glossary is a Fortune:Word glossary document containing entries. Fortune:Word glossaries can be used to store text and keystrokes that can be recalled for use later. A control-glossary entry uses functions accessible only from the Records Processing menu. You can use a control-glossary entry to sort and/or select specific items from a list document.

FOUR TYPES OF DOCUMENTS

The four types of documents used in Records Processing have already been briefly introduced. To summarize, when you use Records Processing, you first create up to three kinds of Fortune:Word documents and then use the Records Processing menu to merge the information in them into a final document. These are the three types of documents you can create to use with Records Processing:

- The list document
- The format document
- The control glossary

After you make your selections from the Records Processing menu, these documents are merged to create a new document called the *output document*.

Sometimes, you will need to use all four kinds of documents to achieve the results you want. At other times, you will use only two or three of the four types. The four kinds of documents are described below briefly. Later chapters contain detailed information on how to create and use each type of document.

The List Document

The list document is a Fortune:Word document containing information grouped into *fields* and *records*. For example, if the list document contains a list of addresses, each separate part of one address is a field, and each complete address is a record. You will learn more about fields and records in Chapter 2. Although an address list is a common kind of list document, such a document can contain a list of information on almost any subject, such as merchandise orders or machine parts. You can merge the information from a list into form letters, or you can use it for record keeping. In addition, you can select information from a list document to use in building tables or other forms.

Suppose, for example, that an administrative assistant in a North American company's international accounts department creates a list containing the names and addresses of the company's clients in Europe. This list document could be used in the following ways:

- Whenever a sales representative goes to Europe, the assistant could use Records Processing to create a list of European clients to help the representative.

- The assistant could combine the same list with a form letter to create mass mailings to all international clients.

- Using client names and sales information, the assistant could use the list to prepare a table showing how much the clients in each country ordered from the company.

The Format Document

The format document is a Fortune:Word document containing the basic structure, format line(s), standard text, header and footer and work page of the final document. The format document includes any text that remains the same from record to record, plus field labels indicating where the system is to insert information from the list document. Records and fields are described in Chapter 2.

A typical format document (see Figure 1-2), of a letter to be used in a mass mailing has the standard body of a letter, including a salutation, text, a closing, and possibly, a date. However, the format document contains special identifying field labels in place of the address, the name in the salutation, and any other information that varies from one record to another in the list document. The field labels indicate where the variable information from the list document is to go in the letter.

The Control Glossary

The control glossary is a glossary document containing an entry or entries consisting of glossary functions specific to Records Processing. You can only use these control-glossary entries from the Records Processing menu. The control glossary does not need to be attached for you to use one of the entries from the Records Processing menu. You use a control glossary if you want to select particular records from a list document or do sorting from the Records Processing menu.

You can create a new control glossary by using the Create New Glossary selection on the Glossary Functions menu, or you can add control-glossary entries to an existing glossary document. If you are not familiar with creating, editing, and verifying glossaries, refer to the *Fortune:Word Glossary User's Guide* for more information.

A control-glossary entry might contain instructions for sorting a list of records alphabetically by last name, or instructions for selecting the records of people who live in a particular city or state from an address list. Figure 1-4 shows a sample control-glossary entry. This entry would select all records for the state of California from a list document and sort them by zip code into an output document.

There are two ways you can use a control-glossary entry from the Records Processing menu to produce an output document:

- Use a control-glossary entry with list and format documents. The control-glossary function is performed first, and the results are then merged with the format document to produce the output document.

- Use a control-glossary entry with a list document. When no format document is used, the field labels are preserved in the output document. In this way, you can select specific records from a master list document to generate smaller, specialized list documents.

Figure 1-5 shows all the possible ways you can combine the three preliminary Records Processing documents to produce a final output document.

```
Doc rp.sort    Page 1  Line 10  Pos 1
word Format 1 Spacing 1 Length 54
+(1►....►1...►....►2...►....►3...►....►4...►....►5...►....►6....◄
entry s◄
{◄
   ►select_record◄
   ►   ►{◄
   ►   ►   ►if (<state> == "CA")◄
   ►   ►   ►save_record◄
   ►   ►{◄
   ►sort<zipcode>◄
{◄
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =




(Document End)
```

Figure 1-4. A Sample Control-Glossary Entry

NOTE: The only time you can use a format document
without a list document or control glossary is when you
are using a format document for Document Assembly that
contains **include** statements, but does not contain any
field labels. See Chapter 3, "Creating the Format
Document," for information on how to use include
statements.

The Output Document

The output document is a Fortune:Word document that contains the final
text that results when Records Processing automatically combines
documents. You assign a name to this document, and Records Processing
creates it automatically as a part of its function. Output documents can
range in length from a large document containing all the letters in a mass
mailing to a small sublist containing only the records selected for a
particular purpose from a larger list.
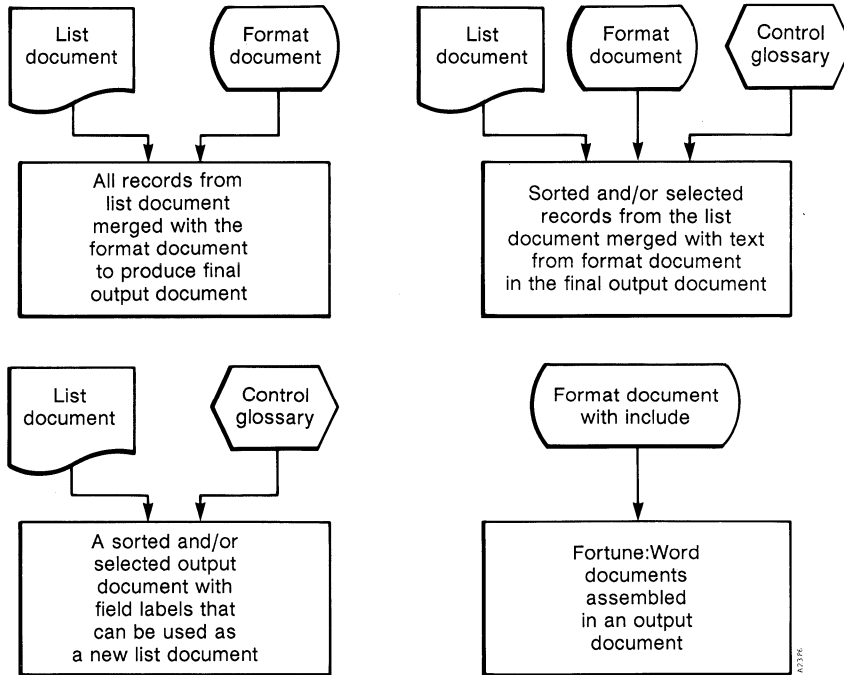
Figure 1-5. *Records Processing Uses from One to Three Documents, Depending on What You Specify*

When Records Processing is used with a mailing list, for example, the system merges the names, addresses, and other variable information from the list document into the standard letter contained in the format document. The resulting output document contains a letter for each person on the mailing list, with the appropriate name and address merged into the body of the letter. You can view the output document on your screen to make sure each letter is correct before you send it to the printer. Alternatively, you can have the letters printed without producing an output document.

The format lines are carried over from the Format document at the end of Records Processing. If you press CANCEL while an output document is being created, the records that have been processed appear in the output document, but the format may not match the format in the source document.

ACCESSING RECORDS PROCESSING

The Records Processing menu is shown in Figure 1-6.

```
                              RECORDS PROCESSING
       Please enter the following document names


              List document      _____
              Format document  _____
              Control glossary   _____
                   entry       _

       Please select one of the following processing options

              Make output document only
              Make output document and print
              Print only
```

Figure 1-6. The Records Processing Menu

To use Records Processing, follow these steps:

1.   Create your list and format documents from the Fortune:Word Main
     menu. If you plan to use a control-glossary entry, create a control
     glossary from the Glossary Functions menu or add a control-glossary
     entry to an existing glossary.

2.   Select *Document Processing Tools* from the Fortune:Word Main menu.

3.   Select *Records Processing* from the Document Processing Tools menu.

4.   Enter the name of the list document you want to use, and press
     RETURN.

5.   Enter the name of the format document you want to use, and press
     RETURN.

6.   If you are using a control glossary, enter the name of the glossary
     document and press RETURN. Otherwise, press RETURN to move to
     the next field.

7.   If you have entered a control-glossary name, type the entry label. Otherwise, press RETURN to move to the next selection.

8.   Select one of the processing options and press EXECUTE. If you have selected an option that creates an output document, enter the name of the output document, and press RETURN or EXECUTE.

If you use Records Processing frequently, you may find it convenient to use the shortcut code **rpr** to access the Records Processing menu (press COMMAND and type **rpr** from any menu).

PRINTING FUNCTIONS

When you combine list, format, and control documents from the Records Processing menu, you can make one of three selections:

- Make output document only
- Make output document and print
- Print only

Make Output Document Only

The first processing selection is *Make output document only*. When you choose this option, the Records Processing documents are combined in an output document on the system disk. This document is assigned the name you enter on the Records Processing menu. If you enter the name of an existing document, you can choose either to overwrite that document or to enter a new name for the output document.

Once the output document has been created, you can edit, print, and archive it as you would any other Fortune:Word document.

If you create large output documents, you may want to archive them for later use. You might also use the Delete Document selection on the Filing menu to remove them. If you delete an output document but save its accompanying list, format, or control glossary documents, you can always use Records Processing again to recreate the output document.

Fortune:Word menus such as the Print Document and Hyphenation and Pagination menus allow you to process documents that contain from 1 to 999 pages. You can edit and archive documents containing more than 999 pages. You cannot create a document containing more than 999 pages while editing. However, Records Processing may create an output document that contains more than 999 pages. If this happens, you must split the long document

up into several parts so that an individual document contains no more than 999 pages before you can print, use Spelling Tools, or hyphenate and paginate it. To do this, use the Create New Document selection from the Fortune:Word Main menu to create a new document, and SHIFT/MOVE the extra pages into that document.

Make Output Document and Print

The second processing selection is *Make Output Document and Print*. This option creates an output document on the system disk and automatically sends a copy of that document to the printer. When printing is completed, the copy is deleted. The settings on the Print Document menu for the format document are used as the default print settings. If no format document is used, the format line from the list document is used. Make sure that your printer is turned on and has enough paper. Once the document is sent to the printer from Records Processing, you can use the Printer Control menu to control printing functions.

Print Only

Suppose you frequently perform the same Records Processing functions and are confident that your output document is correctly formatted. In that case, you do not need to edit the output document before printing it. Thus, you may just want to print the output results without creating a copy of the document on the system disk. To do this, you can use the *Print Only* selection on the Records Processing menu.

Even though you only want to print the output document, a file must be created for the output. Records Processing creates a temporary file beginning with the characters dot and double-u (.W) to contain the information to be printed. When the printing is finished, the temporary file is automatically removed. Before you select this option, make sure that your printer is turned on and has enough paper.

If the printer stops for any reason during the printing, you can control the printing of the temporary file in the same way you would any other document in the print queue.

Of course, once the printing process has been completed, you cannot reprint selected parts of the document. This is because the temporary file has been removed from the system disk. If you think you may want to reprint portions of a Records Processing output document, you should choose one of the two other processing options when creating it.

A RECORDS PROCESSING TUTORIAL

Now that you have an idea of how Records Processing works, here is a brief exercise you can use to combine a list and format document into an output document.

In this exercise, you will type the list document shown in Figure 1-2 and the format document shown in Figure 1-3. For your convenience, the text of these documents is repeated here. Be sure to use the MERGE key to type the symbol " < " and the SHIFT/MERGE key to type the symbol " > ." You can compare the text of the documents on your screen with the figures.

Typing the Documents

1. Create a new document named **add.list** using the *Create New Document* selection from the Fortune:Word Main menu.

2. Type the three addresses exactly as shown below. Remember to use MERGE and SHIFT/MERGE. Compare your document with Figure 1-2 to be sure you have entered the text correctly.

```
<first name>Ronald
<last name>Parsons
<address>4390 Alealoa
<city>Honolulu
<state>HI
<zipcode>99301
<>
<first name>Dorothy
<last name>Adams
<address>389 N. 25th St.
<city>Seattle
<state>WA
<zipcode>96890
<>
<first name>Averil
<last name>Moore
<address>390 N. Birch St.
<city>Fresno
<state>CA
<zipcode>93109
<>
```

3. When the list is complete, leave the document, saving the changes.

4. Create a new document named **add.form** using the Create New Document selection from the Fortune:Word Main menu.

5. Type the form letter exactly as shown below. Remember to use MERGE and SHIFT/MERGE. Compare your document with Figure 1-3 to be sure you have entered the text correctly.

<datel>

<first name> <last name>
<address>
<city>, <state>   <zipcode>

Dear <first name>:

Thank you for your membership in the Central Valley Art Association. As a member, you will receive our newsletter keeping you informed of meetings, workshops, exhibits, and special events.

Sincerely,

Sandra Simons
Public Relations

---

6. Be sure to enter a page break at the end of the document, as shown in Figure 1-3.

7. When the form letter is complete, leave the document, saving the changes.

Merging the Documents

You have created two separate documents that can be merged together with Records Processing. To do this, follow the steps below:

1. Select *Document Processing Tools* from the Fortune:Word Main menu.

2. Select *Records Processing* from the Document Processing Tools menu.

3. Type **add.list** on the *List document* line and press RETURN.

4. Type **add.form** on the *Format document* line and press EXECUTE. Since you are not using a control glossary, you can leave the Control glossary and entry lines blank.

5.   Type **add.out** on the *Please enter output document name* line and press RETURN or EXECUTE.

6.   Return to the Fortune:Word Main menu.

7.   Using the Edit Old Document selection, look at the document **add.out**. If you have done everything correctly, the output document will contains three letters, each with a different address, and each on a separate page. The last page in the document will be blank. If you want to, you can print this document.

This chapter has introduced the basic concepts of Records Processing. In the next two chapters, you will learn more about how to create list and format documents.

Chapter 2
Creating the List Document

The list document is the most important of the three types of Records Processing documents. The entries you make in the list document are used as a data base for all subsequent Records Processing functions. You can use Records Processing without a format document or control glossary, but, in almost every instance, you must have a list document.

> NOTE: The only time you do not need to use a list
> document in Records Processing is when you have a format
> document with **include** statements and no field labels.
> See Chapter 3, "Creating the Format Document," for
> information on how to use include statements.

In this chapter you will learn how to create a list document. You can use the exercises in this chapter to learn how to set up the kinds of lists you might typically create in your work. The list documents you create in this chapter are used with the exercises in Chapters 3 and 4.

This chapter contains information on the following topics:

- The parts of the list document
- Designing the list document
- Typing the list document
- Using glossary entries to create the list document
- Creating an inventory list

THE PARTS OF A LIST DOCUMENT

The list document contains the variable text—the information that changes for each item merged into the format document. You use Fortune:Word to create and edit the list document. List documents are divided into *fields*. Each complete group of fields forms a *record*.

Fields

The individual items in a list document are called *fields*. Each field
has two parts:

- The field label
- The field contents

Field Labels

The *field label* is the name you assign to each field in a record. It
identifies the contents of the field and remains the same from record to
record. You use the same field label in the format document to indicate
where the variable information is to be placed. You also use the field
label to define fields you want to sort and select in a control-glossary
entry.

You choose the field labels and decide what information should be included
in the field contents. The labels you assign will depend on the type of
information in your list document and on the way you want to combine it
with the format document. It is a good idea to choose field labels that
briefly describe the field contents.

Once you decide on a label for a particular field, it is a good practice
to use it consistently in all your list documents. Suppose you decide to
use the field label <zipcode>. If you use <zip> to label the
zip-code field in another list document, you will not be able to use the
original format document with that list document unless you edit it to
change the field label.

Each field label begins with a MERGE and ends with a SHIFT/MERGE. Be
sure you use the Merge key (F6) to create the field labels in any Records
Processing document. Records Processing will not function correctly if
you use the angle brackets at the lower-right corner of the keyboard in
place of MERGE and SHIFT/MERGE. Refer to the section entitled
"Designing the List Document" later in this chapter for suggestions on how
to choose labels for your list document.

> **NOTE:** Field labels should never be broken by Returns or
> page breaks.

You do not need to have the same number of field labels in each record.
However, it is a good idea to make the number of field labels match, even
if many field contents are blank for a particular field label. What
happens if you do not have the same field label in each record? When you
run Records Processing, if that label is used in the format document, you
will have to press EXECUTE each time a record that does not contain that

field label is found, or press CANCEL to end the process. If you have many records that are missing that field label, it will take more time to run Records Processing, and you may not get all the records you want in the output document.

Field Contents

The *field contents* consist of the text you type following a field label. This information, which varies from record to record in the list document constitutes your data base.

When you type the field contents, you should enter only the text, punctuation, and spaces. An entry does not have to be one line. You can also enter Returns, Tabs, and other screen symbols needed within a variable entry. Any text or punctuation surrounding the field contents that does not change for each record should be included in the format document rather than in the field contents in the list document. For example, the comma and space that appear between the name of the city and the name or abbreviation of the state in an address should be entered into the format document. These characters should not be included in the list document as part of the <city> or <state> field contents.

The essential parts of a field and the keys you press to create a field are shown in Figure 2-1.

> REMEMBER: Always press MERGE (F6) to begin a field label and SHIFT/MERGE to end a field label.

```
Merge Symbol (press the Merge key)
  │  Field Label
  │    │  Merge Symbol (hold down Shift key and press the Merge key)
  │    │    │  Field Contents                                    A1078
  │    │    │    │
  │    │  ┌──┴──────────┐
  │    │  │             │
<name>Mr. Averil Moore
```
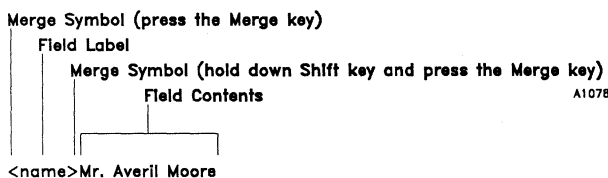
Figure 2-1. The Parts of a Field

When you run Records Processing, you can use the field contents in two ways:

- When you use a list document and a format document, the field labels in the format document determine where the field contents will appear in the output document. In this case, the field labels are never included in the output document. See Chapter 3, "Creating the Format Document," for information on how to create and use format documents.

- When you use a list document and a control glossary without using a format document, the field labels remain with the field contents. In this way, you can sort and select individual records from a master list document to create separate specialty lists. The master list remains unchanged. See Chapter 5, "Creating the Control Document," for information on how to create and use control-glossary entries.

Size Limitations

A field can contain a maximum of 2048 characters, including field labels and merge symbols. Text formatting symbols such as TAB and RETURN are counted as more than one character. See Appendix B in the *Fortune:Word Reference Guide* for more information.

If you use control-glossary entries for sorting and selecting, there are additional restrictions on the sizes of fields and records:

- Each record must contain no more than 2048 characters, including field labels, Merge symbols, and Returns.

- If you use sort, the total length of the contents of all fields to be sorted in a record cannot be greater than 1024 characters, excluding field labels.

See Chapter 5, "Creating Control-Glossary Entries," for information on how to write control-glossary entries to sort and select records.

Records

A group of fields that belong together is called a *record*. Each record contains specific information about, for example, a person, company, or product. This information is saved in the list document. It can be merged with a format document or stored for retrieval later. A record may consist of one letter, one word, one line, or several lines, all pertaining to one particular member of a list.

Records can be any length unless you plan to use a control-glossary entry to select or sort the list document. If you use a control-glossary entry, no record in the list document can contain more than 2048 characters. A list document can have an unlimited number of records (as long as there is enough free space on the system disk).

Each record must end with an end-of-record indicator on a separate line, followed by a RETURN. To create an end-of-record indicator, type MERGE, SHIFT/MERGE, and RETURN (<>).

If you create a list composed only of employee names, each name is considered an individual record. However, a list typically contains records with more information about the list members. For example, the address list of a magazine publisher might contain additional information about the subscribers, such as their business titles and the names of magazines to which they subscribe.

Here are three examples of different types of records that you might have in separate list documents:

<first name>Rose   <garment>Wool suit
<last name>Smith   <style>Three piece
<title>Technical Writer   <fabric>Gray pinstripe
<>   <suppliers>Clothing, Inc.
                                   <price>$375.00
                                   <>

<name>I. M. Smith
<title>Representative
<address>House of Representatives
<city>Washington
<state>DC
<zipcode>10001
<>

Which record structure you choose depends on what requirements you have for the list. You might use the first record in a list of employees, the second in a product list, and the third in an address list.

> **NOTE:** Never put a page break in the middle of a record. When you put page breaks in your list document, always start the top of a new page with the first line of a new record.

Summary

The important information about creating a list document is summarized below:

- Each item in a list document is called a field. A field can contain a maximum of 2048 characters, including field labels and merge symbols.

- Fields have two parts: field labels and field contents.

- Fields labels always precede field contents.

- Each field label must begin with a MERGE and end with a SHIFT/MERGE.

- Never put a RETURN or page break in the middle of a field label.

- A collection of related fields is a record.

- There is no limitation on the size of records unless you use a control-glossary entry.

- If you use a control-glossary entry, each record can contain only 2048 characters.

- If you use sort in a control-glossary, the total length of the contents of all fields to be sorted in a record cannot be greater than 1024 characters, excluding field labels.

- Each record must end with an end-of-record indicator (<>) on a separate line, followed by a RETURN.

- Never put a page break in the middle of a record. If you put page breaks in your list document, always start the top of a page with the first line of a new record.

- The records in a list document do not need to have the same number of fields. However, this is not recommended since unpredictable results can occur.

- Field labels must be consistent from record to record in a list document. For example, if you use <zipcode>, you should not use <zip> as a label for the same field.

DESIGNING THE LIST DOCUMENT

Before you begin typing your list document, you should carefully consider
how you want to use the information. If you want to sort and select
records that contain specific types of information, the list document is
easier to manipulate when such information is in a separate field. You
will have more options if you have a separate field for <state> and
<zipcode> rather than combining them in one field named <state &
zipcode>.

You should also think carefully about the field labels you use. In
addition to being used in the list document, field labels are used in the
format document and in the control-glossary entry. Since the labels
appear without their accompanying information in these other two places,
you should assign field names to the labels that are easy to recognize.
This will make it easier for you to design the format document and to
write the control-glossary entry.

It is a good idea to keep your field labels as short as possible, however,
since field labels count in the total number of allowable characters per
field and record.

You do not need to include all the fields from a list-document record in
the format document. For example, if you are creating a membership
mailing list, you might add such information about each member as
enrollment date and dues paid. Currently, however, you might only want to
use the names and addresses from the list. You can enter all the
information you might ever use into a list document when you first create
it. You can also add more fields and information to a list document at
any time thereafter.

If you want to use information such as a company name more than one time
in a form letter, you only need to type it once for each record in the
list document. Each time you want to use the information in the format
document, you simply repeat the field label. For example, you could use
the field label <company> in the address line, and then refer to the
company by name in the body of the letter by repeating the field label
<company> at the appropriate place.

You can use the same list document with a variety of format documents to
create different output documents, as shown in Figure 2-2. Each of the
format documents might use a different set of field labels from the list
document. Be sure you think through the different ways you might want to
use the list document, and create it with the maximum flexibility
possible.
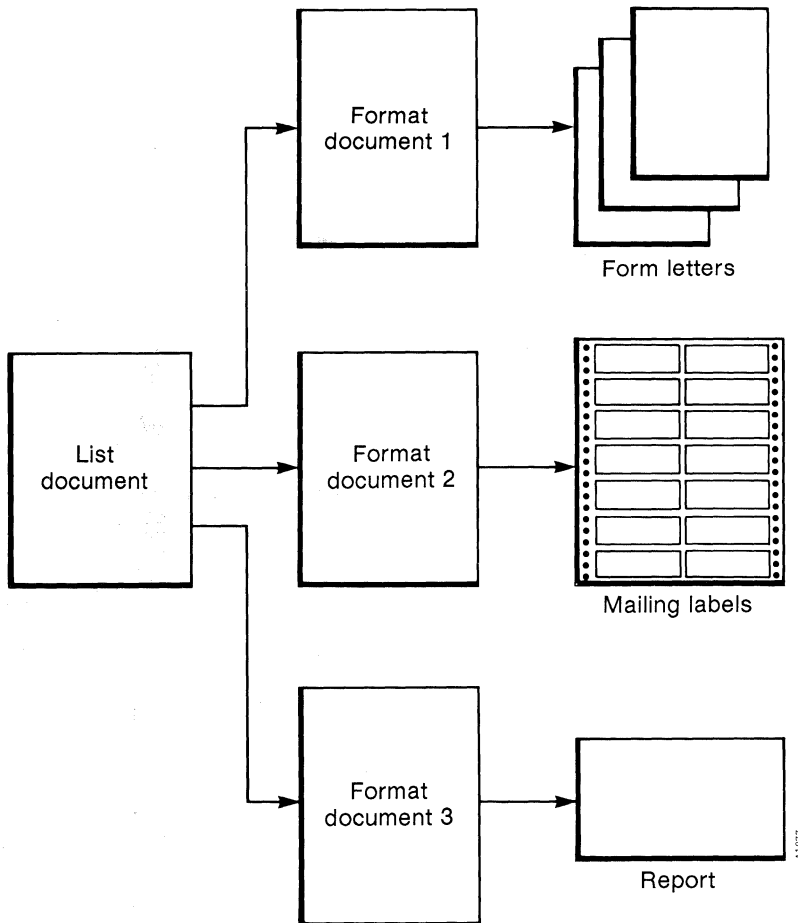
Figure 2-2. You can use the same list document with different format documents

Once you have decided on a naming system for your field labels, you should use it consistently for all the records in that list document. Suppose you decide to label a field <name> in an address list. If you use <Name> to label some of the name fields, Records Processing will be unable to process those records.

You do not need to keep the field labels in the same order from record to record in the list document. Moreover, the order does not have to be the same as in the format document. Nonetheless, a list document can be easier to edit if its field labels are in the same relative position in each record.

Suppose you want to put the following address in a list document:

> Ms. Ruby Katz
> 85 Swanson Road, Apt. 5
> Boxborough, MA   01719

You could assign the following field labels:

> <name>Ms. Ruby Katz
> <address>85 Swanson Road, Apt. 5
> <city>Boxborough
> <state>MA
> <zipcode>01719
> <>

With these field labels, you can easily write a control-glossary entry to sort or select addresses from the list document according to city, state, or zip code. However, if you later decide that you want to sort the list alphabetically by last name, it would be difficult for you to do so without revising the list document.

If you designed the list-document field labels in the following way, you would have a much more flexible data base:

> <honorific>Ms.
> <first name>Ruby
> <last name>Katz
> <street>85 Swanson Road
> <apt>Apt. 5
> <city>Boxborough
> <state>MA
> <zipcode>01719
> <>

The field labels in these examples are spelled out completely for clarity. You can abbreviate field labels in your list and format documents to save space.

> **REMEMBER:** Use field labels that are easy to recognize
> and remember. This can be especially important if you
> have a group of people using the same Records Processing
> documents.

An additional advantage of this selection of field labels is that, in a format document consisting of a letter, you can use the <honorific>, <first name>, and <last name> field labels for the address, and the <honorific> and <last name> field labels for the salutation, by typing **Dear <honorific> <last name>**: in the format document.

REMEMBER: You only need to enter the contents of a field once, even if you intend to use the field more than one time in the format document.

Follow these steps in designing a list document:

1.   After you have decided on the type of record needed for the list, make a sample of the format document. Refer to Chapter 3, "Creating the Format Document," for information on how to set up a format document.

2.   Define any parts of the format document that will vary from record to record. These will be your fields.

3.   Make a list of all the possible fields needed in a record. Write down the field labels, NOT the contents of the fields. The field labels should clearly identify the information contained in the fields.

TYPING THE LIST DOCUMENT

Now that you have designed the list document, you are ready to type it. Follow these steps:

1.   Create a new document, using the Create New Document selection on the Fortune:Word Main menu.

SUGGESTION: Use the word "list" or an abbreviation of it in the name of your list document. Give the format document a similar name, using the word "format" or an abbreviation of it. If you use this convention, it will help you remember what each document is and which list and format documents go together.

NOTE: The format line in the list document does not affect the output document, so you do not need to change it.

2.   Press MERGE.

3.   Type a field label.

4.   Press SHIFT/MERGE.

5.   Press RETURN.

6.   Repeat steps 2 through 6 until you reach the last field label.

7.   Type the end-of-record indicator by pressing MERGE, SHIFT/MERGE, and RETURN. For the list and format documents to merge properly, each record must end with an end-of-record indicator, including the last record in the list document. The last end-of-record indicator in the list document must be followed by a RETURN.

8.   Use the Copy key to copy the field labels, including the end-of-record indicator, and put them at the end of the document.

9.   You can keep on copying the field labels until you have the correct number of records before you enter any of the field contents. Alternatively, you can copy the field labels and then fill in the field contents one record at a time.

      For information about some shortcuts in creating list documents, refer to Appendix C, "Creating a List Document Using Glossary Entries."

      You can also use the Forms Processing feature to create list documents. Refer to the *Fortune:Word Reference Guide* for information about Forms Processing.

10.   Enter the field contents for each field label until you have completed all the records in the list document. Include only the punctuation and spacing that is to be part of the field contents. Do not leave any spaces between the end of the field label and the beginning of the field contents unless you make allowance for them in the format document.

11.   When you have entered all the records, leave the document and save the changes as you normally would.

12.   You can paginate your list document to make it easier to handle. However, be sure you do NOT put a page break in the middle of a record. If you insert page breaks within a record, an extra page break followed by a RETURN will be inserted into your output document. Page breaks BETWEEN records do not affect the pagination of the output document.

      When you use the Hyphenation and Pagination feature, you can set the text length so that page breaks are automatically entered at the correct place. For example, if the records in the list document are each 10 lines long, you can set the text length to 50. The paginated list document will then have 5 records per page.

You can place the label and contents for each field on a separate line in the list, or you can string them together on one or more lines, as shown in the example below. If you do combine fields on one line, it may be more difficult to edit the list document.

> <name>Robert Simmons<address>2390 Wisconsin
> Ave.<city>Washington<state>DC<zipcode>20301
> <>

Even though each record in a list document does not have to contain the same number of fields, if a field is missing and you use a control-glossary entry to sort or select the list, you may miss some records. When a field for a particular record has no information, just type the field label and leave the field contents blank, as shown in this example:

> <name>Mr. Ronald Parsons
> <title>Vice President
> <company>Windward Leasing
> <address1>4390 Alealoa
> <address2>
> <city>Honolulu
> <state>HI
> <zipcode>99301
> <>

In this list document, <address2> is a separate field for an apartment number, even though not all the people on the list live in apartments. If you want, you can later use a control glossary to find all the fields with missing information. See Chapter 5 for information on how to use a control-glossary entry to select records with blank fields.

Exercise

Type the list below, following the procedures outlined above. Save the entries in a document called **add.list**. This list document contains 10 records that can be used with the format document you create in the exercise in Chapter 3. You can also use this list document with some of the control-glossary entries presented in Chapters 5 and 6.

```
<honorific>Mr.
<first name>Ronald
<last name>Parsons
<title>Vice President
<company>Windward Leasing
<address>4390 Alealoa
<city>Honolulu
<state>HI
<zipcode>99301
<memb since>March, 1980
<memb end>March, 1986
<dues>$150.00
<paid>$250.00
<>
<honorific>Mrs.
<first name>Dorothy
<last name>Adams
<title>President
<company>Northwest Canners
<address>389 N. 25th St.
<city>Vancouver
<state>BC
<zipcode>V6K 2Y9
<memb since>September, 1978
<memb end>September, 1986
<dues>$200.00
<paid>
<>
<honorific>Mr.
<first name>Averil
<last name>Moore
<title>President
<company>Westmore Growers
<address>390 N. Birch St.
<city>Fresno
<state>CA
<zipcode>93109
<memb since>November, 1983
<memb end>November, 1985
<dues>$250.00
<paid>$150.00
<>
```

```
<honorific>Mr.
<first name>Arnold
<last name>Yates
<title>Manager
<company>Westfield Chamber of Commerce
<address>Rt. 9, Box 678
<city>Westfield
<state>CA
<zipcode>91293
<memb since>June, 1979
<memb end>June, 1986
<dues>$200.00
<paid>$250.00
<>
<honorific>Ms.
<first name>Leona
<last name>Rogers
<title>Vice President
<company>Central Valley Expeditions
<address>P.O. Box 45
<city>Corvallis
<state>OR
<zipcode>98101
<memb since>August, 1981
<memb end>August, 1986
<dues>$300.00
<paid>$250.00
<>
<honorific>Mr.
<first name>Richard
<last name>Harmon
<title>District Director
<company>California Mining Association
<address>1930 Watt Ave.
<city>Sacramento
<state>CA
<zipcode>93190
<memb since>February, 1980
<memb end>February, 1986
<dues>$250.00
<paid>$250.00
<>
```

<honorific>Ms.
<first name>Margaret
<last name>Overstreet
<title>President
<company>Overstreet Productions
<address>5389 S. 57th St.
<city>Mississauga
<state>Ontario
<zipcode>L4W 1E4
<memb since>July, 1981
<memb end>July, 1986
<dues>$250.00
<paid>$250.00
<>
<honorific>Mr.
<first name>David
<last name>Rossman
<title>Regional Manager
<company>Brimstone Microcircuits
<address>2618 Archway Ave.
<city>Sunnyvale
<state>CA
<zipcode>92167
<memb since>May, 1981
<memb end>May, 1986
<dues>$150.00
<paid>$150.00
<>
<honorific>Ms.
<first name>Joan
<last name>Peters
<title>Vice President
<company>Rightway Publications
<address>128 Maple St.
<city>Hibbing
<state>MN
<zipcode>55746
<memb since>April, 1983
<memb end>April, 1987
<dues>$200.00
<paid>$200.00
<>

```
<honorific>Mr.
<first name>Ray
<last name>Banyon
<title>Vice President
<company>Banyon Sales
<address>909 Washington Ave.
<city>Phoenix
<state>AZ
<zipcode>87190
<memb since>August, 1981
<memb end>August, 1986
<dues>$250.00
<paid>$250.00
<>
<name>
<title>
<company>
<address>
<city>
<state>
<zipcode>
<memb since>
<memb end>
<dues>
<paid>
<>
```

USING GLOSSARY ENTRIES TO CREATE THE LIST DOCUMENT

A helpful shortcut for creating field labels in a list document is to create a glossary entry containing the labels. Then, as you create the list document, recall the entry. If you are not familiar with using glossaries, refer to the *Fortune:Word Glossary User's Guide*.

This section describes how to create a glossary by example that you can then use to help you type the entries into your list document. Refer to Appendix C, "Creating a List Document Using Glossary Entries," for additional suggestions on how to use glossary entries to create a list document.

When you create a glossary to use with Records Processing, it is a good idea to use the initials **rp** in the glossary name to help you identify it as a Records Processing glossary.

> REMEMBER: You can combine control-glossary entries and regular entries in the same glossary. However, you will probably find it more convenient to create a separate glossary to use with Records Processing.

In the next section of this chapter, entitled "Creating an Inventory List," you will create a list document called **parts.list**. One glossary entry that you can use to create this list is presented in the paragraphs that follow.

Creating a Records Processing Glossary by Example

You can use the Glossary by Example feature to create a glossary entry to use when you create your list document. Follow these steps to create a glossary entry you can use to create the list document in the next example:

1.   Select *Glossary Functions* from the Fortune:Word Main menu.

2.   Select *Create New Glossary* from the Glossary Functions menu.

3.   Use **parts.g.rp** as the glossary name.

4.   Without typing anything from the editing screen, press CANCEL and EXECUTE. The glossary will verify, and the messages *An empty glossary is attached* and *Press execute to continue* are displayed.

5.   Press EXECUTE.

6.   Select *Create New Document* from the Fortune:Word Main menu.

7.   Use **parts.list** as the document name.

8.   When the editing screen appears, press MODE and then GL. The words *Glossary entry* flash at the bottom of the screen while you are creating the glossary by example.

9.   Type the labels listed below into the document. Be sure you use the Merge and Shift/Merge keys. If you make typing mistakes, use the Backspace key to correct them. Be sure you terminate the list with an end-of-record indicator (<>) and a RETURN.

          <part no>
          <type>
          <in stock>
          <on order>
          <last order>
          <back order>
          <unit cost>
          <>

10.   When you have finished typing the field labels and the end-of-record indicator, position the cursor on the RETURN following <part no>.

11. Press MODE and then GL. The words *Glossary entry* are no longer displayed at the bottom of the screen. The prompt at the top of the screen is *What entry?*

12. Type **a** and press RETURN.

The glossary entry in **parts.g.rp** should contain the information shown below. Since the Glossary by Example feature does not put each field label on a separate line, the format of the entry will vary. If you want, you can edit the glossary entry to make it look exactly like the example.

```
entry a
{
merge "part no" MERGE return
merge "type" MERGE return
merge "in stock" MERGE return
merge "on order" MERGE return
merge "last order" MERGE return
merge "back order" MERGE return
merge "unit cost" MERGE return
merge MERGE return
up(8) right(9)
}
```

> **NOTE:** There may be keywords such as "backspace" and additional characters in your glossary entry if you made any corrections while typing the entry.

If you want to type the entry directly into a glossary instead of using Glossary by Example, it should look exactly like the example above. For more information on how to create a glossary entry, refer to the *Fortune:Word Glossary User's Guide.*

To use the glossary entry you have created, the glossary **parts.g.rp** must be attached. For information on how to attach a glossary, refer to the *Fortune:Word Reference Guide.* To use the glossary entry, follow these steps:

1. Press GL and type **a**. The field labels are automatically entered into the document, and the cursor moves to the end of the first field label.

2. Type the field contents for each label.

3. Move the cursor to the line below the end-of-record indicator, and repeat steps 1 through 3 until you have entered all the records into the list document.

When you use a glossary entry like this to help you create your list document, you know that all the field labels are the same and are in the same order throughout the list document. This is an advantage because typographical errors in field labels can cause problems when you use Records Processing.

CREATING AN INVENTORY LIST

In this exercise, you will create an inventory list. You will use this document with exercises in Chapter 3 to produce various reports that a company might typically need. You can also use this document to try out some of the control-glossary entries presented in Chapters 5 and 6. Assume that you update the list monthly and generate an inventory report. You can also use the inventory list with a control-glossary entry to generate a smaller list of items and part numbers only—for example, the parts that cost the most or have a certain number in inventory.

Create a parts inventory list, using the information below. You can do this in one of two ways:

- Type all the information directly into a list document, including the field labels and field contents.

- Try using the glossary entry you created in the previous example to enter the field labels in your list document. You can then type in the field contents.

Later on, you will learn how to create an inventory report from the information and how to create and use a control-glossary entry to select specific records from the list document. Follow these steps in creating the parts inventory list:

1. Create a new document and name it **parts.list**.

2. Enter the following records in the list document:

        <part no>T8991Z
        <type>claw hammers
        <in stock>29
        <on order>2
        <last order>1/6/86
        <back order>
        <unit cost>5.67
        <>

```
<part no>78
<type>phillips screwdriver
<in stock>30
<on order>0
<last order>12/22/85
<back order>
<unit cost>2.38
<>
<part no>F890
<type>needle-nose pliers
<in stock>12
<on order>10
<last order>3/9/86
<back order>
<unit cost>4.74
<>
<part no>STY321909
<type>12-piece drills
<in stock>8
<on order>12
<last order>4/23/86
<back order>4
<unit cost>6.99
<>
<part no>U21
<type>tile knife
<in stock>7
<on order>5
<last order>2/27/86
<back order>
<unit cost>3.74
<>
<part no>L1908416J
<type>angle brackets
<in stock>20
<on order>49
<last order>1/27/86
<back order>29
<unit cost>.35
<>
<part no>H892
<type>door springs
<in stock>2
<on order>30
<last order>4/3/86
<back order>28
<unit cost>.98
<>
```

```
<part no>E7416
<type>tool box
<in stock>12
<on order>12
<last order>5/6/86
<back order>
<unit cost>10.45
<>
<part no>T12
<type>ratchet wrench
<in stock>4
<on order>6
<last order>5/25/86
<back order>2
<unit cost>7.12
<>
<part no>W090189
<type>pliers
<in stock>21
<on order>0
<last order>4/25/86
<back order>
<unit cost>3.49
<>
```

3.   Save the list document for later use with a format document.

You will use the document **parts.list** in exercises in subsequent chapters.

Chapter 3
Creating the Format Document

The format document contains the unchanging data and text that merge with the variable data and text from records in the list document. Together they create the output document. Some typical ways to use format documents are listed below:

- Form letters
- Mailing labels or envelopes
- Reports such as an inventory report or a sales report
- Document assembly—a way to automatically combine different Fortune:Word documents

This chapter begins with a brief description of the parts of a format document. The rest of the chapter provides detailed explanations and examples of ways you can use format documents. These format documents can be merged with the list documents you created in the exercises in Chapter 2.

THE PARTS OF THE FORMAT DOCUMENT

Format documents can have as many as five distinct parts, which are listed below. You can use these parts in a format document in any combination.

- Field labels
- Standard text
- Date and time field labels
- Repeat statements
- Include statements

Field Labels

The field labels indicate where information from the field contents of the list document will be inserted. The field labels in the format document must be exactly the same as the field labels in the list document.

However, you do not need to include every field label from the list document in the format document. You can include the same field label more than once.

Standard Text

The standard text is the unchanging text that you want to appear in the final document. The format document should contain both the standard text and the spacing and punctuation marks around the field labels.

Format Lines

Only the format lines in the format document control the format of the text in the output document. This includes any text incorporated from another Fortune:Word document when you use the **include** feature for document assembly.

Page Breaks

Put a page break at the end of the format document to begin each new copy of the format document on a new page in the output document. The format document can have additional page breaks if needed.

> **CAUTION:** If you do not end the format document with a page break, Records Processing will try to put all of the merged text onto one long page. Your system may not have enough memory to do this, resulting in an *Out of memory* message. If you have enough memory on your system, and you want all the output on one page, the format document need not end with a page break.

Header and Footer Pages

Header and footer pages from the format document are included in the output document. Do not use the number symbol (#) for page numbering on a header or footer page in a format document unless you want each page in the output document to have a different page number.

Suppose you have a number symbol (#) on a footer page in a format document that contains a one-page form letter. When you process that format document with a list document that has 23 records, you will generate an output document containing 23 letters. In this case, each distinct letter will be printed with a page number between 1 and 23.

Date and Time Field Labels

There are three special field labels you can use in the format document without putting corresponding field labels in the list document. These field labels use the system clock to generate the date or time in the output document. Type these special labels exactly as shown below. Remember to use the Merge and Shift/Merge keys to begin and end each field label.

<date1>   The system date, in the form *Sep 3, 1986* (month day year), is placed in the output document at the location of the <date1> field label. On European systems, the date is displayed in the format *3 Sep 1986* (day month year).

<time1>   The system time, in the form *10:08 am*, is placed in the output document at the location of the <time1> field label. On European systems, the abbreviation for before or after noon is translated into the selected language.

<time2>   The system time, in the form *1008* and using 24-hour notation, is placed in the output document at the location of the <time2> field label.

> **WARNING:** Records Processing always uses these three field labels to put the system date or time into the output document. If you use one of these labels as a field label in your list document, the field contents will never appear in the output document. If you use date or time field labels to enter dates and times in the list document, do not use these field label names.

If you want a specific date or time to appear in a different style in the output document, you can always type it directly into the format document.

Repeat Statements

Ordinarily, when you type field labels in the format document, each record is merged separately onto a new page in the output document. Sometimes you will want to merge more than one record onto the same page of the format document. To do this you use a **repeat** statement.

Refer to "The Repeat Feature" later in this chapter for information on how to use this feature.

Include Statements

An *include statement* is a command used to place the entire contents of a Fortune:Word document at the location you indicate in the format document. You can use include statements to assemble documents without using a list document. Alternatively, you can use include statements with a list document to generate a series of reports.

Refer to "The Document Include Feature" later in this chapter for information on how to use this feature.

CREATING AND USING THE FORMAT DOCUMENT

You create the format document as you would any Fortune:Word document by using the Create New Document selection on the Fortune:Word Main menu. You can perform all the standard editing functions on a format document.

When you have finished creating the format document, you merge it with the list document by using the Records Processing selection on the Document Processing Tools menu. The system merges each record in the list document with the standard text in the format document, creating an output document in the process. The result is an output document containing individualized letters for each member on the list.

The complete contents of the field in the list document (including spaces, punctuation marks, and screen symbols) are merged into the format document at each field-label position. For example, when you type an address in the format document, the line containing the city, state, and zip code could look like this:

> <city>, <state> <zipcode>

Note that the comma and space between the city and state are part of the format document, as are the two spaces between the state and the zip code.

When you type a salutation in the format document, the line could look like this:

> Dear <honorific> <last name>:

Note that you enter the colon that follows the salutation into the format document, with no space between the end of the field label and the punctuation mark.

When you use the Records Processing menu to merge a format document and a list document, the field labels in the format document are always replaced by the field contents from the list document. In this case, the field labels never appear in the output document.

REMEMBER: The field labels in the format document must match the field labels in the list document.

HOW TO CREATE THE FORMAT DOCUMENT

Here are some rules to follow when creating the format document:

- Include spaces and punctuation marks where required around field labels.

- Begin each field label with a Merge symbol and end it with a Shift/Merge symbol.

- Be sure the field labels you use in the format document match the field labels in the list document.

- Do not include any end-of-record indicators in the format document. If you do, you will see the error message *Label not in list document*, or *Format doc. format error* when you run Records Processing.

- Header and footer pages from the format document are used to create the output document. Do not include a number symbol (#) on a header or footer page unless you specifically want a different page number on each page of the output document.

- Put a page break at the end of the format document to begin each new copy of the format document on a new page in the output document. If you have a long list document and do not put a page break at the end of a format document, you may create an output document too big for system memory.

- The format lines in the format document determine the format for the output document.

- Do not put a return symbol within a field label. If a field label is broken into two lines by a return symbol, that symbol will be interpreted as a space. For example, if the label <newaddress> is broken into two words with a return symbol in between, it will be interpreted as <new address>. When you run Records Processing, you will see the error message *Label not in list document*.

Here are some suggestions for additional ways you can use the format document:

- If you want, you can use the same field label more than once in a format document.

- You do not need to use every field label in the list document in your format document.

- In addition to the page break at the end of the format document, you can insert page breaks at other places if you want a multipage format in the output document.

- These three special labels can be used in the format document without a corresponding label in the list document:

        <date1>
        <time1>
        <time2>

- You can use a repeat statement to create report formats where several records per page are merged into the output document. Refer to the section later in this chapter, "The Repeat Feature" for more information.

- You can use include statements to include the entire contents of other Fortune:Word documents in the output document. Format lines are NOT copied from the included document. You must have the correct format line in the format document. Refer to the section later in this chapter, "The Document Include Feature" for more information.

CREATING A FORM LETTER AND USING RECORDS PROCESSING

In this exercise, you will create a format document that is typical of form letters used in mass mailings. You will use this format document with the list document **add.list** that you created in the exercise in Chapter 2. Figure 3-1 shows the format document you will create.

```
Doc add.form    Page 1    Line 21   Pos 53
word Format 1 Spacing 1 Length 54
1(1).....1.........2.........3........▶4.........5.........6....◀
                                 ▶<date1>◀

◀
<first name> <last name>◀
<address>◀
<city>, <state> <zipcode>◀
◀
Dear <honorific> <last name>:◀
◀
Thank you for joining the International Society to Save the
Skeet.  As you know, skeet are on the endangered species list.
Your contributions are necessary for us to continue in our
campaign to stop the indiscriminate shooting of skeet.◀
◀
We look forward to your continued support.  As a member, you
will receive our newsletter keeping you informed of our
progress.◀
◀
                                  ▶Sincerely,◀
◀
◀
                                  ▶Fred Simmons◀
```

A2387

```
Doc add.form    Page 1    Line 19  Pos 1
word Format 1 Spacing 1 Length 54
+(1).....1.........2.........3........▶4.........5.........6....◀
◀
◀
                                  ▶Fred Simmons◀
                                  ▶Public Relations◀
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1(1).....1.........2.........3........▶4.........5.........6....◀
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =




(Document End)
```

A2388

Figure 3-1. A Typical Form-Letter Format Document

For your convenience, the text of the form letter is repeated below:

<center><date1></center>

<first name> <last name>
<address>
<city>, <state>   <zipcode>

Dear <honorific> <last name>:

Thank you for joining the International Society to Save the Skeet. As you know, skeet are on the endangered species list. Your contributions are necessary for us to continue in our campaign to stop the indiscriminate shooting of skeet.

We look forward to your continued support. As a member, you will receive our newsletter keeping you informed of our progress.

<div align="center">Sincerely,</div>

<div align="center">Fred Simmons
Public Relations</div>

Building the Format Document

When creating the format document yourself, type the standard text that is to appear in the output document. As you type, when you come to a place where there will be variable information, type the field label. Surround it with appropriate spacing and punctuation as if the field label was the actual information to be merged from the list document.

Follow these steps to create the format document for the form letter that you will use with the list document **add.list** you created in the exercise in Chapter 2.

1.   Create a new Fortune:Word document named **add.form**.

2.   Type the letter shown in Figure 3-1, including the field labels.
     Follow the spacing and punctuation between blank lines exactly as it appears in the letter.

REMEMBER: Use MERGE and SHIFT/MERGE symbols to enclose the field labels in the format document.

Notice that the field label <date1> is used in place of the actual date.

Be sure you put an optional page break at the end of the format document.

3.   When you have finished typing the form letter, leave the format document, saving the changes, and go back to the Fortune:Word Main menu.

Now that you have created a list document and a format document, you will use the Records Processing menu to merge them into an output document.

1.   Select *Document Processing Tools* from the Fortune:Word Main menu.

2.   Select *Records Processing* from the Document Processing Tools menu.

3.   Fill in the Records Processing menu as follows:

   a.   Type **add.list** in the *List document* field, and press RETURN to go to the next line.

   b.   Type **add.form** in the *Format document* field, and press EXECUTE.

   c.   For this exercise, you want to view the resulting output document before printing it. Leave *Make output document only* highlighted and press EXECUTE.

   d.   When you see the message *Please enter output document name*, type **add.out** and press EXECUTE. Note the prompts on the lower part of the screen. They tell you which record is currently being merged.

         NOTE: Be sure you enter the names of the list and format documents in the correct places on the Records Processing menu. If you put the names on the wrong lines, Records Processing will not work correctly.

4.   When the Document Processing Tools menu reappears, the system has completed the Records Processing procedure. Press CANCEL to go back to the Fortune:Word Main menu.

5.   Use the *Edit Old Document* selection to look at the newly created output document, **add.out**, observing the following information on the Document Summary screen:

- Note that, except for the document name, the information at the top of the Document Summary screen is the same as for the format document.

- The statistics on the lower part of the screen show the date and time the output document was created, and the number of pages, lines, and characters in the output document.

- The work time and number of keystrokes are the same as for the format document.

In the output document, each letter is on a separate page and contains the address and other information merged from the list document into the standard text from the format document. You can edit the output document to make changes to the letters before you print them. If the results are not what you expect, you can revise your format or list documents and start again, overwriting the output document with the new output results.

6.   You can print the document **add.out** as you would print a regular Fortune:Word document.

THE REPEAT FEATURE

Ordinarily, when you type field labels in the format document, each record is merged separately onto a new page in the output document. Sometimes you will want to merge more than one record onto the same page of the format document. To do this you use a repeat statement.

You can use the repeat statement in two different ways:

- To repeat different field labels vertically
- To repeat the same field labels horizontally

Repeating Different Field Labels Vertically

To repeat different field labels vertically use the following syntax, where *n* is the number of times on each page that the field labels in the repeat statement are repeated, and <fielda>, <fieldb>, <fieldc>, and <fieldd> are different field labels within a record.

```
<repeat(n)
<fielda>        <fieldb>        <fieldc>        <fieldd>
>
```

For example, suppose you want to use the information in the **parts.list** document you created in Chapter 2 to generate a table in your output document. The format document can look like the one below.

```
<repeat(5)
<part no>       <type>      <in stock>          <on order>
>
```

When you merge the format and list documents, the first line in the output document contains the field contents from record one in the list document, the second line contains the contents from record two, and so on. The first five records appear on the first page of the output document, the second five records on page two, and so on.

An exercise later in this section shows how to use a repeat statement to create a report, complete with table headings, from the list document **parts.list**.

Repeating the Same Field Labels Horizontally

If you want to repeat a record horizontally across the page use a repeat statement with the syntax shown below. With this syntax, four records are repeated across the page in each column. The number between the field label name and the SHIFT/MERGE symbol indicates the column position in which the field contents appear in the output document. You must add a comma and the number of the column at the end of each field label to instruct Records Processing that a different record is to be merged at each location.

```
<repeat(n)
<fielda,1>      <fielda,2>      <fielda,3>      <fielda,4>
<fieldb,1>      <fieldb,2>      <fieldb,3>      <fieldb,4>
<fieldc,1>      <fieldc,2>      <fieldc,3>      <fieldc,4>
<fieldd,1>      <fieldd,2>      <fieldd,3>      <fieldd,4>


>
```

To print addresses on side-by-side labels, the format document can look like the example below:

```
<repeat(10)
<first name,1>  <last name,1>          <first name,2>  <last name,2>
<title,1>                              <title,2>
<company,1>                            <company,2>
<city,1>,  <state,1>  <zipcode,1>      <city,2>,  <state,2>  <zipcode,2>

>
```

Using this format document, each page in the output document contains 20 addresses. The first column contains records 1, 3, 5, 7, 9, 11, 13, 15, 17, and 19. The second column contains records 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20.

> **NOTE:** Do NOT add the column numbers to the field labels on the list document. The column numbers appear only in the format document to tell Records Processing how to position the records on a page.

For further examples of how to use the repeat feature, refer to the section entitled "Using the Repeat Feature for an Inventory Report" later in this chapter and to Chapter 4, "Special Format Documents."

These are the procedures to follow when you use the repeat feature:

- The repeat statement must begin with a MERGE and end with a SHIFT/MERGE. Press MERGE to create the left-merge symbol that begins the repeat statement. Type **repeat(n)**, where *n* is the number of times you want the information repeated on each page. Press RETURN. Type the field labels, and press RETURN to end the last line. End the repeat statement with a SHIFT/MERGE, and press RETURN.

- If you want to include column headings on each page of the output document, you can type them once in the format document. You may want to use different format lines for the headings and the field labels to center the information correctly in the columns.

- Check to be sure the format line has tabs in the correct positions. You can extend the format line to make a wide document if you wish. If you use tabs, the first character of the field contents prints where the field-beginning indicator (left-merge symbol) is placed in the format document.

- To determine tab placement, scan your list document to see how much room to allow for field contents.

- If you use decimal tabs before the field labels, use different format lines for the headings and the field labels. When you use decimal tabs, the contents of each field align at the tab position on the decimal or on the last character if there is no decimal.

- You should line up columns in the format document by using tabs or decimal tabs between field names instead of spaces. If you use spaces between the field labels in the format document, the columns in the output document will not line up correctly unless the field contents in each record have the same number of characters. Be sure to remove all unnecessary tabs from the format line.

- Insert a page break at the end of the format document so that the next group of records will begin on the next page.

Producing an Inventory Report Using Repeat

In this exercise, you will produce an inventory report using the list document **parts.list** that you created in the previous chapter. First, you must create the format document. Follow these steps to use the repeat feature in a format document:

1.   Create a new Fortune:Word document named **parts.form**.

2.   Set tabs in the format line at positions 20, 45, and 65, deleting all other tab stops. Extend the right margin to position 80.

3.   Set up the headings for the report and the beginning of the repeat statement as follows using spaces between the column headings:

       PART NUMBER   TYPE   IN STOCK   ON ORDER
       <repeat(5)

4.   Type the first field label at the left margin. Put a TAB in front of each of the other field labels, so that each is below its column heading. Press RETURN, SHIFT/MERGE, and RETURN to enter a field-ending symbol to finish the repeat statement. The result should look like this:

       PART NUMBER   TYPE   IN STOCK   ON ORDER
       <repeat(5)
       <part no>   <type>   <in stock>   <on order>
       >

5.   Enter a page break. When you have finished, save the document and return to the Fortune:Word Main menu.

6.   Select *Document Processing Tools* from the Fortune:Word Main menu. Then select *Records Processing* from the Document Processing Tools menu. This time, you will print the output document instead of just examining it on the screen.

7.   Make the following selections from the Records Processing menu:

   a.   Type **parts.list** for the list document, and press RETURN.

   b.   Type **parts.form** for the format document, and press RETURN three times.

   c.   Press SPACE to move the marker to *Make output document and print*, and press EXECUTE.

   d.   Type **parts.out** and press RETURN or EXECUTE to initiate Records Processing.

8.   When the merging has been completed, the output document begins printing. You will also see a copy of **parts.out** on the screen. Note that each page of the output document is set up in columns with five lines of information.

9.   Edit the format document **parts.form**, changing the number in parentheses following the word **repeat** to 10.

10. Select *Records Processing* again to see how the change in the format document affects the output.

THE DOCUMENT INCLUDE FEATURE

An *include statement* is a command used to place the entire contents of a Fortune:Word document at the location you indicate in the format document. You can use include statements to assemble documents without using a list document. Alternatively, you can use include statements with a list document to generate a series of reports.

The syntax for the include statement is:

   <include = *document*>

The include statement must begin with a MERGE and end with a SHIFT/MERGE. If the document to be included is in a different library, use a full pathname following the equal sign. This statement causes the text from the editing screen of the Fortune:Word document *document* to be merged with the format document at the place indicated. Page breaks are not copied. Text from header, footer, work, and footnote pages are not included. If the document to be included requires a different format line, that format line must be placed in the format document. Refer to the *Fortune:Word Reference Guide* for detailed information on how to create and use format lines.

You can include the same document more than one time in a format document if you want to repeat the information. The text from the included document is copied into the output document. The source include document is not changed when you use it in this way.

To create an output document that includes the text from several other documents, all you need is a format document and an output-document name.

> NOTE: You cannot use an include argument within a repeat statement.

Using Include Statements for Document Assembly

You can use include statements for document assembly. For example, a law office can create a series of documents, each containing a paragraph of standard legal text. The documents might be named by standard paragraph numbers, such as **para.1a, para.1b, para.2a, para.3a**.

> NOTE: The syntax and requirements for using the
> *Document Assembly* selection from the Document
> Processing Tools menu are exactly the same as using a
> Records Processing format document that does not contain
> any field labels.

When a legal document is prepared, the variable information can be typed directly into the format document. When a standard paragraph is called for, you type an include statement, using the syntax shown above. For example, <include = para.1b>. You can immediately type another include statement to include another paragraph, such as <include = para.3a>, or continue to type new text, including another standard paragraph later in the document. The documents can be called up in any order, and you can use the name of a document more than once if you want to repeat the information.

In this way you create a format document that you use with Records Processing to automatically combine frequently used paragraphs with new text. When you use the document assembly feature in this way, with all of the include statements in one format document, there is no limit to the number of documents you can include, as long as your system has enough memory.

> REMEMBER: The entire body of text from a document named
> in an include statement appears in the output document.
> You cannot use the include statement to assemble
> different parts of the same document.

You can create three kinds of format documents using include statements:

- Documents that contain include statements only

- Documents that contain both text and include statements

- Documents that contain text, include statements, and field labels
  when you want to combine the format document with a list document

> **REMEMBER:** When you use include statements, format
> lines, page breaks, header, footer, work, and
> footnote pages are NOT transferred along with a
> document.

Suppose you have put field labels in a Fortune:Word document and then use
an include statement to incorporate the document into a Records Processing
format document. In this case, the field labels are treated as regular
text and will appear in the output document. If you want, you can
construct the included documents so that they contain field labels around
key information such as <county name> or <plaintiff's name>. Once
the full document has been assembled, you can use it as a format document
to merge those fields from a list document.

> **NOTE:** The only time you can use a format document
> without a list document from the Records Processing menu
> is when the format document contains include statements
> and does not contain any field labels.

Using Nested Include Statements

In addition to putting include statements in a format document, you can
put include statements in Fortune:Word documents that are themselves
named in include statements. All the documents will then be combined
automatically before they are merged with the format document and placed
in the output document. This technique is called *nesting*.

You can create up to 16 levels of nested include statements in one format
document. The format document itself must contain at least one include
statement to indicate to Records Processing how the other documents are to
be assembled. There is no limit to the number of Fortune:Word documents
you can merge with include statements that are on the same level (that is,
not nested).

Each of the documents that you name in a succession of include statements
on the same level might have several include statements within it; these
levels are counted separately in determining the maximum levels of
nesting. Figure 3-2 shows one possible way you can use nested include
statements. If you exceed the maximum number of levels of nesting, you
will get an error message. Each level of nesting must have at least one
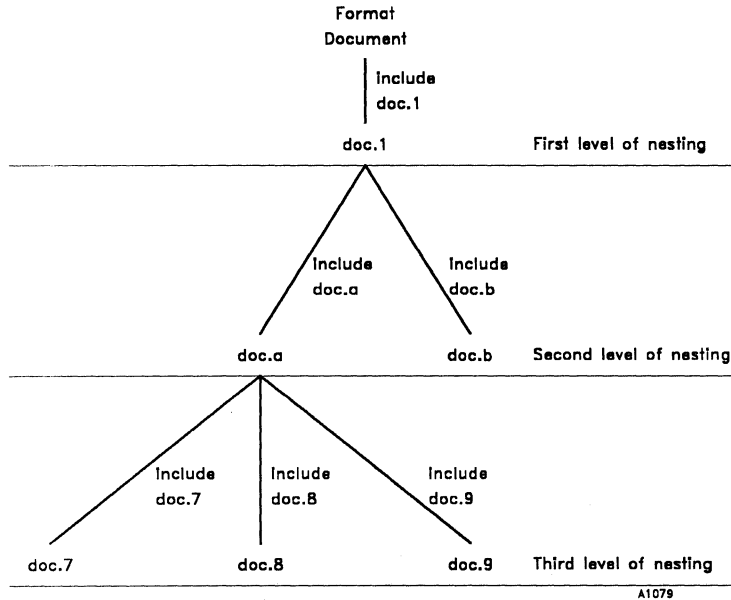include statement that leads to the next level.

```
                          Format
                         Document
                              │  Include
                              │  doc.1

                          doc.1                    First level of nesting
   ──────────────────────────┴─────────────────────────────────────
                             ╱ ╲
                            ╱   ╲
                      Include     Include
                      doc.a       doc.b
                     ╱               ╲
                  doc.a               doc.b       Second level of nesting
   ──────────────────────────────────┴──────────────────────────────
                        ╱   │   ╲
                 Include  Include  Include
                 doc.7    doc.8    doc.9
               ╱            │         ╲
           doc.7          doc.8        doc.9      Third level of nesting
   ─────────────────────────────────────────────────────────────────
                                              A1079
```

Figure 3-2. *Up to 16 levels of nested include statements are allowed in Records Processing*

> **REMEMBER:** When you use an include statement to merge a document, the *total* text of the document is copied into the resulting output document. The format of such a document is governed by the preceding format line in the format document.

If you want to use more than one level of nesting, type include statements in any of the Fortune:Word documents that are named in the format document. When you do this, all the documents named are automatically assembled before they are placed in the output document. You can use up to 16 levels of nesting.

Procedures for Using an Include Statement

These are the procedures to follow to use an include statement in a format document:

1.   Type the standard text up to the place where you want to put the included document.

2. Check to be sure that the current format line in the format document is correct for the information you want to copy from the other document. The format line from the included document is *not* copied when the documents are merged.

3. Press MERGE.

4. Type **include** = *document*, where **document** is the name of the document you want to include. You can type the name of a document from the same library, or you can use a pathname.

5. Press SHIFT/MERGE. This is an example of an include statement:

    <include = 1985stats>

6. Continue to type the text in the format document until it is complete. If you want to add text from other documents, type the include statements at the appropriate places in the format document.

Producing a Memo Using Include

You can use document assembly to combine many Fortune:Word documents. You can also use document assembly to copy one document, such as a table or report, into a format document. The following exercise uses the output document **parts.out**, which you created in the previous exercise.

> **NOTE:** Before you begin this exercise, edit the document **parts.out**, so that it looks like Figure 3-4. All the text should be on one page, and the document should contain no page breaks.

1. Create a new document, and name it **parts.inc**.

2. Type the memo shown in Figure 3-3. Be sure you include a format line that has tabs at positions 20, 45, and 65 and a right margin at 80 immediately above the include statement.

3. Select *Document Processing Tools* from the Fortune:Word Main menu.

4. Select *Records Processing* from the Document Processing Tools menu.

```
Doc parts.inc   Page 1    Line 1    Pos 1
word Format 1 Spacing 1 Length 54
+(1).....▶...▶....▶2...▶....▶3...▶....▶4...▶....▶5...▶....▶6.........7.........◀
To:       ▶Randall Smith, Vice President◀
◀
From:     ▶Donna Harrison, Manager, Inventory Control◀
◀
Date:     ▶<date1>◀
◀
Subject: ▶Month-End Stock Count◀
◀
_____◀
◀
The following list reflects the month-end inventory count.  The number amount
listed under ON ORDER will be restocked unless you specify otherwise.  Please
reply within five days.◀
◀
1(2).....1......... ▶.........3......... 4....▶....5.........6....▶....7.........◀
<include = parts.out>◀
◀
1(1).....▶...▶....▶2...▶....▶3...▶....▶4...▶....▶5...▶....▶6.........7.........◀
Parts with a zero in the ON ORDER column will not be restocked.◀
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

(Document End)                                                           A2589
```

Figure 3-3. Format document with include statement

5.   Press RETURN to move the cursor from the *List document* line to the
     Format document selection.

6.   Type **parts.inc**, and press EXECUTE. The document that will be
     included **parts.out** is shown in Figure 3-4.

7.   Leave the *Make output document* selection highlighted, type **inc.out**
     for the name of the output document, and press EXECUTE.

8.   When Records Processing is done, look at **inc.out**. It should look
     like the example in Figure 3-5.

     NOTE: You never type the name of an included document on
     the Records Processing menu. The document name(s) only
     appear in the format document for the first level of
     include, and in the documents to be nested if you use
     more than one level of nesting.

```
Doc parts.out   Page 1   Line 1   Pos 1
word Format 2 Spacing 1 Length 54
+(2).....1......... ▶.........3......... 4....▶....5.........6....▶....7.........◀
PART NUM            TYPE              IN STK            ON ORD◀
◀
T8991Z          ▶claw hammers        ▶29               ▶2◀
78              ▶phillips screwdrivers ▶30             ▶0◀
F890            ▶needle-nose pliers   ▶12              ▶10◀
STY321909       ▶12-piece drills      ▶8               ▶12◀
U21             ▶tile knife           ▶7               ▶5◀
L1908416J       ▶angle brackets       ▶49              ▶20◀
H892            ▶door springs         ▶2               ▶30◀
E7416           ▶tool box             ▶12              ▶12◀
T12             ▶ratchet wrench       ▶4               ▶6◀
W090189         ▶pliers               ▶21              ▶0◀
◀

= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =




(Document End)
```

A2390

Figure 3-4. The document to be included

```
To:      Randall Smith, Vice President

From:    Donna Harrison, Manager, Inventory Control

Date:    April 1, 1986

Subject: Month-End Stock Count

----------------------------------------------------------------------------

The following list reflects the month-end inventory count.  The number amount
listed under ON ORDER will be restocked unless you specify otherwise.  Please
reply within five days.

PART NUM              TYPE                IN STK             ON ORD

T8991Z                claw hammers        29                    2
78                    phillips screwdrivers  30                 0
F890                  needle-nose pliers  12                   10
STY321909             12-piece drills     8                    12
U21                   tile knife          7                     5
L1908416J             angle brackets      49                   20
H892                  door springs        2                    30
E7416                 tool box            12                   12
T12                   ratchet wrench      4                     6
W090189               pliers              21                    0

Parts with a zero in the ON ORDER column will not be restocked.
```

Figure 3-5. The final output document, with the document parts.out included

Chapter 4
Creating Format Documents for Address Labels, Envelopes, and File Cards

In Chapter 3 you learned how to create format documents for form letters and reports and how to use the include and repeat features.

You can use many different format documents with the same list document as long as the field labels match. Once you have created a form letter for a mass mailing, you will want to use the same list document to print labels or envelopes for each letter. You might also want to print the information on file cards to help you track the response to the mailing more easily. These three additional types of format documents are described in this chapter:

- Mailing labels
- Envelopes
- File cards

MAILING LABELS

Have you ever had to type the same mailing labels over and over again? If you have, you will be glad to know that once you have set up an appropriate Records Processing format document, you can create mailing labels easily and correctly every time.

Standard mailing labels are available from stationery suppliers on tractor-fed, continuous-form paper with labels placed one, two, three, or more across and in sizes from 1 inch deep and 3 inches across to 2 inches deep and 4 inches across. Of course, you can also have labels custom-made. Use the procedure suggested below to help you determine which kind to purchase. Be sure to check your printer manual to determine the widest paper you can use in your printer before you order labels.

Determining the Label Size to Purchase

If you do not already have continuous-form labels, follow these guidelines to determine the size you need to order.

1. Determine the number of lines you want to print on a label. If you have printed your list document, you can find this out quite easily.

2. Most standard labels have a spacing of 6 lines per inch down the page. The number of lines you want to print divided by 6 lines per inch tells you how deep the label should be in inches. For instance, if you want to print 10 lines on a label, then at 6 lines per inch your label should be at least 1 2/3 inches deep. Be sure to allow room for margins around the printed information for a professional look.

3. Now you need to determine how wide the labels should be. At a pitch of 10, you can fit 10 characters in one inch across the label. Go back and look at the list document, and determine how long the longest line of information is that you want to print. If the longest line is 40 characters, then dividing that by 10 gives you 4 inches plus the left and right margins. You should probably get a label 4 1/2 inches wide.

4. Using these calculations, determine which size of label you need to purchase.

> **NOTE:** You can change the number of lines and characters that will be printed on a label by modifying the lines-per-inch and pitch settings on the Print Document menu.

Determining Vertical Layout

Once you have obtained the labels, you need to determine how to set up the format document. Start with the vertical placement down the page of labels. Take out the notes you used to determine which size labels to buy. Also get a sheet of labels if they are 2-, 3-, or 4-across or several sheets if they are 1-across.

1. Lay the labels flat. When you feed them into the printer, you should set up the first label so that the printer will begin printing the first line at the correct place. Therefore, you do not need to calculate a top margin, but rather the distance between the labels horizontally and vertically.

2.   Place a mark on the first label where the upper-left corner of the
     first character of the first name should print, as shown in
     Figure 4-1.

3.   Now, go to the next label down and mark where the upper-left corner
     of the first character of the second name should print. Measure the
     distance between the two marks, as shown in Figure 4-2.

     In this example, the distance is 1 inch. At 6 lines per inch, this
     is equivalent to six lines. Only 4 lines from the list document will
     be printed on the label. Subtracting 4 lines from 6 lines, there are
     2 lines left over to create the bottom margin of one label and the
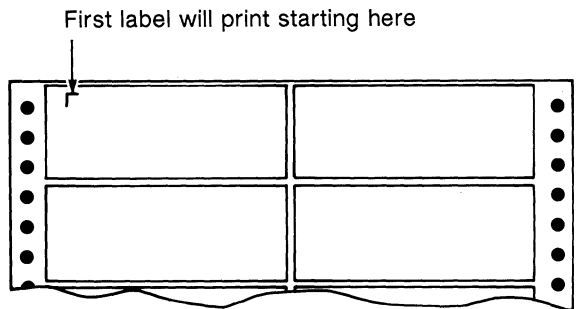   top margin of the next.

First label will print starting here

Figure 4-1. Mark the position where printing will start on the first label

Measure this distance

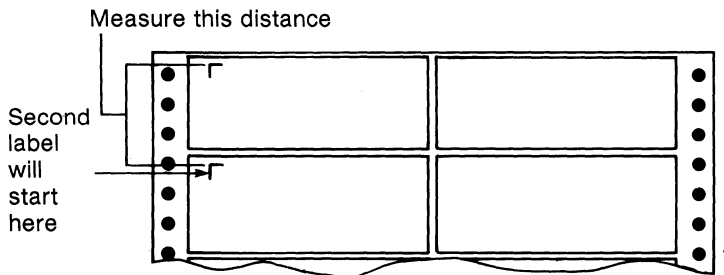Second
label
will
start
here

Figure 4-2. Measuring the distance between labels

When you create the format document, it should have the field labels from the list document followed by the two extra returns, a SHIFT/MERGE symbol, and another RETURN. This ending symbol determines when Records Processing should look for another record from the list document. If you are printing only one label across, you do not need to put column numbers after the field names.

```
<repeat(10)
<name>
<company>
<address>
<city>, <state> <zipcode>

>
```

SUGGESTION: If you have addresses that contain different numbers of lines, you can set up the format document to print one address on a page. In this case you would type just the field labels. You do not need to use a repeat statement. Then you should change the page length on the Print Document menu so that it matches the number of lines on each label. That way, the first line of an address will always print in the same location on each label. For example, a 2-inch label printed at 6 lines per inch would have a page length of 12 lines.

Determining Horizontal Layout

If you are using 1-across labels, you do not have to be concerned with horizontal placement. However, if you use any other type, you must also set up the format for the labels across the sheet.

REMEMBER: The leftmost merge symbol for a field label (but not for the repeat statement) determines where the first character of information from the list document will print.

1. Using the same sheet of labels you used to determine the vertical placement, put a corner mark where the first character will print on the second label to the right.

2. Measure the distance between the marks on the first and second labels. Then divide this figure by 10 for 10 pitch. If the distance is 3 inches, this means that there will be 30 characters between the beginning character of one label and the beginning character of the label next to it. This is where the calculation you did for the label size is most crucial. If your longest field has 60 characters in it, then it obviously will not fit on 3-inch labels. Figure 4-3 shows how to mark and measure the distance between horizontal labels.

3. In calculating the horizontal measurement, remember that you will line up the leftmost edge of the label (not of the label backing) as position 0 on your printer. Therefore, you will probably want to print the first label starting at column 2. Consequently, you should specify this as the left margin on the Print Document menu. If you want 30 characters between labels, remove all the tabs from the format line except for one at position 32. If the labels are 3-across, put a second tab in the format line at position 62. Use the Tab key to put the field labels in the correct positions.

4. Now that you have the spacing and the format line set up correctly, type a repeat statement, and the field labels. For a format with two labels across, the finished format document should look like the example below.

    When you have more than one label across a page, you must add a comma and the column number between the end of the field label name and the SHIFT/MERGE symbol, as shown in the example. Type ,1 before the SHIFT/MERGE symbol for each field label in column 1. Type ,2 before the SHIFT/MERGE symbol for each field label in column 2, and so on. Do not add these numbers to the field labels in the list document. Since this format row is going to repeat for each row of labels, you do not need to specify more than one repeat.
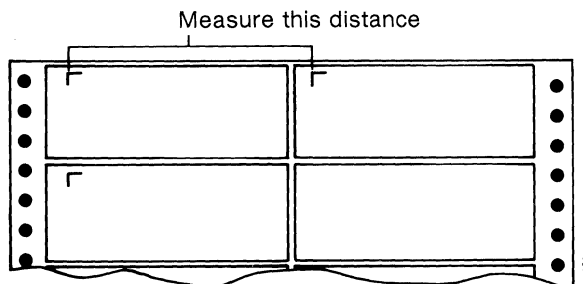
Measure this distance

Figure 4-3. Marking and measuring the distance between horizontal labels

```
<repeat(1)
<name,1>                            <name,2>
<address,1>                         <address,2>
<city,1>, <state,1> <zip,1>         <city,2>, <state,2> <zip,2>


>
```

Printing Labels

You may want to test your label layout by printing on paper first.

1. Load your printer using a forms tractor and paper wide enough to accommodate the printing. Measure your sheets of labels and mark on the first sheet of paper that you print where the edges of the labels would be. Load the paper so that the left edge of the first label across is at position 0.

2. Create a new document, and copy enough records into it from the list document to equal at least one sheet of labels. You do not need to print the whole list to determine if the spacing is correct.

3. Using the Records Processing menu, indicate the name of this short list document, the format document, and an output document. Make sure your paper is properly aligned before you start. Choose the *Print only* option from the menu, and press EXECUTE.

4. As the names print in label format, check to see how well they line up with your marks on the paper. When the printing is finished, analyze it carefully. If the spacing is not correct, check your measurements. Then check the locations of the tabs in the format document. Be sure you have removed any unneeded tabs. Make any other changes to the format document as necessary.

Hints on Printing Labels

*   Read your printer instructions carefully before you load and print labels. Labels attached to a backing are thicker than ordinary paper. You may need to adjust your printer platen and rollers to accommodate this extra thickness. Otherwise, your labels might peel off the backing and get stuck beneath the platen. The printer setting for 3 or 4 carbon copies usually allows sufficient room. Try this setting by rolling some labels through on the forms tractor before you begin printing.

- You can get more information onto a label by changing the lines-per-inch and pitch settings on the Print Document menu when you print the labels. Be sure to allow for these changes when you calculate the positioning in the format document. Try printing a sample on paper before you use labels.

- If your labels do not print correctly, or if something gets jammed, use the Printer Control feature to stop the printing. If you are only partly done with the list, you may want to start over. If only a few labels did not print correctly, copy those records into a smaller list document and print them separately.

- If you have to reprint labels, be sure to reposition the labels correctly in the printer.

Summary

The procedure for creating labels is summarized below:

1. Analyze your list document to determine what size of labels you want to purchase.

2. Set up the format document.

3. Copy some records from the list document into a smaller document to use for testing the format.

4. Load some paper into your printer, and print the short list document using the Records Processing menu.

5. Make any necessary adjustments to the format document or Print Document menu.

6. Try the printing again.

7. If the printing on paper is successful, follow the directions in your printer manual for adjusting the printer to accommodate the labels.

8. Print the labels.

ENVELOPES

If you want to print envelopes for your form letters, you can use the original list document to print the name and address information on labels and then attach the labels to the envelopes. Alternatively, you can print the address list directly on envelopes. To print on envelopes, you need either:

- Envelopes set up as continuous forms
- Single envelopes that you will feed individually

Many forms suppliers have custom-made envelopes that either come with tear-off strips like continuous-forms paper or are attached to a backing like mailing labels. Figure 4-4 shows these two kinds of continuous-form envelopes. If you do not have such envelopes, you will have to feed envelopes one by one or use labels.

Continuous-Form Envelopes

No matter what kind of continuous-form envelope you are using, the key factor is the format document. If you have already set up a format document for mailing labels, then you will find the format document for envelopes even easier. That is because only one envelope is printed at a time.

1. Determine which fields you want printed on the envelope. Set up the format document with the repeat feature, and give it a name that you will recognize, for example, **env.form**. A sample format is shown below. Note that since envelopes only come one-across, the repeat is 1 and the field labels do not have column numbers.

   ```
   <repeat(1)
   <name>
   <company>
   <address>
   <city>, <state> <zipcode>
   >
   ```

2. You need to calculate the number of lines from the first line of one name and address to the first line of the following one on the next envelope. Measure the distance in inches between the lines just as you would for labels. Multiply that figure by the number of lines per inch you will set on the Print Document menu. If the distance is 4 inches and you are printing 6 lines per inch, the number of lines would be 24. Subtract from that amount the total number of lines in one name and address. The difference is equal to the number of returns you must add after the name and address in the format document. Continuing the example, if the name and address take 4 lines, then you should add 20 returns before ending the repeat. Your format document is ready.

3. Try printing a sample on paper before you print on envelopes. Then compare it against the envelopes to see if the spacing is correct.
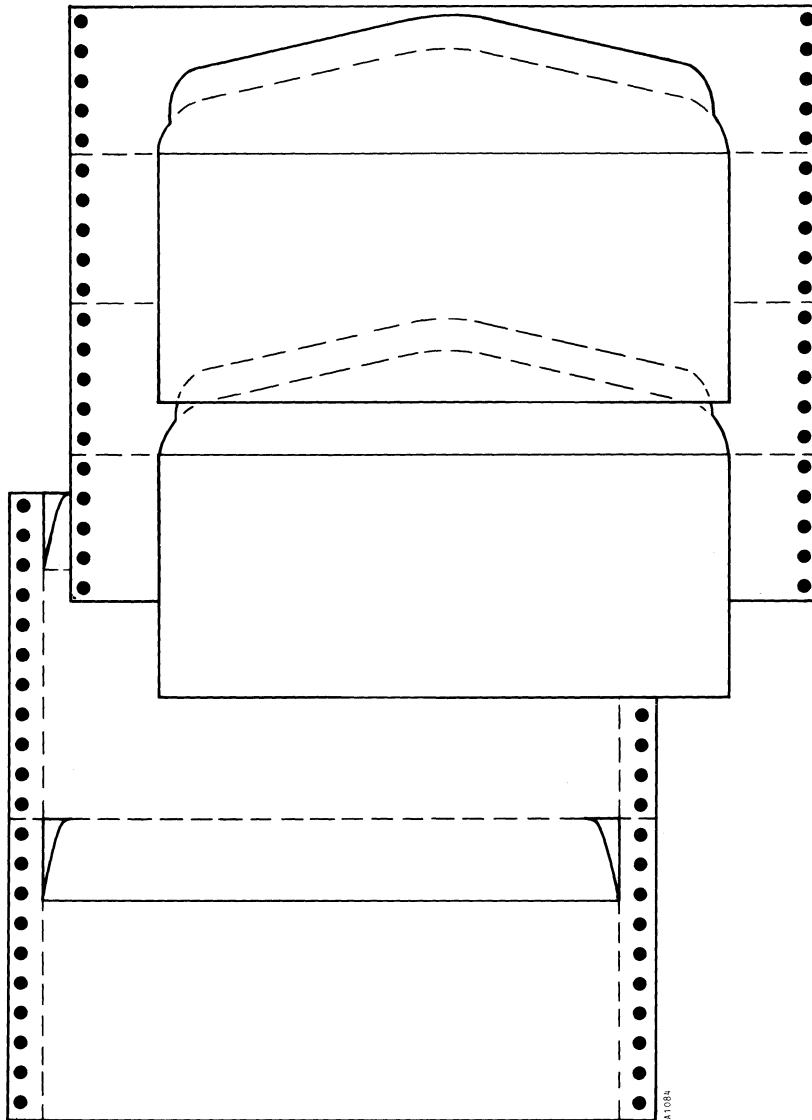
Figure 4-4.   Continuous-form envelopes

Refer to the discussion on "Determining Vertical Layout for Labels"
earlier in this chapter for more information if you have difficulty
determining the proper vertical spacing for envelopes.

Single Envelopes

If you will be feeding envelopes individually, set up the envelope format document with an optional page break at the end. You must change the *Feed* selection on the Print Document menu to *Sheet*. There are two ways you can change the Print Document settings for Records Processing documents:

- You can send the format document to the printer, making the necessary changes on the Print Document menu. You must send the document to the printer to register the changes, but you do not have to print the entire document. Once the document has been sent to the printer, the changes on the Print Document menu are recorded. Any output document that is generated by Records Processing will have these Print Document settings.

- You can generate an output document with Records Processing and change the Print Document settings when you send that output document to the printer.

When the *Feed* selection is set to *Sheet*, the printer stops after each envelope is printed so that you can remove it and insert a new one. When the new envelope is properly positioned, from the Printer Control screen press GO TO PAGE to print the next envelope. If you are not familiar with how to print single sheets, refer to the *Fortune:Word Reference Guide* for more information.

FILE CARDS

You may have a card file on your desk with the names, addresses, and phone numbers of people you contact regularly. You may also have found that, after a while, the list becomes outdated and you must continually revise it. If you have incorporated your address list into a list document for other purposes such as form letters, you can print the same information on cards. Forms suppliers offer both regular and rotary-style continuous-forms cards in 1- to 3-across formats and in 3-by-5- or 4-by-6-inch sizes. Examples of these file cards are shown in Figure 4-5. Before you set up a format document to print file cards, decide what size cards you want and purchase some.
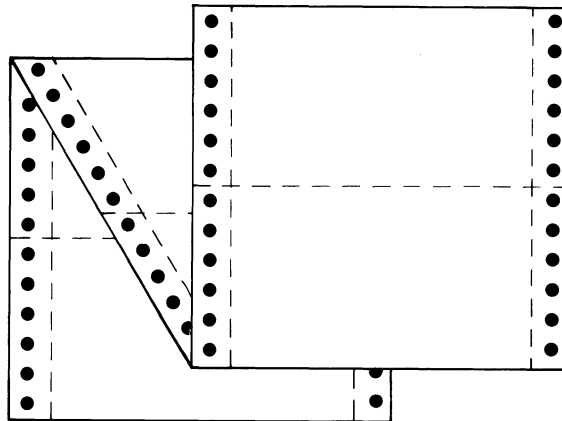
Regular file card
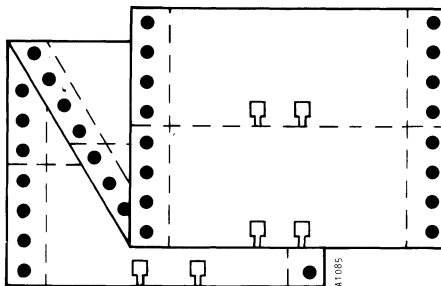
Rotary file card

Figure 4-5.   Regular and rotary file cards can be fed through a printer and printed with Records Processing

The Format Document for Continuous Form Cards

The format document for continuous-form card is just like the format document for mailing labels. Since you can put the fields anywhere you want in a format document, you might want to have the name and phone number on the first line, followed by the address and other pertinent information. If you find that your address list does not contain adequate information for the file cards, you may want to add other fields.

1.   Measure from the first line of one card to the first line of the next card below to determine the total number of lines between cards. Include as many returns as necessary.

2.   If you will be printing more than one card across, set up the format
      for the first card, and then measure the cards to determine where to
      start the information for the next card. Remember that the
      left-margin setting on the Print Document menu determines where the
      first character will print horizontally.

3.   When you have completed the format document, run a test of it on
      paper with a few records to see if your calculations are right. If
      not, go back and adjust the format document and/or the Print
      Document settings, and then try another test.

4.   When you are satisfied that the format is correct, load the cards
      into your printer. Be sure to adjust the printer for the thickness
      of the cards before you begin printing them.

Chapter 5
Creating Control-Glossary Entries

You have learned how to perform many useful Records Processing functions using list and format documents. When you learn how to create and use control-glossary entries, you will be able to do even more with Records Processing. You can select specific records from your list documents based on criteria that you specify. You can then use the records in your list documents as a data-base management system.

WHAT IS A CONTROL GLOSSARY?

A *control glossary* is a glossary with entries consisting of special Records Processing instructions for manipulating records in the list document. If you are not familiar with glossaries, refer to the *Fortune:Word Glossary User's Guide* for detailed information on how to create, write, verify, attach, and use glossary entries. You can use control-glossary entries to perform functions such as sorting lists and selecting records from lists according to the criteria you set up. A glossary can contain both regular glossary entries and control-glossary entries. You can use a control-glossary entry in two ways:

- You can sort and select records from a list document into an output document, keeping the field labels. You can then use the output document as an additional list document for further Records Processing functions.

- You can sort and select records from a list document and merge the results with a format document to produce an output document. For example, you can produce an output document containing form letters for persons who live in a specific zip code area.

HOW A CONTROL GLOSSARY DIFFERS FROM A REGULAR GLOSSARY

Glossary entries that use Records Processing control functions can only be used from the Records Processing menu. Some regular glossary functions are not accessible through Records Processing, and some Records Processing

functions cannot be used in a regular glossary entry. Appendix B, "Comparison of Glossary Keywords and Functions," in the *Fortune:Word Glossary User's Guide* contains a complete list of the glossary keywords and functions that can be used for regular glossaries and Records Processing.

CONTROL-GLOSSARY SYNTAX

There are rules for the way each Records Processing control-glossary function should be written within an entry. These rules are called the syntax of the function. When you type a statement within an entry, you must follow the proper syntax.

Below is a summary of the syntax guidelines that apply to all control-glossary entries. You will learn the syntax of specific functions later in this chapter. Refer to the *Fortune:Word Glossary User's Guide* for more detailed information on glossary syntax.

- Begin each entry with an entry label consisting of the word **entry** followed by a one-character label, as shown in this example:

      entry s

- Begin the body of the entry with a left brace ({), and end it with a right brace (}), as shown in this example:

      entry s
      {
          sort<company>
      }

- You may include comments anywhere in the entry. A comment must begin with a /* and end with a */, as shown in this example:

      entry s /*sort list document alphabetically by company name*/
      {
          sort<company>
      }

- Make sure that the glossary verifies successfully. If it does not, you will not be able to access any of the entries in it.

- You can have each entry perform a separate function, or a single entry may include as many simple or complex combinations of functions as you need. As you become more familiar with control-glossary entries, you can combine many functions into one entry.

- A glossary entry contains statements, arguments, and expressions that describe what the entry is supposed to do.

  - A *statement* is a declaration of purpose. A programming statement may be a single keyword or a whole series of words that consist of keywords, variables, functions, and strings. Statements belonging to conditional and control statements must be enclosed in braces { }. Other types of statements do not need braces.

  - A *conditional statement* tests to see if certain conditions are met before a function is performed. Examples of conditional statements are **if** and **while**.

  - An *argument* is an instruction that tells a particular function what to act on. For example, in the control-glossary entry example given below, "<last name>" is the argument for the "sort" function. It tells Records Processing what is to be sorted.

  - An *expression* is a part of an argument. In an argument composed of values inside parentheses, the values are expressions. For example, if you want to move the cursor down three lines, you can type **down(3)**. In this case, the argument is **(3)** and the expression is the number **3**.

- When you type two-word variable names such as **select_record**, the two words must be joined by an underbar (_). To type the underbar, press SHIFT/hyphen. Do not use MODE _ to type the underbar.

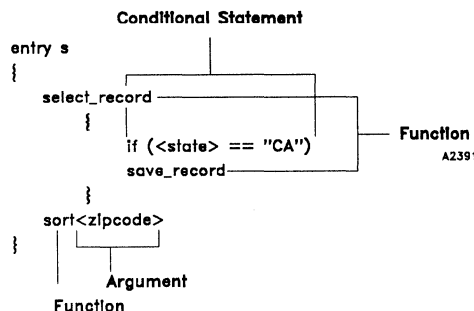Figure 5-1 shows the parts of a control-glossary entry.

```
                        Conditional Statement
    entry s                      |
      {              ┌───────────────────────┐
        select_record ────────────────────┐  │
          {        │                       │  │
            if (<state> == "CA")        ───┤  ├── Function
            save_record────────────────────┘      A2391
          }
        sort<zipcode>
      }    │  ┌────────┐
           │  │        │
           └──┤        │
           Argument
      Function
```

Figure 5-1. The parts of a control-glossary entry

The example below shows the difference between a regular glossary entry using keywords and a control-glossary entry.

Regular glossary entry	Control document entry
entry a	entry b
{	{
insert "word processing" execute	**sort** <last name>
}	}

If you are not sure how to create a glossary, refer to the *Fortune:Word Glossary User's Guide* for instructions. A control glossary follows the same rules for creation and verification as a regular glossary. You cannot create a control-glossary entry by using Glossary By Example.

CONTROL-GLOSSARY FUNCTIONS

The glossary language contains functions for two Fortune:Word applications:

- Glossary
- Records Processing

Refer to Appendix B, "Comparison of Glossary Keywords and Functions," in the *Fortune:Word Glossary User's Guide* for a list of the functions and keywords you can use with glossary entries. Table 5-1 is an alphabetical list of the functions you can use in control-glossary entries. The "Use of Function" column gives you the specific use for each function.

Table 5-1. Functions Used in Records Processing Control-Glossary Entries

Function	Use of Function
abs	String manipulation
ascending	Sort
call	Control
cat	String manipulation
descending	Sort
do	Control
else	Conditional
error	Display
exit	Control
false	Logical
if	Conditional
index	String manipulation

Table 5-1. (continued)

Function	Use of Function
jump	Control
key	Interactive
keys	Interactive
len	String manipulation
max	String manipulation
min	String manipulation
num	String manipulation
occur	String manipulation
prompt	Display
round	String manipulation
save_record	Record selection
seg	String manipulation
select_record	Record selection
sort	Sort
status	Display
sub	String manipulation
substr	String manipulation
thru	Range
true	Logical
truncate	String manipulation
while	Conditional

Table 5-2 lists the Records Processing control-glossary functions alphabetically by type.

Table 5-2.   Records Processing Control-Glossary Functions Grouped by Function Type

Conditional	Display
else	error
if	prompt
while	status
Control	Interactive
call	key
do	keys
exit	
jump	Logical
	false
	true

Table 5-2.   (continued)

Range	String manipulation
thru	abs*
	cat
Record selection	index
save_record	len
select_record	max
	min
Sort	num
ascending	occur
descending	round*
sort	seg
	sub
	substr
	truncate*

---

*Mathematical string-manipulation functions. The functions **max** and **min** can also be used as mathematical functions.

For complete information on the functions in the left-hand column, refer to the *Fortune:Word Glossary User's Guide*. The way Records Processing uses the functions in the right-hand column are described in this guide.

Each type of Records Processing function is explained in this chapter, beginning with the functions that are the easiest to use and progressing to the more advanced functions.

THE SORT FUNCTION

The **sort** functions are the simplest control-glossary functions to use. The **sort** functions are shown in Table 5-3.

Table 5-3. Sort Functions and Syntax

Function	Syntax
ascending	**sort ascending** *<field>*
descending	**sort descending** *<field>*
sort	**sort** *<field1>* *<field2>* *<field3>* *<field4>*
(Default to	
ascending)	

Sorting puts the records in a list document in a different alphabetical or numerical order based on the field or fields you specify in the control-glossary entry. The order of the fields within each record is *not* changed by the sorting process. Once you have created a control-glossary entry for sorting, you can use it in two ways:

- From the Records Processing menu, you can use a control glossary and a list document (without a format document) to sort the list document and change the order of its records. In this case, the field labels will remain in the output document.

- From the Records Processing menu, you can use a control glossary with both list and format documents to perform the sorting and merge the contents into the output document at the same time.

These are the guidelines for sorting:

- You can sort the records in a list either alphabetically or numerically, in either ascending (a–z) or descending (z–a) order. If you do not specify a descending sort, the sort will always be in ascending order.

- When the sort is being performed on only one field, the maximum record length is 2048 characters, including the field label.

- You can use a maximum of four fields at one time to sort the list document. If you sort, the total length of the contents of all fields to be sorted in a record, excluding the field labels, cannot be greater than 1024 characters.

- The sort function arranges the records in the list document according to the case-insensitive ASCII collating sequence shown in Appendix B. If you want to sort records containing non-English characters, such as accented characters, you must create an operating system file named **/usr/help/chtable.***XX*, where *XX* represents a language. Refer to Appendix E in the *Fortune:Word Reference Guide* for information on how to create this file.

Figure 5-2 shows a list document to be sorted, the control entry used to sort it, and the output document that results from the sorting. In this example, the list-document records are rearranged in alphabetical order by the **state** field. Since no format document is used, the field labels are included in the output document.
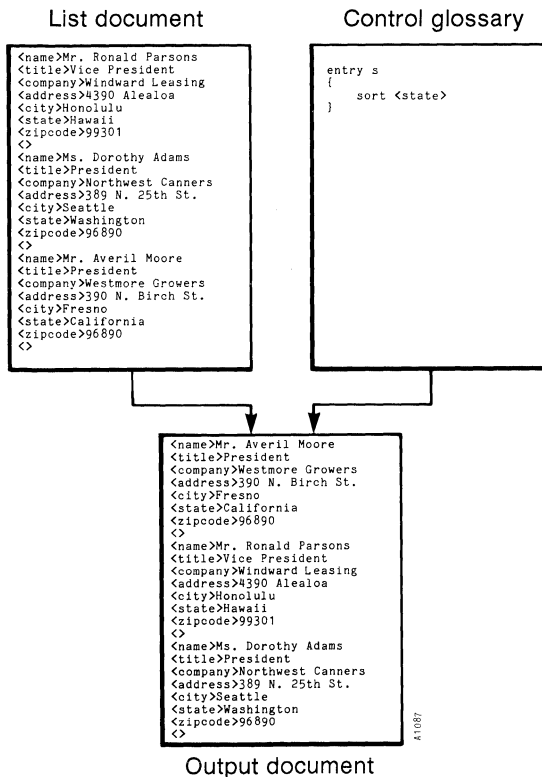
List document   Control glossary

```
<name>Mr. Ronald Parsons
<title>Vice President
<company>Windward Leasing
<address>4390 Alealoa
<city>Honolulu
<state>Hawaii
<zipcode>99301
<>
<name>Ms. Dorothy Adams
<title>President
<company>Northwest Canners
<address>389 N. 25th St.
<city>Seattle
<state>Washington
<zipcode>96890
<>
<name>Mr. Averil Moore
<title>President
<company>Westmore Growers
<address>390 N. Birch St.
<city>Fresno
<state>California
<zipcode>96890
<>
```

```
entry s
{
    sort <state>
}
```

```
<name>Mr. Averil Moore
<title>President
<company>Westmore Growers
<address>390 N. Birch St.
<city>Fresno
<state>California
<zipcode>96890
<>
<name>Mr. Ronald Parsons
<title>Vice President
<company>Windward Leasing
<address>4390 Alealoa
<city>Honolulu
<state>Hawaii
<zipcode>99301
<>
<name>Ms. Dorothy Adams
<title>President
<company>Northwest Canners
<address>389 N. 25th St.
<city>Seattle
<state>Washington
<zipcode>96890
<>
```

Output document

*Figure 5-2. The **sort** function rearranges records according to the order you specify and by the particular field you choose*

When you sort a list, you need to specify a particular field label, or *key field*, on which the sort will be performed. For example, suppose you want to sort an address list in alphabetical order according to last name. In this case, you would specify the field label for the last name as the key field for the sort in the control-glossary sort entry, as shown below:

```
entry 1
{
        sort <last name>
}
```

Ascending and Descending Order

When you sort a list in ascending order by a field that contains alphabetical characters, the characters are sorted alphabetically from a to z. A descending sort results in the reverse: characters are sorted from z to a.

An ascending numerical sort arranges the numbers from smallest to largest: for example, 15, 25, 100, 5000. A descending numerical sort would arrange the numbers in this way: 5000, 100, 25, 15.

> **NOTE:** The **sort** function sorts numerical fields in numerical order. Fields that contain a combination of alphabetic and numeric characters are sorted according to the case-insensitive ASCII collating sequence. For example, 1a, 2a, 3a, 10a, 20a, 30a would be sorted in this way: 1a, 10a, 2a, 20a, 3a, 30a.

Sorting on More than One Field at a Time

You can specify up to four key fields for sorting. This example sorts on three fields:

```
entry s
{
    sort<last name>  <first name>  <mid initial>
}
```

In this case, the sort is done on the first field, then the second, and third. With this example, a docket calendar would be sorted first alphabetically by last name, then by first name, and then by middle initial. Figure 5-3 shows an output document resulting from using this entry.

The sorting process is always performed on all fields specified in the sort-function entry. However, if you specify four key fields, the system will only sort a record using the second key field if the first key field is the same for two or more records. Sorting on the third key field occurs only if two or more records have the same information in the first two key fields, and so on for the fourth key field.

Sort Syntax

The basic syntax for the **sort** function is the word **sort** followed by the label of the key field in the list document, as in this example:

```
sort <field>
```

```
                              Docket Calendar

Attorney        Date      Ct/Item       Case #  Client         File #  Comments


Fred Q. Bailey  09/09/86  Sup. Ct.      45679   Tom Rogers     10098   Trial at 9:30 a.m.
Fred Q. Bailey  05/27/85  Circuit Ct.   45611   Miller Bank    10021   Brief due
Fred Q. Bailey  12/12/85  Sup. Ct.      45605   Dance Disco    10096   Judgement
Fred Q. Bailey  12/07/85  Sup. Ct.      45610   Dance Emporium 10097   Judgement
Fred Q. Bailey  12/25/85  Probate       45632   Sigmond Denver 10053   Estate tax return due
Fred Q. Bailey  10/06/85  Sup. Ct.      45600   Dance Emporium 10093   Judgement
Fred Q. Bailey  01/08/85  Limit         45678   Smiling Plants 10027   Expires on fraud
Fred T. Bailey  09/21/85  Limit         45634   Smiling Plants 10047   Expires on cause of action
Fred T. Bailey  01/04/86  District      45690   Amber Garage   10033   Motion-Temp. Ord. 2:00 p.m.
Fred T. Bailey  02/03/85  Probate       45649   Miller Bank    10099   Annual account due
Fred T. Bailey  02/14/86  Probate       45637   Miller Bank    10063   Hearing-Petition-2:00 p.m.
Sarah W. Luchini 05/06/85 Circuit Ct.   45665   Happy Cars     10026   Brief due
Sarah W. Luchini 09/09/85 Sup. Ct.      45644   Beach Homes    10038   Trial 2:00 p.m.
Sarah W. Luchini 05/04/85 Probate       45602   Happy Cars     10013   Estate tax return due
Sarah W. Luchini 11/04/85 Probate       45692   Happy Cars     10014   Estate tax return due
Sarah W. Luchini 01/08/85 District      45663   Beach Homes    10085   Trial brief due
Sarah W. Luchini 06/05/86 Sup. Ct.      45671   Miller Bank    10056   Judgement lien expires
Sarah W. Luchini 03/06/86 District      45609   Wilt Still     10054   Motion-new trial 2:00 p.m.
Sarah W. Luchini 11/02/86 District      45661   Huge Trailers  10025   Trial brief due
Sarah W. Luchini 07/19/85 Federal Ct.   45670   Bonnie White   10024   Motion-continuance 10:00 a.m.
Sarah W. Luchini 11/18/85 District      45662   Beach Homes    10084   Trial brief due
Ashley R. Miller 05/09/85 Sup. Ct.      45685   Amber Garage   10078   Trial 10:00 a.m.
Ashley R. Miller 10/04/85 District      45650   Tom Smith      10011   Motion-new trial 9:30 a.m.
Ashley R. Miller 06/05/85 Probate       45666   Smiling Plants 10023   Inheritance tax return due
Ashley R. Miller 10/03/86 Limit         45633   Happy Cars     10012   Expires on trespass
Ashley R. Miller 05/25/86 Limit         45604   Smiling Plants 10022   Expires on Contract
Dennis C. Rodeen 08/16/86 District      45687   Mathew Shirts  10068   Answer to interrogs
Dennis C. Rodeen 08/20/85 Firm                                 20000   Prepare quarterly returns
Dennis C. Rodeen 12/06/86 Limit         45660   Shoes Inc.     10076   Brief due
Dennis C. Rodeen 04/09/85 District      45620   Amber Garage   10044   Answer Due
Dennis C. Rodeen 05/07/86 District      45664   Western Pants  10017   First day-default judgment
Dennis C. Rodeen 11/12/86 Firm                  Happy Cars     20000   Auto lease expires
Dennis C. Rodeen 11/03/85 Limit         45639   Shoes Inc.     10077   Brief due
```

Figure 5-3. An Output Document Sorted on Three Fields

A complete control-glossary entry using the sort function should look like
this:

```
entry a
{
      sort <field>
}
```

Of course, you can include a **sort** statement as part of a more complicated
entry. You will see some examples of these later in this chapter.

> **REMEMBER:** Each glossary entry must have an entry label,
> and it must begin and end with a set of braces.

In a **sort** statement, the default order for sorting is ascending. You do
not have to specify a sorting order unless you want to sort in descending
order. You can specify ascending order in the sort argument if you want,
though this is not required. The examples below produce the same result:

```
entry a
{
      sort <field>
}


entry b
{
      sort ascending <field>
}
```

To sort a list in descending order, you must type the word **descending**
between the sort argument and the field label, as in this example:

```
entry c
{
      sort descending <field>
}
```

To sort more than one field at a time, follow the sort function with the
labels of the key fields to be used in sorting. Your only option is to
specify either ascending or descending order for all the fields. You
cannot combine ascending and descending order in the same sort statement.
You can only have one **sort** statement in an entry.

```
entry e
{
      sort <last name> <state> <city> <zipcode>
}


entry f
{
      sort descending <last name> <state> <city> <zipcode>
}
```

Using Sort to Change the Order of a List Document

To put a list into a different order, you can sort a list document without using a format document. You do this by choosing the *Make output document only* option on the Records Processing menu and then specifying a name for the output document. You can overwrite an output document that has already been created. However, Records Processing will not permit you to sort and then overwrite a list document you have specified on the Records Processing menu. If you want to replace the old list document with the new, sorted list document, follow these steps:

1. Give the output document a temporary name such as **list.sort**.

2. Rename the **list.sort** document using the original list-document name.

You cannot use a document name that you have entered on the Records Processing menu as an output document name.

THE SELECTION FUNCTIONS

The control-glossary functions **select—record** and **save—record** must be used together. You can use them to choose specific records from a list document. Just as with the control-glossary entry for sorting, you can produce two types of output documents:

*   You can produce a secondary list that includes all the field labels by using a control-glossary entry with a list document only.

*   You can produce your final output, selected and merged with the format document, by using a control-glossary entry, a list document, and a format document.

The output documents generated with the **select—record** and **save—record** functions and those generated only with the **sort** function are different. When you use the **select—record** and **save—record** functions, you specify which records you want to select; only the records that meet the criteria you have specified are merged into the output document. When you use a control-glossary entry to sort a list document, all the records in the original document are placed into the output document; only the order in which they appear is changed.

An Example of Using the Selection Functions

The easiest way to understand how **select__record** and **save__record** work is by
looking at an example. Suppose a company has a mailing list of 300
customers from all over the United States. The Marketing Department wants
to mail a form letter only to those people who live in California. To
expedite the mailing, they want to have the records for California
residents sorted by zip code.

The glossary entry they use looks at the field label <state> for each
record in the list document. Each record that contains a **CA** in the
<state> field is put aside for sorting by zip code. Out of the 300
records, 110 meet the criteria of **CA**. Records Processing then sorts these
records into numerical order by zip code and merges them with the format
document to create the letters. All of the 300 customer records remain in
the list document, unchanged by this selection process.

Here is how the control-glossary entry would look:

```
entry s      /*select CA records*/
{
       select__record
             {
                    if (<state> == "CA")
                    save__record
             }
       sort <zipcode>
}
```

RELATIONAL OPERATORS

Relational operators define how one field should relate to another
field. You must always include a relational operator in an **if** statement
in addition to the field label and field criteria. The relational
operator explains the type of criteria the field being searched for must
match in order for a record to be selected. You have already seen several
examples of **if** statements in Chapter 4 using the relational operator ==,
which means equal to. The nine relational operators are shown in
Table 5-4, which lists their symbolic form and meaning. You can use these
with either numeric or character fields.

Table 5-4. Relational Operators

Symbol	Meaning
==	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
!=	Not equal to
|	Or
&	And
!	Not

Below are some examples of how these operators are used. Continuing with the example above, if our fictional company wanted to send letters to every state EXCEPT California, the entry would look like this:

```
entry s      /*select records other than CA*/
{
        select_record
            {
                    if (<state> != "CA")
                    save_record
            }
        sort <zipcode>
}
```

In this case, all states that were not equal to **CA** would be selected, saved, and sorted by zip code.

The next example selects and saves all records that contain either a state abbreviation equal to **MN** or a company name equal to **XYZ**:

> **NOTE:** When you use multiple arguments with an **if** statement, the entire argument must be enclosed in parentheses. In addition, each part of the argument has its own matching parentheses.

```
entry c
{
      select_record
           {
                 if((<state> == "MN") | (<company name> ==
                 "XYZ"))
                 save_record
           }
}
```

The next example selects and saves only those records in which the zip
code field is equal to **87190**:

```
entry Z
{
      select_record
           {
                 if(<zipcode> == 87190)
                 save_record
           }
}
```

Selecting Records by Suppressing a Blank Field

Suppose you have an inventory list document. You may have a field labeled
<back order>. If an order has been placed but has not yet been
filled, this field has an entry. If the order was filled, the field is
blank. The entry below can be used to select all records in which the
field <back order> is not blank:

```
entry n
{
   select_record
      {
            if(<back order> != "")
            save_record
      }
}
```

If you want to generate a list that contains records for only the items on
the list that have been back ordered, you can use the entry below. Note
that the difference between these two entries is that entry n uses the
assignment operator not equal to (!=), and entry N uses the assignment
operator equal to (==).

```
entry N

{
   select_record
      {
            if(<back order> == "")
            save_record
      }
}
```

The Syntax of Selection Functions

Look carefully at the example above. You will notice that there is an
extra set of braces in this control-glossary entry. The select_record
statement requires an argument, and that argument must be included within
its own set of braces. Once you have typed the opening brace, you have to
state what conditions must be met for a record to be selected. To do
this, you use the conditional statement if.

The if statement has its own syntax. The word if must be followed by an
argument enclosed in parentheses. In that argument, you specify the field
label you want to use for the selection, the criteria for the selection,
and the relationship between the two. In this case, you want the contents
of that field label to be equal to a blank. To do this you type two
quotes with nothing between them (""). If the contents of a field is a
combination of alphabetic and numeric characters or spaces, or contains
only alphabetic characters or spaces, you must put quotation marks around
the specified text. You do not need to enclose numeric expressions in
quotes.

You can combine as many ifs as you wish into one statement. It is useful
to line up the braces and indent the if statement, as shown in the
example. Although this is not necessary, it can be very helpful if the
glossary does not verify and you must go back and check the entry for
errors.

Another convenient way of keeping track of multiple if statements is to
construct a flowchart like the one shown in Figure 5-4. If you are
familiar with flowcharts, try using this method to chart the results of
your if statements.

The *Fortune:Word Glossary User's Guide* gives you detailed information on
if and other conditional statements.
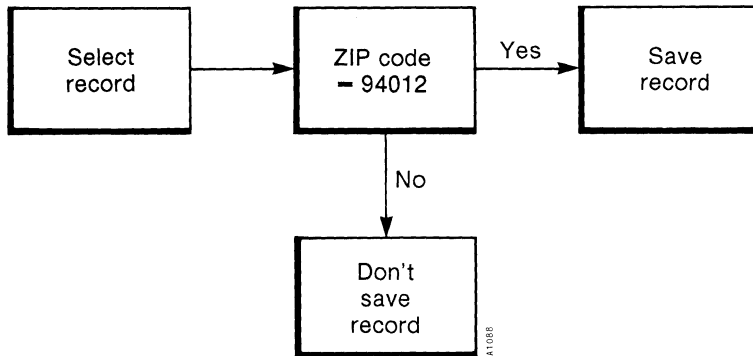
Figure 5-4. *Use a simple flowchart to track the results of* if *statements you use with* select_record *and* save_record

THE THRU FUNCTION

You can use the function **thru** to specify a range for a particular field in the list document. Suppose you want to send a mailing to each address in a list document that contains a zip code starting with the number 9. You could use the **thru** function in a control-glossary entry to select these records. This is how such an entry might look:

```
entry t
{
        select_record
            {
                if(<zipcode> == 90000 thru 99999)
                save_record
            }
}
```

All records containing zip codes from 90000 through 99999 would be selected by this entry.

You can also use the **thru** function to make alphabetical selections from a list document. Suppose you want to send a reminder notice to all members on a list whose last name begins with the letter **B**. You can easily write a control-glossary entry to do this. You do not have to presort your list into alphabetical order and then enter the actual names that begin and end the alphabetical listing. Instead, you can specify the range by the beginning letter alone. The control-glossary entry would look like this:

```
entry B
{
        select_record
            {
                    if(<last name> == "B" thru "C")
                    save_record
            }
}
```

Suppose you had the following last names in your list document:

```
<last name>Adams
<last name>Banyon
<last name>Barry
<last name>Bergman
<last name>Bunyan
<last name>Charles
<last name>Cooper
```

Using the sample control-glossary entry above, the records containing the last names Banyon, Barry, Bergman, and Bunyan would be selected. The **thru** function in the example was constructed to select from **B** through **C**. Why would the last names Charles and Cooper not be selected? The **thru** function would cause the search to end with records containing only the letter **C**. Since the two last names beginning with **C** contain additional characters, they would not be selected.

Suppose the entry were changed to select **B** thru **Ci**, as shown below:

```
entry B
{
        select_record
            {
                    if(<last name> == "B" thru "Ci")
                    save_record
            }
}
```

The records containing the last names Banyon, Barry, Bergman, Bunyan, and Charles would be selected. This is because Records Processing was instructed to select from **B** through **Ci**, and the last name **Charles** is included in that range.

If you want to select records containing the last names beginning with the letters **B** and **C**, the entry could look like the one below. This example also instructs Records Processing to sort the last names in alphabetical order.

```
entry B
{
      select_record
           {
                 if(<last name> = = "B" thru "D")
                 save_record
           }
      sort<last name>
}
```

The **thru** function is easy to use, and it can be extremely helpful in making specific selections of records with fields that fall within an alphabetic or numeric range.

SUMMARY

As you can imagine, there are many possible combinations of statements for sorting and selecting records. Here is a review of the major points:

- Follow the **sort** function with the field label to be sorted. Be sure to enclose the field label within Merge and Shift/Merge symbols.

- Use the **select_record** and **save_record** functions with an if statement to define which records you want to choose.

- The **select_record** statement has the following syntax:

```
select_record
     {
            if(argument)
            save_record
     }
```

In addition to **if** statements, there are 13 string-manipulation functions you can use in a **select_record** argument for even more specific sorting and selecting. Refer to the section on "String Manipulation Functions" in Chapter 6, "Advanced Control-Glossary Functions," for more information about these.

USING THE CONTROL-GLOSSARY ENTRY

You enter the control glossary name and entry label on the Records Processing menu when you wish to sort or select records from a list document. The control glossary may be used with a list document alone, or it may be used along with both list and format documents. Records are selected and sorted from the list document, merged with the format document, and saved in the output document.

Try writing a control-glossary entry to sort the address list called **add.list** you created in the list-document exercise. Then enter **add.list, add.form,**—the form letter you created in the format-document exercise—and your new control-glossary name and entry label on the Records Processing menu. Press EXECUTE and specify an output-document name. The letters will be output in the sort order you specified in the control glossary.

In this chapter, you have learned how to use the sorting, selection, and thru functions, and you have been introduced to the conditional function **if**. Chapter 6, "Advanced Control-Glossary Functions," contains information on arithmetic operators and string manipulation functions.

Chapter 6
Advanced Control-Glossary Functions

The control-glossary functions you learned about in Chapter 5 allow you to sort and select records from a list document using criteria you specify. You may find that you can use these functions to satisfy all your Records Processing needs.

However, Records Processing has additional control-glossary functions that you can use to create even more sophisticated entries. The Records Processing functions discussed in this chapter require a special syntax and may operate in a slightly different way than functions with the same names in regular glossary entries.

Table 5-1 contains an alphabetical list of all the functions you can use as part of a control-glossary entry. Regular glossary functions are not discussed in this manual. Refer to the *Fortune:Word Glossary User's Guide* for information on these functions. The only functions discussed in this chapter are those that are unique to Records Processing.

- *Arithmetic operators* are used to compare the numerical contents of the fields in a list document.

- *String-manipulation functions* are used to test all or part of the contents of a field for specific conditions.

The following sections do not contain lengthy examples or sample documents and exercises. Instead, you will find a description of each advanced control function, a statement of its syntax, and a brief example of how to use it. Information that may vary each time you use the function is indicated by italics. For example, if the syntax statement includes *<field>*, you would substitute the appropriate field label from your list document between the Merge and Shift/Merge symbols in your control-glossary entry.

ARITHMETIC OPERATORS

You can use arithmetic operators to analyze and assess values within alphabetical or numeric strings of data. They do NOT change the field contents. You use the arithmetic operators in conditional statements with the **select_record** and **save_record** functions. If the conditions are met, the record is selected. The arithmetic operators are listed in Table 6-1.

Table 6-1. Arithmetic Operators

+	addition
-	subtraction
*	multiplication
/	division
%	remainder

A field used with an arithmetic operator can contain numbers, a leading dollar sign, commas, and a decimal point. If they contain any other characters, the record will not be selected. Following is a discussion of the operators, including a description of each, its syntax, and an example.

> NOTE: You can perform arithmetic operations on any number of fields.

Addition

Syntax: if ((<*field1*> + <*field2*> + <*field3*>) == *n*)
             save_record

The *addition operator* (+) adds the contents of numeric fields. That value is then compared to *n*, using the relational operator ==   If the result of this addition matches the sum represented by the number *n*, the record is saved. Here is an example of addition within an if statement:

Sample list-document contents:

    <number1>15
    <number2>10

Sample control-glossary entry:

```
entry a        /*addition of fields*/
{
        select_record
            {
                    if((<number1> + <number2>) == 25)
                    save_record
            }
}
```

Result: This record is saved.

Subtraction

Syntax: if ((<*field1*> - <*field2*>) == *n*)
            save_record

The *subtraction operator* (-) subtracts the contents of the second field
from the contents of the first. If the result matches the number
represented by *n*, the record is saved. Here is an example of
subtraction within an **if** statement:

Sample list-document contents:

```
<number1>15
<number2>15
```

Sample control-glossary entry:

```
entry s       /*subtraction of fields*/
{
        select_record
            {
                    if((<number1> - <number2>) == 25)
                    save_record
            }
}
```

Result: This record is *not* saved.

Multiplication

Syntax: if ((<*field1*> * <*field2*>) == *n*)
            save_record

The *multiplication operator* (*) multiplies the contents of one field by
the contents of another. If the result matches the number represented by
n, the record is saved. Here is an example of multiplication within an
if statement:

Sample list-document contents:

```
<number1>5
<number2>1
```

Sample control-glossary entry:

```
entry m      /*multiplication of fields*/
{
        select_record
            {
                    if((<number1> * <number2>) == 5)
                    save_record
            }
}
```

Result: This record is saved.

Division

Syntax: if ((*<field1>* / *<field2>*) == *n*)
            save_record

The *division operator* (/) divides the contents of the first field by the
contents of the second. If the result matches the number represented by
n, the record is saved. Here is an example of division within an if
statement:

Sample list-document contents:

```
<number1>4
<number2>1
```

Sample control-glossary entry:

```
entry d      /*division of fields*/
{
        select_record
            {
                    if((<number1> / <number2>) == 4)
                    save_record
            }
}
```

Result: This record is saved.

Remainder

Syntax: if ((*<field1>* % *<field2>*) == *n*)
            save_record

The *remainder operator* (%) divides the contents of the first field by
the contents of the second, then tests to see if the resulting remainder
(if there is one) matches the number represented by *n*. If there is a
match, the record is saved. Here is an example of the use of the
remainder function within an **if** statement:

Sample list-document contents:

        <number1>5
        <number2>2

Sample control-glossary entry:

```
entry r      /*remainder of fields*/
{
        select_record
            {
                    if((<number1> % <number2>) == 3)
                    save_record
            }
}
```

Result: The remainder of this division is 1 (5/2 = 2 with a remainder
            of 1), therefore the record is *not* saved.

STRING-MANIPULATION FUNCTIONS

In programming languages, a *string* is a group of characters that belong
together and that are not part of an executable command. Strings can be
alphabetic (made up of letters, sometimes with numbers and/or special
characters, including spaces) or *numeric* (made up of numbers, sometimes
with a few special characters). The characters in an alphabetic string
are usually enclosed in quotation marks, and the characters in a numeric
string are not.

In Records Processing, a string is defined in a different way. Strings
are made up of the characters in the field contents that follow a field
label in the list document. The character strings in a list document

should not be enclosed in quotation marks unless you want quotation marks to appear in the output document. The string may consist of information such as a person's name, a phone number, a part number, or a company name.

When you use the glossary programming language to write control-glossary entries, you specify the Records Processing strings that will be examined in the list document. You do this by using the field label or labels that identify the field contents to be compared.

Any alphabetic or alphanumeric information that you use in a conditional statement within the glossary entry to compare against these Records Processing strings must be enclosed in quotation marks.

To see the difference between using a field label to define a string for manipulation and using a quoted string in a control-glossary entry, look at the sample control-glossary entry shown in Figure 6-1. In this entry, the field label <title> defines the string in the list document. The contents of each field that follow the field label <title> are compared against the text that appears between the quotation marks. The quoted string "Vice President" tells the control glossary what specific text you want to match. The syntax and function of this entry are explained in the next section of this chapter, which is entitled "Index."

```
entry v    /*select all titles containing the string Vice President*/
{
    select_record
        {
                if(index(<title>, "Vice President"))
                save_record
        }
}
```

Figure 6-1. String syntax

There are 13 string-manipulation functions that you can use in Records Processing control-glossary entries. When you manipulate strings, you do not actually make any changes to the list, format, or output documents.

You can use the string-manipulation functions to improve your flexibility and control in using your list documents. You can select records from a list document by looking at a specific location or segment of the field contents. You can also compare the contents of two different fields. In addition, you can select records that contain no entries in a specific field. Once you understand how these functions work and how easy they are to use, you will find that your ability to sort and select records from a list document is greatly expanded.

> NOTE: The control-glossary functions that manipulate
> strings can only be used to select specific records from
> a list document. They do not change the text in the
> list, format, or output documents.

The 13 string-manipulation functions are shown below. The syntax and a
brief description of each function are given in Appendix A.

- Absolute (**abs**)
- Concatenate (**cat**)
- Index (**index**)
- Length (**len**)
- Maximum (**max**)
- Minimum (**min**)
- Numeric (**num**)
- Occurrence (**occur**)
- Round (**round**)
- Segment (**seg**)
- Substitute (**sub**)
- Substring (**substr**)
- Truncate (**truncate**)

The most commonly used functions are explained on the following pages, in
the order you are most likely to use them.

INDEX

Suppose you want to select from an address list only those last names that
begin with the letter "B." Also suppose you want to send letters to just
the vice presidents on a mailing list. You can use the **index** function to
do both of these things.

The syntax for **index** is as follows:

      index(*<field>*,*"string"*,*n*)

The **index** function searches the contents of a specified field—for example,
<name> or <title>—for the string of characters you type between
the quotation marks, starting at the character position specified by the
number *n*. If you do not specify a character position, or if you type
"1" in the last position, the **index** function searches from the beginning
of the field contents. If the character string is found, the record is
selected and placed in the output document. If the string is not found,
the record is not selected. If no record in the list document contains
the string, the message *No records selected, Press CANCEL to stop*
appears on the Records Processing menu.

The control-glossary entry shown below searches a list document for last names beginning with the letter **B**, selects only those records, and sorts the records numerically by zip code in the output document. The list document used with this entry must have a separate field, called <last name> that contains only the last names of the members on the list.

Since this example searches the contents of the <last name> field from the beginning, if there is an uppercase **B** anywhere in that field, the record will be selected. You could also use the **substr** function to make this kind of a selection (described later in this chapter).

```
entry b      /*Selects records for last names beginning with the letter
             B and sorts them by zipcode*/
{
      select_record
            {
                  if(index(<last name>,"B",1))
                  save_record
            }
      sort<zipcode>
}
```

Another way to use the **index** function would be to select all records that have the words "Vice President" in the title field. The control-glossary entry would look like the following, assuming that you have a list document with the field label <title>.

```
entry v      /*Selects all titles containing the string Vice President*/
{
      select_record
            {
                  if(index(<title>, "Vice President"))
                  save_record
            }
}
```

This control-glossary entry selects all records that contain the string **"Vice President"**. This could include titles such as "Executive Vice President" and "Vice President, Operations." If you want to sort the records alphabetically by company name, you could add a **sort** statement to this control-glossary entry. If any word or characters are included between the words "Vice" and "President," the record is not selected. For example, a record with the title "Vice-President, Operations" would not be selected since the hyphen is not included in the string.

Note that in both of the above examples the field is searched from the beginning. The first example specifies that the search should begin at character position "1," though you could, of course, specify any character position in the string. In the second example, since no number is entered after the string, the **index** function automatically searches the entire field.

> **REMEMBER:** Using comments—the text enclosed between "/*" and "*/" on the first line of each entry—can help remind you what each entry in the control glossary is designed to do.

OCCUR (Occurrence)

The **occur** function counts the total number of segments in a field. You specify the delimiter that divides one segment from the next. The delimiter can be any character. The most commonly-used delimiters are a space, comma, slash, dash, or some other punctuation mark.

The syntax for **occur** is as follows:

occur(*<field>*, *"delimiter"***)**

You can use this function to limit the number of items selected from a list or to select records that contain no information. For example, you could select titles that are composed of two words. Note that the delimiter in this example is a space.

```
entry o      /*Selects all titles that contain two words*/
{
      select_record
            {
                  if(occur(<title>, " ") == 2)
                  save_record
            }
}
```

Suppose you want to select all the records from a list that contain information in a specific field. If you have an inventory list document with a separate field to indicate if an item has been back ordered, the control-glossary entry to select all addresses that contain an entry in the <back order> field would look like this:

```
entry W    /*Select all records that contain back orders*/
{
       select_record
             {
                    if(occur(<back order>, " ") != 0)
                    save_record
             }
}
```

If you want to select records that do *not* contain any information in a
particular field, the entry should look like the one below. This entry
will select records that do not contain any information in the
<back order> field. Note that the only difference between the two
entries is the relational operator. In the previous example, the **occur**
statement is set not equal to 0. In this example, the **occur** statement is
set equal to 0.

```
entry w    /*Select all records that do not contain back orders*/
{
       select_record

             {
                    if(occur(<back order>, " ") == 0)
                    save_record
             }
}
```


NUM (Numeric)

The **num** function can be used to determine if the information contained in
a field is a number. In addition to numbers, the field may contain a
leading dollar sign, commas, a decimal point, and/or leading or trailing
minus or plus signs.

The syntax for **num** is as follows:

   num <*field*>

You can use the **num** function to sort a list document that contains both
Canadian and U.S. zip codes. Since Canadian zip codes contain alphabetic
as well as numeric characters, you can use control-glossary entries like
the ones shown below to separate a list into U.S. and Canadian addresses.

You can use entry z to select addresses with U. S. zip codes, which only contain numbers. You can use entry Z to select addresses with Canadian zip codes, which contain alphabetic as well as numeric characters. You can also use the **num** function to select records if a field contains a number and skip them if it is blank.

```
entry z      /*Select numeric zip codes*/
{
     select_record
          {
               if(num(<zipcode>) == true)
               save_record
          }
}

entry Z      /*Select zip codes containing alphabetic characters*/
{
     select_record
          {
               if(num(<zipcode>) == false)
               save_record
          }
}
```

LEN (Length)

The **len** function counts the number of characters in a field.

The syntax for **len** is as follows:

len(<*field*> == *n*)

The length is calculated from the first character after the specified field label to the last character before the next field label. If the field label contains any returns, each return is counted as a character.

The entry shown below tests to see if the fields for name or company contain more than 25 characters. You can use it to check whether all addresses in a list document will fit on the labels you are using. If you use a control-glossary entry like this and all records are selected, then you know all the addresses will fit on the labels.

```
entry Q     /*Select short records using len function*/
{
        select_record
            {
                if((len(<name>) < 25) & (len(<company>) <
                25))
                save_record
            }
}
```

If all records are not selected, you can use the following glossary to make a new list document containing the addresses that will not fit on the standard labels. You can then edit the long fields, breaking up the long lines with returns. Then you can use the list documents separately or recombine them into the same document.

Remember, however, that if you split titles or company names onto two lines, these addresses may contain too many lines to print on the labels you have selected. If you use such a list document with your format document, the number of lines in the output document will change.

```
entry q     /*Select long records using len function*/
{
        select_record
            {
                if((len(<name>) >= 25) | (len(<company>)
                >= 25))
                save_record
            }
}
```

Notice the differences between the two entries. Entry Q selects records only if each field contains less than 25 characters. Entry q selects records if the length of the contents of either specified field is greater than or equal to 25.

MAX (Maximum)

The **max** function compares fields and selects records that are either larger numerically or occur later in the alphabet. You can use **max** to compare any number of fields.

The syntax for **max** is as follows:

```
max(<field1>,<field2>)
```

If you have a list document containing membership information for an organization, you can use the **max** function to determine whether an amount of money that is due has been paid, either in part or in full. You can then send reminder letters to only those on the list who have not paid the full amount. You do this by comparing the number in the <dues> field with the number in the <paid> field. Here is a control-glossary entry that looks like it will select the records you want. Note that the field label <dues> to the right of the assignment operator (==) indicates which field is to be used as the maximum value. Different types of members may pay dues from a sliding scale.

```
entry m    /*Select records of members who have not paid the full
            amount due using the max function*/
{
       select_record
           {
                  if(max(<dues>, <paid>) == <dues>)
                  save_record
           }
}
```

If you try this entry, you will notice that it also selects those records in which the <dues> and <paid> fields contain the same number, which would happen if the member had paid the full dues. However, you only want to send reminders to members who have not paid the full amount. You can modify the entry to select only the records of those who have not paid their full dues. The modification adds a statement that tests to be sure that the dues is not equal to the amount paid.

```
entry M    /*Select records of members who have not paid the full
            amount due using the max function*/
{
       select_record
           {
                  if((max(<dues>, <paid>) == <dues>) &
                  (<dues> != <paid>))
                  save_record
           }
}
```

MIN (Minimum)

The **min** function is similar to the **max** function. You can use **min** to compare fields and select records that either are smaller numerically or occur earlier in the alphabet. You can use **min** to compare any number of fields.

The syntax for **min** is as follows:

    min(*<field1>*,*<field2>*)

Using the **min** function, you can create a control-glossary entry that would allow you to send "thank you" letters to those who have paid their dues and/or paid more than the dues, for example, if someone prepays. In this case, you want to select a record if the number in the <paid> field is either equal to or greater than the number in the <dues> field. Consequently, the modifications made to the previous entry are not necessary. The following entry performs this operation:

```
entry n      /*Select records of members who have paid their dues or
             made an additional payment using min function*/
{
     select_record
         {
              if(min(<dues>, <paid>) == <dues>)
              save_record
         }
}
```

    REMEMBER: You can also use **max** and **min** to select the highest or lowest field in an alphabetic sequence.

SEG (Segment)

The next two functions discussed, **seg** and **substr**, are used to test for specific characters at specific locations in a field. The **seg** function counts the number of segments in the field; you define what a segment is. You can use the **seg** function to select records that contain specific characters at the same position in the segment string. You determine the delimiter, or character that divides the segments of the string. The delimiter can be any character, but is often an ampersand (&), a hyphen (-), or a space.

The syntax for **seg** is as follows:

    seg(*<field>*,*"delimiter"*,*n1*,*n2*)

The numbers of the segments with which the search begins and ends are given in positions *n1* and *n2*. Suppose, for example, that you have a Social Security number such as 980-56-8765 and define the delimiter as a hyphen (-). In this case, "980" is segment 1, "56" is segment 2, and "8765" is segment 3.

If you have a list document containing Social Security numbers, you can use the **seg** function to select all records that have the number "58" in the middle position of the Social Security–number field. In using the **seg** function, you have to count the number of segments in the defined field and identify what segment(s) you want to check by entering the segment number(s). In this case, there are three segments and you want to begin and end your search with the second one, as shown in this entry:

```
entry S      /*Select social security numbers containing the number 58
             in the middle segment*/

{
      select_record
          {
                if(seg(<number1>,"-",2,2) = = "58")
                save_record
          }
}
```

SUBSTR (Substring)

The **substr** function is similar to the **seg** function. You can use **substr** to test for a particular sequence of characters at a location defined by the positions of characters in a string.

The syntax for **substr** is as follows:

substr(<*field*>, *n1*, *n2*)

The substring function tests the characters between character positions *n1* and *n2* in the specified field.

The difference between **seg** and **substr** is that **seg** tests by segments, whereas **substr** tests by character locations. Suppose you want to write a control-glossary entry to do the same thing using **substr** that entry S above does using **seg**. To do so, you have to count the number of characters in the defined field and enter the numbers of the beginning and ending positions in order to identify what locations you want to check. A Social Security number has the format 000-00-0000. You can easily determine that the locations you wish to select are from character position 5 to character position 6, as shown in this entry:

```
entry g      /*Test for a social security number with 58 in the middle
             segment, using substr*/
{
     select_record
         {
              if(substr(<number>,5,6) == 58)
              save_record
         }
}
```

As another example, if you have a list document that contains phone numbers with area codes, you can select all the records from the list with area code 415. The area code must be in the first three positions of the field. The entry would look like this:

```
entry G      /*Test for a 415 area code in the first three positions
             using substr*/
{
     select_record
         {
              if(substr(<number>,1,3) == 415)
              save_record
         }
}
```

REMEMBER: You do not have to enclose numeric expressions in quotes.

If the phone numbers in the list document are in the form (415) 593-9000, the positions to be tested are 2, 3, and 4. The entry would look like this:

```
entry f      /*Test for a 415 area code starting at position two using
             substr*/
{
     select_record
         {
              if(substr(<number>,2,4) == 415)
              save_record
         }
}
```

Earlier in this section, you learned how to use the **index** function to select all the records that contain names beginning with the letter **B**. You can use the **substr** function to make sure that you will only select records containing the letter **B** in the first position in the name string. Such an entry would look like this:

```
entry B      /*Select for company names beginning with the letter B
             using substr*/

{
     select_record
         {
             if(substr(<company>,1,1) == "B")
             save_record
         }
}
```

CAT (Concatenate)

The **cat** function *concatenates,* or combines, the contents of two or more
fields. Unlike the **cat** function in regular glossary, the contents are
combined only for comparison purposes; they cannot be transferred together
to an output document. This function provides another way to compare the
information in two different fields of a record.

The syntax for **cat** is as follows:

cat(*<field1>*,*<field2>*)

Suppose you have a membership list, and you want to select those members
whose membership begins in May and ends in December. You could select
those records from the list document by using this control-glossary entry:

```
entry C      /*Compare the information contained in the two specified
             fields with the text string in quotes. If there is a
             match, the records are saved*/
{
     select_record
         {
             if(cat(<month start>,<month end>) ==
             "MayDecember")
             save_record
         }
}
```

COMBINING STRING-MANIPULATION FUNCTIONS

So far, the examples in this chapter have shown how each
string-manipulation function is used separately. You can also combine
functions in a control-glossary entry to increase the precision with which
you select records.

One example of how you can combine functions is given below. You have
seen how to use the **index** function to find records containing the title
"Vice President." You have also learned how to use the **occur** function to
test for a two-word title. It is likely that some vice-presidential
titles will contain other words, such as "Executive Vice President," or
"Vice President and Treasurer." Suppose you want a list that contains
only the records from your list document containing the two-word title
"Vice President." In that case, you can combine the **index** and **occur**
functions in one control-glossary entry, as shown below:

```
entry G     /*Select all titles that are two words long and contain the
            string Vice President, and sort alphabetically by company
            name*/
{
    select_record
        {
                if((occur(<title>," ") == 2) &
                (index(<title>, "Vice President")))
                save_record
        }
    sort<company>
}
```

NOTE: Notice that the complete argument for the **if**
statement is enclosed in parentheses, and that each
function within the argument is enclosed in parentheses.
The number of right and left parentheses must match for
the glossary to verify.

A SAMPLE INTERACTIVE CONTROL GLOSSARY

If you are familiar with regular glossary functions, you can use them in
control-glossary entries to create interactive entries. The example shown
below defines a variable, uses the **keys** function, and displays prompts on
the Records Processing menu screen. If you want to try this entry, type
it exactly as shown. Refer to the *Fortune:Word Glossary User's Guide*
for information about these functions.

Suppose you have a mailing list that contains some addresses in each
state. Each time you use the list document, you want to send a mailing to
the addresses in only one state. You can write an interactive
control-glossary entry that allows you to enter a two-letter state
abbreviation from the Records Processing menu. Thus, you can select
addresses from the list document according to the state you specify from
the Records Processing menu. Consequently, you do not need a separate
control-glossary entry for each state.

Such an entry could look like this:

```
entry k
{
        call prompt("Enter state:   ")
        response = keys

        select_record
            {
                    if(<state> == response)
                    save_record
            }
}
```

In this entry, the variable *response* is set equal to **keys** to allow you
to enter the state abbreviation from the Records Processing menu. The
call prompt statement displays a message on the Records Processing menu to
remind you to enter the state abbreviation before Records Processing
begins.

Chapter 7
Error Messages and Troubleshooting

You may encounter problems with Records Processing at two stages:

- During Records Processing
- During control-glossary verification

In this chapter, you will learn about the problems that commonly arise during Records Processing and about the kinds of problems you can have during control-glossary verification.

> NOTE: The format lines are carried over from the Format document at the end of Records Processing. If you press CANCEL while an output document is being created, the records that have been processed appear in the output document, but the format may not match the format in the source document.

INFORMATION MESSAGES IN RECORDS PROCESSING

When you use Records Processing, you fill in information on the Records Processing menu. If you use a control-glossary entry, the system checks to be sure the control glossary you specify is available, and that it contains an entry with the label you entered. Once Records Processing begins, information messages appear on the screen. Which messages are displayed depends on the choices you made on the Records Processing menu and the functions you included in the control-glossary entry.

- If your control glossary is performing the select_record function, you see the message *Select records phase*.

- If a sort is being done, you see the message *Sort phase*.

  > NOTE: If you are not using a control glossary, the two messages above are not displayed.

- When the resulting list has been completed and the merging process begins, you will see the messages *Merge phase* and *Processing record* followed by a number. This number tells you which record on the list is being merged. If the list has been sorted, the number reflects the order of the records in the sorted list, not the order in the list document.

ERROR MESSAGES IN RECORDS PROCESSING

At any point in the processing procedure, an error message may appear on the screen along with instructions to *Press execute to continue* or *Press EXECUTE to continue or CANCEL to stop*. If you press EXECUTE, Records Processing will skip that field and merge the remaining fields and records. If you press CANCEL, the process will cease. Any records processed thus far will be saved in the output document. However, the format line is not copied into the output document until the entire process has been completed. Thus, if you interrupt the process at any point before completion, the output document will not have the format line you created in your format document.

Depending on the message you see, you will need to find a mistake in the list, control, or format document and correct it before going on. Following the steps outlined below will help you locate errors.

Errors in the Selection and Sort Phases

1. Make a note of which phase was interrupted and what the error message was.

2. Look up the error message in the list that follows.

3. Examine the list document and control glossary to locate the problem. If you have printed a copy of these two documents, you may find the printed copies easier to use for troubleshooting.

4. Correct the problem. If it is in the list document, save the changes. If it is in the control glossary, reverify it.

5. Try to run Records Processing again.

Errors in the Merge Phase

1. Make a note of the number of the record that was being processed when the error occurred. This number on the screen refers to the order of the record in the sorted or selected list, not in the original list document.

2. You need to determine whether the problem is in the sorted or selected list or in the format document. As a test, start up Records Processing again but do not specify a format document. This will produce the sorted or selected list. If this works correctly, then the problem is in the format document. Check that document, taking into account the original error message you saw, and make the required corrections.

3. If your test indicates that the error is in the sort or select phases, again make a note of the number shown with the message *Processing record*. As above, this refers to the record number in the sorted or selected list that is your output document. Edit the output document and look at that record to determine the error. Depending on what it is, you may need to go back and make corrections either to the list or to the control glossary.

PROBLEM	SOLUTION

Can't access (control, format, list, or output) document

You do not have read permission for the document.	Change the document permissions. Refer to Appendix C of the *Fortune:Word Reference Guide* for information on how to change document permissions.

Can't open include document

The name of a document in an include statement in a format document cannot be opened for use.	Check any documents used in an include statement in the format document to be sure you have permission to use the document you specified.
The include document is being edited.	Check to be sure the document is not in use.
The include document you specified is not in your current library, or you have specified an incorrect pathname for the include document.	Check to be sure the include document is in your library. If it is not, specify a full pathname, or move it to your current library.
	Check to be sure that the document name is spelled correctly in the include statement.

PROBLEM	SOLUTION
Can't open temp file The results of a select or sort are placed in a temporary file. In this case, this file cannot be opened because the system disk is full or because you do not have permission to use the library specified.	Use the Filing menu to remove some documents from the system disk and try again, or change the permissions for the library.
Can't write to output document Your system disk is full.	Use the archiving functions on the Filing menu to remove some documents, and then try again.
Control glossary format error The control glossary has a syntax error in it.	Edit the control glossary, locate the error, and fix it.
Document in use On a multiuser system, someone else is using a document you have specified.	When the document is available, start over again.
Field size greater than 2048 characters A field in the list document is greater than 2048 characters.	Return to the list document, shorten the field, and try again.
Format document format error A field label is missing a < or >.	Return to the format document and make the correction.
The format document contains an end-of-record indicator (<>).	
The number of times in a repeat statement is zero.	
The include statement is not followed by a document name.	

PROBLEM   SOLUTION

Function in control glossary unavailable
An entry in the control   Return to the control glossary
glossary contains a function   and make corrections.
or command that is not
available.

Function 'substitute' has invalid arguments
The substitute argument you   Check your use of the substitute
specified has invalid   function and make changes.
arguments.

Glossary entry not found
The glossary entry you   Return to the control glossary
specified in a call statement   and find the proper name of the
is not in the control   entry you want.
glossary.

The glossary contains an   Edit the glossary and make the
entry that has not been   necessary corrections, or remove
verified, so none of the   the entry that does not verify.
entries in that glossary are
available for use.

Label not in list record   Compare the labels of the format
The format document contains   document with the labels in the
a field label that does not   list document and make
exist in the list document.   corrections. Or press EXECUTE to
                                           continue, then edit the output
                                           document to see which field(s)
                                           were not merged.

List and output documents have same name
You have specified the same   Type an output document name that
name for the list and output   is different from the list
documents.   document name.

List document format error
The list document does not   Return to the list document and
end with a < >, or some   make the necessary changes.
<'s and >'s are missing
from records.

The record size is larger
than 2048 characters, and the
control glossary is sorting
or selecting.

PROBLEM   SOLUTION

The size of a field is larger
than 2048, and the control
glossary is not sorting or
selecting.

The names of the list   Return to the Records Processing
document and format document   menu and make sure the names of
have been typed in the wrong   the list document and format
place on the Records   document are entered in the
Processing menu.   correct fields.

No records selected
No records were selected   If this is unexpected, check the
during the select-records   control and list documents to
phase.   determine why this happened.

Not available on this system
Accessing any Fortune:Word   The module is not installed on
menu function.   your system. Reinstall
                                     Fortune:Word.

Out of memory
Your records processing has   Check the length of the records
used up all available memory   and shorten them if possible,
on your Fortune system. You   break a complex control-glossary
may be working with long   entry up into several entries, or
documents, using a very   paginate the list document.
complex control-glossary
entry, or the list document
has no page breaks.

Substitute function has incorrect arguments
The arguments associated with   Return to the control glossary
a substitute function in the   and correct the error.
control glossary are
incorrect.

Too many fields to sort
More than four sorting fields   Return to the control glossary
are specified in the **sort**   and correct the **sort** statement.
statement in the control
glossary.

Too many levels of nesting
The include statement in the   Return to the format document and
format document contains more   make corrections.
than 16 levels of nesting.

PROBLEM	SOLUTION

Total of sort fields greater than 1024 characters

More than one field at a time is being sorted, and the total number of characters in all the fields to be sorted is greater than 1024.

Check the list document, locate the long fields, shorten them, and try again.

Trying to divide by zero

An entry in the control glossary contains an argument calling for a division by zero, or the field is not numeric.

Return to the control glossary and correct the entry, or use a numeric field.

Trying to sort more than once

You placed more than one **sort** statement in a control-glossary entry.

Return to the control glossary and delete the duplicate **sort** statement from the entry. Remember you can include up to four fields in one **sort** statement.

Trying to select_record more than once

The control-glossary entry specifies more than one select_record statement.

Return to the control glossary and delete the inappropriate select_record statement(s).

Record size greater than 2048 characters

When sorting by one field, the record is larger than 2048 characters.

Return to the list document and correct the record.

A field in the record is larger than 2048 characters.

ERRORS IN CONTROL GLOSSARIES

A control glossary must be verified before you can use any entry in it successfully. It does not need to be attached. The verification process checks your entries to make sure that the function names are spelled correctly and that the function syntax is correct. If there are any problems with the contents of the control glossary, it will not verify. When a glossary does not verify correctly, messages that tell you where to look for errors are saved on the work page of the control glossary.

If you encounter problems in verifying a control glossary, follow these steps:

1.   Press RETURN from the End-of-Edit Options screen.

2.   Go to the work page of the control glossary by pressing GO TO PAGE and typing **w**.

3.   New error messages are added to the bottom of the work page. Locate the error message received during the last editing session by checking the date and time of the error messages. Refer to the section entitled "Errors in Verification" that follows for more information.

4.   Go to the appropriate page in the glossary, correct the errors, and reverify the glossary.

Do NOT use a control glossary that has not verified correctly. If a glossary document contains one unverified entry, none of the entries in that glossary can be accessed. If you try to use such a glossary, you will encounter error messages during Records Processing such as *Press cancel to stop* and *Glossary entry not found*.

Errors in Verification

The error messages listed below will only appear on the work page of a control glossary that could not be verified. Any new error messages will be added to the bottom of the work page if you have not deleted earlier error messages. Each verification error message listed on the work page of a glossary shows the numbers of the page and line on which the error occurred. Return to the entry, correct the problem, and reverify the glossary. The alphabetical list of verification error messages below also indicates possible errors that could generate each message.

> NOTE: When you type two-word variable names such as **select_record**, the two words must be joined by an underbar (_). To type the underbar, press the Shift key, and then type _ (SHIFT/hyphen). Do not use MODE _ to type the underbar.

Can't start another entry here
   An ending brace is missing in a previous entry.

Illegal glossary entry name
   There are too many characters in the entry label; an illegal symbol appears in the entry label.

Illegal multiple keystrokes

You have specified a number argument for a keyword that does not recognize multiple keystrokes, for example *insert*(5). Numbers are only used with keywords such as *return, tab,* and *space*.

Illegal sorting statement inside select_record

The **select_record** argument must be enclosed in a pair of braces and must not include a **sort** statement.

Improper use of function

Function is not preceded by a **call** statement; function is misspelled; function cannot be used in the statement.

Keystrokes not allowed

A keyword or function name is misspelled; a syntax statement is incorrect.

Multiply defined entry name

You have two entries in the control glossary with the same entry label.

Routine not recognized

You have used a function in an illegal way, for example to call a control-glossary entry that does not exist.

Save_record allowed only inside select_record

You have used the **save_record** function without preceding it with a **select_record** statement; you have used the **save_record** function outside the **select_record** braces.

Select_record statement cannot be nested

You have included a second **select_record** statement within a **select_record** statement.

Undefined label

You used a **jump** function to jump to a nonexistent identifier label.

Unexpected variable

You have misspelled a keyword.

Unknown symbol .....

The control glossary contains a symbol that is not recognized as part of a control glossary.

Errors in Syntax

Most of the errors you encounter during the verification process are caused by improper syntax in the functions used in control-glossary entries. To help you find and correct such problems, the system generates very specific messages concerning syntax errors. The general form of a syntax error message is as follows:

> Syntax error on page *n*, line *n*: *x*

On your screen, page *n* and line *n* are replaced by the numbers of the page and line on which the error occurred. The *x* is replaced by the symbol, arithmetic operator, or control-glossary function that you used incorrectly or accidentally omitted.

To correct a syntax error, check the error message against the list of verification error messages. Refer to Chapters 3 and 4 of this guide for the proper syntax for the function, operator, or symbol you used incorrectly.

Using Arguments Within Functions

Certain control-glossary functions must include at least one or more arguments for the syntax to be correct. If you use an incorrect number of arguments with a particular function, the system will give you this message:

> *n* argument(s) expected for (function name)

On the screen, the name of the incorrectly-used function is given. The *n* is replaced by the number of arguments this function requires. Refer to Chapter 6, "Advanced Control-Glossary Functions," for the proper syntax for each function. Refer to the *Fortune:Word Glossary User's Guide* for information on the correct syntax for other glossary functions.

Appendix A
String-Manipulation Functions

You can use 13 string-manipulation functions in Records Processing control-glossary entries. Table A-1 lists these functions in alphabetical order.

Table A-1. String-Manipulation Functions

Description	Function
abs	absolute
cat	concatenate
index	index
len	length
max	maximum
min	minimum
num	numeric
occur	occurrence
round	round
seg	segment
sub	substitute
substr	substring
truncate	truncate

All except **abs, round, sub,** and **truncate** are described in detail in Chapter 6, "Advanced Control-Glossary Functions." For quick reference, all 13 functions are listed below in alphabetical order with their correct syntax and a brief description.

> **REMEMBER:** These functions can only be used to select records. They do not change the list, format, or output documents.

ABS (Absolute)

Syntax: **abs** <*field1*>

The **abs** function gives the *absolute value* of a number—its value without a plus or minus sign.

CAT (Concatenate)

Syntax: cat(<*field1*>,<*field2*>)

In Records Processing, the **cat** function is used to compare the contents of two or more fields. It does not combine or concatenate the contents of the fields.

INDEX

Syntax: index(<*field*>,"*string*", *n*)

The **index** function searches the specified field for the indicated string ("*string*") starting at the character specified by position *n*. If no match exists, the record is not selected.

LEN (Length)

Syntax: len(<*field*> == *n*)

The **len** function counts the number of characters in a field.

MAX (Maximum)

Syntax: max(<*field1*>,<*field2*>)

The **max** function can be used with either alphabetic or numeric text. With alphabetic text, **max** determines which string would occur last in the alphabet. With numeric text, it determines which number is larger. There is no limit to the number of fields you can compare.

MIN (Minimum)

Syntax: min(<*field1*>,<*field2*>)

The **min** function can be used with either alphabetic or numeric text. Used with alphabetic text, **min** determines which string occurs first in the alphabet. With numeric text, it determines which number is smaller. There is no limit to the number of fields you can compare.

NUM (Numeric)

Syntax: num *<field>*

The **num** function determines if a field is numeric or not. If the field is numeric, the record is selected. If it is not numeric, the record is not selected.

OCCUR (Occurrence)

Syntax: occur(*<field>*,*"delimiter"*)

The **occur** function counts the total number of segments in the field as defined by the delimiter.

ROUND

Syntax: round(*<field>*,*n*)

The **round** function rounds a floating-point number to the number of decimal places specified by n.

SEG (Segment)

Syntax: seg(*<field>*,*"delimiter"*,*n1*,*n2*)

The **seg** function looks at the segment of the field between the segment numbers $n1$ and $n2$. The delimiter you specify defines the segments of the field.

SUB (Substitute)

Syntax: sub(*<field>*,*"delimiter"*,*n1*,*n2*,*"string"*)

The **sub** function replaces text segments $n1$ and $n2$ in the field delimited by a particular character with the text of the string.

SUBSTR (Substring)

Syntax: substr(*<field>*,*n1*,*n2*)

The **substr** function tests the text between character positions $n1$ and $n2$ in the specified field. If the text matches, the record is selected.

TRUNCATE

Syntax: **truncate(** *<field>* **,***n***)**

The **truncate** function truncates a floating-point number to the number of decimal places specified by n.

Appendix B
ASCII Collating Sequence

Table B-1 shows the case-insensitive ASCII collating sequence that is used by the **sort** function. If you want to sort records containing non-English characters such as characters with accents, you must create an operating-system file named /usr/help/chtable.*XX*, where *XX*, represents a language. Refer to Appendix E in the *Fortune:Word Reference Guide* for instructions on how to create such a file.

Table B-1. Case-Insensitive ASCII Collating Sequence

space	3	b, B	u, U
!	4	c, C	v, V
"	5	d, D	w, W
#	6	e, E	x, X
$	7	f, F	y, Y
%	8	g, G	z, Z
&	9	h, H	{
'	:	i, I	|
(	;	j, J	}
)	<	k, K	-
*	=	l, L	
+	>	m, M	
,	?	n, N	
-	@	o, O	
.	[	p, P	
/	]	q, Q	
0	^	r, R	
1	—	s, S	
2	a, A	t, T	

Appendix C
Creating a List Document Using Glossary Entries

This appendix contains suggestions on how you can use regular glossary entries to save time when you create a list document for Records Processing. Refer to the *Fortune:Word Glossary User's Guide* for information on the functions used in this appendix.

Several glossary entries are presented here. Programming styles differ, and there are many ways to write glossary entries that perform the same functions. You can experiment with these glossary entries, or write your own. Some of these entries are more complicated than others. When you type these entries into a glossary, be sure to copy them exactly. Then try them out to be sure they are working correctly before you use them for creating a list document.

When you create a glossary to use with Records Processing, it is a good idea to use the initials **rp** in the glossary name to help you identify it as a Records Processing glossary.

> **REMEMBER:** Although you can combine control-glossary entries and regular entries in the same glossary, you may find it more convenient to create a separate glossary for use with Records Processing.

The glossary entries in this appendix use the field labels from the **parts.list** document used as an example in Chapter 2.

Creating Individual Glossary Entries for Each Field Label

You can create a separate glossary entry for each of the field labels in a list document. Each time you want to enter a field label, press GL and type the entry label assigned to that field label.

The advantage of this approach is that you can use the appropriate entries to put the field labels in the format document as well as in the list document. The disadvantage of this approach is that it is a slower way to create a list and that you must remember the entry letter for each field label you want to recall.

You can use the Glossary by Example feature to create the entry for each field label. If you prefer, you can type the entries directly into the glossary.

An Interactive Glossary Entry

You can write interactive glossary entries. With such an entry, the system stops and allows you to enter the field contents after displaying each label. You cannot use Glossary by Example to create interactive glossary entries; you must type them directly into a glossary. Create the entry named **b** shown below in the glossary called **parts.g.rp**. This entry generates the field labels one by one, pausing after each label to allow you to interactively enter the contents for each field. When you have typed the contents of a field, press EXECUTE to continue to the next part of the entry.

The entry below is written using the glossary programming language. You can use the entry without understanding just what it means if you type it carefully, exactly as shown. The names between each merge and MERGE are field labels. You can change these to tailor this glossary entry to your needs, adding or deleting lines to suit the particular list you wish to create.

You can use either tabs or spaces to indent glossary text. When you type single or double quotes, use the key to the left of the RETURN. Refer to the *Fortune:Word Glossary User's Guide* for more complete information on how to write glossary entries.

```
entry b
{
        call posmsg(2,48,"Press EXECUTE to continue")
        insert "<part no>" call keysin return
        "<type>" call keysin return
        "<in stock>" call keysin return
        "<on order>" call keysin return
        "<last order>" call keysin return
        "<unit cost>" call keysin return
        "<>" return execute
        call clrpos(2,48,26)
        call prompt("Repeat entry?  y/n")
        question = key
[loop]
                if((question == "y") | (question == "Y"))
                    {
                            call prompt("")
```

(**entry b** continued on next page)

(**entry b** continued)

```
                          call b
                }
                if((question == "n") | (question == "N"))
                {
                          call prompt("")
                          exit
                }
                if((question != "y") | (question != "Y") |
                (question != "n") | (question != "N"))
                {
                          call prompt("Try again:  y/n")
                          question = key
                          jump loop
                }
        }
```

> **REMEMBER:** A glossary must be verified and attached before you can use it.

Once the glossary is verified and attached, create your list document. With the list document on the editing screen, follow these steps to use entry b:

1. Press GL and type **b** to start the glossary entry.

2. The first field label is displayed on the editing screen by the system. Type the contents for the first field and press EXECUTE. A RETURN is automatically generated at the end of each line when you press the Execute key; you do not need to press the Return key yourself.

    You can use the Backspace key to go back and type over any errors, but you cannot use the Insert and Delete keys while you are entering field contents using this glossary entry.

3. The next field label appears on the editing screen. Type the contents for the field, and press EXECUTE.

4. Continue typing the field contents and pressing EXECUTE until you reach the end of a record.

5. At the end of each record, the system asks if you want to continue with another record. You can type **y** for **yes** and continue until you have typed all the records.

6. When the prompt appears at the top of the screen after you have entered the last record, type **n** for **no** to exit from the glossary entry.

A Second Interactive Glossary Entry

Another interactive glossary entry you can use to create list documents is given below.

When you use this entry, the field contents are displayed on the bottom line of the screen as you type them. You can use the Backspace key to go back and correct typographical errors. Press RETURN or EXECUTE to go on to the next field label. The field contents must be all on one line. When you have entered the last record, type the word "end" as the field contents for the first field label. The system will then delete this unnecessary field label and return you to the editing screen. You can press CANCEL at any time to stop this entry.

```
entry 1
{
call prompt("")
[loop]

"<part no>"
call prompt("Type entry and ")
call posmsg(2,50,"\034H Press EXECUTE:  \034Id")
call posmsg(25,1,'When done type "end"')
call status("")
a = keys
call prompt("")
call clrpos(2,50,31)
        if(a == "end")
                {
                        goto left delete return execute
                        call clrpos(25,1,21)
                        exit
                }
call feed(a)
return

"<type>"
call prompt("Type entry and ")
call posmsg(2,50,"\034H Press EXECUTE:  \034Id")
call status("")
a = keys
call prompt("")
call clrpos(2,50,31)
call feed(a)
return
```

(entry 1 continued on next page)

(**entry** 1 continued)

```
    "<in stock>"
    call prompt("Type entry and ")
    call posmsg(2,50,"\034H Press EXECUTE:  \034Id")
    call status("")
    a = keys
    call prompt("")
    call clrpos(2,50,31)
    call feed(a)
    return

    "<on order>"
    call prompt("Type entry and ")
    call posmsg(2,50,"\034H Press EXECUTE:  \034Id")
    call status("")
    a = keys
    call prompt("")
    call clrpos(2,50,31)
    call feed(a)
    return

    "<last order>"
    call prompt("Type entry and ")
    call posmsg(2,50,"\034H Press EXECUTE:  \034Id")
    call status("")
    a = keys
    call prompt("")
    call clrpos(2,50,31)
    call feed(a)
    return

    "<unit cost>"
    call prompt("Type entry and ")
    call posmsg(2,50,"\034H Press EXECUTE:  \034Id")
    call status("")
    a = keys
    call prompt("")
    call clrpos(2,50,31)
    call feed(a)
    return

    "<>" return
    jump loop
    }
```

Using a Glossary to Create an Entry for Creating List Documents

As you can see from the examples given above, there are many ways to use glossary entries to help you produce your list documents. You can experiment with different glossary entries until you find one that best suits your needs.

In addition to using a glossary entry to generate the field labels for your list document, you can use a glossary entry to create another glossary entry that will, in turn, generate your field labels. If you make many list documents with different field labels, you may find it convenient to use the glossary entry shown below.

With this glossary entry, you type all the commands in glossary programming language only once. When you want to create a glossary entry to use in creating a list document, you just call up this glossary entry. It causes the system to generate the required commands. All you have to do is type the field labels you want. This process produces a glossary entry that you can then use to create your list document.

```
entry c      /*c = create.  Use this glossary to create an entry for a
             list document.*/

{
"entry "
call prompt("Enter entry label")
label = key
call feed(label) return
"{" return
"call prompt(" quote quote ")" return
"[repeat]" return

[loop]
"merge "
call prompt("Enter field name and")
call posmsg(2,50,"\034H Press EXECUTE:  \034Id")
variable = keys
call prompt("")
call clrpos(2,50,31)
call status('When done, type "end"')
if(variable == "end")
        {
                call status("")
                jump end
        }
```

(**entry c** continued on next page)

(**entry c** continued)

```
' ' quote call feed(variable) quote " MERGE" return
"call prompt(" quote "Type entry and " quote ")" return
"call posmsg(2,50," quote "\\034H Press EXECUTE:  \\034Id" quote
")" return
"call posmsg(25,1," quote "When done type 'end'" quote ")" return
"call status(" quote quote ")" return
"a = keys" return
"call prompt(" quote quote ")" return
"call clrpos(2,50,31)" return
        tab "if(a == \qend\q)" return
            tab(2) "{" return
                    tab(3) "goto left delete return execute" return
                    tab(3) "call clrpos(25,1,21)" return
                    tab(3) "exit" return
            tab(2) "}" return
"call feed(a)" return
"return" return

jump loop
[end]
' ' "MERGE return" return
"jump repeat" return
"}" return
exit
}
```

To use this glossary entry, follow these steps:

1.   Edit an old glossary or create a new one.

2.   Type the entry exactly as shown above.

3.   Verify the glossary and attach it.

4.   Edit an old glossary or create a new one that you will use to create the list document.

5.   Press GL and type c.

6.   Type a one-letter entry label.

7.   When the glossary prompts you to do so, type the first field label you want to use in the list document and press EXECUTE. It takes a few moments for the text of the glossary entry to appear on the screen.

8. When the glossary prompts you to do so, type the next field label and press EXECUTE. After you have entered each field label one time and the glossary is asking you to enter another field label, type **end** and press EXECUTE. The glossary entry will be terminated.

9. Verify and attach the glossary that contains the new entry.

10. Create or edit a document that will contain your list.

11. Press GL, and type the label you have assigned to the glossary entry that will generate your field labels.

12. When you use this entry, the field contents are displayed on the bottom of the screen as you type. You can use the Backspace key to go back and correct typographical errors. When the field contents have been entered correctly, press RETURN or EXECUTE to display the next field label.

13. When you have entered the last record, type the word **end** as the field contents for the first field label and press EXECUTE. The system automatically deletes this unnecessary field label from the glossary entry and returns you to the editing screen.

14. Save the text you have entered in the list document.

Index

occur

    5-5, 5-6, 6-7, 6-9, 6-10,

    A-1, A-3

prompt

    5-5, 5-6, 6-19

round

    5-5, 6-7, A-1, A-3

save_record

    5-5, 5-12, 5-13, 5-15,

    5-19, 6-2, 7-9

seg

    5-5, 5-6, 6-7, 6-14, 6-15,

    A-1, A-3

select_record

    5-5, 5-12, 5-13, 5-15,

    5-16, 5-19, 6-2, 7-1, 7-7

    through 7-9

selection

    5-12, 5-13, 5-16

sort

    2-4, 5-4 through 5-12,

    5-19, 7-1, 7-4, 7-6, 7-7,

    7-9, B-1

status

    5-5, 5-6

sub

    5-5, 5-6, 6-7, A-1, A-3

substr

    5-5, 5-6, 6-7, 6-15 through

    6-17, A-1, A-3

thru

    5-5, 5-17, 5-18

true

    5-5, 5-6

truncate

    5-5, 5-6, A-4

while

    5-5

G

glossary

    attach

        C-7, C-8

    by example

        2-16 through 2-19, 7-9

    control

        1-4, 1-6, 1-7, 1-9, 1-10, 2-4,

        5-1, 5-4, 5-7, 5-19, 5-20, 6-6,

        6-18, 7-1, 7-4 through 7-9

    functions

        1-6, 1-9, 2-17, 6-1

    interactive

        C-2, C-4

greater than

    5-14, 6-12, 7-4, 7-7

H

header page

    3-2

headings, column

    3-12, 3-13

N

nesting (16 levels)

3-16, 3-17, 7-6

999 pages, maximum document size

1-10

num function

5-5, 5-6, 6-7, 6-10, 6-11,

A-1, A-3

numeric

6-5, 6-7, 6-10, 6-11, A-1,

A-3

O

occur function

5-5, 5-6, 6-7, 6-9, 6-10,

A-1, A-3

occurrence

6-7, 6-9, A-1, A-3

operators

addition

6-2

arithmetic

6-1, 6-2

division

6-4

multiplication

6-4

relational

5-13, 5-14, 6-2

remainder

6-5

subtraction

6-3

output document

1-3, 1-7, 1-8, 1-10, 1-11, 3-2,

3-3, 3-5, 3-9 through 3-12,

3-15, 3-21, 4-10, 5-1, 5-10,

5-12, 7-1, 7-2, A-1

P

page break

2-2, 2-5, 2-6, 2-11, 3-2, 3-5,

3-6, 3-16, 3-17, 4-10, 7-6

parts.form

3-13, 3-14

parts.g.rp

2-17, 2-18, C-2

parts.inc

3-18, 3-19 parts.list

2-17, 2-19, 2-21, 3-11, 3-13,

3-14, C-1

parts.out

3-14, 3-18, 3-19, 3-21

print only

1-10, 1-11, 4-6

printer control

1-11, 4-7, 4-10

printing
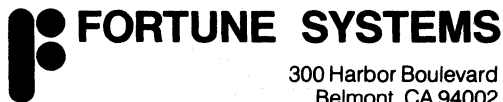
functions

1-10

FORTUNE SYSTEMS

300 Harbor Boulevard
Belmont, CA 94002